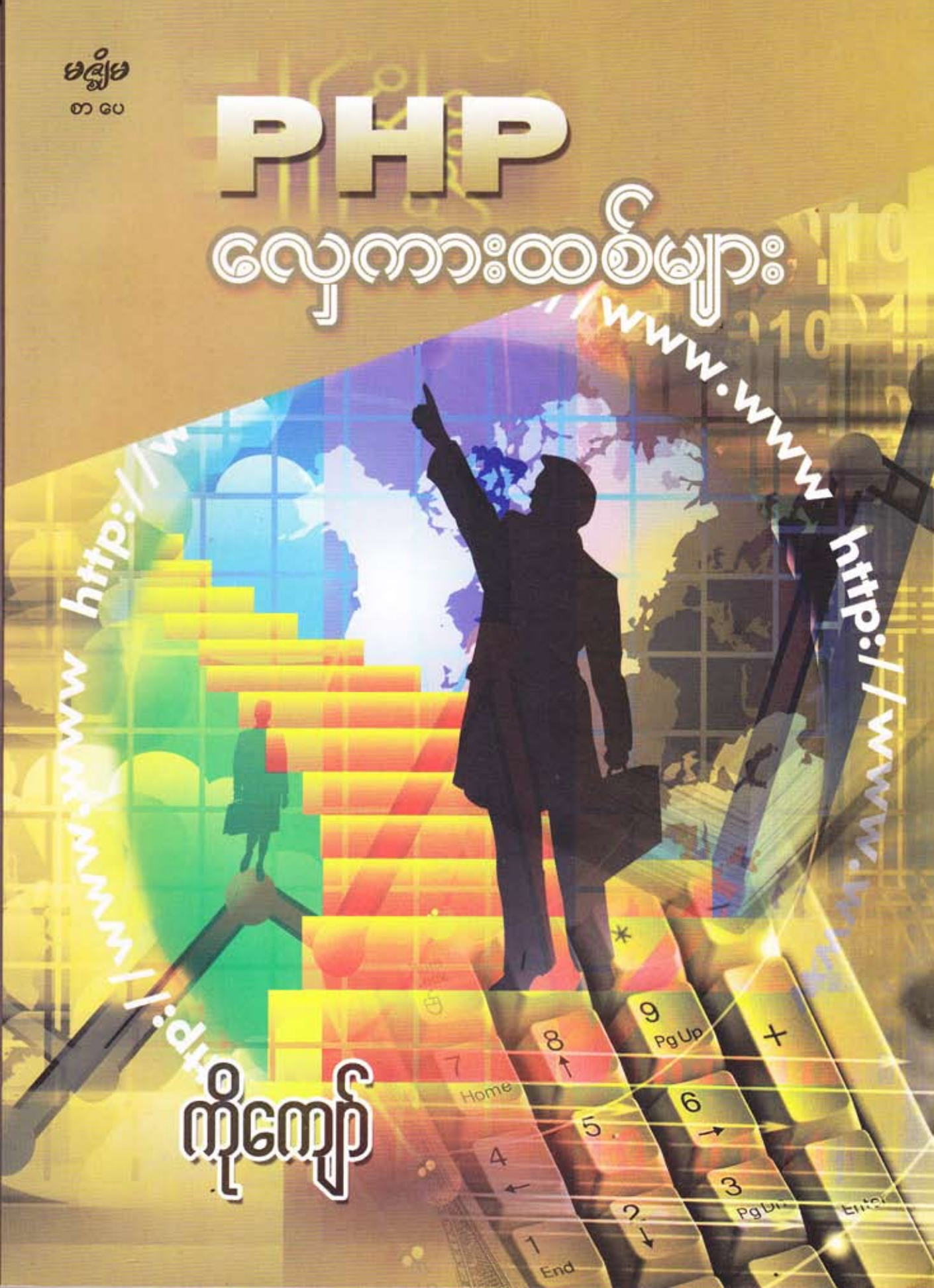


# PHP

## လေ့ကျင့်စဉ်များ

တိုကျော်



## မာတိကာ

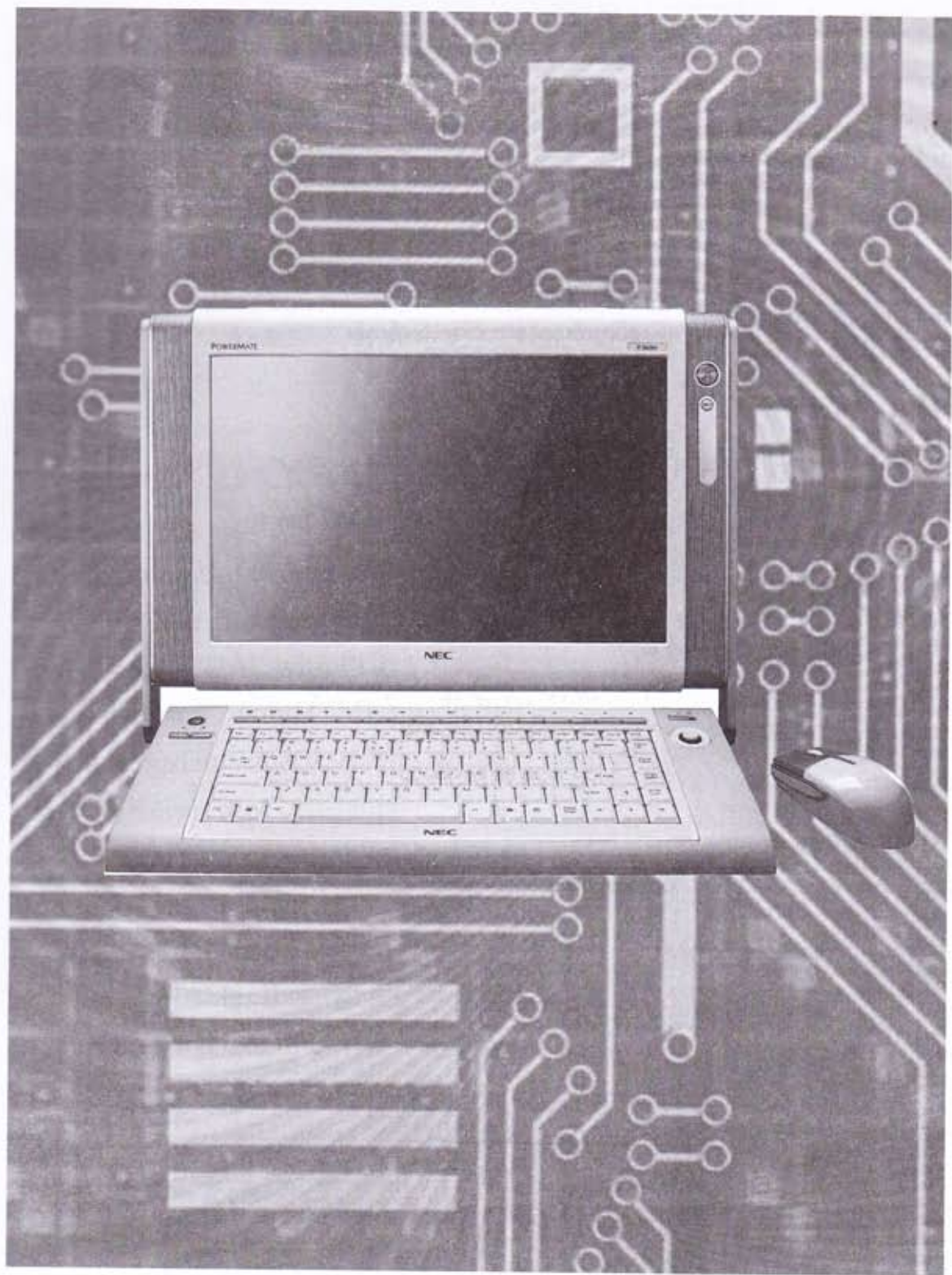
### Chapter(1), Configure your installation

|                                |    |
|--------------------------------|----|
| Introduction . . . . .         | .1 |
| Requirements . . . . .         | .2 |
| Starting with browser. . . . . | .6 |
| How to use notepad. . . . .    | .8 |

### Chapter (2), Creating PHP pages using PHP

|  |     |
|--|-----|
| How PHP fits with HTML. . . . .                              | .9  |
| The important of coding practices . . . . .                  | .10 |
| Creating your first program. . . . .                         | .12 |
| Integrating HTML with PHP . . . . .                          | .15 |
| Considerations with HTML inside PHP . . . . .                | .15 |
| Using Constants and Variables to add Functionality . . . . . | .17 |
| Overview of constants. . . . .                               | .17 |
| Overview of variables. . . . .                               | .19 |
| Using variables. . . . .                                     | .19 |
| Passing variables between pages . . . . .                    | .22 |
| Passing variables through a URL . . . . .                    | .22 |
| Special characters in URLs . . . . .                         | .25 |
| Passing variables with SESSIONS . . . . .                    | .26 |
| Comparison operators. . . . .                                | .31 |
| Passing variables with cookies . . . . .                     | .33 |





NEC Powermate

|   |     |
|---|-----|
| Passing information with Forms . . . . .          | .37 |
| Using forms to get information . . . . .          | .37 |
| Using if / else arguments . . . . .               | .42 |
| Using if statements . . . . .                     | .42 |
| Using if and else together . . . . .              | .44 |
| While and do-while . . . . .                      | .46 |
| do-while loop. . . . .                            | .47 |
| FOR loop . . . . .                                | .47 |
| FOREACH.. . . . .                                 | .48 |
| BREAK. . . . .                                    | .50 |
| Switch . . . . .                                  | .50 |
| Continue . . . . .                                | .52 |
| return . . . . .                                  | .53 |
| Understanding ARRAYS . . . . .                    | .53 |
| Incrementing / Decrementing operators : . . . . . | .55 |
| Logical operators . . . . .                       | .55 |
| Using INCLUDES for Efficient code . . . . .       | .57 |
| Using FUNCTIONS . . . . .                         | .59 |
| All about Arrays . . . . .                        | .64 |
| Sorting Arrays . . . . .                          | .67 |
| Concatenate . . . . .                             | .79 |



## CHAPTER (3) , Using PHP with MySQL

|  |      |
|--|------|
| Overview of MySQL structure and Syntax . . . . .       | .81  |
| MySQL structure . . . . .                              | .81  |
| Field types .. . . .                                   | .82  |
| Choosing the right field type . . . . .                | .85  |
| null or not null . . . . .                             | .86  |
| Indexes . . . . .                                      | .87  |
| Auto-increment . . . . .                               | .87  |
| Type of MySQL Tables and Storage Engine . . . . .      | .88  |
| MySQL syntax and commands . . . . .                    | .89  |
| How PHP fits with MySQL . . . . .                      | .90  |
| Connecting to the MySQL Server . . . . .               | .91  |
| Looking at the ready-made Database . . . . .           | .92  |
| Querying the database . . . . .                        | .99  |
| WHERE . . . . .  | .100 |
| Logical operators . . . . .                            | .101 |
| Working with PHP and arrays of Data(foreach) . . . . . | .104 |
| Using foreach to create table . . . . .                | .106 |
| Working with two tables . . . . .                      | .107 |
| Referencing Two Tables . . . . .                       | .107 |
| Referencing Individual Tables . . . . .                | .108 |

**CHAPTER (4), Using Tables to Display Data**

|                                       |      |
|---------------------------------------|------|
| Creating a table . . . . .            | .113 |
| Using heredoc syntax . . . . .        | .119 |
| Adding links to table . . . . .       | .127 |
| Creating movie review table . . . . . | .136 |

**CHAPTER (5), FORM Elements: Letting the User Work with Data**

|                                   |      |
|-----------------------------------|------|
| First Form . . . . .              | .147 |
| Multiple submit buttons . . . . . | .157 |
| Basic input testing . . . . .     | .157 |
| Ternary operator . . . . .        | .158 |
| Linking forms together . . . . .  | .159 |

**CHAPTER(6), Letting the user Edit the DATABASE**

|   |      |
|---|------|
| Creating admin page . . . . .                         | .169 |
| Inserting a Record in a Relational Database . . . . . | .173 |
| Deleting record . . . . .                             | .181 |
| Editing Data in a Record . . . . .                    | .186 |



# Chapter 1

## Configuring Your Installation

### Introduction

Great website တွေကို creat လုပ်ပေးနိုင်တဲ့ technologies တွေက အများကြီးရှိပါတယ်။ အဲဒီထဲကမှ သင်ဟာ Apache/MySQL/PHP (တစ်ခါတစ်ရံ AMP လို့သုံးနှုံးပါတယ်) ကို ရွေးချယ်ပြီး ပီးလေ့လာကြမှာဖြစ်ပါတယ်။ သင့်အနေနဲ့ အရင်ကတည်းက မလေ့လာဖူးတာဖြစ်ဖြစ် နည်းနည်းပဲ ပီးပါးလေ့လာဖူးရင်ဖြစ်ဖြစ် ဒီစာအုပ်ကို ဖတ်ရှုနိုင်ပါတယ်။ Apache, MySQL, PHP ဆိုတဲ့ ၃ ခုလုံးဟာ တစ်ပိုင်းခြင်းစီဖြစ်ဖြစ် သုံးခုတွဲဖြစ်ဖြစ် တော်တော်လေးကို complex ဖြစ်ပါတယ်။ ယခုစာအုပ်ထဲမှာ အဲဒီ ၃ ခုလုံးနဲ့သက်ဆိုင်တဲ့ အကြောင်းအရာအားလုံးရဲ့ details ကို ဖော်ပြနိုင်ဖို့ဆိုတာ တော်တော်လေးကို မလွယ်ကူပါဘူး။ မဖြစ်နိုင်ပါဘူး။ ဒီစာအုပ်ရဲ့ အဓိက ရည်ရွယ်ချက်က ခုနက ပြောခဲ့တဲ့ AMP တစ်ခုခြင်းစီကို အခြေခံကနေ နားလည်လွယ်အောင် ရည်ရွယ်ပြီးရေးထားတာပါ။ တစ်ချို့နေရာတွေမှာ coding အားလုံးကို အသေးစိတ်လိုက်မရှင်းထားတာက ရှေ့သင်ခန်းစာမှာ ရှင်းထားခဲ့ပြီးရင် နောက်သင်ခန်းစာမှာထပ်မရှင်းတော့ပါဘူး။ အဓိက ရည်ရွယ်ချက်ကို focus လုပ်နေတာကနေ သွေဖီသွားမှာစိုးလို့ပါ။

နောက်တစ်ခုက ဒီစာအုပ်ကို ဖတ်မယ်ဆိုရင် စာဖတ်သူဟာ HTML, CSS ကို နားလည် ထားသူ သို့မဟုတ် တီးမိခေါက်မိရှိထားသူဖြစ်ဖို့လိုပါတယ်။ HTML နဲ့ CSS နဲ့ပါတ်သတ်တာတွေကို လုံးဝ ထည့်မရှင်းထားပါဘူး။ တစ်ခြား နောက်ထပ် Program တစ်ခုခုကို လည်း သင် မသိထားရင် ရပါတယ်။ သိထားတဲ့သူတွေကတော့ programming concept ရတာပေါ့။ ဆိုလိုတာကတော့ APACHE/MySQL/PHP newbie တွေအတွက် ရည်ရွယ်ပါတယ်။ တကယ်လို့များ ကျွန်တော် intermediate level အတွက် စာအုပ် ထုတ်ဖြစ်ခဲ့မယ်ဆိုရင် အဲဒီစာအုပ်မှာ အောက်ပါအခန်းများ ပါဝင်လာမှာဖြစ်ပါတယ် -

1. Image upload
2. Create thumbnails
3. Handling and avoiding errors
4. Validating user input

အခု စာအုပ်မှာ အဓိက ပါတာတွေကတော့ PHP ကို စတင်ဖို့အတွက် မိမိစက်ထဲမှာ web server တစ်ခု တည်ဆောက်တာနဲ့ ကျန်တဲ့အပိုင်းကတော့ PHP/MySQL အဓိက ဖြစ်ပါတယ်။ အခုစာအုပ်ကို သင်က ဆုံးခန်းတိုင်အောင် ဖတ်ပြီးပြီဆိုပါက သင့်အနေဖြင့် movie review dynamic website တစ်ခုကို အခြေခံကျကျ ဖန်တီးနိုင်မှာဖြစ်ပါတယ်။ ဘာတွေပါလဲဆိုရင် movie name တွေပါမယ်။ အဲဒီ movie တစ်ခုခြင်းစီရဲ့ details တွေပါမယ်။ user တွေရဲ့ comments တွေပါမယ်။ အဲဒီ comments တွေနဲ့အတူ movie ratings တွေပါ ကြယ်လေးတွေနဲ့ ပြတတ်လာမယ်။ ပြီးရင် နောက်ထပ် movie အသစ်တွေ ထပ်ပေါင်းထည့်နိုင်မယ်။ ရှိပြီးသားဟာတွေကို Edit လုပ်နိုင်မယ်။ delete လုပ်နိုင်မယ်။ အဲဒါတွေ အားလုံးအတွက် concepts တွေကို သေချာရှင်းပြထားပါတယ်။ တစ်ချို့နေရာတွေမှာ screen

shot နဲ့ ပြပါတယ်။

Requirements

ခု သင်ခန်းစာအတွက် လိုအပ်တာတွေကတော့

၁။ Browser

၂။ Text editor ( ကျွန်တော်ကတော့ notepad++ သုံးပါတယ်)

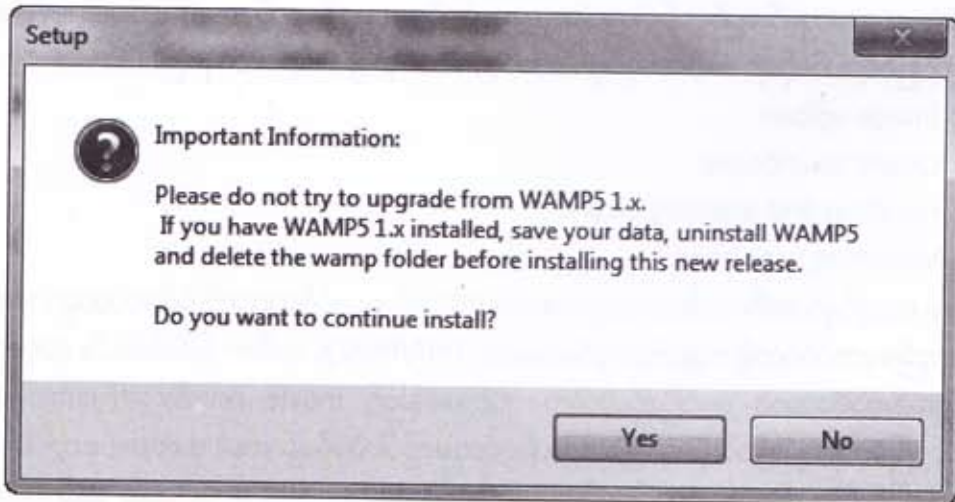
၃။ Wamp server

၄။ developer tools bar (firefox addon)

တို့ဖြစ်ပါတယ်။ အဲဒါတွေ အားလုံးကို ဒီစာအုပ်နဲ့အတူ ပါလာမယ့် CD ချပ်ထဲကနေ ရနိုင်ပါတယ်။

Wamp server installation

ပထမဆုံး CD ထဲက WampServer2.0c ဆိုတဲ့ .exe file ကို double click လုပ်ပါ။

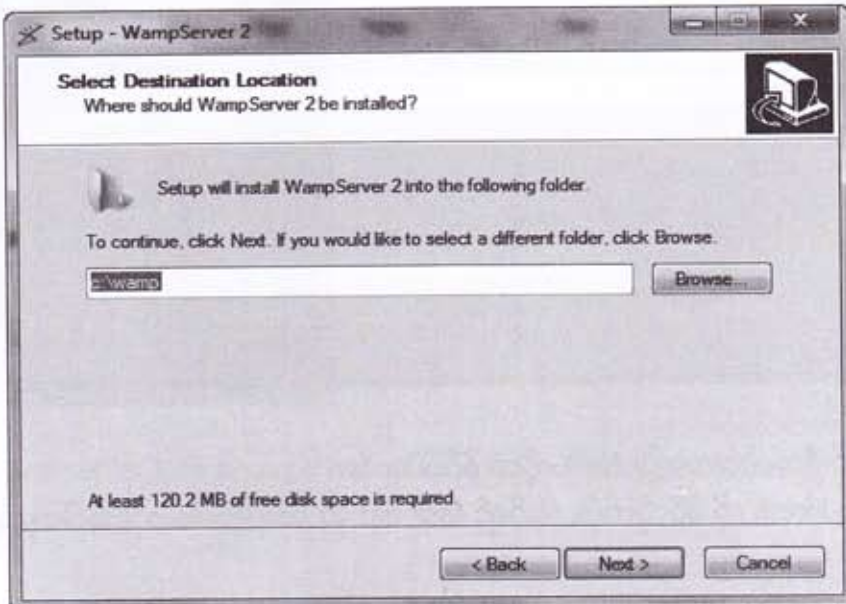


Yes ကို နှိပ်ပါ။





Next ကို နှိပ်ပါ။

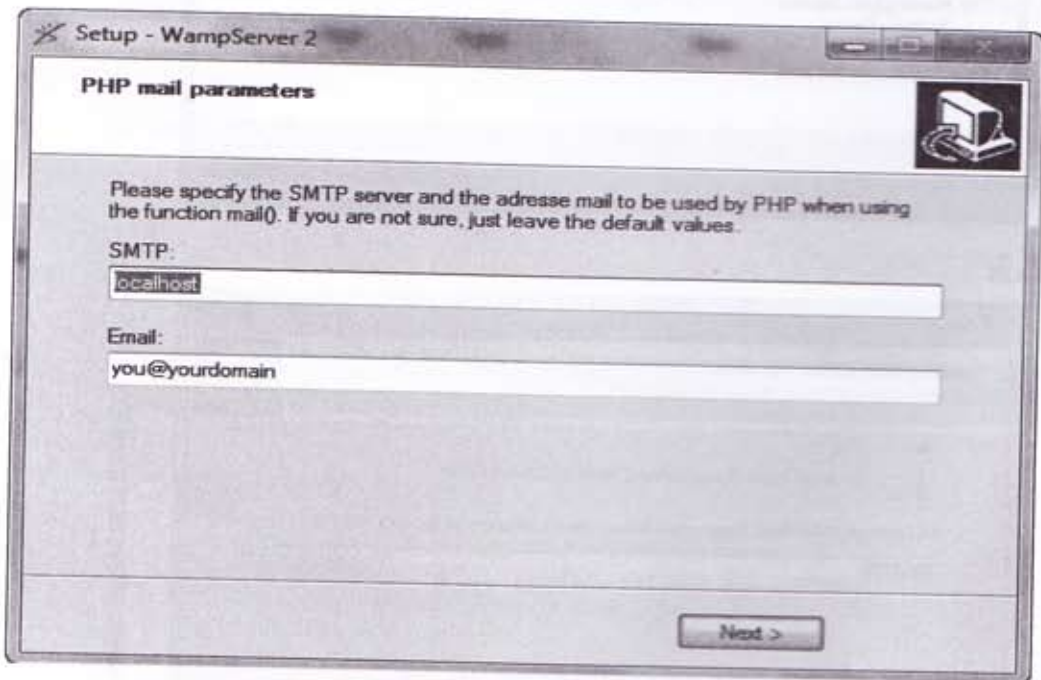


default အနေနဲ့ c အောက်မှာ wamp ဆိုတဲ့ folder နာမည်နဲ့ install လုပ်မှာပါ။ Next ကိုဘဲ နှိပ်ပါ။  
နောက်တစ်ခါ Quick launch တို့ဘာတို့ရွေးခိုင်းရင်လည်း နှစ်သက်သလို ရွေးပြီး Next လုပ်ပါ။

ပြီးရင်တော့ Install ကို နှိပ်ပါ။ ဒါဆိုရင် သူ့ဟာသူ install လုပ်နေပါလိမ့်မယ်။

ပြီးသွားရင် default browser ရွေးခိုင်းပါလိမ့်မယ်။ ကျွန်တော်ကတော့ firefox ကိုဘဲရွေးခဲ့ပါတယ်။

နောက်တစ်ဆင့်အနေနဲ့ mail parameters ကို ရွေးခိုင်းပါတယ်။



လောလောဆယ် ဒီအတိုင်းထားပြီး Next ကိုဘဲ နှိပ်ပါတယ်။

ပြီးသွားရင်တော့ Finish ကို နှိပ်လိုက်ပါ။ ဒါဆိုရင် task bar မှာ အောက်ပါအတိုင်းပေါ်နေပါလိမ့်မယ်။

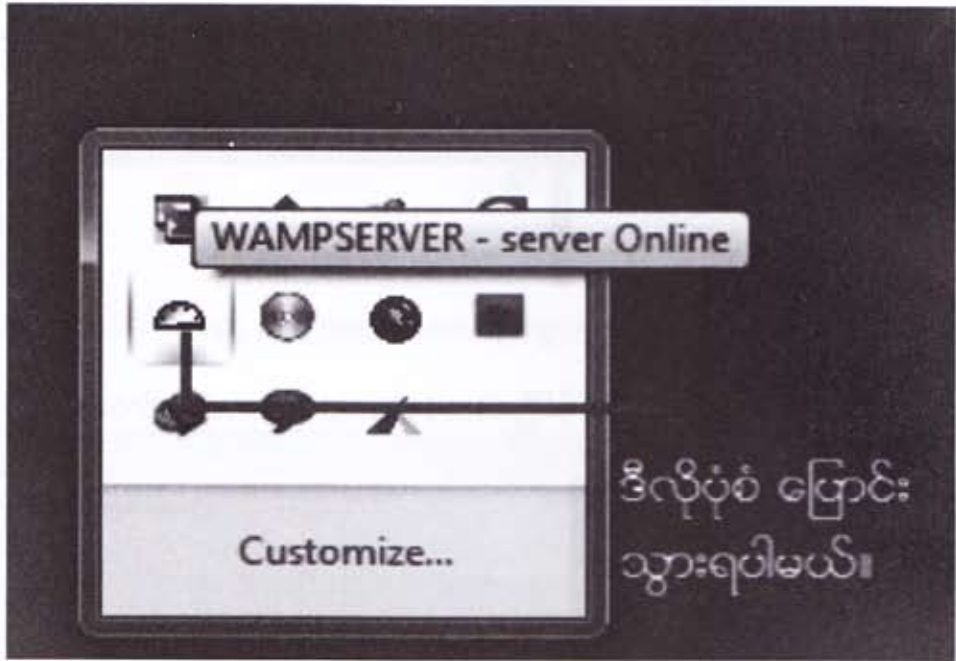




အဲဒီ မြားပြထားတဲ့ icon ပေါ်မှာ left click နှိပ်ပြီးတော့ put online ကို ကလစ်လိုက်ပါ။



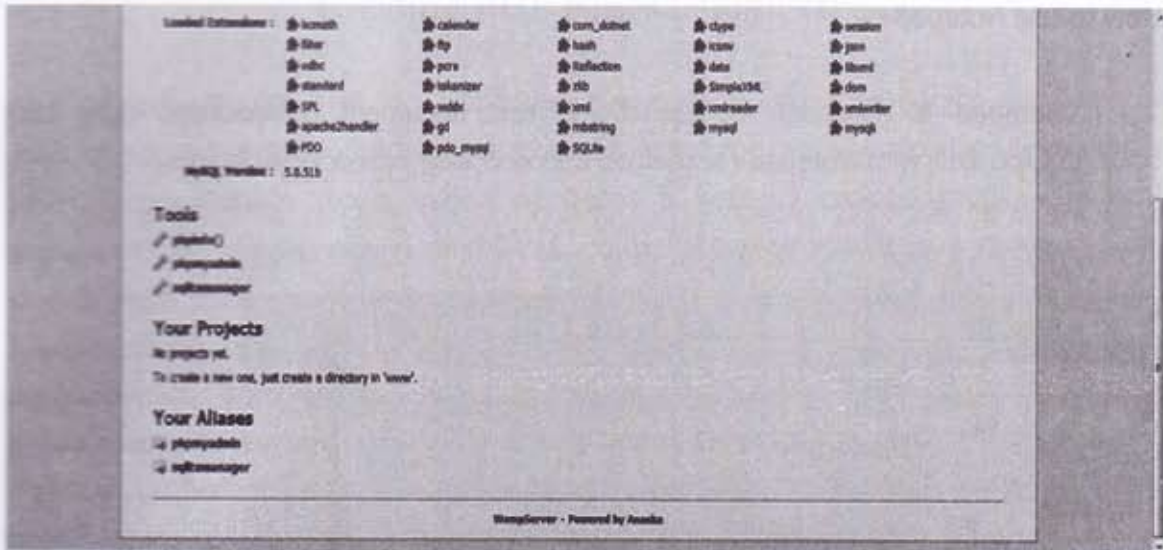
အဲဒါပြီးရင် wamp server icon လေးဟာ အောက်ပါပုံအတိုင်းပြောင်းသွားပါလိမ့်မယ်။



အားလုံးပြီးသွားပြီဆိုရင် သင့်အနေနဲ့ wamp server ကို အောင်မြင်စွာ install လုပ်ပြီးသွားပါပြီ။ သင် ယခုပြုလုပ်လိုက်တဲ့ ဆာဗာထဲမှာ PHP 5.2.6 , Apache 2.2.8 , MySQL 5.0.51b တို့ ကိုလည်း install လုပ်ပြီးသား ဖြစ်သွားပါမည်။

**Starting with browser**

firefox browser မှာ localhost or 127.0.0.1 လို့ ရိုက်ထည့်လိုက်ပါက အောက်ပါအတိုင်း တက်လာရပါမည်။  
ဒါမှမဟုတ်လည်း wamp server icon အပေါ်မှာ left click လုပ်ပြီး localhost ကို click လုပ်ရင်လည်း တက်ပါတယ်။



Your projects ဆိုတဲ့နေရာမှာ ခုလောလောဆယ် အောက်ပါအတိုင်းမြင်ရမှာဖြစ်ပါတယ်။

No projects yet.

To create a new one, just create a directory in 'www'.

ကျွန်တော်တို့လုပ်သမျှ projects အားလုံးသည် c:\wamp\www ဆိုသော folder အောက်တွင်သာ ပြုလုပ်ရမည် ဖြစ်ပါသည်။

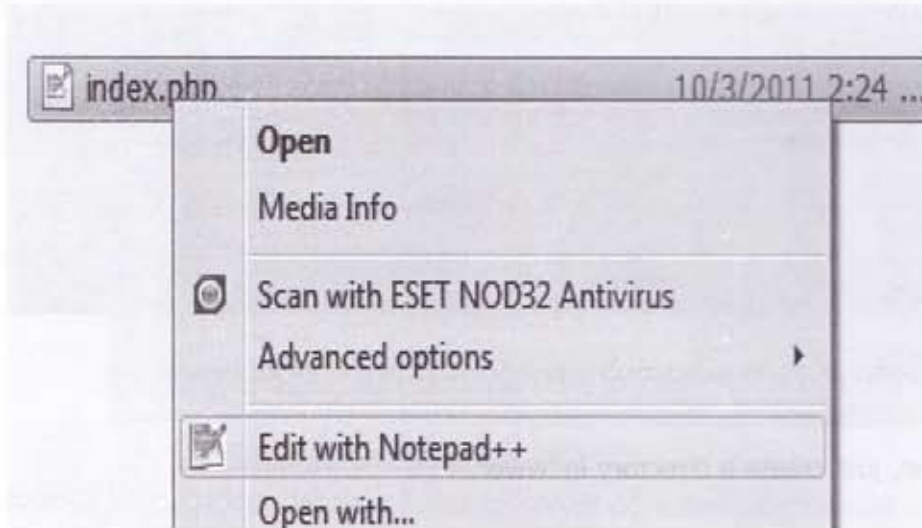
နောက်ထပ် phpMyAdmin ကို ခေါ်လိုလျှင် wamp server icon ပေါ်မှ phpMyAdmin ကို click လုပ်ပြီး ခေါ်နိုင်ပါသည်။

Default Database username မှာ root ဖြစ်ပြီး password မှာ blank ဖြစ်ပါသည်။ ယခုစာအုပ်တွင် Default အတိုင်းသာ သုံးသွားမည်ဖြစ်ပါသည်။



How to use Notepad++

Notepad++ ကို install လုပ်ပြီးပြီဆိုပါက text document တစ်ခုပေါ်တွင် right click လုပ်ကြည့်ပါက Edit with Notepad++ ဆိုသော စာတမ်းကိုတွေ့မြင်ရမည်ဖြစ်ပါသည်။



သင်က php page တစ်ခုကို စတင်ပြီး create လုပ်လိုပါက www folder ထဲတွင် text document တစ်ခု လုပ်လိုက်ပါ။ ပြီးရင်အဲဒီ text document ပေါ်ကို right click လုပ်ပြီးတော့ Edit with Notepad++ လုပ်လိုက်ပါ။ ဒီလိုဆိုရင် notepad++ program တက်လာပါမယ်။ တက်လာတဲ့အခါမှာ Language မှာ PHP ကို ရွေးပေးပါ။ ပြီးရင်save as လုပ်ပြီးတော့ နာမည်တစ်ခုနဲ့ သိမ်းလိုက်ရင် php file တစ်ခုကို ရလာမှာဖြစ်ပါတယ်။ default အနေနဲ့ PHP ရဲ့ open and close tag တွေကို အနီနဲ့ ဖော်ပြပြီး comment တွေကို အစိမ်းနဲ့ပြပါတယ်။ ပြင်ချင်တယ်ဆိုရင်တော့ settings/style configura-tor မှာ ပြင်လို့ရပါတယ်။

## Chapter (2) Creating PHP pages using PHP

ဒီအခန်းမှာတော့ PHP ရဲ့ အခြေခံအချက်အလက်များကို ပြောသွားမှာဖြစ်ပြီး တစ်ချိန်တည်းမှာဘဲ ပြောသွားတဲ့အချက်အလက်တွေနဲ့ web တစ်ခုကို အဆင့်ဆင့် တည်ဆောက်သွားမှာဖြစ်ပါတယ်။ အားလုံးပြီးသွားတဲ့အခါမှာ Movie review တွေရယ် ၊ ဒီ web ကို လာတဲ့ လူတွေရဲ့ comment တွေရယ် movie details တွေကို MySQL နဲ့ တွဲပြီး ပြနိုင်မှာဖြစ်ပါတယ်။ အသေးစား dynamic web page တစ်ခု တည်ဆောက်တယ်လို့ မှတ်ယူလို့ရပါတယ်။ သင့်အနေနဲ့ PHP program ကို ဒီအတိုင်း သီအိုရီ ချည်း သက်သက်လေ့လာ နေရင် ပျင်းမှာစိုးပါတယ်။ ကိုယ် သိထားသလောက်က ဘယ်လောက်ထိ ရေးလို့ရသလဲ ဘယ်လောက်ထိ ဖန်တီးလို့ရသလဲဆိုတာကို သိထားဖို့လိုပါတယ်။ ခု အခြေခံအချက်အလက်တွေနဲ့ တင် သင် ထင်မှတ်မထားလောက်အောင် လုပ်နိုင်မှာကို တွေ့ရှိရမှာပါ။ ကျွန်တော်ဟာ ဒီစာအုပ်နဲ့ အတူ CD တစ်ချပ် တွဲထည့်ပေးထားပြီးတော့ အဲဒီ CD ထဲမှာ Exercise files တွေပါတယ်။ အကြံပြုလိုတာကတော့ အဲဒီ CD ထဲက ဖိုင်တွေကို တတ်နိုင်သမျှ နားလည်အောင်ကြည့်ပါ။ copy paste လုပ်ပြီး သင်မလေ့ကျင့်ပါနဲ့။ Programming ဆိုတာမျိုးကလည်း တစ်ခါရေးနားမလည် နှစ်ခါရေး ရင် ရူးသွားတာမျိုးမဟုတ်ပါဘူး။ ထပ်ခါထပ်ခါရေးလေ ပို နားလည်လေ ဖြစ်စေတာပါ။

PHP ကို အခြား language တွေဖြစ်တဲ့ HTML တို့လို text editor နဲ့ ရေးလို့ရပါတယ်။ ဒါပေမယ့် HTML နဲ့ မတူတာက PHP မှာက .php ဆိုတဲ့ extension နဲ့ အဆုံး သတ်တာဖြစ်ပြီး Web စာမျက်နှာ ပေါ်တာတွေက HTML တွေဖြစ်တယ်ဆိုတာ မှတ်ထားဖို့လိုပါတယ်။

### How PHP fits with HTML

#### The Rules of PHP syntax

PHP ကို စရေးပြီဆိုရင် ထိပ်ဆုံးမှာ <?php နဲ့ စပြီးတော့ နောက်ဆုံးမှာ ?> အဆုံးသတ်ရပါတယ်။ ယျေဘူယျအားဖြင့် PHP statement တွေရဲ့ အဆုံးကို semicolon (;) နဲ့ အဆုံးသတ်ပေးရပါတယ်။ အောက်က ဥပမာကို ကြည့်ပါ။

```
<?php
    $num = 1+2;
    echo $num;
?>
```

ကိုယ်ရေးနေတဲ့ ပရိုဂရမ်ဟာ တဖြည်းဖြည်းရှည် လာပြီးရှုပ်ထွေးမှုရှိလာရင် နောက်တစ်ခါ ပြန်ကြည့်တဲ့အခါ ကိုယ်ရေးထားတဲ့ဟာကို ကိုယ့်ဟာကို ပြန်မမှတ်မိ ဖြစ်နေတတ်ပါတယ်။ ဒါကြောင့် စတင်ရေးပြီးဆိုကတည်းက ဘယ် အကြောင်းအရာ ကို ဘာကြောင့်ရေးထားတယ်ဆိုတာကို ပရိုဂရမ်ထဲမှာ ထည့်ရေးထားဖို့ အကြံပြုလိုပါတယ်။ ဒါကို Comment ရေးတယ်လို့လဲခေါ်ပါတယ်။ Comment ရေးရင် တစ်ကြောင်းတည်းဆိုရင် double forward slashes (//) ကို အသုံးပြုပြီးရေးရပါတယ်။

```
ဥပမာ-
<?php
    // Adding number
    $num = 1+2;
    echo $num;
?>
```

ဒီလို ရေးတဲ့အခါမှာ comments တွေကို ပရိုဂရမ်က ထည့် မ run ပါဘူး။ တကယ်လို့ ကိုယ်မှတ်ထားတာက ငါးကြောင်း လောက်ဖြစ်သွားမယ်ဆိုရင် /\* comment \*/ ကို သုံးလို့ရပါတယ်။ ဒါတွေကို မှတ်ထားရင်ရပါပြီ။ ကဲ... အခု ဆိုရင် သင်ဟာ PHP ကို ဘယ်လို စရေးရမယ်ဆိုတာ သိပြီးသွားပြီမဟုတ်လား။ သိပ်မခက်ပါဘူး။

**The importance of coding practices**

သင်ဟာ ပရိုဂရမ် တစ်ခု စရေးတော့မယ်ဆိုရင် Code structure တွေကို အရင် ဆုံးလေ့လာ သင် ပါတယ်။ သင်ရေးလိုက်တဲ့ Code တွေဟာ Browser ပေါ်မှာ ကောင်းမွန်စွာအလုပ်လုပ်နေသမျှ သင့်ရဲ့ code structure ကို သိပ်ဂရုစိုက်စရာမလိုပါဘူး။ တကယ်လို့ အဲသည်လိုမဟုတ်တော့ဘဲ တစ်ခုခု error ပြု နေပြီဆိုရင် Code error ကို ရှာရတော့မယ်။ အဲဒီအခါမှာစရေးကတည်းက ရှုပ်ထွေးနေအောင်ရေးခဲ့မယ်ဆိုရင် အလွယ်တကူ ရှာဖို့ ခက်ခဲမှာပါ။ အောက်က နမူနာပုံစံကိုကြည့်ကြည့်ပါ။



Example(2) ကိုကြည့်လိုက်ရင် ပိုရှည်တယ်ထင်ပေမယ့် တကယ်တမ်းကြတော့ syntax error or other troubleshooting လုပ်တဲ့အခါမှာ ပိုမိုမြန်ဆန်စွာလုပ်နိုင်ပါတယ်။

ဒါဟာ ပရိုဂရမ်ကို ခုမှ စရေးမယ့် သူတွေအတွက်တကယ်ကို အရေးကြီးပါတယ်။ ကြည့်လိုရေးလိုက်ရင် အကျင့်ကြီးပါသွားတတ်ပါတယ်။

Example (1)

```

< ?php
    //check to make sure the first name is equal to MgMg before granting access

    if ($_POST['fname'] == 'MgMg') {
        echo '<p> Hi ' . $_POST['fname'] . '</p>';
    } else {
        echo '<p> Your name isn't MgMg so you cannot enter the web site. </p>
    }
?>

```

Example(2)

```

< ?php
    //check to make sure the first name is equal to Joe before granting access
    if ($_POST['fname'] == 'Joe')
    {
        echo '<p>';
        echo 'Hi';
        echo $_POST['fname'];
        echo '</p>';
    }
    else
    {
        echo '<p>';
        echo 'Your name\'s not MgMg so you cannot enter the web site!';
        echo '</p>';
    }

```

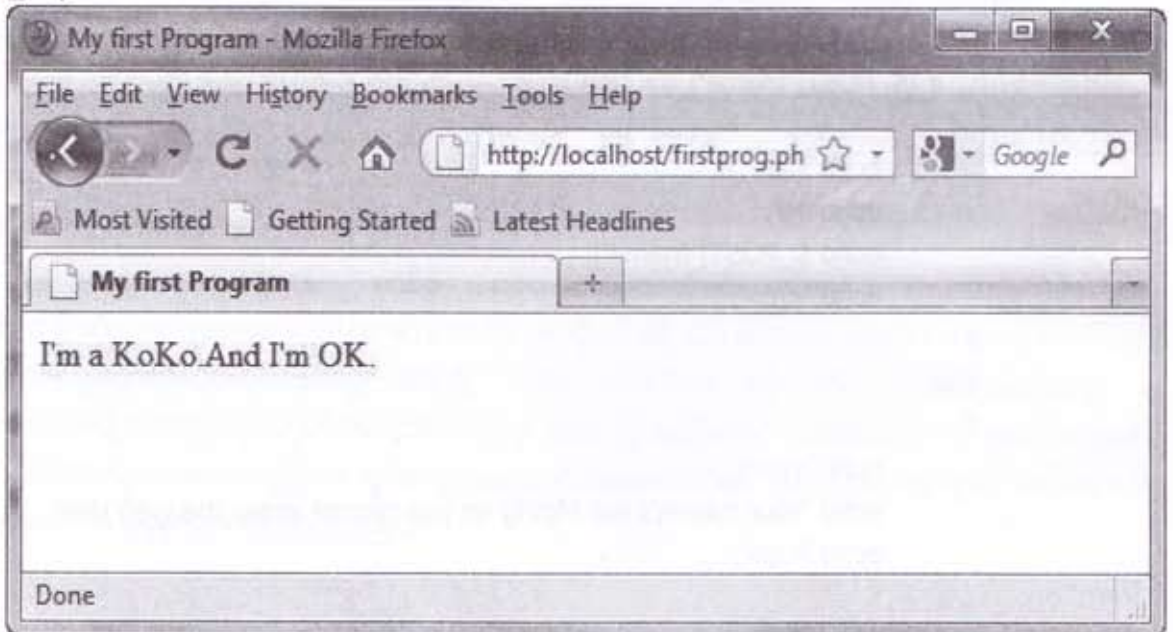
## Creating your first program

Using (echo)

1. အောက်မှာ ပြထားတဲ့ ကုဒ်တွေကို notepad++ ထဲ ကူးထည့်ပြီး firstprog.php ဆိုပြီး Save လုပ်ပါ။

```
<html>
  <head>
    <title>My first Program</title>
  </head>
  <body>
    <?php
      echo "I'm a KoKo.And I'm OK.";
    ?>
  </body>
</html>
```

OK . အဲဒီ firstprog.php ဆိုတဲ့ ဖိုင်ကို web browser ကနေ ကြည့်ကြည့်ရင် အောက်ကပုံအတိုင်း မြင်ရပါမယ်။



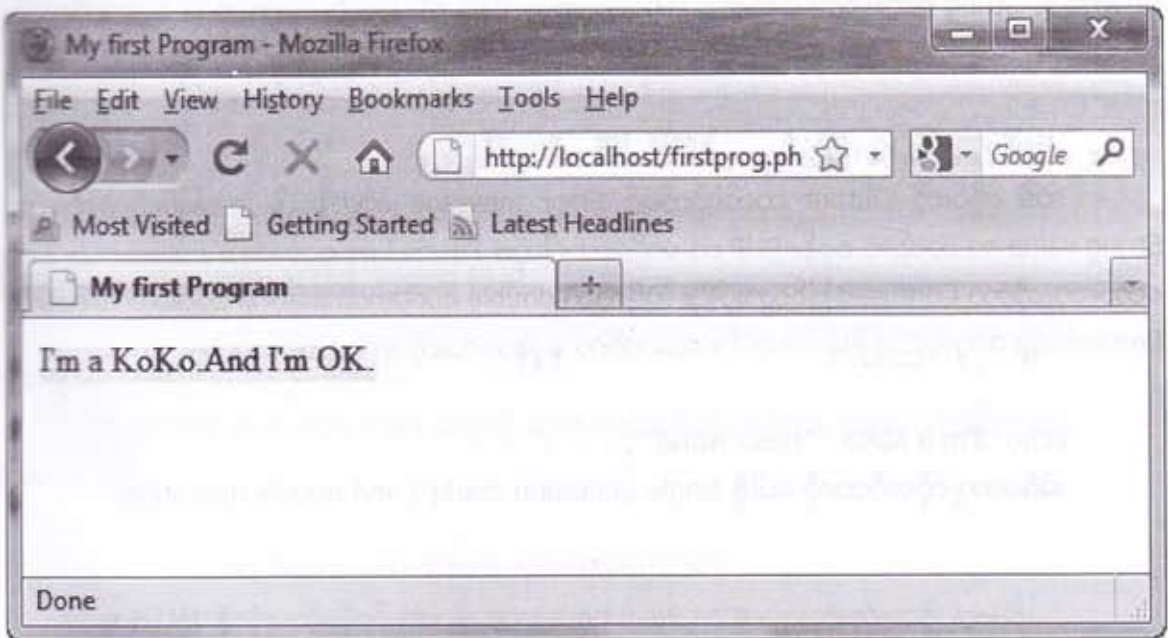
ကဲ အခု နောက်ထပ် တစ်ကြောင်းလောက် ထပ်ပေါင်း ရေးကြည့်ရအောင်။ ထပ်ပေါင်းရေးတဲ့ စာကြောင်းကို Highlight လုပ်ပေးထားပါတယ်။

```

<html>
  <head>
    <title>My first Program</title>
  </head>
  <body>
    <?php
      echo "I'm a KoKo.";
      echo 'And I\'m OK.';
    ?>
  </body>
</html>

```

File ကို save လုပ်ပြီးတော့ Browser ကနေ ပြန်ကြည့်ကြည့်ပါ။ အောက်ပါ ပုံအတိုင်း မြင်ရပါလိမ့်မယ်။





How it works?

Browser ကနေ PHP program ကို call လုပ်လိုက်ရင် အရင်ဆုံး PHP code တွေကို line by line ရှာပါတယ်။ ဆိုလိုတာက <?php ?> ကြားက ကုဒ်တွေကို ရှာတာပေါ့။ တစ်ချိန်တည်းမှာဘဲ အဲဒီ code တွေကို process လုပ်ပါတယ်။ ဆာဗာကို PHP code တွေဟာ တစ်ကြောင်းတည်း ပုံစံနဲ့ သွားတာပါ။ ဆာဗာဆီကို PHP code တွေဟာ မှန်မှန်ကန်ကန်ရောက်သွားပြီဆိုရင် ကျန်တဲ့ HTML code တွေကို browser ဆီပို့ပေးပါတယ်။ တစ်ခါတည်း php sections တွေပါ ပို့ပေးတာပါ။

အထက်က ပရိုဂရမ် မှာ Coding ပိုင်းမှာ သတိထားမိမယ်ဆိုရင် single quotation marks (') လေးကို သတိထားမိမှာပါ။

Text string တွေကို ဖော်ပြရာမှာ single quotation mark(') နဲ့ double quotation mark("") ကို လိုအပ်ချက်ပေါ်မူတည်ပြီး သုံးသွားရမှာပါ။ နောက်ပိုင်း သင်ခန်းစာတွေမှာ ပရိုဂရမ် ရေးရင်း ရေးရင်း ပိုမိုလေ့လာနိုင်ပါတယ်။ အောက်က နမူနာ ပုံစံလေးကို ကြည့်ကြည့်ပါ။

```
echo " I'm a KoKo";
```

အဲဒီဟာကို output အနေနဲ့ ထုတ်လိုက်ရင် I'm a KoKo ဆိုပြီး ရလာမှာပါ။ ဘာဖြစ်လို့လဲဆိုတော့ text string ရဲ့ အစနဲ့ အဆုံး ကို " နဲ့ ခံပြီး သုံးထားလို့ပါ။ double quotation mark (") နှစ်ခုကြားမှာ ရှိတဲ့ဟာက String တွေဖြစ်တယ်ဆိုတာ PHP က သူ့အလိုလို သိပါတယ်။ အောက်က နမူနာကိုလည်း ကြည့်ပါဦး။

```
echo 'I'm a KoKo';
```

အဲဒီ လိုင်းကို Output ထုတ်လိုက်ရင် Error message ရလာမှာပါ။ ဘာဖြစ်လို့လဲဆိုတော့ String value က ဘယ်က စမှန်း PHP က မသိတော့ပါဘူး။ I'm က ' က စတာလား။ ဒါမှမဟုတ် 'I က စတာလားဆိုတာ Confused ဖြစ်သွားပါပြီ။ အဲဒီလိုဘဲ double quotation mark နဲ့ စပေမယ့် အလားတူ Problem မျိုး တွေလည်းဖြစ်နိုင်ပါတယ်။ အောက်က နမူနာလေးကို လေ့လာကြည့်ပါ။

```
echo "I'm a KoKo . "Hello world"";
အဲဒီတော့ လိုအပ်သလို ပေါ်ဖို့ single quotation mark(') and double quotation
```

mark (") တွေကို လိုအပ်သလို အသုံးပြုရမှာပါ။ အထက်ပါ code ကို မှန်မှန်ကန်ကန် Run ဖို့ အောက်ပါအတိုင်း ရေးရင် ရပါတယ်။

```
echo "I'm a KoKo . \"Hello world\"";
```

\ သုံးထားတာကို သတိပြုမိမှာပါ။ ဘာအတွက် သုံးတာလဲဆိုရင် ဒီမှာက Hello world ချဲ့ရှေ့နှဲ့နောက်က (") ကို plain character အဖြစ် ယူလိုတဲ့အတွက်ဘဲဖြစ်ပါတယ်။ အခုနေ ရှုပ်နေမယ်ဆိုတာ သိပါတယ်။ နောက် အခန်းတွေ ရောက်လာရင် ခု လေ့ကျင့်ခန်းတွေက ထမင်းစားရေသောက် တာထက်ကို ပိုမို လွယ်ကူ သွားစေမယ်ဆိုတာ အာမခံပါတယ်။

**Integrating HTML with PHP**

အောက်မှာ ပြထားတဲ့ နမူနာလေးတွေကို ကြည့်လိုက်ရင် HTML ကို PHP ထဲ ထည့်သုံးရတာ ဘယ်လောက် လွယ်ကူပြီး လှပ တယ်ဆိုတာ ကိုယ်တိုင် လုပ်ကြည့်လိုက်ပါ။

```
<html>
  <head>
    <title>My first Program</title>
  </head>
  <body>
    <?php
      echo " <h1>I'm a KoKo.</h1>";
      echo " <h2>I'm a KoKo . \"Hello world\"</h2>";
    ?>
  </body>
</html>
```

**Considerations with HTML inside PHP**

HTML ကို PHP ထဲမှာ ထည့်သုံးတော့မယ်ဆိုရင် အောက်မှာ ပြထားတဲ့အချက်တွေကို သေချာလေး ဂရုစိုက်စေချင်ပါတယ်။

၁။ double quotation mark တွေကို သုံးရာမှာ ဂရုစိုက်ပါ။ back slash (\) ခံပြီးသုံးပါ။

ဥပမာ-

```
echo "<p style=\"font-size:80%;\">";
```

၂။ HTML နဲ့သုံးမယ်ဆိုရင် single quotation mark ကို အများဆုံးသုံးသင့်ပါတယ်။

ဥပမာ

```
echo '<p style="font-size:80%;">';
```

၃။ PHP code တွေထဲကို HTML code တွေ တအားများနေပြီဆိုရင် သတိထားပါ။ code တွေရောထွေးပြီး ရှုပ်ထွေး နေတတ်ပါတယ်။ အဲဒီအခါမှာ ကိုယ့်ရဲ့ HTML code တွေကို Strictly ဖြစ်အောင် ပြန်စဉ်းစားပြီး ပြန်ရေးပါ။ အောက်က ဥပမာ ၂ ခုကို နှိုင်းယှဉ်ကြည့်ရင် သိနိုင်ပါတယ်။ Example(1)

```
<?php
echo ' <table style="font-family: Arial,sans-serif; font-size: 80%; ;
echo 'width: 100%;">';
echo ' <tr> ';
echo ' <td style="width: 50%;">';
echo 'First Name:';
echo ' </td > ';
echo ' <td style="width: 50%">';
echo $_POST['fname'];
echo ' </td > ';
echo ' </tr > ';
echo ' </table > ';
?>
```

Example(2)

```
<table style="font-family: Arial,sans-serif; font-size: 80%; width: 100%;">
<tr>
<td style="width: 50%;">
First Name:
</td >
<td style="width: 50%" >
<?php echo $_POST['fname'];?>
</td>
</tr>
</table>
```



အထက်က ဥပမာ ၂ ခုကို ယှဉ်ကြည့်မယ်ဆိုရင် အဓိက ရည်ရွယ်ချက်က \$\_POST['fname'] ဆိုတဲ့ ဟာကို ထုတ်ပေးဖို့ဘဲဖြစ်တယ်ဆိုတာ တွေ့ရမယ်။ ဒီလိုပုံစံမျိုးမှာ HTML code အားလုံးကို PHP ပုံစံ ထဲ ထည့်ရေးမယ်အစား(Example 1လို) Example 2 လို HTML ထဲမှာ လိုအပ်တဲ့ PHP code ကိုဘဲ ထည့်ရေးရင် ပိုပြီးရိုးရှင်းပါတယ်။ ဆာဗာ အတွက်ကတော့ အထက်က Example ၂ ခုထဲက ကြိုက်တဲ့ဟာနဲ့ ရေး ။ အရေးမကြီးဘူး။ ပရိုဂရမ်မာတွေအတွက်ဘဲကိုယ်ရေးလိုက်တဲ့ဟာ Format ကျကျ ရေးချင်တယ်။ debugging လုပ်ဖို့ လွယ်လွယ်ဖြစ်ချင်တယ်ဆိုရင် example 2 လိုရေးဖို့ အကြံပြုလိုပါတယ်။

**Using Constants and Variables to add Functionality**

ခုဆိုရင် echo function ဘယ်လို အလုပ်လုပ်သလဲဆိုတာ နားလည်သွားပြီလို့ယူဆပါတယ်။ echo function ဟာ HTML နဲ့ တူတူပါဘဲ။ ခု ရှင်းပြမယ့် Constants and Variables တွေကို ထည့်သုံး အသုံးပြုတတ်မှ တကယ် True power of PHP ကို ရရှိမှာဖြစ်ပါတယ်။

**Overview of constants**

Constant ဆိုတဲ့ သဘောတရားကို ကျောင်းသားတိုင်း သိကြမှာပါ။ သူ့ကို အသုံးပြုမယ်ဆိုရင် (၁) မသုံးခင်မှာ သူ့ရဲ့ တန်ဖိုးကို ကြိုတင်သတ်မှတ်ဖို့လိုပါတယ်။

(၂) Constant တွေကို နာမည်ပေးတဲ့အခါမှာ အစဇာလုံးတွေကို letter (A to Z, a to z) or \_(underscore) နဲ့စရပါတယ်။

(၃) Constant နာမည်တွေကို Number နဲ့ စလို့ မရပါဘူး။

(၄) Constant တွေဟာ case-sensitive ဖြစ်ပါတယ်။ သိသာထင်ရှားအောင် စာလုံး အကြီးတွေနဲ့ ရေးလေ့ရှိပါတယ်။

(၅) Constant တွေကို PHP function တစ်ခုဖြစ်တဲ့ define() နဲ့ သတ်မှတ်ပြီးတော့ သတ်မှတ်ပြီးသွားရင် သူ့ကို ပြန်ပြောင်းလို့မရပါဘူး

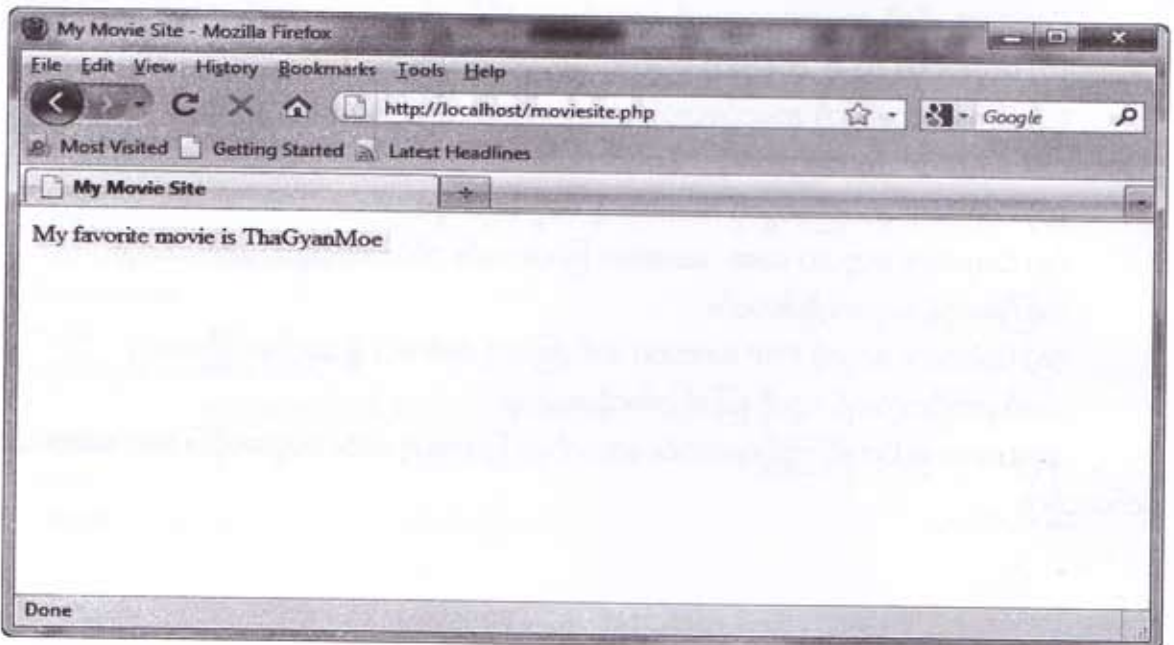
နမူနာလေး လေ့ကျင့်ကြည့်ရအောင်။ အောက်မှာ ပြထားတဲ့ code တွေအတိုင်း text editor ထဲ ရိုက်ထည့်ပါ။

```

<html>
  <head>
    <title> My Movie Site </title>
  </head>
  <body>
    <?php
      define ('FAVMOVIE', 'ThaGyanMoe');
      echo 'My favorite movie is ';
      echo FAVMOVIE;
    ?>
  </body>
</html >

```

အဲဒီ ပရိုဂရမ်လေးကို လေ့လာမယ်ဆိုရင် FAVMOVIE ဆိုတဲ့ constant ကို သတ်မှတ်ပြီး သူ့ရဲ့ တန်ဖိုးကို ThaGyanMoe လို့ ထားလိုက်တာပါ။ အဲဒီတော့ ဘယ်လိုဖြစ်သွားမလဲဆိုရင် define('name','value') ဖြစ်သွားတာပါ။ အောက်က ပုံအတိုင်း Result ထွက်လာပါမယ်။



### Overview of Variables

Constant ရဲ့ ပြောင်းပြန်က Variable ပေါ့။ Variable တွေရဲ့ တန်ဖိုးဟာ ပြောင်းနိုင်တယ်။ Variable တွေကို သုံးဖို့ define သတ်မှတ်ပေးစရာမလိုသလို declare လုပ်ဖို့လည်း မလိုပါဘူး။ Variable တွေဟာ ဘာနဲ့ တူသလဲဆိုရင် information တွေကို သိမ်းထားတဲ့ container တစ်ခုပါဘဲ။ လိုတဲ့နေရာကနေ ခေါ်သုံးနိုင်တယ်။ ပြီးတော့ အဲဒီ information ကိုလည်း ထပ်ပြောင်းနိုင်တယ်။ Variable တွေကို ရေးတော့မယ်ဆိုရင်

(၁) \$ နဲ့ စရမယ်။

(၂) Variable တွေဟာ case-sensitive ဖြစ်ပါတယ်။ (\$dateEntered နဲ့ \$Dateentered) တို့ရဲ့ တန်ဖိုးဟာ မတူပါဘူး

(၃) Variable name တွေရဲ့ အစဟာလည်း letter(A to Z, a to z) or \_ (underscore) ဖြစ်ရပါမယ်။

(၄) Variable name တွေရဲ့ အစဟာ number မဖြစ်ရပါဘူး (0-9)။

### Using Variables

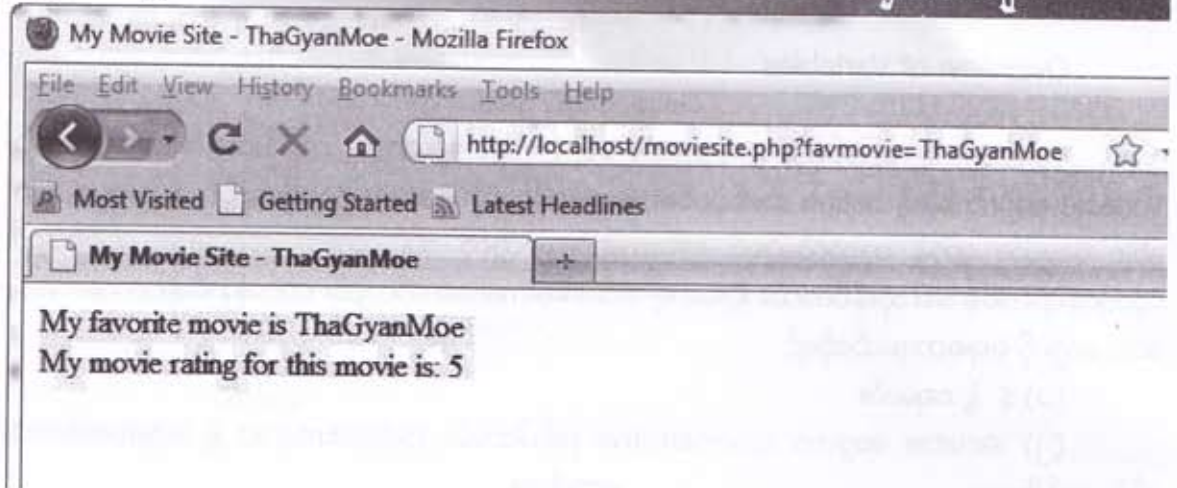
ခုနက moviesite.php ကို အနည်းငယ် ပြင်ရေးကြည့်ပါမယ်။ Variables လေး ထည့်သုံးကြည့်မှာပေါ့။ ပြင်ထားတဲ့ စာကြောင်းတွေကို Highlight ပြထားပါတယ်။

```

<html>
  <head> <title> My Movie Site </title> </head>
  <body>
    <?php
      define ('FAVMOVIE', 'ThaGyanMoe');
      echo 'My favorite movie is ';
      echo FAVMOVIE;
      echo '<br/>';
      $movierate = 5;
      echo 'My movie rating for this movie is: ';
      echo $movierate;
    ?>
  </body>
</html >

```





### How it works

ပထမဆုံး \$movierate ဆိုတဲ့ variable ရဲ့ တန်ဖိုးကို 5 လို့ သတ်မှတ်လိုက်ပါတယ်။ Number တွေကို တန်ဖိုးသတ်မှတ် ရင် single quotation mark or double quotation mark မလိုပါဘူး။ တကယ်လို့ သင်ဟာ အောက်ပါအတိုင်း ရေးမယ်ဆိုရင် movierate တန်ဖိုးဟာ ဂဏန်း 5 မဟုတ်ဘဲ Character 5 ဖြစ်သွားမှာပါ။

```
$movierate='5';
```

ဒီတော့ တန်ဖိုးတွေကို သတ်မှတ်ရင် integer သတ်မှတ်ပြီး ထားရင် နောက်ပိုင်း mathematical calculation လုပ်ရင် ပိုမိုလွယ်ကူပါလိမ့်မယ်။ အောက်မှာ ပြထားတဲ့ နမူနာကို ကြည့်လိုက်ရင် သိသာပါတယ်။

```
<?php
    $mamamovierate = 5;
    $nyinyimovierate = 7;
    $fathermovierate = 2;
    $mothermovierate = 1;
    $avgmovierate = (($mamamovierate + $nyinyimovierate +
                    $fathermovierate+ $mothermovierate) / 4);
    echo 'The average movie rating for this movie is: ';
    echo $avgmovierate;
?>
```

ဒီပရိုဂရမ်လေးကို လေ့လာကြည့်ရင် ရှိရှိရှင်းရှင်းလေးပါဘဲ။ သင်္ချာ ဖော်မြူလာအတိုင်း ဝုဇာပေါင်းကို အရေအတွက် နဲ့ စားလိုက်ရင် average value ကို ရတာဘဲလေ ။ မဟုတ်ဘူးလား။ တစ်ခု သတိထားရမှာက () တွေကို ရေးရင်ဖြစ်ပါတယ်။ ( ဖွင့်ပြီးရင် ပြန်ပိတ် ) ရပါတယ်။ + -\*/ တွေပါလာပြီဆိုတော့ ဒီအကြောင်းကို နောက်ပိုင်းမှာ ဆက်ရှင်းပြပါမယ်။ ခု PHP ထဲမှာ ပါတဲ့ built-in mathematical functions တွေကို နည်းနည်းလောက် တီးခေါက် ကြည့်ရအောင်။

rand([\$min , \$max]) : Returns a random number.

ceil(\$value) : Returns the next highest integer by rounding the value upwards.

floor(\$value) : Returns the next lowest integer by rounding the value downwards.

number\_format(\$number[\$decimal\_places[\$decimal\_point, \$thousands\_sep]])

အောက်မှာ ပြထားတဲ့ နမူလေးနဲ့ ယှဉ်တွဲ လေ့လာကြည့်ပါ။ ပြီးရင် ကိုယ့်ဟာကို တစ်ခြား တန်ဖိုးတွေကို ရေးပြီး လေ့ကျင့်ပါ။ Praticce makes perfect တဲ့။

rand([\$min , \$max]) : Returns a random number.

ceil(\$value) : Returns the next highest integer by rounding the value upwards.

floor(\$value) : Returns the next lowest integer by rounding the value downwards.

number\_format(\$number[\$decimal\_places[\$decimal\_point, \$thousands\_sep]])

```
$price = 12345.67
```

```
number_format($price); // returns 12,345.67
```

```
number_format($price,2,','); // returns 12345.67
```

အခုက အမြည်းလေးကျေးထားတဲ့သဘောပါဘဲ။ နောက်ပိုင်းမှာ လက်တွေ့လေ့ကျင့်ခန်းတွေ လုပ်ရင်းနဲ့ ပိုပြီး နားလည် လာပါမယ်။ သက်ဆိုင်ရာ နေရာရောက်ရင် အသေးစိတ်ရှင်းပြမှာပါ။

**Passing variables between pages**

တကယ်လို့ သင့်ရဲ့ website ထဲကို user တွေကို ဝင်ခွင့်ပေးမယ်ဆိုရင် login form လိုလာပြီပေါ့။ Login လုပ်ပြီးသွားရင် သင့်အနေနဲ့ ဝင်လာတဲ့ user ကို နှုတ်ဆက်ချင်တယ်။ အဲဒီ နှုတ်ဆက်စာကိုလည်း user က ဘယ် စာမျက်နှာကိုဘဲ သွားသွား ပေါ်နေစေချင်တယ်ဆိုရင် ပေါ်စေချင်တဲ့ ဟာကို variable တစ်ခု သတ်မှတ်ပြီး အဲဒီ variable ကို စာမျက်နှာတစ်ခုကနေ နောက်တစ်ခုကို passing လုပ်ဖို့ လုပ်ရမှာဘဲဖြစ်တယ်။ အဲဒီလို pass လုပ်ဖို့ အခြေခံ အားဖြင့်ပေါ့ ၄ မျိုးရှိပါတယ်။

- (1) Pass the variables in URL
- (2) Through a session
- (3) Via a cookie
- (4) with an HTML form

အဲဒီ ထဲက ဘယ်ဟာကို သုံးမယ်ဆိုတာတော့ ကိုယ် ပြုလုပ်တဲ့ site ရဲ့ လိုအပ်ချက်ပေါ်မှာမူတည်ပြီးဘဲ ဆုံး ဖြတ်ကြပါတယ်။

**Passing Variables through a URL**

ဥပမာပေါ့ဗျာ -

```
http://www.mydomain.com/news/articles/showart.php?id=12345
```

သူက variables ကို URL ထဲကနေ ဖြတ်ပြီးဖော်ပြတာပါ။ showart.php ကို ခေါ်လိုက်ရင် articles ထဲက id နံပါတ် 12345 ကို ခေါ်တင်လိုက်တာပါ။

အဲဒီလို variable တွေကို URL ထဲမှာ ဖော်ပြတဲ့အခါမှာ နှစ်ခု ကို တစ်ပြိုင်တည်း လုပ်ချင်ရင် ampersand(&) သင်္ကေတကို သုံးပြီး နှစ်ခုကို ပေါင်းပြီး ဖော်ပြလိုရတယ်။ ဥပမာ-

```
http://www.mydomain.com/news/articles/showart.php?id=12345&lang=en
```



သို့မို့ သဘောတရားကတော့ id နံပါတ် 12345 ဖြစ်ရမယ့် အပြင် language ကလည်း en ဆိုတဲ့ english ဖြစ်ရမယ်ပေါ့ဗျာ။ ခုကတော့ ဒီလို လုပ်လို့ရတယ်ဆိုတာဘဲ မှတ်ထားဦးနော်။ ကိုယ်က ခုမှ လေ့လာတာဆိုတော့ အခြေခံလောက်တွေဘဲ လေ့လာနေတာလေ။ ဒီလို variable တွေကို URL ထဲမှာ ဖော်ပြတဲ့ နည်းက တကယ်တမ်း တော့ Security အပိုင်းက စဉ်းစားရင် တအားကို အားနည်းပါတယ်။ sensitive ဖြစ်တဲ့ information တွေကို ခုလိုသာ ပြပေးနေရင် Security အတွက် ဘယ်လိုလုပ် စိတ်ချနိုင်ရတော့မှာလဲ။ သူ့ရဲ့ အားနည်းချက်လေးတွေကို လေ့လာကြည့်ရအောင်။

- ၁။ ယူဇာအားလုံးက သင်သုံးထားတဲ့ variables တွေကို မြင်ရမယ်။
- ၂။ ယူဇာတွေဟာ URL ထဲက variable တွေကို ပြောင်းပြီးတော့ တစ်ခြား နည်းလမ်းတွေနဲ့ သင် မြင်ချင်တဲ့ information တွေကို ယူနိုင်တယ် ။

ဘာဘဲဖြစ်ဖြစ်ပေါ့။ security ပိုင်း အားနည်းတယ်ဆိုပေမယ့် တစ်ခြား သုံးလိုရတဲ့ နေရာလေးတွေလည်း ရှိသေးတယ်လေ။ ခုလိုမျိုး variable တွေကို PASS လုပ်ဖို့ PHP မှာ ရှိပါတယ်။ \$\_GET array ဖြစ်ပါတယ်။ URL မှာ ပေါ်မယ့် variable name ကို key အဖြစ် ယူလိုက်မှာပါ။ URL မှာ variable က id ဖြစ်တယ်ဆိုရင် \$\_GET['id'] ဖြစ်ပြီးတော့ နောက် variable က lang ဆိုရင် \$\_GET['lang'] ပေါ့။ ဒါဆိုရင် လေ့ကျင့်ခန်း နမူနာလေး လုပ်ကြည့်ရ အောင် ။ ခုနက ရေးခဲ့တဲ့ moviesite.php ကို နည်းနည်းလေး ပြင်ရေးမယ်ဗျာ။

```

<html>
  <head>
    <title> My Movie Site - <?php echo $_GET['favmovie'];
?> </title>
  </head >
  <body >
    <?php
      //delete this line: define('FAVMOVIE', 'The Life
of Brian');
      echo 'My favorite movie is ';
```

```

        echo $_GET['favmovie'];
        echo ' <br/> ';
        $movierate = 5;
        echo 'My movie rating for this movie is: ';
        echo $movierate;
    ?>
</body>
</html>

```

ပြင်ပြီးသွားပြီလား။ OK. ဒါဆိုရင် နောက်ထပ် document အသစ်တစ်ခုရေးမယ်။ text editor မှာ new page ခေါ်ပြီးတော့ အောက်ပါအတိုင်း ရေးပါ။ ပြီးရင် movie1.php ဆိုပြီး save လုပ်ပါ။

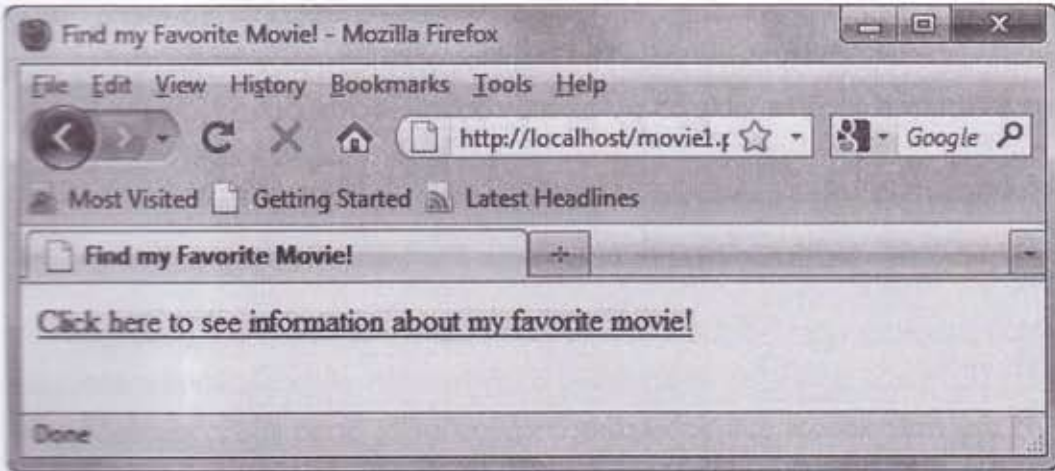
```

<html>
  <head>
    <title> Find my Favorite Movie! </title>
  </head>
  <body>
    <?php
      echo ' <a href="moviesite.php?favmovie=Stripes"> ';
      echo 'Click here to see information about my favorite
movie!';

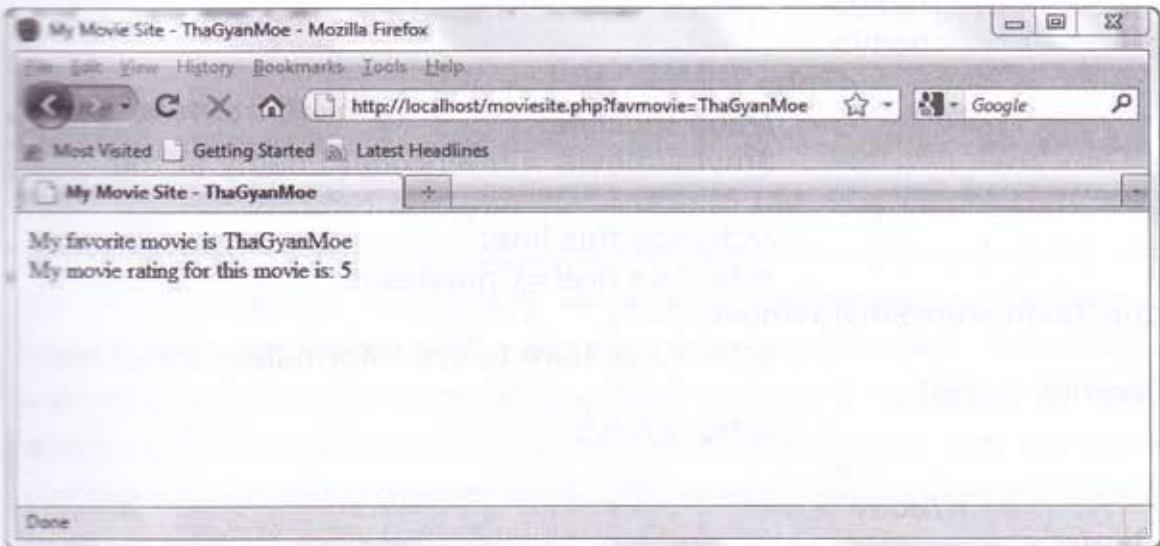
      echo ' </a> ';
    ?>
  </body>
</html>

```

ပြီးရင် browser ကနေ movie1.php ကို ခေါ်လိုက်ရင် အောက်က ပုံ အတိုင်းပေါ်လာ ပါမယ်။ ကျွန်တော် ဒီစာအုပ်မှာ HTML အကြောင်းကို အသေးစိတ်ရှင်း မနေပါဘူး။ PHP ကို လေ့လာနေပြီဆိုတော့ HTML ကိုလည်း အတော် အသင့် နားလည် နေပြီလို့ ယူဆပါတယ်။



ပုံမှာ မြင်တဲ့အတိုင်းဘဲ သူက click ပါတဲ့ အဲဒီတော့ click လိုက်ပေါ့။ ဒါဆိုရင် အောက်ကပုံအတိုင်း ထွက်လာမယ်။





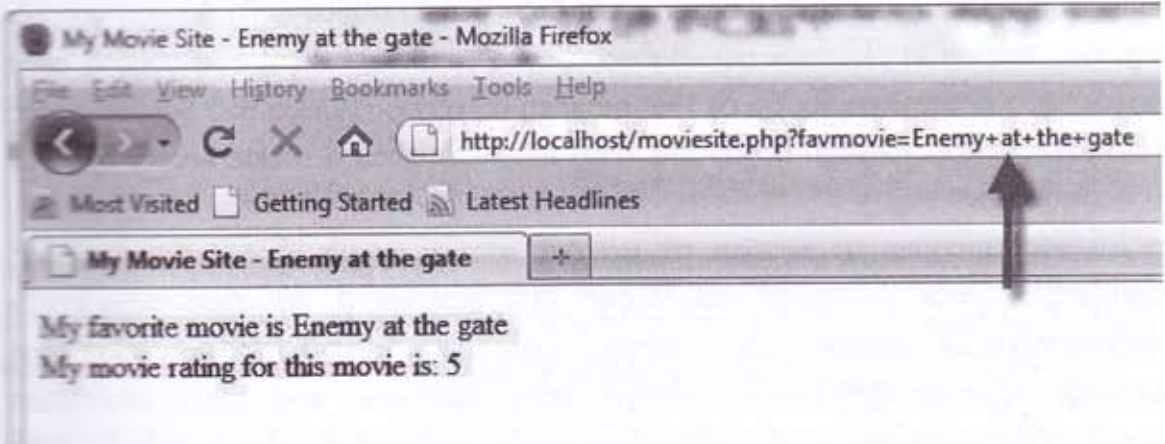
ပုံမှာမြင်ရတဲ့အတိုင်းဘဲ \$favmovie = TheGyanMoe ဖြစ်သွားတာပါ။ တစ်ခါတည်း browser ရဲ့ Tab ရဲ့ title ကိုလည်း ကြည့်လိုက်ပါဦး My Movie Site - ThaGyanMoe ဆိုပြီး မြင်ရမှာပါ။ ဘာဖြစ်လို့လဲဆိုတော့ title tag မှာ \$favmovie ဆိုတဲ့ variable ကို ခေါ်သုံးထားလိုက်လို့ဖြစ်ပါတယ်။

Special characters in URLs

URL မှာ variable တွေမှာပါတတ်တဲ့ special character တွေဖြစ်တဲ့ space, ampersands(&) စတဲ့ character များ ကို ဒီအတိုင်း ဖော်ပြရာမှာ problem ရှိပါတယ်။ ဒီလို ဟာတွေကို PHP မှာ ကျော်လွှားနိုင်ပါတယ်။ သူ့မှာပါတဲ့ urlencode() ဆိုတဲ့ function ကို သုံးပြီး special character problem ကို ဖြေရှင်းနိုင်ပါတယ်။ အခု နည်းနည်းလောက် ထပ်ပေါင်း လေ့ကျင့်ကြည့်ရအောင်။ movie1.php ကို အောက်က ကုန်တွေအတိုင်း ထပ် ပြင်ရေးကြည့်မယ်လေ။

```
<html>
  <head>
    <title> Find my Favorite Movie! </title >
  </head>
  <body>
    <?php
      //add this line:
      $myfavmovie = urlencode('Enemy at the
gate');
      //change this line:
      echo "<a href=\"moviesite.
php?favmovie=$myfavmovie\">";
      echo 'Click here to see information about my
favorite movie!';
      echo '</a>';
    ?>
  </body >
</html>
```

save လုပ်ပြီး ပြန် ဖွင့်ကြည့်ရအောင်။ ဒါဆိုရင် အောက်ပါပုံအတိုင်းမြင်ရမယ်။ URL box ထဲကို သေရာကြည့်ပါ။



ဒီစံနစ်ရစ်လေးမှာဆိုရင် \$myfavmovie = Enemy at the gate ဆိုပြီး ပေးခဲ့တာပါ။ word တွေကြားက space တွေကို မရှိစေချင်တဲ့အတွက် urlencode() ဆိုတဲ့ function ကို တစ်ခါတည်း တွဲပေးလိုက်တာပါ။ ခေတ်ပါဘူး။ ဟုတ်တယ်မလား။ ကဲ ..နောက်တစ်ခန်းကို ဆက်ကူး လိုက်ကြရအောင်။

### Passing variables with SESSIONS

ကျွန်တော် အထက်မှာ ပြောခဲ့တဲ့အတိုင်း variable ကို URL ကနေ passing လုပ်တဲ့အခါမှာ အရေးကြီးတဲ့ ဒေတာတွေ၊ saved လုပ်ထားတဲ့ old information တွေကို ယူသုံးလိုက်လို့ ကိုယ့် site အတွက် dangerous အခြေအနေကို မရောက်နိုင်တဲ့ အခြေအနေတွေမှာ သုံးလို့ရပါတယ်။ တကယ်လို့များ user တစ်ယောက်ရဲ့ personal data တွေကိုသာဆိုရင် အထက်ပါနည်းနဲ့ မသုံးသင့်ပါဘူး။ ကိုယ့်အကြောင်းကို ဘာဖြစ်လို့ လူတိုင်းကို ပေးသိရမှာလဲ ? မဟုတ်ဘူးလား။ Personal data တွေဆိုတာ username, password, address, email, phone number, credit card number စတာတွေပါပါတယ်။ အဲဒါတွေကို private အနေနဲ့ သုံးဖို့ Cookies ကို သုံးရပါတယ်။ နောက်ပိုင်းမှာတော့ cookies အကြောင်းကို ထပ်ပြီး ဖြည့်စွက်ဆွေးနွေးသွားပါမယ်။

what is a session? Session ဆိုတာ တကယ်တော့ variable ကို temporary သတ်မှတ်လိုက်တာဘဲဖြစ်ပါတယ်။ သူ့ရဲ့ သက်တမ်းက ဘယ်လောက်ခံလဲဆိုတော့ Browser ကို shutdown ချလိုက်တဲ့အထိပါဘဲ။ browser လည်း ပိတ် သူလည်း ပျောက် ပါဘဲ။ Session မှာလည်း သူ့ဟာသူ session id ပါပါတယ်။ ဒါမှ ကိုယ်လုပ်လိုက်တဲ့ (browser ပေါ်မှာ) အရာတစ်ခုကို သူက သူ့နံပါတ်နဲ့သူ မှတ်ထားတာဖြစ်ပါတယ်။ PHP program က လိုအပ်တဲ့အခါမှာ အဲဒီ Session တွေကို refer ပြန်လုပ်ပါတယ်။ Session တွေကို URL ထဲမှာ Pass လုပ်လို့ရသလို cookie နဲ့လည်း တွဲဖက်ပြီး လုပ်လို့ရပါတယ်။

PHP program မှာ session ကို ခေါ်သုံးဖို့ session\_start() ဆိုတဲ့ function လေးကို ခေါ်သုံးလိုက်ယုံပါဘဲ။ ဘယ်နားမှာ ရေးရလဲဆိုရင် PHP program ရဲ့ ထိပ်ဆုံးမှာ ရေးရပါတယ်။ တစ်ခုတော့ရှိတယ်။ session ထဲမှာ သင်က ဘယ် data တွေကို သိမ်းမှာလဲဆိုတာတော့ ပထမဆုံး စဉ်းစားထားဖို့လိုပါတယ်။ အဓိကက တော့ username and login information တွေကို သိမ်းတာများပါတယ်။ ကဲ ..ပြောနေကြပါတယ်။ ကျွန်တော်တို့ ပရိုဂရမ် တစ်ပုဒ်လောက် ချရေးလိုက်မယ်။ movie1.php ကိုဘဲ နည်းနည်း ထပ်ပေါင်းရေးကြည့်မယ်။ အောက်က အတိုင်းပေါ့ -

```
<?php
    session_start();
    $_SESSION['username'] = 'MgMg12345';
    $_SESSION['authuser'] = 1;
?>
<html>
    <head>
        <title> Find my Favorite Movie! </title >
    </head>
    <body>
        <?php
            $myfavmovie = urlencode('Enemy at the gate');
            echo " <a href=\"moviesite.
php?favmovie=$myfavmovie\"> ";
```



```

        echo 'Click here to see information about my favorite mov-
ie!';

        echo ' </a> ';

    ?>
</body>
</html>

```

နောက်ပြီးတော့ moviesite.php ကိုလည်း နည်းနည်းတော့ပြင်ရေးရမယ်လေ။ ဘာလို့ဆိုတော့ သူ့ဆီက information တွေ ယူရမှာကိုး ။ session သုံးပြီးဆိုတော့ သူက ဒီအတိုင်း မပေးတော့ဘူးလေ။ သတ်မှတ်ထားတဲ့ data တွေကို တူမတူ တိုက်စစ်မယ်။ တူရင်ပေးမယ်။ မတူရင် မပေးဘူးပေါ့။ အဲဒီလို စစ်ဖို့အတွက် moviesite.php ကို နည်းနည်းလေး ပြင်ရေးမယ်လေ။

```

<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['authuser'] != 1) {
    echo 'Sorry, but you don\'t have permission to view this page!';
    exit();
}
?>
<html>
<head>
<title> My Movie Site - <?php echo $_GET['favmovie']; ?> </title>
</head >
<body>
    <?php
        echo 'Welcome to our site, ';
        echo $_SESSION['username'];
        echo '! <br/>';
        echo 'My favorite movie is ';
        echo $_GET['favmovie'];
        echo ' <br/> ';
        $movierate = 5;
        echo 'My movie rating for this movie is: ';
        echo $movierate;
    ?>

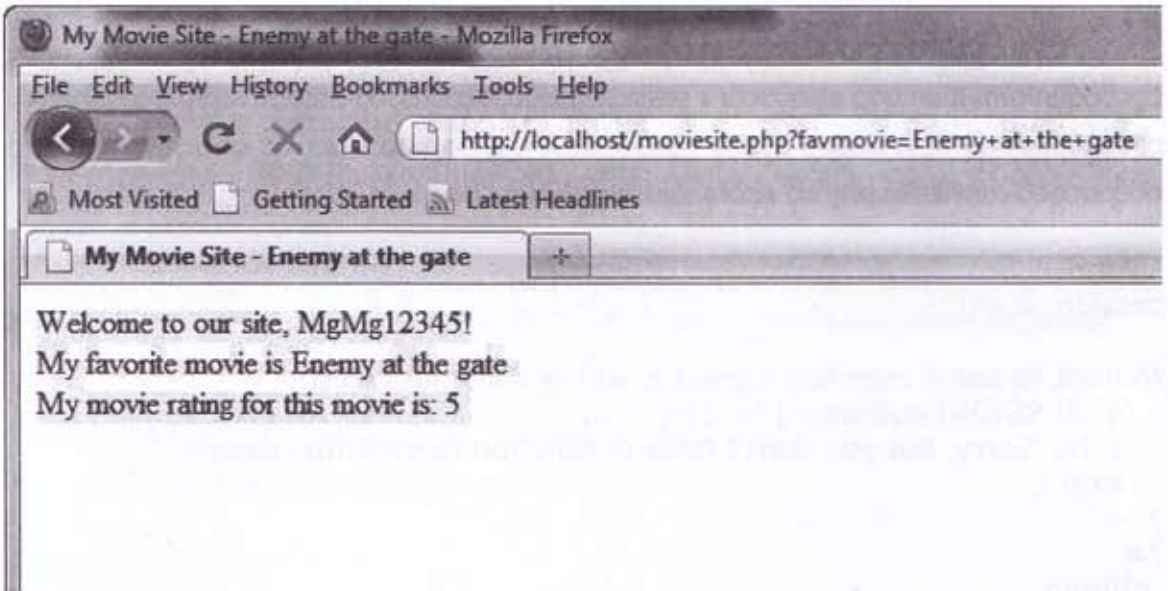
```

```

?>
</body>
</html>

```

Save လုပ်ပြီးတော့ movie1.php ကို ဖွင့်ပြီး click လုပ်လိုက်ရင် အောက်က ပုံအတိုင်းထွက်လာပါမယ်။



**How it works**

အရေးကြီးတာကို မမေ့နဲ့နော်။ session ကို ရေးမယ်ဆိုရင် ကိုယ်ရေးမယ့် script ရဲ့ ထိပ်ဆုံးမှာ ရေးရတယ်။ ခု ဒီ ပရိုဂရမ် မှာဆိုရင် movie1.php ကနေ request လုပ်လာတာကို moviesite.php ရဲ့ ထိပ်ဆုံးက session က if နဲ့ စစ်ပါတယ်။ if ရဲ့ အလုပ်လုပ်ပုံက အောက်ပါအတိုင်းဖြစ်ပါတယ်။ သူ့ရဲ့ condition က if(conditional statement){statement} ဖြစ်ပါတယ်။ ဥပမာ လေး တစ်ခု နဲ့ လုပ်ကြည့်မယ်လေ။ အစဆိုတော့ လွယ်လွယ်ကူကူပေါ့ဗျာ။

```

<?php
    if (1+3 ==4){
        echo 'Summation is true.';
    }
?>

```

အဲဒါကို run လိုက်ရင် browser ကနေ Summation is true ဆိုတာကို ပြမှာပါ။ အလုပ်လုပ်ပုံက  
 # ဆိုတဲ့အတိုင်း ဖြစ်လျှင်ပေါ့ ဗျာ... 1+3==4 ဖြစ်လျှင် Summation is true ဆိုတာကို echo နဲ့ output  
 ထုတ်ထားတာပါ။ 1+3==5 ဆိုရင် ဘာမှ ပြမှာမဟုတ်ပါဘူး။ ဟုတ်မှမဟုတ်တာကိုး။ ဘာဖြစ်လို့ ==  
 နှစ်ခါသုံးရတာလဲ ဆိုရင် PHP မှာ သုံးရမယ့် Comparison operators ကို ရှင်း ပြပါမယ်။

**Comparison operators**

- == :Values are equal
- === : Values are identical
- != :Values are not equal
- !== :Values are not identical
- < :value 1 is less than value 2
- > :value 1 is greater than value 2
- <= :value 1 is less than or equal to value 2
- >= :value 1 is greater than or equal to value 2

ဒီနေရာမှာ သေချာ မှတ်ထားရမှာက equal ဆိုတာနဲ့ identical ဆိုတာ မတူပါဘူး။ identical  
 မှာဆိုရင် variable ရဲ့ value ရော datatype ရော တူပါတယ်။ equal မှာကတော့ value ဘဲတူတာပါ။  
 အခု moviesite.php မှာ ပြင်ရေးလိုက်တဲ့ ဟာကို ပြန်ရှင်းပါမယ်။



if(\$\_SESSION['authuser']!=1) ဆိုတာက movie1.php file ကနေ request လုပ်လာတာမှာ authuser =1 မဟုတ်ဘူးဆိုရင် ဆိုတဲ့ အဓိပ္ပာယ်ရပါတယ်။ လိုရင်းပြောရရင် 1 ဖြစ်မှ ရမယ်ဆိုပေါ့ဗျာ။ 1 မဟုတ်ဘူးဆိုရင် {} ကွင်းထဲကဟာကို ဆက် Run မှာပါ။ အဲဒီမှာ သူက echo သုံးပြီး သင်က 1 မဟုတ်တဲ့အတွက် ဒီ page ကို ကြည့်ခွင့်မရှိဘူး ဆိုပြီး output ထုတ်ပေးတာပါ။ ပြီးရင် exit() function သုံးပြီး ပြန်ထွက်သွားပါတယ်။

တကယ်လို့ 1 ဖြစ်ခဲ့မယ်ဆိုရင် ကျန်တဲ့ code တွေကို ဆက် run မှာပါ။ အခုက 1 ဖြစ်နေတော့ ပထမဆုံး welcome to our site လာမယ်။ နောက်ပြီးရင် echo \$\_SESSION['username'] ကို ကျွန်တော်တို့က MgMg12345 လို့သတ်မှတ်ခဲ့တဲ့အတွက် MgMg12345 ဆိုပြီးပြမယ်။ ကျန်တာကတော့ ရှေ့က အတိုင်းဘဲ။

If , else if, while , do while, switch and break , For စတဲ့ statement တွေကို သူ့ရဲ့ သက်ဆိုင်ရာနေရာ ရောက်မှ ကျွန်တော် ရှင်းပြသွားပါမယ်။ Theory နဲ့ Pratical ကို တစ်ခါတည်း လုပ်သွားတာပေါ့။ ဒါမှမဟုတ်ရင် ခင်ဗျား ပျင်းသွားမှာပေါ့။



Passing variables with cookies

Cookies တွေမှာ ဘာတွေကို သိမ်းသလဲဆိုရင် website ကို လာရောက်တဲ့သူရဲ့ computer information အနည်းငယ်ကို သိမ်းပါတယ်။ Theory အရဆိုရင်တော့ Cookies တွေဟာ အသုံးပြုသူရဲ့ IP , operating system တွေကို intercept လုပ်နိုင်ပါတယ်။ ဒါပေမယ့် cookies တွေက storing information ဘဲလုပ်ပါတယ်။ တစ်ချို့သူတွေက တော့ ဒီ cookies technology နဲ့ အသုံးပြုသူရဲ့ browsing habits တွေ ကို track လုပ်ပါတယ်။ ပြီးတော့ ဒီ cookies တွေကို commonly name directory ထဲမှာ သိမ်းထားတာဖြစ်တဲ့အတွက် hacker တွေက hack လုပ်ပြီးတော့ သော်လည်းကောင်း တစ်ခြားသူက မိမိကွန်ပျူတာထဲက cookies တွေကို တစ်နည်းနည်းဖြင့်လာယူသွားလျှင်သော် လည်းကောင်း သင့်ရဲ့ အရေးကြီးဒေတာတွေ ပါသွားနိုင်ပါတယ်။ ဒါကြောင့် တစ်ချို့သူတွေကတော့ Browser က cookies ကို disable ပေးထားလေ့ရှိပါတယ်။ ဒီလိုမျိုး ဖြစ်ပျက်နိုင်တာတွေကြောင့် တကယ် အရေးကြီး ဒေတာတွေ (ဥပမာ - Credit card information ) စတာတွေကို ကွန်ပျူတာထဲမှာ သိမ်းဆည်း မထားဖို့ အကြံပေးလိုပါတယ်။

Cookies နဲ့ပတ်သတ်ပြီး ပိုသိချင်ရင် security risk တွေကို ပို သိချင်ရင် အောက်က website တို့ သွားရောက် လေ့လာနိုင်ပါတယ်။

[www.w3.org/Security/faq/wwwsf2.html#CLT-Q10](http://www.w3.org/Security/faq/wwwsf2.html#CLT-Q10)

ဒီလိုဆိုရင် ဘာဖြစ်လို့ developer သမားတွေက cookies ကို သုံးနေကြသေးလဲ။ cookies ကို သုံးခြင်းဖြင့် သတ်မှတ်ထားတဲ့ အချိန်ထိ information ကို သိမ်းထားလို့ရပါတယ်။ Session ကို သာသုံးမယ်ဆိုရင် Browser ကို ပိတ်လိုက်တာနဲ့ သူ့ရဲ့ life ကလည်း ကုန်ပြီလေ။ cookies ကတော့ အဲလို မဟုတ်ပါဘူး။ Session ကိုသာ cookies နဲ့ တွဲသုံးမယ်ဆိုရင် browser life ထက် ပို ကြာရှည်ခံအောင်သုံးနိုင်ပါတယ်။

ဟုတ်ပြီ။ ခု cookie ကို ဘယ်လို သတ်မှတ်မလဲ။ လွယ်ပါတယ်။ setcookie() ပေါ့။ ဒါလည်း PHP function တစ်ခုပါဘဲ။ cookie တစ်ခု သတ်မှတ်မယ်ဆိုရင် အောက်က အချက်တွေကို အရင်ဆုံး ဧည့်စား ဖို့လိုပါတယ်။

1. Cookie name
2. Value of the cookie name (eg. username)
3. Cookie life time (in seconds) UNIX timestamp ပေါ်မှာ အခြေခံထားပါတယ်။

ဒါပေမယ့် ခုလို ရေးလို့ရပါ တယ်။ time()+60\*60\*24\*365 ဒါဆိုရင် သူ့ရဲ့ သက်တမ်းက ဝ နှစ်တာဖြစ်ပါတယ်။

4. Path ( cookie ကို သိမ်းမယ့်နေရာ)

5. Domain ( ဒီ cookie ကို လက်ခံနိုင်တဲ့ domain) (ဒီဟာက optional ဖြစ်ပါတယ်။ မသတ်မှတ်လည်းရတယ်)

6. Cookie မှာ secure connection HTTPS ကို on ပေးထားရမယ်။

အဲဒီတော့ ဒီလိုပုံစံဖြစ်မှာပေါ့။

```
setcookie ($name[, $value[, $expire[, $path[, $domain[, $secure]]]])
```

ခုလည်း \$\_COOKIE['cookienam'] ကို သုံးမှာပါဘဲ။ ခု လုပ်မယ့် ပရိုဂရမ်မှာဆိုရင် MgMg12345 ဆိုတဲ့ user ဟာ ဒီ site ကို လာတိုင်း သူ့ရဲ့ username ကို type လုပ်စရာမလိုဘဲ အော်တို ဖြည့်ပေးမှာဖြစ်ပါတယ်။ movie1.php file ကိုဘဲ နည်းနည်းလေး ထပ် ပြင် ရေးကြည့်တာပေါ့။

```
<?php
    setcookie('username', 'MgMg', time() + 60);
    session_start();
    //delete this line: $ SESSION['username'] = 'MgMg12345';
    $_SESSION['authuser'] = 1;
?>
<html>
  <head>
  <title> Find my Favorite Movie! </title>
  </head>
  <body>
    <?php
      $myfavmovie = urlencode('Enemy at the gate');
      echo " <a href=\"moviesite.php?favmovie=$myfavmovie\" > ";
      echo 'Click here to see information about my favorite movie!';
      echo ' </a> ';
    ?>
  </body>
</html>
```



ပြီးတော့ ထုံးစံအတိုင်း moviesite.php file ကို နည်းနည်းလေး ထပ်ပြင်ရေးမယ်

```

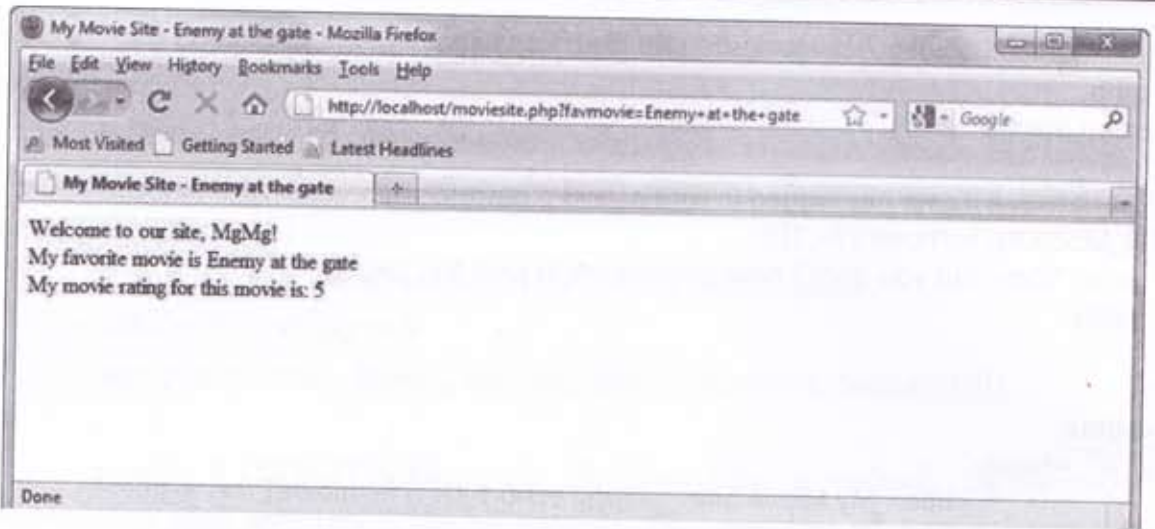
<?php
session_start();

//check to see if user has logged in with a valid password
if ($_SESSION['authuser'] != 1) {
    echo 'Sorry, but you don\'t have permission to view this page!';
    exit();
}

<html>
    <head>
        <title> My Movie Site - <?php echo $_GET['favmovie']; ?> </title>
    </head >
    <body >
        <?php
            echo 'Welcome to our site, ';
            echo $_COOKIE['username'];
            echo '! <br/> ';
            echo 'My favorite movie is ';
            echo $_GET['favmovie'];
            echo ' <br/> ';
            $movierate=5;
            echo 'My movie rating for this movie is: ';
            echo $movierate;
        ?>
    </body >
</html>

```

ဒီလိုဆိုရင် အောက်ပါပုံတွေအတိုင်းမြင်ရမှာပါ။



အထက်က ပရိုဂရမ်ဟာ cookies life time ကို 60 seconds သဲ သတ်မှတ်ထားတာပါ။  
 ကိုယ့်ဟာကို စိတ်ကြိုက် ပြင်ကြည့်ပြီးလေ့ကျင့်ကြည့်ပါ။

Passing information with Forms

Form ပုံစံကို နည်းနည်းလောက် လေ့လာကြည့်ရအောင် ။ သူ့ကို စတင်တော့မယ်ဆိုရင်

(၁) <form> နဲ့ စတင်ပြီး action and method ဆိုတဲ့ attribute ပါပါတယ်။ action အနေနဲ့

URL တစ်ခုကို ပေးနိုင်ပြီး method အနေနဲ့ GET or POST တစ်ခုခုကို သုံးနိုင်ပါတယ်။

(၂) input fields များပါပါတယ်။ အသုံးများတာတွေကတော့

TEXT: min 2 characters to 2000 characters အထိ သုံးနိုင်ပါတယ်။ ဒီထက်

ပိုများများသုံးချင်ရင်တော့ **textarea** ဆိုတဲ့ ဟာကို သုံးရပါမယ်။

Check box : user တွေကို Choice လုပ်ဖို့ရန်အတွက်ဖြစ်ပါတယ်။ choice ကို

တစ်ခုထက်မက ရွေးချယ်နိုင်မှာဖြစ်ပါတယ်။

Radio: radio button ကလည်း list ထဲကနေ ရွေးချယ်ဖို့ဘဲဖြစ်ပါတယ်။ ဒါပေမယ့်

သူက တစ်ခုဘဲ ရွေးလို့ရမှာဖြစ်ပါတယ်။

Select: ဒီကောင်ကိုတော့တော်တော်များများက Drop down box အဖြစ်သိကြပါတယ်။

Passwords: သူ့ကိုတော့ user က ဘာရိုက်တယ်ဆိုတာကို Hide

လုပ်ထားရမှာဖြစ်ပါတယ်။

Using forms to get information

movie1.php file ကိုဘဲ နည်းနည်းလေး ထပ်ပြီး ပြင်ရေးကြည့်မယ်လေ။ ပြင်ရင်းပြင်ရင်း

တော်တော်တောင် များလာပြီနော်။ ကျွန်တော်ပြောချင်တာက များလာပြီဆိုတော့ Comment

လေးတွေ သေချာရေးဖို့ ပါ။ အောက်မှာ ပြထားတဲ့ Code လေးတွေကို သေချာ လေ့လာကြည့်ပါ။



```
<?php
```

```
//delete this line: setcookie('username', 'MgMg', time() + 60);
session_start();
$_SESSION['username'] = $_POST['user'];
$_SESSION['userpass'] = $_POST['pass'];
$_SESSION['authuser'] = 0;

//Check username and password information
if (($_SESSION['username'] == 'MgMg') and
    ($_SESSION['userpass'] == '12345')) {
    $_SESSION['authuser'] = 1;
} else {
    echo 'Sorry, but you don\'t have permission to view this page!';
    exit();
}
```

```
?>
```

```
<html>
```

```
<head >
```

```
<title> Find my Favorite Movie! </title >
```

```
</head >
```

```
<body>
```

```
<?php
```

```
$myfavmovie = urlencode("Enemy at the gate");
echo " <a href=\"moviesite.php?favmovie=$myfavmovie\" > ";
echo "Click here to see information about my favorite movie!";
echo " </a> ";
```

```
?>
```

```
</body>
```

```
</html>
```

ပြီးသွားရင် moviesite.php ကို နည်းနည်း ထပ်ပြင်ရေးပါဦးမယ်။

```
<?php
```

```
session_start();
//check to see if user has logged in with a valid password
if ($_SESSION['authuser'] !=1 ) {
    echo 'Sorry, but you don\'t have permission to view this page!';
    exit();
}
```

```

    }
  >
</html>
</head>
<title> My Movie Site - <?php echo $_GET['favmovie']; ?> </title>
</head>
<body>
  <?php
    echo 'Welcome to our site, ';
    //delete this line: echo $_COOKIE['username'];
    echo $_SESSION['username'];
    echo '!<br/>';
    echo 'My favorite movie is ';
    echo $_GET['favmovie'];
    echo ' <br/> ';
    $movierate = 5;
    echo 'My movie rating for this movie is: ';
    echo $movierate;
  ?>
</body>
</html>

```

ပြီးသွားရင် Login form တစ်ခု တည်ဆောက်ပါမယ်။ text editor မှာ new document လုပ်ပါ။ File name ကို login.php လို့ နာမည်ပေးပါမယ်။

```

<?php
session_unset();
?>
<html>
  <head>
    <title>Please Log In</title>
  </head>
  <body>
    <form method="post" action="movie1.php">
      <p> Enter your username:
      <input type="text" name="user"/>
      </p>
      <p> Enter your password:

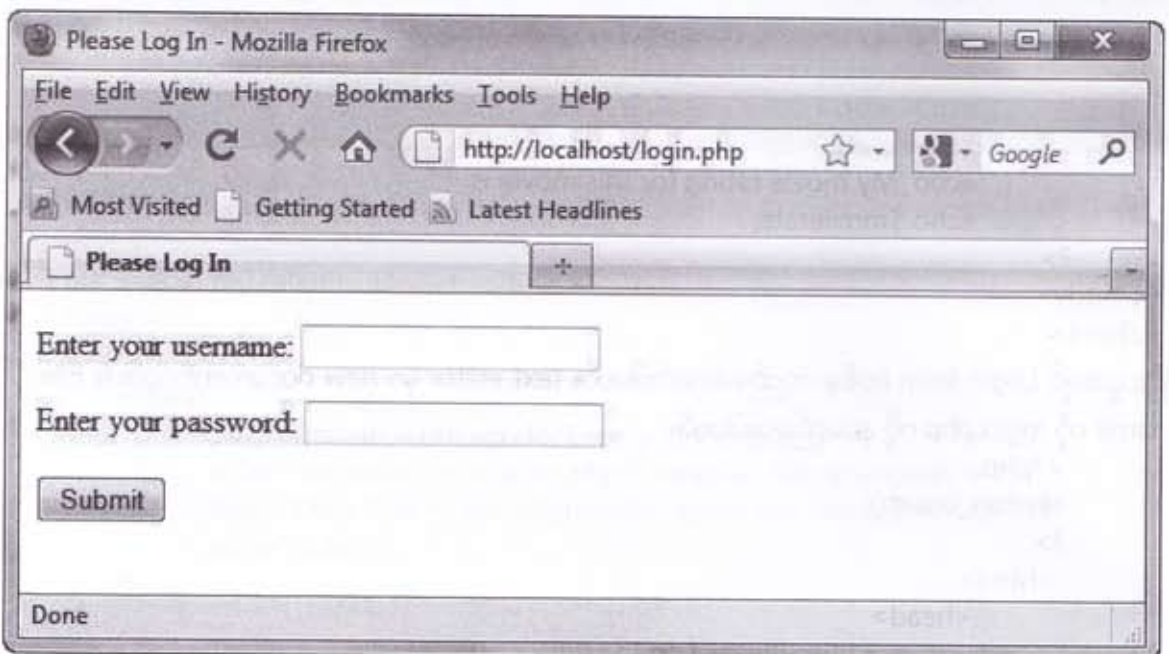
```

```

<input type="password" name="pass"/>
</p>
<p>
<input type="submit" name="submit" value="Submit"/>
</p>
</form>
</body>
</html>

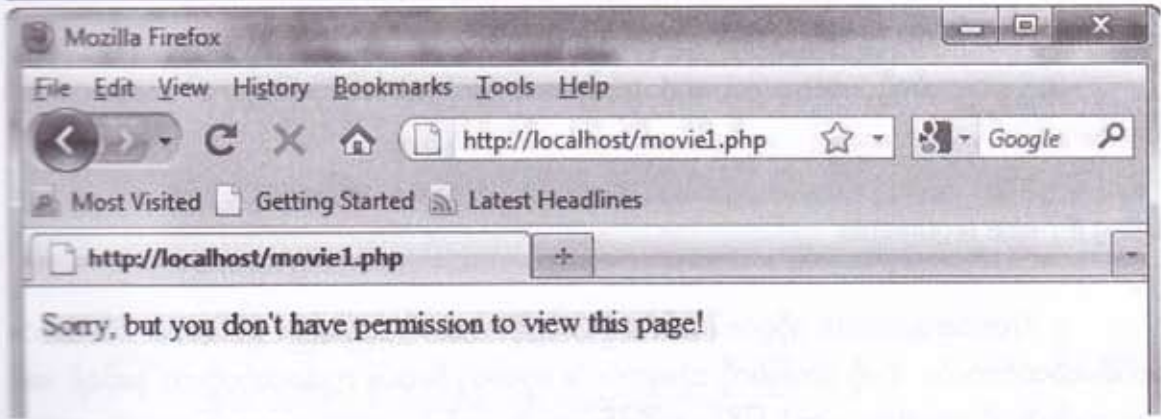
```

ပြီးသွားရင် login.php file ကို ခေါ်ကြည့်ပါမယ်။ အောက်ကပုံအတိုင်း မြင်ရမှာပါ။



username နေရာမှာ mgmg12345 password ကို 12345 ဆိုပြီးဝင်ကြည့်ပါ။  
ဝင်လို့ရမှာမဟုတ်ပါဘူး။ username မှားနေတဲ့အတွက် ဖြစ်ပါတယ်။ Authorization scripts  
အလုပ်လုပ်တယ်ပေါ့ဗျာ။





အခု login name "MgMg" and password "12345" နဲ့ဝင်ကြည့်ပါ။ ဝင်လို့ရသွားတာကို တွေ့ရပါမယ်။

**How it works**

ပထမဆုံးအလုပ်လုပ်တာက login.php လေ ။ ဟုတ်တယ်မလား။ သူ့ထဲမှာ ဘယ် page တတ်လာတဲ့ session တွေထဲက variables တွေကို unset လုပ်လိုက်ပါတယ်။ အဲဒါကို session\_unset() ဆိုတဲ့ function နဲ့ လုပ်လိုက်တာပါ။ ပြီးရင် variables ၂ ခုဖြစ်တဲ့ username and password (user and password respectively) ကို မေးပါတယ်။ ပြီးရင် ရလာတဲ့ variables ၂ ခုကို movie1.php ကို ပို့ပေးပါတယ်။ အဲလို ပို့တဲ့နေရာမှာ POST method ကို သုံးပြီးပို့ပါတယ်။ အဲဒါကြောင့် movie1.php မှာ \$\_POST syntax ကို တွေ့ရတာဖြစ်ပါတယ်။ ခု ဒီလောက် သိရင် ရပါပြီ။ နောက် အခန်းတွေမှာ လက်တွေ့လေ့ကျင့်ခန်းတွေ လုပ်ရင်းနဲ့ ပိုပြီးနားလည်လာပါမယ်။

movie1.php fileက တကယ်တော့ အောက်ပါလုပ်ငန်းစဉ်များကို လုပ်ဆောင်သွားပါတယ် -

- (၁) Session ကို စတင်ပေးတယ်။ default အားဖြင့် login က ရလာတဲ့ variables တွေကို register လုပ်ပေးတယ်။
- (၂) Username and password ကို accept လုပ်နိုင်လား မလုပ်နိုင်ဘူးလား စစ်ပေးတယ်။
- (၃) တကယ်လို့ username and password ၂ ခုလုံးကို လက်ခံတယ်ဆိုရင် autheruser ကို 1 အဖြစ် သတ်မှတ် ပေးပြီးတော့ other pages တွေကို access ပေးပါတယ်။

(like moviesite.php)

(၄) တကယ်လို့ username and password က မှားတယ်ဆိုရင် error message ကို ပြပေးပါတယ်။

Using if / else arguments

ခု if/elsearguments ကိုရှင်းပါမယ်။ ရှေ့မှာတုန်းကတော့ နည်းနည်း ရှင်းခဲ့သေး တယ်လေ။ မှတ်မိမယ်ထင်တယ်။ if ကို ဘာဖြစ်လို့ သုံးရတာလဲ။ ဟုတ်ကဲ့ ဒီလိုပါ။ ကျွန်တော်တို့ဟာ ခုဆိုရင် val- ues အမျိုးမျိုးကို variables တွေနဲ့ ပြနိုင်လာပြီဖြစ်ပါတယ်။ တစ်ချို့ variable တွေကို specific action လုပ်ချင်တဲ့အခါမျိုးမှာ if ကို သုံးပါတယ်။ ဥပမာ \$password ဆိုတဲ့ variable ပေါ့။ user ကို password တောင်းမယ်။ user က ထည့်မယ်။ မှန်ရင် site ထဲကို ဝင်ခွင့်ပေးမယ်။ မမှန်ဘူးဆိုရင် user ကို kick out ဒါမှမဟုတ် site ထဲကို ဝင်ခွင့်မပေးဘူး။ if ကို else ထပ်ပေါင်း ပေးလိုက်မယ်ဆိုရင် the whole range of possible actions တွေကို ထိန်းနိုင်မှာပါ။

Using if statements

Basic if statement က အောက်ဖော်ပြပါ အတိုင်းဖြစ်ပါတယ်  
if (condition) action to be taken if true.

ဥပမာ အနေနဲ့ -

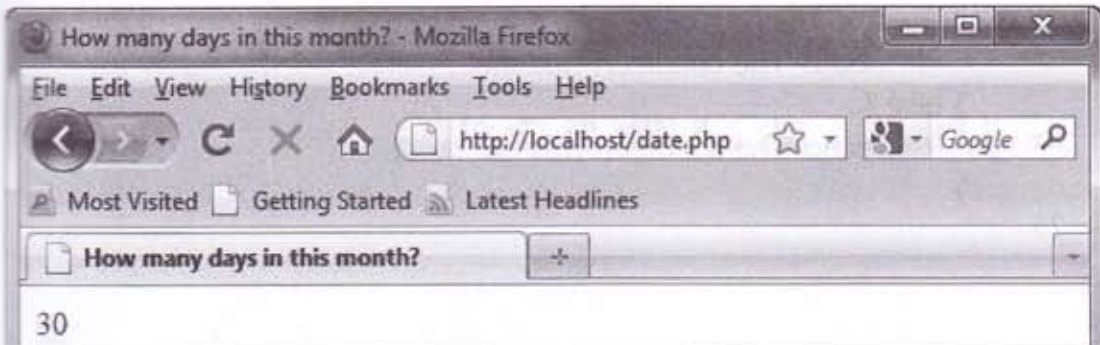
```
if ($stockmarket > 1000) echo 'Wharr.... Time to Party';
```

တကယ်လို့ simple statement ထက် ပို ရှည်တယ်ဆိုရင်တော့ {} ကို သုံးရမှာပါ။ ဒီ အဖွင့် အပိတ်တွေကို သေချာ ဂရုစိုက်ဖို့လိုပါတယ်။ အောက်ဖော်ပြပါ နမူနာကို ကြည့်ပါ။

```
if ($stockmarket > 10000) {
    echo 'Wharr .....Time to Party!';
    $mood = 'happy';
    $retirement = 'potentially obtainable';
}
```

အခု ကျွန်တော်တို့ လက်တွေ့လေ့ကျင့်ဖို့ လုပ်ပါမယ်။ text editor ကိုဖွင့်ပါ။ new document လုပ်ပါ။ date.php လို့ နာမည်ပေးပါ။ ပြီးရင် အောက်က code တွေအတိုင်း လိုက်ရေးကြည့်ပါ။

```
<html>
<head>
<title> How many days in this month? </title>
</head>
<body>
<?php
    date_default_timezone_set('Asia/Yangon');
    $month = date('n');
    if ($month == 1) { echo '31'; }
    if ($month == 2) { echo '28 (unless it\'s a leap year)'; }
    if ($month == 3) { echo '31'; }
    if ($month == 4) { echo '30'; }
    if ($month == 5) { echo '31'; }
    if ($month == 6) { echo '30'; }
    if ($month == 7) { echo '31'; }
    if ($month == 8) { echo '31'; }
    if ($month == 9) { echo '30'; }
    if ($month == 10) { echo '31'; }
    if ($month == 11) { echo '30'; }
    if ($month == 12) { echo '31'; }
?>
</body>
</html>
```





ကျွန်တော် ဒီစာအုပ်ရေးနေတဲ့ အချိန်က စက်တင်ဘာလ ဖြစ်တဲ့အတွက် 30 လို့ ပြနေတာပါ။

**How it works**

အရင်ဆုံး date\_default\_timezone\_set ဆိုတဲ့ date function တစ်ခု နဲ့ ကိုယ်ရောက်ရှိနေတဲ့ ဒေသပေါ်မူတည်ပြီး time zone သတ်မှတ်တာပါ။ အမေရိကား နယူးယောက် မှာဆိုရင် ('America/New\_York') လို့ သတ်မှတ်ရမှာပါ။ ကျွန်တော်တို့ ရေးလိုက်တဲ့ script က \$month ဆိုတဲ့ variable ကို ယူပါတယ်။ ပြီးတော့ PHP date function တွေထဲက တစ်ခုဖြစ်တဲ့ date('n') ဆိုတဲ့ value နဲ့ assign လုပ်ပေးလိုက်ပါတယ်။ ဒီ script က ဆာဗာရဲ့ အချိန်ကို ကြည့်ပါတယ်။ ဒီနေရာမှာ တစ်ခု ပြောချင်တာက date function ကိုက 1 for january , 2 for february စသဖြင့် သိပါတယ်။ တော်နေကြာ ဘယ်က ဘယ်လို သိတာလဲ ဦးကျောက်စားနေမှာစိုးလို့ပါ။ date function ကိုလည်း နောက်ပိုင်း အခန်းတွေမှာ အကျယ်တဝန် ရှင်းပါဦးမယ်။

ဒီ script မှာ if statement သုံးပြီး မှန်တဲ့ အထိ စစ်ပါတယ်။ ပထမဆုံး if statement က False ဖြစ်သွားတယ်ဆိုရင်နောက် ထပ် statement တစ်ခုကို ချက်ခြင်းစစ်ပါတယ်။ ဒီလိုနဲ့ True ဖြစ်တဲ့အထိ စစ်ပါတယ်။

**Using if and else together**

သူ့ရဲ့ပုံစံလေးကတော့အောက်ပါအတိုင်းဖြစ်ပါတယ်။ အကျယ်တဝန်တော့မရှင်းတော့ဘူးလေ။ တစ်ခု မဟုတ်ရင် တစ်ခု ဆိုတဲ့ အဓိပ္ပာယ်ရပါတယ်။

```
<?php
    $test = 5;
    if($test < 15) {
        echo 'The condition was met';
    } else {
        echo 'The condition was not met';
    }
?>
```

အထက်က ပရိုဂရမ်လေးကို ကြည့်မယ်ဆိုရင် \$test ဆိုတဲ့ variable တန်ဖိုးကို 5 လို့ သတ်မှတ်လိုက်တယ်။ ပြီးတော့ if statement နဲ့ စစ်ကြည့်လိုက်တယ်။ စစ်ကြည့်တယ်ဆိုတာ ကိုယ် output လုပ်ချင်တဲ့ data အတွက် ဖြစ်ပါတယ်။ \$test က 15 ထက်ငယ်တယ်ဆိုရင် The condition was met ကို ထုတ်ပေးမှာပါ။ မငယ်ဘူးဆိုရင်တော့ The condition was not met ဆိုတဲ့ စာကြောင်းကို ထုတ်ပေးမှာပါ။ \$test = 20 လို့သာပေးခဲ့မယ်ဆိုရင် ဒုတိယ output ကို ထုတ်ပေးမှာပါ။

တကယ်လို့ ကိုယ်ပေးချင်တဲ့ message က အခြေအနေ ၂ ခုထက်ပိုရင် ဘယ်လိုလုပ်မလဲ ။ ရပါတယ်။ အဲဒီအခါမှာ if ရယ် else ရယ်ကြားမှာ else if ကို ထည့်သွင်း အသုံးပြုနိုင်ပါသေးတယ်။ အောက်က နမူနာလေးကို လေ့လာကြည့်ရအောင်။

```
<?php
    $age = 20;
    if($age < 18) {
        echo "Not old enough to vote or drink! <br />";
    } else if ($age < 21) {
        echo "Old enough to vote, but not to drink. <br />";
    } else { // If we get here, $age is >= 21
        echo "Old enough to vote and drink! <br />";
    }
?>
```

အပေါ်က code ကိုလေ့လာကြည့်ရင် အသက် ၁၈ နှစ်အောက်ဆိုရင် vote or drink ၂ ခုလုံး လုပ်လို့မရဘူး။ တကယ်လို့ ၂၁ နှစ်အောက်ဆိုရင်တော့ ( ၁၈ နှစ် နဲ့ ၂၁ နှစ်ကြားဖြစ်သွားတာပေါ့) ထောက်လို့မရဘူး။ အောက်ဆုံး တစ်ခုက တော့ ၂ ခုလုံးလုပ်လို့ရပြီပေါ့။ \$age=20 ဆိုတဲ့ နေရာမှာ အသက် အမျိုးမျိုးကို ပြောင်းလဲ လေ့ကျင့်ကြည့်ပါ။

While and do-while

while loop က ဘယ်လို လုပ်ပေးလဲဆိုရင် code block တစ်ခုကို True မဖြစ်မချင်း repeat လုပ်ပေးပါတယ်။ နမူနာလေးနဲ့တွဲလေ့လာလိုက်ရင် ပိုနားလည်သွားမှာပါ။

```

<?php
    $i = 0;
    while($i<3) {
        echo "Count is at $i. <br />";
        ++$i;
    }
?>

```

အထက်ကပုံစံမှာဆိုရင် ကျွန်တော်တို့က counter variable \$i ကို သုံးထားပါတယ်။ သူ့ရဲ့ ကနဦးတန်ဖိုးကို သုည 0 ပေးထားလိုက်ပါတယ်။ နောက်ပြီးတော့ while loop ထဲထည့် သုံးလိုက်ပါတယ်။ ဘယ် အခြေအနေထိ ဒီကုဒ် ကို Run မှာလဲဆိုတော့ <3 ထိ ပါ။ ဘယ်က စမှာလဲဆိုတော့ သုည ကပါ။ while နဲ့စစ်မယ်။ ပထမဆုံးက သုည ..အိုကေ မှန်တယ် ဘာဖြစ်လို့လဲဆိုတော့ 3 ထက်ငယ်တယ်လေ။ ဒီတော့ ပထမဆုံး output အနေနဲ့ Count is at 0. ဖြစ်မယ်။ ဒုတိယအနေနဲ့ 1 တတိယအနေနဲ့ 2 အထိမှန်တယ်။ စတုတ္ထအနေနဲ့ 3 ကြတော့ မမှန်တော့ဘူး။ ဘာဖြစ်လို့လဲဆိုတော့ <3 ဆိုတာ 3 နဲ့ ညီလို့မရဘူး။ ငယ်ကိုငယ်ရမှာ။ ဒီလိုမျိုး ပထမက 0 ဒုတိယက 1 တတိယက 2 ဆိုတာ ဘယ်လို လုပ် ကွန်ပျူတာက သိလဲ။ ဟုတ်ကဲ့... ++\$i ဆိုတဲ့ဟာနဲ့ ဖော်ပြတာပါ။ ထွက်လာတဲ့ တန်ဖိုးကို 1 ပေါင်း ပေးတာပါ။

ဒီ script ကို run လိုက်ရင် Browser မှာ ဒီလို ပေါ်မှာပါ။

Count is at 0.  
 Count is at 1.  
 Count is at 2

မှတ်ချက်။ သင့် အနေနဲ့ မှတ်ထားရမှာက condition က မကိုက်ညီနိုင်ဘူးဆိုရင် loop က execute လုပ်ပေးမှာ မဟုတ်ပါဘူး။ \$i=4 ဆိုရင် ဘာ output မှ ထွက်လာမှာမဟုတ်ပါဘူး။



do-while loop

```

<?php
    $i=10;
    do{
        echo "The count is at $i.";
        ++$i;
    }while(i<5);
?>

```

ဒီပုံစံမှာဆိုရင် do ဆိုတော့ လုပ်ပါပေါ့။ ဘာလုပ်ခိုင်းလဲဆိုရင် output အနေနဲ့ \$i value ကို ထုတ်ခိုင်းတယ်။ ပြီးတော့ while loop ကို ဆက်ဆင်းတယ်။ အဲဒီမှာကြတော့ ဘာ output မှ ထွက်တော့ဘူးလေ။ initial value ကိုက 10 ဖြစ်နေပြီ။ while မှာပေးထားတဲ့ condition က < 5 ဆိုတော့ ဘာ မှ ထွက်လာမှာမဟုတ်ဘူး။ Output အနေနဲ့ The count is 10. ဆိုပြီး ပေါ်လာမှာပါ။

FOR loop

for loop ဆိုတာ PHP program ရဲ့ အသုံးများဆုံး statement တွေထဲက တစ်ခု အပါဝင်ဖြစ်ပါတယ်။ သူ့ အနေနဲ့ expression ၃ ခု ကို လက်ခံနိုင်ပါတယ်။ Expression 1 အနေနဲ့ loop ရဲ့ အစ တန်ဖိုးကို ဖော်ပြတာဖြစ်ပြီး Expression 2 အနေနဲ့ကတော့ expression 1 အပေါ်အခြေခံကာ iteration method သုံးပြီး တွက်ချက်မှာ ဖြစ်ပါတယ်။ True value တွေကိုဘဲ တွက်မှာဖြစ်ပြီး False ဖြစ်သွားတဲ့အခါမှာ ရပ်သွားမှာဖြစ်ပါတယ်။ Expression 3 အနေနဲ့ကတော့ iteration တစ်ခုစီရဲ့ end of iteration ဖြစ်ပါတယ်။ ဒီလိုပြောနေတော့ နည်းနည်း ရှုပ်သွားသလို ခံစားရမယ်။ မရှုပ်ဘူးနော်။ အောက်က နမူနာပုံစံလေးကိုကြည့်လိုက်။ လွယ်မှလွယ်။ ပြောဖို့ကျန်သွားတယ်။ Expression တစ်ခုစီမှာ အစိတ်အပိုင်း တစ်ခု ထက်မက ပါနိုင်ပါတယ်။ ပါလာခဲ့ရင် comma (,) နဲ့ ခံပေးရပါမယ်။ Expression တစ်ခုနဲ့တစ်ခုကြားမှာတော့ semicolon (;) နဲ့ ခံရေးပေးရပါတယ်။

```
<?php
    for($i=0; $i<5; ++$i){
        echo "The count is $i <br />";
    }
?>
```

ဒါဆိုရင် အဲဒီ script ကို run မယ်ဆိုရင် browser က အောက်ပါအတိုင်း ပြပေးမှာပါ။

```
The count is 0
The count is 1
The count is 2
The count is 3
The count is 4
```

### FOREACH loop

foreach ဆိုတဲ့ အတိုင်း သူလည်းဘဲ တစ်ခုချင်းစီအတွက် သုံးတာပေါ့ဗျာ။ ဒီ foreach က arrays နဲ့ တွဲသုံးရင် very powerfull ဖြစ်တယ်ဗျ။ နမူနာလေးနဲ့ဘဲပြမယ်ဗျာ။ ရှေ့ခန်း က array ကိုသေချာ နားလည်ခဲ့ရင် လွယ်ပါတယ်။

```
<?php
    $bands = array(
        "IronCross", "Emperor", "LazyClub", "Aurora"
    );
    foreach($bands as $band) {
        echo $band, "<br />";
    }
?>
```

ဘယ်လို အလုပ်သွားသလဲဆိုရင် \$bands ဆိုတဲ့ variable နဲ့ array တစ်ခု နဲ့ တည်ဆောက်လိုက်တယ်။ ပြီးတော့ foreach နဲ့ loop ပတ်လိုက်တယ်။ \$bands as \$band ဆိုတာမှာ as = alias name (နာမည်ခွဲ ပေါ့ ဗျာ) ။ array ထဲက value တစ်ခု စီကို variable တစ်ခု စီ အဖြစ်

သတ်မှတ်လိုက်တယ်။ အဲဒါတော့ ဘယ်လိုဖြစ်သွားမလဲ။ မြင်သာအောင် ရှင်းပြရရင် \$band = Iron-Cross နောက်တစ်ခါ ထပ်ပတ်ပြန်တော့ \$band = Emperor ..and so on ပေါ့ဗျာ။ အဲဒါတော့ output ထုတ်လိုက်ရင် Browser က အောက်ပါအတိုင်း ပြပါလိမ့်မယ်။

- IronCross
- Emperor
- LazyClub
- Aurora

foreach loop ရဲ့ လှပပြီး အသုံးဝင်တဲ့ ပုံစံလေးတစ်ခုကို နည်းနည်းလေး ထဲထဲဝင်ဝင် ဆယ်လေ့လာကြည့်ရအောင်။ အောက်က ပုံစံလေးကိုကြည့်ကြည့်ပါ။

```
<?php
    $person = array(
        'name' => 'Kokyaw',
        'age' => 23,
        'passion' => 'craft beer'
    );
    foreach($person as $key => $value) {
        echo "His $key is $value. <br />";
    }
?>
```

အထက်က script ကို run လိုက်ရင် အောက်ပါ အတိုင်း browser က ထုတ်ပေးမှာပါ။

His name is Kokyaw.  
 His age is 23.  
 His passion is craft beer.



**Break**

ဘယ် loop မှာ မဆို break function သုံးပြီး လိုချင်တဲ့ နေရာမှာ ရပ်တန့်လို့ရပါတယ်။ ဥပမာ-

```

while($i<10) {
    if($i == 7) {
        break; // Exit the while loop
    }
}

```

ဒီမှာဆိုရင် while and if ကို တွဲသုံးသွားတာ တွေ့ရပါမယ်။ break က loop မဟုတ်ဘဲ loop ကို အခြေအနေတစ်ခုမှာ ရပ်တန့်ပေးတယ်ဆိုတာကို မှတ်ထားဖို့လိုပါတယ်။ \$i ==7 ဆိုတဲ့ နေရာမှာ loop က ရပ်သွားမှာပါ။ ရပ်ပြီးသွားတာနဲ့ loop ကိနေ ထွက်သွားမှာ ဖြစ်ပါတယ်။

**Switch**

Switch statement ကို condition 2 ခုထက် ပိုတဲ့ အခါမျိုးမှာ သုံးလေ့ရှိပါတယ်။ သူ့ကို case နဲ့ တွဲပြီး သုံးရပါတယ်။ မေးစရာရှိလာပါတယ်။ if ကို သုံးရင်ရော မရဘူးတဲ့လား။ ရပါတယ် လို့ဘဲဖြေရမှာဘဲ။ ဒါပေမယ့် condition အများကြီးအတွက်ဆိုရင် switch ကို သုံးရတာ ပိုပြီး လွယ်ကူမြန်ဆန်ပါတယ်။ case တစ်ခုချင်းစီဟာ ဘာနဲ့ သွားတူသလဲဆိုရင် if function နဲ့ သွားတူပါတယ်။ switch မှာ expression ကို လက်ခံပြီး case တစ်ခုချင်းစီနဲ့ စစ်ပါတယ်။ မှန်သွားပြီဆိုတဲ့ case မှာ code ကို execute လုပ်ပါတယ်။ case တစ်ခုချင်းစီကို break statement or else statement နဲ့ separate လုပ်ထားရပါမယ်။ နမူနာလေး တစ်ခုနဲ့ ပြပါမယ်။ date ကို check လုပ်တဲ့ script လေးတစ်ခုပေါ့။ ရိုးရိုးရှင်းရှင်းလေးနဲ့ နားလည်လွယ်ပါတယ်။

```

<?php
    $day = date('w');
    switch($day)
    {
        case '0':
            echo "It's Sunday!";
            break;

```

```

case '1':
    echo "It's Monday!";
    break;
case '2':
    echo "It's Tuesday!";
    break;
case '3':
    echo "It's Wednesday!";
    break;
case '4':
    echo "It's Thursday!";
    break;
case '5':
    echo "Woohoo! It's Friday!";
    break;
case '6':
    echo "It's Saturday!";
    break;
default:
    echo "That's no day I recognize...";
    break;
}
?>

```

ကျွန်တော်တို့ အပေါ်က script မှာဆိုရင် date() function ကို ယူသုံးခဲ့ပါတယ်။ \$date ဆိုတဲ့ variable ရဲ့ value ကို date() နဲ့ ပေးခဲ့တာပါ။ w ဆိုတာက week ကို ပြောတာပါ။ နောက်ပိုင်း အခန်းတွေမှာ လက်တွေ့လုပ်ရင်း နဲ့ ပိုနားလည်မှာပါ။ ခုကတော့ ဒီလောက် သိထားရင် ရပါပြီ။

Continue

continue ရဲ့ အလုပ်လုပ်ပုံက break ရဲ့ အလုပ်လုပ်ပုံနဲ့ ဆင်တူပါတယ်။  
ခြားနားတာလေးတစ်ခုဘဲရှိတယ်။ အဲဒါက ဘာလဲဆိုတော့ continue ဟာ လက်ရှိ အခြေအနေမှာဘဲ  
end ဖြစ်ပါတယ်။ ပြီးရင် သူက Loop က နေ ပြန်မထွက် သွားပါဘူး။ ဆက်ပြီးတော့ loop လုပ်ပါတယ်။  
ဥပမာ အနေနဲ့ ကျွန်တော်တို့ စုံ ဂဏန်းသီးသန့် ထုတ်ကြည့် ရအောင် ။

```
<?php
for($i=0; $i<=10; ++$i) {
    /*
    * If the modulus of $i and 2 is not zero (which evaluates
    * to false), we continue
    */
    if($i%2) {
        continue;
    }
    echo $i, " ";
}
?>
```

ဒီမှာဆိုရင် for နဲ့ if ကို တွဲသုံးထားပါတယ်။ For loop ထဲမှာ if statement ထည့်သုံးပြီးတော့  
အခြေအနေကို စစ်ခိုင်းတာပါ။ \$i%2 ဆိုတာက \$i တန်ဖိုးကို ၂ နဲ့ စားလို့ ပြတ် သွားရင် continue  
(loop ကို ဆက် ပြီးတော့ ပတ်ခိုင်းတာပါ) လုပ်ခိုင်းပါတယ်။ " " ကို ထည့်သုံးလိုက်တာက space  
ခြားစေချင်လို့ပါ။ သူ့ကို run လိုက်ရင်

0 2 4 6 8 10  
ဆိုပြီးရပါမယ်။



return

return statement ဟာဆိုရင် PHP မှာ most usefull statement လို့တောင် ပြောလို့ရပါတယ်။ သူ့ရဲ့ သဘောတရားကတော့ function တစ်ခုအတွင်း မှာထည့်သုံးပြီး အဆုံးကိုရောက်သွားပြီဆိုရင် function ကို stop လုပ်လိုက်ပြီး ရလာတဲ့ argument ကို function ရဲ့ value အဖြစ် ပြန်ထုတ်ပေးတာဘဲ ဖြစ်ပါတယ်။ တကယ်လို့ function ရဲ့ အပြင်ဘက်မှာ သုံးမယ်ဆိုရင်တော့ script ကို stop လုပ်တာဘဲဖြစ်ပါတယ်။ ဒီ return အကြောင်းကို နောက်ပိုင်းမှာ user defined function အခန်းရောက်ရင် ဆက်ရှင်းပါမယ်။ ခုတော့ ဒီလောက်သိထားဖို့ဘဲလိုပါတယ်။

Understanding ARRAYS

array ကတော့ PHP ရဲ့ powerfull datatypes ထဲက တစ်ခုဘဲဖြစ်ပါတယ်။ သူ့ရဲ့ map information ( map ဆိုတာက one to one function လိုပါဘဲ) က key နဲ့ value ဆိုတာကို သုံးပါတယ်။ နံပါတ် ၁ key သည် နံပါတ် ၁ value ပေါ့။ ဒီတော့ array ဆိုတဲ့ အထဲမှာ information အများကြီးကို variable တစ်ခုချင်းစီအနေနဲ့ သိမ်းဆည်းထား လို့ရတယ်။ သုံးချင်တဲ့အခါ အကုန်ထုတ်သုံးလို့ရသလို လိုအပ်တာလေးတွေဘဲ လည်း ထုတ်သုံးလို့ရတယ်လေ။ array ဆိုတာ သေတ္တာ နဲ့ လည်းတူပါတယ်။ သူ့ထဲမှာ ပစ္စည်းတွေထည့်ထားမယ်။ ကိုယ်က လိုအပ်တဲ့အခါ လိုအပ်တာကို ထုတ်ယူမယ်။ နံပါတ် ၁ သေတ္တာထဲမှာ ခဲတံ ၊ ပေတံ တွေဘဲ ထည့်ထားမယ်လို့ မှတ်ထားရင် နောင် ကို လိုတဲ့အခါ ချက်ချင်း သွားယူလို့ရမယ်လေ။ နံပါတ် ၂ ထဲမှာ ပုဆိုး ထားမယ်ဆိုရင် နောင် ကို ပုဆိုးလိုတဲ့အခါ ဘယ် လိုက်ရှာရမှန်း မဖြစ်တော့ဘူး ။ သိနေပြီ နံပါတ် ၂ ထဲမှာ ဆိုတာကို ။ အဲဒီထဲကမှ ဘာ အရောင် ဘာ အမျိုးအစား ဆိုတာတော့ ထပ်ရွေး ရမှာပေါ့။ အောက်က script ကို ကြည့်ပါ။ ရိုးရိုး တန်းတန်းရေးထားတာပါ။

```
<?php
    $entryTitle = "Sample Title";
    $entryDate = "September 9, 2011";
    $entryAuthor = "Kokyaw";
    $entryBody = "Today, I wrote a blog entry.";
?>
```

သာမန်အားဖြင့် ဒီလို ရေးချင်းမျိုးဟာ ဘာမှမဖြစ်ပေမယ့် function တစ်ခုမှာ သုံးတော့မယ်ဆိုရင်တော့ confuse ဖြစ်နိုင်ပါတယ်။ ဒီလို problem မျိုးကို ကျော်လွှားနိုင်အောင် array ကို သုံးပြီး ရေးလို့ရပါတယ်။ နမူနာလေးကို ကြည့်ပေးပါ။

```
<?php
    $entry = array(
        'title' => 'Sample Title',
        'date' => 'September 9, 2011',
        'author' => 'Kokyaw',
        'body' => 'Today, I wrote a blog entry.'
    );
?>
```

အထက်ပါ script ကို လေ့လာကြည့်ပါ။ \$entry ဆိုတဲ့ variable တစ်ခုထဲမှာ data အကုန်လုံးကို save လုပ်ထားလိုက်ပါပြီ။ ဟုတ်ပြီ။ ဒါဆိုရင် အဲဒီ data တွေကို ဘယ်လို access လုပ်မလဲ။ Access လုပ်ဖို့ရာ ကျွန်တော်တို့က key ကို ပေါင်း ပေးရမှာပါဖြစ်ပါတယ်။ key ကို ပေါင်းပေးတဲ့အခါမှာ variable ရဲ့ အဆုံးမှာ [] square brackets ထဲမှာ ထည့်ပေးရမှာ ဖြစ်ပါတယ်။ ဒီတော့ output ကို ဘယ်လိုထုတ်မလဲပေါ့။ လွယ်လွယ်လေးပါ။

```
<?php
    echo $entry['title'];
?>
```

ဒါဆိုရင် သူ့ရဲ့ value ဖြစ်တဲ့ Sample Title ဆိုတာ ထွက်လာမှာပေါ့။ title က key ပေါ့။

အိုကေနော်။ ခုက ဒီလောက်ဘဲ။ နောက်ပိုင်း သင်ခန်းစာမှာ All about arrays ဆိုပြီး ဆက်ရှင်းဦးမယ်။ ခု တစ်ခါတည်းရှင်းလိုက်ရင် မျက်စိရှုပ်ကုန်မှာစိုးပါတယ်။

Incrementing / Decrementing operators

ကျွန်တော်တို့ PHP program ရေးတဲ့အခါ တစ်ချို့ အပိုင်းလေးတွေမှာ ဘဲ ပေါင်းခြင်း ဘဲ နှုတ်ခြင်း တွေ ရေးဖို့ လိုအပ်ပါတယ်။ loop ပတ်တဲ့အခါမျိုးမှာ ရေးဖို့ ဆိုရင် PHP မှာ လွယ်လွယ်ကူကူ ရေးလို့ရအောင် ဝီမံပေးထားပါတယ်။ value တစ်ခုကို 1 ပေါင်းပေးဖို့ variable ရဲ့ ရှေ့မှာ ဖြစ်ဖြစ် နောက်မှာ ဖြစ်ဖြစ် ++ နှစ်ခုကို ရေးလိုက်ယုံပါဘဲ။ နှုတ်ခြင်းရင်တော့ -- (minus) နှစ်ခုကို ရေးရမှာပါ။ variable ရဲ့ ရှေ့မှာ ထည့်ရင် prepending လို့ခေါ်ပြီး နောက်မှာ ထည့်ရင်တော့ postpending လို့ခေါ်ပါတယ်။

```
<?php
    $age = 5;
    ++$age; // New value is 6
    $age++; // New value is 7

    --$age; // New value is 6
    $age--; // New value is 5

    $bar = 4;
    // Echo a prepended value
    echo ++$bar; // Output is 5, new value is 5

    // Echo a postpended value
    echo $bar++; // Output is 5, new value is 6
?>
```

Logical Operators

logical operators တွေကို ရှင်းပြတဲ့အခါမှာ control structure တွေမပါဘဲ ရှင်းပြရဖို့ တော်တော် ခက်ပါတယ်။ ဆက်စပ်နေတာကိုး ။ logical operators တွေကတော့ အောက်ပါအတိုင်းဖြစ်ပါတယ်-

- && (AND) [Returns true if both expressions are true]
- || (OR) [Returns true if at least one expression is ture]





XOR [Returns true if one expression is true, but not the other]

! [Returns true if the expression is not true]

ထုံးစံအတိုင်းလေး လေ့ကျင့်ကြည့်ကြရအောင်။

```
<?php
```

```
    $age = true;
    $bar = false;
    // Print the statement if $foo AND $bar are true
```

```
    if($age && $bar) {
        echo 'Both $age and $bar are true. <br />';
    }
```

```
    // Print the statement if $age OR $bar is true
    if($age || $bar) {
        echo 'At least one of the variables is true. <br />';
    }
```

```
    // Print the statement if $age OR $bar is true, but not both
    if($age xor $bar) {
        echo 'One variable is true, and one is false. <br />';
    }
```

```
    // Print the statement if $bar is NOT true
    if(!$bar) {
        echo '$bar is false. <br />';
    }
```

```
?>
```

ဒီလေ့ကျင့်ခန်းမှာ \$bar = true လို့ ပြောင်းပြီး ကိုယ်တိုင် ထပ်လေ့လာကြည့်ပါ။

Using INCLUDES for Efficient code

ခု ဆိုရင် သင်ဟာ code တော်တော်များများကို ( တူတာတွေရော မတူတာတွေရော ) တော်တော်လေး လေ့ကျင့်ခဲ့ ပြီးပြီနော်။ သင့် အနေနဲ့ code အတူတူ တွေကို ထပ်ခါတလဲလဲ ပြန်ရိုက်နေရတာ စိတ်အနှောက် အယှက်ဖြစ်နေမှာဘဲ။ ဒီတော့ ခုလိုမျိုးမဖြစ်ရအောင် တူရာတူရာ code တွေကို သီးသန့် လုပ်ထားမယ်။ လိုအပ်တဲ့အခါ လိုအပ်တဲ့နေရာ ရောက်မှ ခေါ်သုံးမယ်။ ဒီလိုလုပ်ရင် မရဘူးလား။ သိပ်ရတာပေါ့။ PHP မှာ include() ဆိုတဲ့ statement လေးတစ်ခု ခေါ်သုံးလိုက်ယုံဘဲ။ ဘယ်လို အလုပ်လုပ်သလဲဆိုရင် script ကို parse လုပ်တဲ့အခါမှာ parser ကခုန်က include ထဲက ရေးထားတဲ့ဟာကိုပါ page ထဲကို insert လုပ်ပေးပါတယ်။ အားလုံး ပြီးသွားတော့မှ browser ပေါ်ကို တင်ပေးတာပါ။

include statement ထဲမှာ any file type extension ကို သုံးလို့ရပါတယ်။ တစ်ချို့ကတော့ .inc ဆိုတာကို သုံးပါတယ်။ သင့်အနေနဲ့ .php ဆိုတဲ့ extension ကိုလည်း သုံးလို့ရပါတယ်။ ကျွန်တော်တို့က php လေ့လာနေတာဆိုတော့ .php နဲ့ သုံးကြမယ်ဗျာ။

ဟုတ်ပြီ ဘယ်လို နေရာတွေမှာသုံးမလဲ ။ page တစ်ခုမှာ အမြဲလိုလို ပါနေတတ်တာက header and footer တွေမဟုတ်လား။ စာမျက်နှာတိုင်း နီးပါးမှာ ပါတတ်ကြပါတယ်။ web pages 20 ရှိရင် အခါ ၂၀ ရေးမလား။ ဘယ်ရေးမလဲ။ တစ်ခါဘဲရေးပြီး အခါ ၂၀ ခေါ်သုံးမှာပေါ့။ dynamic website မှာဆိုရင် တစ်ခါရေးပြီး ကြိုက်သလောက် ခေါ်သုံးလို့ရတယ်နော်။

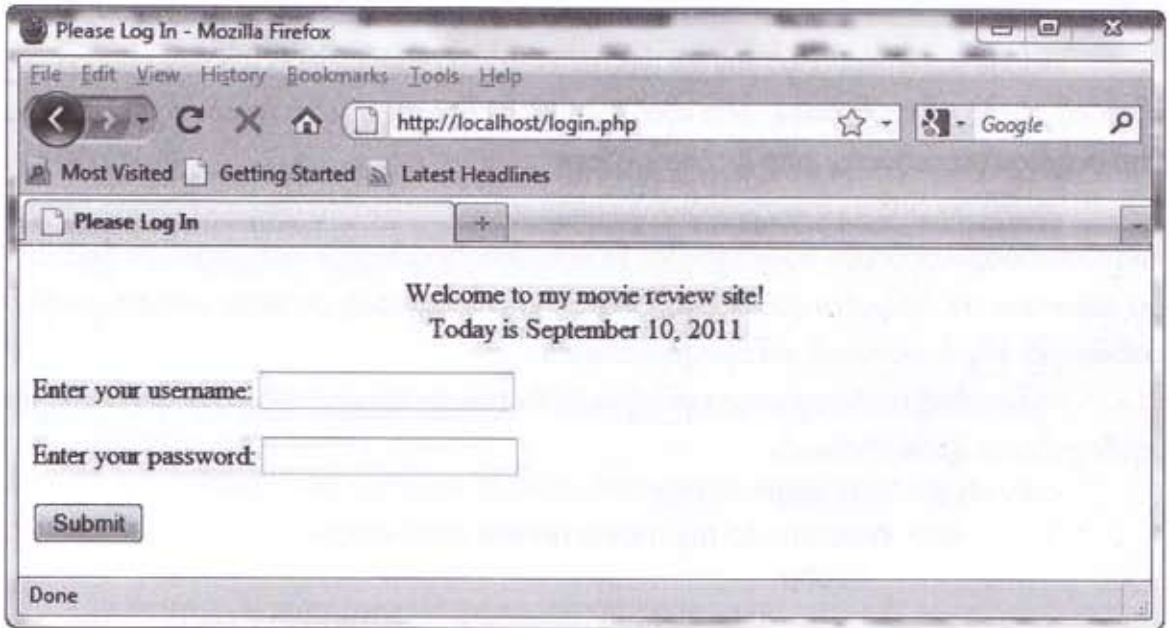
ကျွန်တော်တို့ လက်တွေ့ လေ့လာကြည့်ရအောင်။ header file လေးတစ်ခု ရေးလိုက်မယ်။ ရိုးရိုး လွယ်လွယ်လေး နဲ့ဘဲစ လိုက်မယ်။

```
<div style="text-align: center">
  <p> Welcome to my movie review site! <br/>
  <?php
      date_default_timezone_set('Yangon/Asia');
      echo 'Today is ';
      echo date('F d');
      echo ', ';
      echo date('Y');
  ?>
<br/>
</div >
```

ဒီ script ကို run လိုက်ရင် အောက်ပါအတိုင်း browser က ထုတ်ပေးမှာပါ။

Welcome to my movie review site!  
Today is September 10, 2011

ဟုတ်ပြီ ဒီscript ကို ခေါ်သုံးမယ်။ ဘယ်လို ခေါ်ရသလဲဆိုရင်<?php include 'header.php'; ?> လို့ ခေါ်လိုက်ရင် ပါပြီ။ မလွယ်ဘူးလား။ လွယ်တယ်နော်။ ကျွန်တော်တို့ ရေးခဲ့တဲ့ movie site က 3 page ရေးခဲ့ပြီးပြီလေ။ moviesite.php, movie1.php, login.php ဆိုတဲ့ဟာတွေပေါ့။ အဲဒီမှာ <body> tag ပြီးတာနဲ့ ခုနက <?php include 'header.php'; ?> ကို ပေါင်းထည့်ပေးလိုက်ပါ။





Using FUNCTIONS

PHP မှာ သူ့ရဲ့ built in function များဟာ ပြည့်စုံများပြားစွာ ပါဝင်တည်ရှိပါတယ်။ ဒါပေမယ့်လည်း တစ်ခါတစ်လေမှာ ကိုယ်လိုချင်တဲ့ ပုံစံရအောင် function တွေကို ကိုယ့်ဟာကိုလည်း သတ်မှတ်ပြီးရေးလို့ရပါသေးတယ်။ function ကို ရေးချင်ရင် function ဆိုတာနဲ့ စပါတယ်။ ဥပမာ-

```
<?php
function sayHello(){
    echo "Hello, Mingalarpar";
}
//Execute the function
sayHello();
?>
```

sayHello ဆိုတာကတော့ function name ဖြစ်ပါတယ်။ () ထဲမှာတော့ argument တွေထည့်လိုက် ထည့်နိုင်ပါတယ်။ ဥပမာ-

```
<?php
function meet($name){
    echo "Hello, Mingalarpar $name";
}
//Execute the function
meet("Kokyaw");
?>
```

ဘယ်လို အလုပ်လုပ်သလဲဆိုရင် function တွေကို လေ့လာကြည့်မယ်ဆိုရင် အောက်ကနေ ပြန် အလုပ်လုပ်တာလို့ အလွယ် မှတ်ထားလို့ရပါတယ်။ meet("Kokyaw") ဆိုတဲ့ function ကို တွေ့မယ်။ အဲဒီကနေမှာ သူ့ရဲ့ function block ဆီကို သွားမယ်။ function block မှာ စေခိုင်းထားတဲ့အတိုင်း အလုပ်လုပ်မယ်။ Kokyaw = \$name ဖြစ်သွားမယ်။ နောက်ဆုံး Hello, Mingalarpar Kokyaw ဆိုတာ ထွက်လာတာပါဘဲ။

Function block တစ်ခုကို တစ်နေရာမှာ name တစ်ခုနဲ့ရေးပြီးရင် အဲဒါနဲ့ နာမည်တူ နောက်ထပ် function တစ်ခုကို ထပ်ရေးခွင့်မရှိပါဘူး။ Function တွေကိုယ်တိုင်က miniprogram ဖြစ်ပြီး သူတို့ဟာ အပြင်က variable တွေကို မသိနိုင်ပါဘူး။ အပြင်က variable တွေကို သူတို့ထဲကို ထည့်သွင်း နိုင်ဖို့ဆိုရင် global ကို သုံးရမှာပါ။ Global ဆိုတဲ့အထဲမှာ default အနေနဲ့ \$\_POST, \$\_GET, \$\_SESSION, \$\_COOKIE and so on ပေါ့ဗျာ။

Function အကြောင်း နည်းနည်း သိသွားပြီဆိုတော့ ကျွန်တော်တို့လက်တွေ့ သုံးကြည့်ရအောင်။ movie1.php ကို နည်းနည်းလေး ပြင်ရေးမယ်နော်။ ပြင်ရင်း ပြင်ရင်း တော်တော်လေး ပြင်မိလာပြီ။ concept မပြတ်အောင်တော့ သတိထားနော်။

```

<?php
    //delete this line: setcookie('username', 'MgMg', time() + 60);
    session_start();
    $_SESSION['username'] = $_POST['user'];
    $_SESSION['userpass'] = $_POST['pass'];
    $_SESSION['authuser'] = 0;

    //Check username and password information
    if (($SESSION['username'] == 'MgMg') and
        ($SESSION['userpass'] == '12345')) {
        $_SESSION['authuser'] = 1;
    } else {
        echo 'Sorry, but you don\'t have permission to view this page!';
        exit();
    }
?>
<html>
<head >
<title> Find my Favorite Movie! </title >
</head >
<body>
    <?php include 'header.php'; ?>

</html>

```

```

<?php
    $myfavmovie = urlencode('Enemy at the gate');
    echo " <a href=\"moviesite.php?favmovie=$myfavmovie\" >
";
    echo "Click here to see information about my favorite movie!";
    echo " </a > ";
?>
<br/>
<a href="moviesite.php?movienum=5">Click hereto see my top 5
movies.</a>
<br/>
<a href="moviesite.php?movienum=10">Click hereto see my top 10
movies.</a>
</body>
</html>

```

Highlight ပြထားတဲ့ထဲက movienum ဆိုတာကို သတိထားမိမှာပါ။ အဲဒီကောင်ကို moviesite.php မှာ \$\_GET method နဲ့ ပြန်ခေါ်ပြီးသုံးမယ်ဆိုတာ သတိရစေချင်တယ်။ တစ်ခေါက်ထဲဖတ်ပြီး နားလည်ဖို့ မှတ်မိဖို့ တော်တော် ခက်ပါလိမ့်မယ်။ ဒီတော့ တစ်ခါဖတ်ပြီး တစ်ခါ ချ ရေးကြည့်ပါ။ လက်တွေ့လုပ်ကြည့်ပါ။ များများ ရေးပါ။ ကိုယ့်ဟာကို သင့်တော်သလို ပြန်ပြောင်း ရေးကြည့်ပါ။ ဒီလိုဆိုရင် ပိုပြီးနားလည်လာမှာပါ။

ကျွန်တော်တို့ moviesite.php ကို နည်းနည်းလေး ထပ်ပေါင်းရေးကြည့်မယ်။ function လေးကိုလည်း လက်တွေ့ ကျကျလေ့ကျင့်ရင်းပေါ့။

```

<?php
session_start();
//check to see if user has logged in with a valid password
if ($_SESSION['authuser'] !=1 ) {
    echo 'Sorry, but you don\'t have permission to view this page!';
    exit();
}
?>

```



```

<html>
  <head>
    <title> My Movie Site - <?php echo $_GET['favmovie']; ?> </title>
  </head>
  <body>
    <?php include 'header.php'; ?>
    <?php
      function listmovies_1() {
        echo '1. ThaGyanMoe <br/> ';
        echo '2. Saving Private Ryan <br/> ';
        echo '3. Pearl Harbour <br/> ';
        echo '4. Enemy at the gate <br/> ';
        echo '5. Matrix <br/> ';
      }

      function listmovies_2() {
        echo '6. Terminator 2 <br/> ';
        echo '7. Star War <br/> ';
        echo '8. Tokyo Drifts <br/> ';
        echo '9. Sixteen Candles <br/> ';
        echo '10. Caddyshack <br/> ';
      }
    }
    if (isset($_GET['favmovie'])) {
      echo 'Welcome to our site, ';
      //delete this line: echo $_COOKIE['username'];
      echo $_SESSION['username'];
      echo '!<br/>';
      echo 'My favorite movie is ';
      echo $_GET['favmovie'];
      echo ' <br/> ';
      $movierate = 5;
      echo 'My movie rating for this movie is: ';
      echo $movierate;
    }
  }

```

```

} else {
    echo 'My top ';
    echo $_GET['movienum'];
    echo ' moiveis are: ';
    echo '<br />';
    listmovies_1();
    if($_GET['movienum']==10){
        listmovies_2();
    }
}
?>
</body>
</html>

```

**How it works**

ခုဒီမှာဆိုရင် If / else statement ကို ပါ သုံးထားပါတယ်။ နားမလည်ရင် ရှေ့မှာ ရှင်းပြထားတာနဲ့ သေချာ ပြန်ဖတ်ကြည့်ဖို့ တိုက်တွန်းပါတယ်။ ကျွန်တော်တို့ login ဝင်ပြီးတာနဲ့ movie1.php က user ကို link ခု ပေးထားပါတယ်။ favourite movie , Top 5 and Top 10 movies တွေဆိုပြီးတော့ ပေးထားပါတယ်။ ဘယ် link ကို ဘဲ ရွေးရွေး ( 5 or 10 ) \$movienum ဆိုတဲ့ variable ရဲ့ value ကို ဘဲ ရွေးချယ်တာပါ။ ပြီးတာနဲ့ movieslite.php အလုပ်လုပ်ပါတယ်။ အဲဒီမှာ သူက function ၂ ခုကို set up လုပ်ပါတယ်။ listmovies\_1 and listmovies\_2 တွေပေါ့။

if (isset(\$\_GET['favmovie'])) { ဆိုတာကို ကျွန်တော်တို့ ထပ်ပေါင်း ရေးခဲ့ပါသေးတယ်။ ဒီမှာ isset ဆိုတာက ဘာလဲဆိုတော့ -

isset function က variable သတ်မှတ်ထားတာ ရှိလား မရှိဘူးလားဆိုတာ စစ်တာပါ။ သူ့ရဲ့ ( variable ) ရဲ့ value ကိုတော့ မစစ်ပါဘူး။ ဆိုလိုတာက favmovie ဆိုတာ သတ်မှတ်ထားတာရှိလား ။ ရှိရင် ပြမယ်။ မရှိရင် မပြဘူးပေါ့။

ALL about arrays

ရှေ့မှာလည်း array အကြောင်းကို အတန်အသင့်ရှင်းပြခဲ့ပြီးပါပြီ။ ခု နည်းနည်းလေး ထပ်ပေါင်းရှင်း ပြပါဦးမယ်။ array နဲ့ one variable အောက်မှာ multiple data ကို store လုပ်လို့ရတယ်ဆိုတာ သိခဲ့ပြီးပြီနော်။ ထပ်ရှင်းပြဦးမယ်။

```
<?php
    $husband = array(
        'firstname' => 'MgBa',
        'lastname' => 'Oo',
        'age' => 30
    );
    echo $husband['firstname'];
?>
```

ဒီမှာ ထပ်လေ့လာရမှာက = ကို မသုံးဘဲ => ကို သုံးထားတာပါ။ ဒါဟာ ဘာအဓိပ္ပာယ်လဲဆိုတော့ key တွေရဲ့ value ကို assign လုပ်တဲ့အခါမှာ => ကို သုံးရတာပါ။ ဒီမှာဆိုရင် firstname က key ဖြစ်ပြီး သူ့ value က MgBa ဆိုတဲ့ဟာပါ။ ဒီတော့

array (key => value); ဖြစ်သွားတာပေါ့။

နောက်ပြီးတော့ ခုလိုမျိုး variable တစ်ခုထဲမှာ store မလုပ်ဘဲ တစ်ခုချင်းစီလုပ်မယ်ဆိုလည်း အောက်ပါအတိုင်း array ကို တိုက်ရိုက် လုပ်လို့ရပါတယ်။ သူ့ရဲ့ key ကို ဘယ်လို reference လုပ်ရသလဲဆိုရင်-

```
<?Php
    $husband['firstname'] => 'MgBa';
    $husband['lastname'] => 'Oo';
    $husband['age'] => 30;
?>
```

ဒီလိုပုံစံတွေကို သင်ဟာ array အခန်းမသင်ခင်ကတည်း အသုံးပြုခဲ့ဖူးပါတယ်။ စဉ်းစားမိပါသလား။ ကျွန်တော်တို့ ရှေ့မှာသုံးခဲ့တဲ့ \$\_POST, \$\_GET, \$\_COOKIE, \$\_SESSION တို့တွေလေ။ သူတို့ကလည်း array တွေပါ။



သင့်အနေနဲ့ array တွေထဲမှာ နောက်ထပ် array တွေ ထပ်ပေါင်းထည့်လို့ရပါသေးတယ်။ အဲဒါကို ဘာလို့ ခေါ်လဲဆိုရင် Multidimensional arrays လို့ခေါ်ပါတယ်။ ဥပမာတစ်ခုနဲ့ ရှင်းပြမယ်။ စားပွဲတစ်လုံးမှာ လူနှစ်ယောက် ထိုင်နေတယ်။ တကယ်လို့ နောက်ထပ် စားပွဲ တစ်လုံး ထပ်ဆက်ပြီး လူနောက်ထပ် ၂ ယောက်ကို ထပ်ပေါင်း ထိုင်မယ်။ သူတို့ရဲ့ data တွေကို ဘယ်လို စုမလဲ။ စနစ်တကျစုမယ်ဆိုရင် table 1 မှာ ဘယ်သူနဲ့ ဘယ်သူ Table 2 မှာက ဘယ်သူနဲ့ဘယ်သူ ... စသဖြင့် စုရမယ်လေ ဟုတ်တယ်မလား။ အဲဒါကို Code အနေနဲ့ ဒီလိုလေးရေးလို့ရပါတယ်။

```
<?php
    $table[1] = array(
        'husband' => array (
            'firstname' => 'MgBa';
            'lastname' => 'Oo';
            'age' => 30;
        ),
        'wife' => array (
            'firstname' => 'MohMoh';
            'lastname' => 'Thu';
            'age' => 29;
        )
    );
?>
```

ဟုတ်ပြီ။ တစ်ယောက်ယောက်က မေးလိုက်မယ်။ Table 1 မှာ ထိုင်နေတဲ့ သူတွေရဲ့ firstname တွေက ဘာလဲ ဟေ့ ဆိုရင် သင်က အောက်ပါအတိုင်း ထုတ်ပေးလိုက်ယုံဘဲ။

```
<?php
    echo $table[1]['husband']['firstname'];
    echo ' and ';
    echo $table[1]['wife']['firstname'];
?>
```

ဒီလိုဆိုရင် Output အနေနဲ့ MgBa and MohMoh ဆိုပြီး ထုတ်ပေးမှာပါ။ (and ရဲ့ ရှေ့နဲ့နောက်မှာ space bar တစ်ချက်စီ ခြားထားတာကို သတိပြုပါ။ )

နောက်တစ်ခုက သင့် အနေနဲ့ simple store အနေနဲ့ list တစ်ခုကို သိမ်းချင်တယ်။ key တွေကိုလည်း မသတ်မှတ်ချင်ဘူး။ ဒီလိုဆိုရင် PHP က သူ့ဟာသူ numeric keys တွေကို auto assign လုပ်သွားမှာပါ။ သတိပြုရမှာက auto လုပ်တဲ့အခါမှာ 0 (သုည) က စပါတယ်။ ဥပမာ-

```
<?php
    $flavours[] = 'Blue berry';
    $flavours[] = 'Sweety';
    $flavours[] = 'Strawberry';
    $flavours[] = 'Candy';
```

?>

ဒါကို PHP က key ကို ဘယ်လို assign လုပ်လိုက်သလဲဆိုရင်

```
$flavours[0] = 'Blue berry';
$flavours[1] = 'Sweety';
$flavours[2] = 'Strawberry';
$flavours[3] = 'Candy';
```

ဆိုပြီး auto သူ့ဟာသူ သတ်မှတ်လိုက်ပါတယ်။ သင့်အနေနဲ့က ဒါကို မြင်ရမှာမဟုတ်ဘူးနော်။ စိတ်ထဲမှာ မှတ်ထားဖို့ဘဲ။ ဒီကောင်ကို ဘယ်လိုခေါ်သုံးရမလဲဆိုတော့

```
<?php
    echo $flavours[0];
    echo ' , ' ;
    echo $flavours[3];
```

?>

ဆိုရင် output အနေနဲ့က Blue berry , Candy ဆိုပြီး ထွက်လာမှာပါ။ အခုချိန်ထိ သိပ် အခက်အခဲရှိမယ် မထင်ဘူး။

Sorting Arrays

ကျွန်တော်တို့က array တွေရဲ့ output တွေကို sorting လုပ်လို့ရပါတယ်။ လုပ်လို့ရတဲ့ နည်းတွေထဲက ခု အခြေခံနဲ့ပတ်သတ်ပြီး နည်းနည်း ပြောသွားပါမယ်။

၁။ sort(\$array) : ဒါက array value တွေကို ကြီးစဉ်ငယ်လိုက်စီတာပါ။ ascending value order ပေါ့။

၂။ rsort(\$array): ဒါက array value တွေကို ငယ်စဉ်ကြီးလိုက်စီတာပါ။ descending value order ပေါ့။

ထုံးစံအတိုင်း quick test လေးလုပ်ကြည့်မယ်။

```
<?php
    $flavours[] = 'Blue berry';
    $flavours[] = 'Sweety';
    $flavours[] = 'Strawberry';
    $flavours[] = 'Candy';

    sort($flavours);
    print_r($flavours);
?>
```

ဒီမှာ print\_r ဆိုတဲ့ function လေးအကြောင်း နည်းနည်းရှင်းပြပါမယ်။ သူ့ရဲ့ တာဝန်ကတော့ variable တွေကို ထုတ်ပြဖို့ဘဲဖြစ်ပါတယ်။ ဒီလိုဆိုရင် user တွေအနေနဲ့ variable ကို မြင်ရမှာပါ။ ဘယ်မှာ အသုံးပြုတာများသလဲဆိုရင်တော့ array တွေကို ပြန်စစ်တဲ့အခါမှာ သုံးတာများပါတယ်။

ခုတစ်ခါ ကျွန်တော်တို့ ရေးနေတဲ့ moviesite.php ကို array ပေါင်းမယ်။ sorting လုပ်မယ်။ foreach သုံးမယ်။ moviesite.php မှာ highlight ပြထားတဲ့နေရာတွေကို ပြင်ပါမယ်။



```
<?php
```

```
session_start();
```

```
//check to see if user has logged in with a valid password
```

```
if ($_SESSION['authuser'] !=1 ) {
```

```
    echo 'Sorry, but you don\'t have permission to view this page!';
```

```
    exit();
```

```
}
```

```
?>
```

```
<html>
```

```
<head>
```

```
<title> My Movie Site - <?php echo $_GET['favmovie']; ?> </title>
```

```
</head>
```

```
<body>
```

```
<?php include 'header.php'; ?>
```

```
<?php
```

```
$favmovies = array (
```

```
    'ThaGyanMoe',
```

```
    'Saving Private Ryan',
```

```
    'Pear Harbour',
```

```
    'Enemy at the gate',
```

```
    'Matrix',
```

```
    'Terminator',
```

```
    'Star war',
```

```
    'Tokyo Drifts',
```

```
    'Sixteen Caddles',
```

```
    'Caddyshack'
```

```
);
```

```
//delete this lines
```

```
/*function listmovies_1() {
```

```
    echo '1. ThaGyanMoe <br/> ';
```

```
    echo '2. Saving Private Ryan <br/> ';
```

```
    echo '3. Pearl Harbour <br/> ';
```

```
    echo '4. Enemy at the gate <br/> ';
```

```
    echo '5. Matrix <br/> ';
```

```
}
```

```

function listmovies_2() {
    echo '6. Terminator 2 <br/> ';
    echo '7. Star War <br/> ';
    echo '8. Tokyo Drifts <br/> ';
    echo '9. Sixteen Candles <br/> ';
    echo '10. Caddyshack <br/> ';
}
*/
// End of deleted lines

if (isset($_GET['favmovie'])) {
    echo 'Welcome to our site, ';
    //delete this line: echo $_COOKIE['username'];
    echo $_SESSION['username'];
    echo '!<br/>';
    echo 'My favorite movie is ';
    echo $_GET['favmovie'];
    echo ' <br/> ';
    $movierate = 5;
    echo 'My movie rating for this movie is: ';
    echo $movierate;
} else {
    echo 'My top 10 favourites movies are :<br /> ';
    if (isset($_GET['sorted']))(
        sort($favmovies);
    )
    echo '<ol>';
    foreach($favmovies as $movie){
        echo '<li>';
        echo $movie;
        echo '</li>';
    }
    echo '</ol>';
}
?>
</body>
</html>

```

နောက်တစ်ခု movie1.php ကို ပြင်ဆင်ပါမယ်။

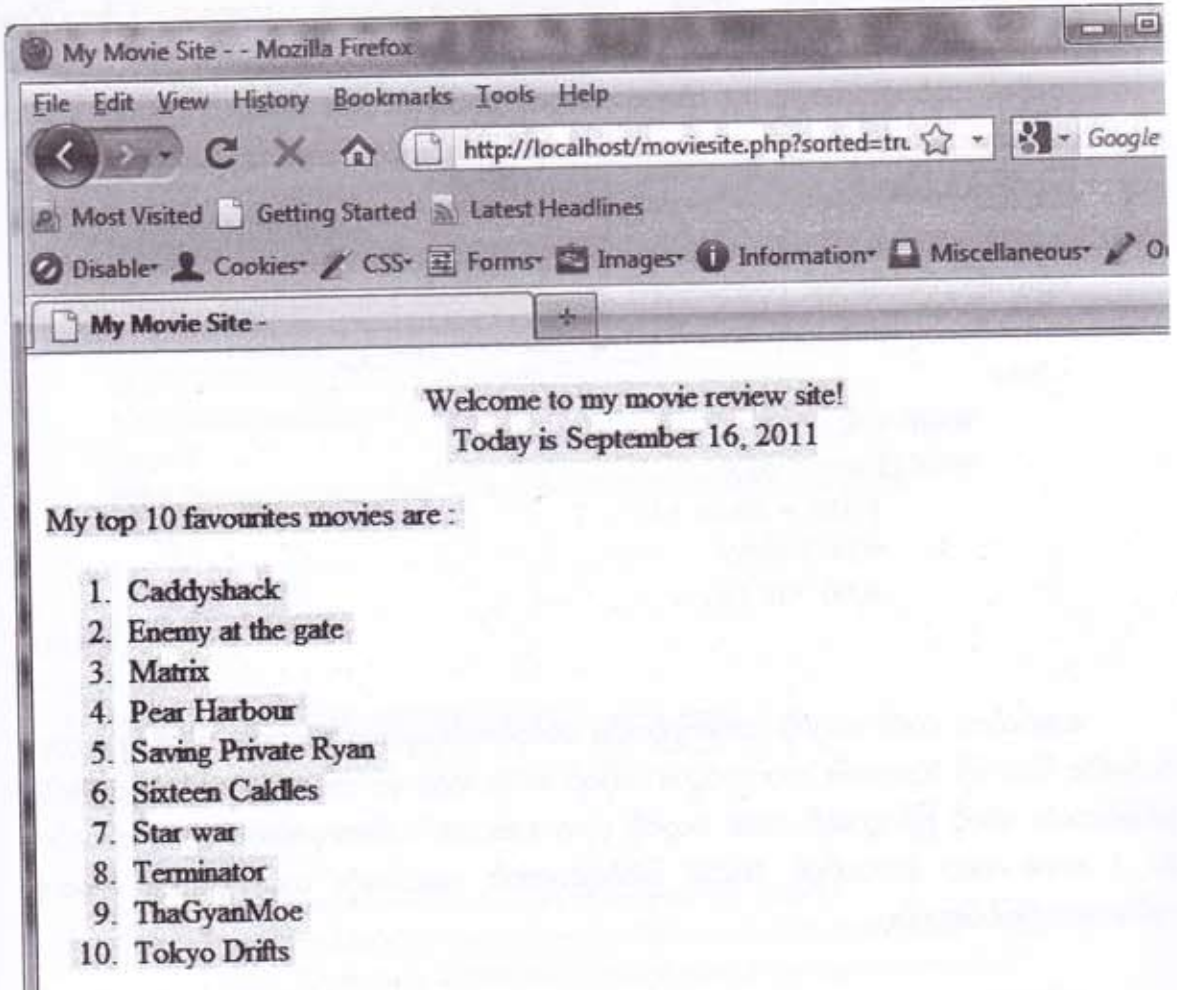
```
<?php
//delete this line: setcookie('username', 'MgMg', time() + 60);
session_start();
$_SESSION['username'] = $_POST['user'];
$_SESSION['userpass'] = $_POST['pass'];
$_SESSION['authuser'] = 0;
//Check username and password information
if (($_SESSION['username'] == 'MgMg') and
    ($_SESSION['userpass'] == '12345')) {
    $_SESSION['authuser'] = 1;
} else {
    echo 'Sorry, but you don\'t have permission to view this page!';
    exit();
}
?>
<html>
<head >
<title> Find my Favorite Movie! </title >
</head >
<body>
<?php include 'header.php'; ?>
<?php
    $myfavmovie = urlencode('Enemy at the gate');
    echo " <a href=\"moviesite.php?favmovie=$myfavmovie\" > ";
    echo "Click here to see information about my favorite movie!";
    echo " </a > ";
?>
<br/>
<!-- delete this lines
<a href="moviesite.php?movienum=5">Click hereto see my top 5 movies.</a>
<br/>
<a href="moviesite.php?movienum=10">Click hereto see my top 10 movies.</a>
End of deleted lines -->
<a href="moviesite.php">Click here to see top 10 movies</a> <br />
```



<a href="moviesite.php?sorted=true">Click here to see my top 10 movies sorted alpha-  
betically</a>

</body>

</html>



How it works

ပထမဆုံး movielist ကို \$favmovies ထဲကို array function နဲ့ ထည့်လိုက်တယ်။ ဒါဆိုရင် သင်ဟာ foreach function ကိုသုံးပြီး array ထဲကဟာတွေကို တစ်ခုချင်းစီ ထုတ်ပြန်နိုင်ပြီပေါ့။ ပုံမှာပြထားတာကတော့ array ကနေ ထုတ်ပေးလိုက်တာတွေကို sorting ပြန်လုပ်ပြီးမှ ထုတ်ပြထားတာပါ။ \$\_GET['sorted'] ဆိုတဲ့ variable ကို သုံးခဲ့တာ တွေ့ပါလိမ့်မယ်။

ခုဆိုရင် သင်ဟာ movie list ထဲကဟာတွေကို ထုတ်ပြနိုင်ပါပြီ။ ဒါပေမယ့် သင်က ၂ ခု ၃ ခုဘဲ ထုတ်ပြချင်တယ်ဆိုရင်ရော ... ??? ဘယ်လိုလုပ်ကြမလဲ။ ဟုတ်ကဲ့.. ကျွန်တော်တို့ ဆက်ပြီး ကြိုးစားကြည့်ကြပါမယ်။

စဉ်းစားကြည့်မယ်ဆိုရင် လိုသလောက် အရေအတွက်ကိုဘဲ ထုတ်ပြဖို့ဆိုရင် while loop ကို သုံးရမယ်လေ။ ဒီတော့ while loop လေးကို နည်းနည်းလောက် ပြန်ကြည့်ရအောင်။

```
<?php
    $num = 0;
    while ($num < 5) {
        $num = $num + 1;
        echo $num;
        echo '<br />';
    }
?>
```

အောက်က code တွေကို ထပ်ကြည့်ပါဦး။ အလုပ်လုပ်ပုံချင်းက အတူတူဘဲ။ ကွဲသွားတာက do/while loop ကို သုံးတာပါ။ ဘာကွာသွားလဲဆိုရင် while loop မှာ condition တစ်ခုခုမှာ FALSE ဖြစ်နိုင်တယ်။ အဲလို ဖြစ်သွားရင် code တွေကို လုံးဝ execute လုပ်ပေးမှာမဟုတ်ဘူး။ တကယ်လို့ do / while loop နဲ့သာဆိုရင် FALSE ဖြစ်ခဲ့ရင်တောင် အနည်းဆုံး တစ်ကြိမ်တော့ execute လုပ်ပေးမှာဘဲဖြစ်ပါတယ်။

```
<?php
    $num = 0;
    do {
        $num = $num+1;
```

```

echo $num;
echo '<br />';
}while($num < 5)

```

?>

ခု while loop ကို အသုံးပြုကြည့်ကြရအောင်။ movie1.php ကို နည်းနည်း ပြင်ရေးပါမယ်။

<?php

```

//delete this line: setcookie('username', 'MgMg', time() + 60);
session_start();
$_SESSION['username'] = $_POST['user'];
$_SESSION['userpass'] = $_POST['pass'];
$_SESSION['authuser'] = 0;

```

```

//Check username and password information
if (($_SESSION['username'] == 'MgMg') and
($_SESSION['userpass'] == '12345')) {
$_SESSION['authuser'] = 1;
} else {
echo 'Sorry, but you don\'t have permission to view this page!';
exit();
}

```

?>

```

<html>
<head >
<title> Find my Favorite Movie! </title >
</head >
<body>

```

```

<?php include 'header.php'; ?>

```

```

<?php
    $myfavmovie = urlencode('Enemy at the gate');
    echo " <a href=\"moviesite.php?favmovie=$myfavmovie\" > ";
    echo "Click here to see information about my favorite movie!";
    echo " </a > ";

```

?>



```

<br/>
<!-- delete this lines
<a href="moviesite.php?movienum=5">Click hereto see my top 5 movies.</a>
<br/>
<a href="moviesite.php?movienum=10">Click hereto see my top 10 movies.</a>
End of deleted lines -->
<!-- delete this lines
<a href="moviesite.php">Click here to see top 10 movies</a><br />
<a href="moviesite.php?sorted=true">Click here to see my top 10 movies sorted
alphabetically</a>
End of deleted lines -->
<br />
Or choose how many movies you would like to see?<br />
<form action = "moviesite.php" method="POST">
    <p>Enter movies number up to 10:<input type="text" name="num"
        .   maxlength="2" size="2" /><br />
        Check to sort them alphabetically: <input type="checkbox" name="sorted"
/>
        </p>
        <input type="submit" name="submit" value="Submit" />
    </form>
</body>
</html>

```

moviesite.php ကိုလည်း နည်းနည်းလေး ပြင်ရေးဖို့ လိုပါတယ်။ အောက်ပါအတိုင်းလေးပြင်ပေးပါ။

```

<?php
    session_start();

    //check to see if user has logged in with a valid password
    if ($_SESSION['authuser'] !=1 ) {
        echo 'Sorry, but you don\'t have permission to view this page!';
        exit();
    }
?>
<html>
<head>
<title> My Movie Site - <?php echo $_GET['favmovie']; ?> </title>
</head>
<body>
<?php include 'header.php'; ?>
<?php
    $favmovies = array (
        'ThaGyanMoe',
        'Saving Private Ryan',
        'Pear Harbour',
        'Enemy at the gate',
        'Matrix',
        'Terminator',
        'Star war',
        'Tokyo Drifts',
        'Sixteen Candles',
        'Caddyshack'
    );

    //delete this lines
    /*function listmovies_1() {
        echo '1. ThaGyanMoe <br/> ';
        echo '2. Saving Private Ryan <br/> ';
        echo '3. Pearl Harbour <br/> ';
        echo '4. Enemy at the gate <br/> ';
        echo '5. Matrix <br/> ';
    }

```

```

function listmovies_2() {
    echo '6. Terminator 2 <br/> ';
    echo '7. Star War <br/> ';
    echo '8. Tokyo Drifts <br/> ';
    echo '9. Sixteen Candles <br/> ';
    echo '10. Caddyshack <br/> ';
}
*/
// End of deleted lines
if (isset($_GET['favmovie'])) {
    echo 'Welcome to our site, ';
    //delete this line: echo $_COOKIE['username'];
    echo $_SESSION['username'];
    echo '!<br/>';
    echo 'My favorite movie is ';
    echo $_GET['favmovie'];
    echo ' <br/> ';
    $movierate = 5;
    echo 'My movie rating for this movie is: ';
    echo $movierate;
} else {
    echo 'My top ' . $_POST['num'] . ' favourites movies<br /> ';
    if (isset($_POST['sorted'])) {
        sort($favmovies);
        echo 'sorted alphabetically: ';
    }
    echo ' are : <br />';
    // delete these lines
    /*
    echo '<ol>';
    foreach($favmovies as $movie){
        echo '<li>';
        echo $movie;
        echo '</li>';
    }
    echo '</ol>';
    //end of deleted lines

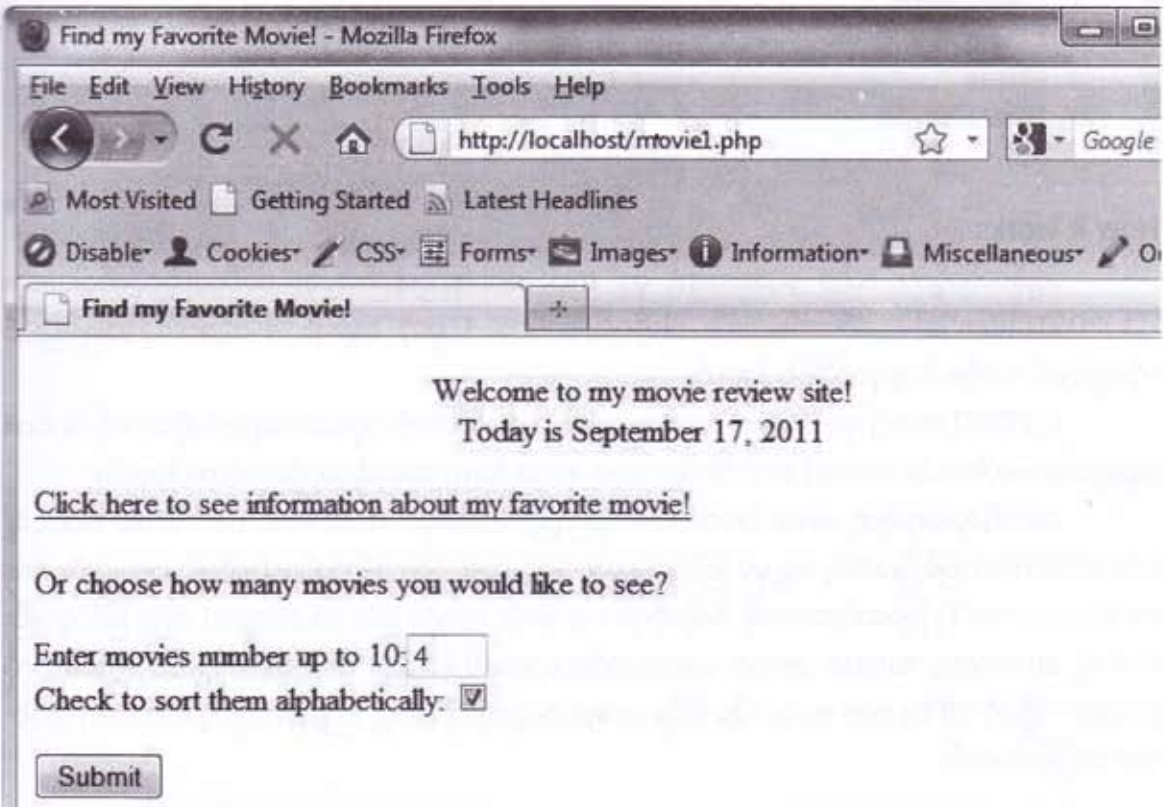
```

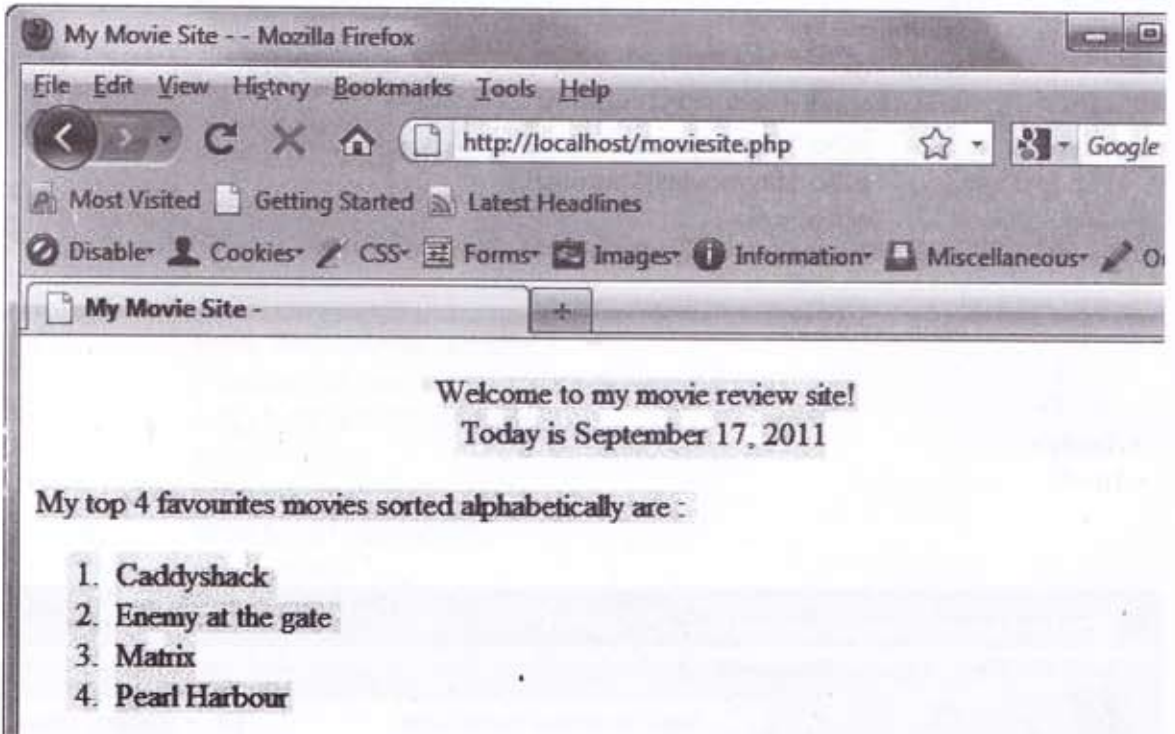


```

*/
$numlist = 0;
echo '<ol>';
while($numlist < $_POST['num']){
    echo '<li>';
    echo $favmovies[$numlist];
    echo '</li>';
    $numlist = $numlist + 1;
}
echo '</ol>';
}
?>
</body>
</html>

```





### How it works

ကျွန်တော်တို့က echo statement မှာ trick လေးနည်းနည်းလေး ထပ်ထည့်လိုက်ပါတယ်။ '\$\_POST['num']. ' ဆိုတာလေးပါ။ ဒါဆိုရင် သင်က 4 ခုကို ရွေးရင် 4 လို့ပေါ်နေမှာဖြစ်ပြီး 5 ကိုရွေးခဲ့ရင် 5 လို့ပေါ်နေမှာဘဲဖြစ်ပါတယ်။

\$\_POST['num'] မှာ movie အရေအတွက် ဘယ်လောက် ထုတ်ပေးရမလဲဆိုတာကို သင်က ရွေးချယ်ပေးရပါတယ်။ movie1.php file မှာ user input form အနေနဲ့ထည့်ပေးထားပါတယ်။

နောက်ဆုံးရေးခဲ့တဲ့ while block လေးကိုကြည့်ပါ။ အဲဒီမှာ increment 1 က while block ရဲ့ last statemnet ဖြစ်ပြီးတော့ ရှေ့မှာ ပြခဲ့တဲ့ နမူနာတွေ အရ စဉ်းစားကြည့်မယ်ဆိုရင် ပထမဆုံး Ans က \$favmovies[1] ဖြစ်သင့်တာပေါ့။ ဒီလိုဆိုရင် ပထမဆုံး movie title က skipped over ဖြစ်သွားပြီ။ PHP ရဲ့ automatic number arrays သဘောတရား အရ 0 (သုည) က စတာမဟုတ်ဘူးလား။ ဒီလို problem မျိုးကို off by one error လို့ခေါ်ပြီး arrays တွေသုံးရင် တွေ့ ကြုံရလေ့ရှိတဲ့ common problem ဘဲဖြစ်ပါတယ်။

ဒါကြောင့်မို့လို့ pointer variable တွေယူတဲ့နေရာမှာ သေချာ စဉ်းစားပြီးမှ ယူသင့်တယ်။  
\$numlist = 0 က စခွဲတော့ ပထမဆုံး အဖြေက \$favmoives[0] ဖြစ်သွားမယ်။ ပြီးရင် ၁ ပေါင်းမယ်။  
ဒါလေးကို သေချာစဉ်းစားပြီး လေ့လာဖို့ အကြံပြုလိုပါတယ်။

**concatenate**

concatenate ဆိုတာက quote တွေကနေ virtually undetected ဖြစ်အောင် လုပ်ဆောင်ပေးတာပါ။ . (dot) လေး ခံပြီးသုံးရပါတယ်။ အောက်ပါနမူနာလေးကိုကြည့်ပါ။  
echo 'My top ' . \$\_POST['num'] . 'movies';



## Exercises

**For chapter (2)**

1. Go back to your `date.php` file, and, instead of displaying only the number of days in the current month, add a few lines that say:
 

The month is \_\_\_\_\_ .

There are \_\_\_\_\_ days in this month.

There are \_\_\_\_\_ months left in the current year.
2. On your movie web site, write a file that displays the following line at the bottom center of every page of your site, with a link to your e-mail address.
 

This site developed by: ENTER YOUR NAME HERE .
3. Write a program that displays a different message based on the time of day. For example, have
 

the site display "Good Morning!" if it is accessed in the morning.
4. Write a program that formats a block of text (to be input by the user) based on preferences chosen by the user. Give your user options for color of text, font choice, and size. Display the output on a new page.
5. In the program you created in step 4, allow your users the option of saving the information for the next time they visit. If they choose "yes," save the information in a cookie.
6. Using functions, write a program that keeps track of how many times a visitor has loaded the page.

## Chapter(3) Using PHP with MySQL

အခုဆိုရင် သင်ဟာ PHP နဲ့ တော်တော်လေးကို ရင်းနှီးခဲ့ပြီးပြီနော်။ function တွေသုံးတတ်နေပြီ။ include တွေဘာတွေနဲ့တောင် တွဲသုံးတတ်နေပြီ။ ဒီအခန်းမှာတော့ PHP ကို MySQL database type နဲ့ ဘယ်လို ဆက်စပ် အသုံးပြုရသလဲဆိုတာကို သင်ပေးသွားမှာပါ။ ဒါမှ dynamic website တစ်ခုကို အခြေခံကျကျ လုပ်နိုင်မှာပါ။ သင့် အနေနဲ့ MySQL ကို သိပြီးသားဖြစ်ချင်လည်းဖြစ်မယ်။ မသိရသေးသူတွေအတွက် MySQL ကို အခြေခံက စပြီး ဖြည်းဖြည်းချင်း ရှင်းလင်းသွားပါမယ်။ ဒီသင်ခန်းစာပြီးသွားရင် သင့်အနေနဲ့ -

- MySQL database ကို နားလည်သွားပါမယ်။
- MySQL database ထဲက Data တွေကို ကြည့်နိုင်မယ်။
- သင့်ရဲ့ website ကနေ database ကို လှမ်းချိတ်ဆက်နိုင်ပါမယ်။
- သင့် website ရဲ့ လိုအပ်တဲ့နေရာမှာလိုအပ်တဲ့ data ကို ဘဲ သီးခြား ထုတ်ယူနိုင်မယ်။
- Problems တစ်ချို့ကို သင်ကိုယ်တိုင် ဖြေရှင်းတတ်လာမယ်။

### Overview of MySQL structure and Syntax

database ဆိုတာ information တွေကို Store လုပ်ထားတဲ့ဟာဘဲဖြစ်ပါတယ်။ တစ်စုံတစ်ယောက်က အကြောင်းအရာတစ်ခု သို့မဟုတ် တစ်ခုထက်ပိုတာတွေကို record လုပ်ထားမယ် ပြီးရင် ကျန်တဲ့သူတွေက access လုပ်နိုင်ရမယ်။ ဒေတာဘေ့စ်ထဲမှာ name , address, phone number, medical record, police record, email address, sale transaction, music or video informations, and more... တွေ မှတ်ထားလေ့ရှိပါတယ်။ သင့် အနေနဲ့ကတော့ ကျွန်တော်တို့ ရှေ့သင်ခန်းစာမှာလုပ်ခဲ့တဲ့ movie ကိုဘဲ ဆက်လုပ်သွားရမှာပါ။ MySQL commands တွေကို command prompt သို့မဟုတ် PHP ကနေ ပေးနိုင်ပါတယ်။ ကျွန်တော်တို့ ခုက စပြီး PHP နဲ့ဘဲ ပေးမှာပါ။

### MySQL structure

MySQL structure ကို နည်းနည်းလေး ရှင်းပြပါဦးမယ်။ MySQL database က relational database system ပါ။ relational database system မှာ information တွေကို separate လုပ်ပြီး tables တွေထဲမှာ သိမ်းဆည်းထားလို့ရတယ်။



ဒါမှမဟုတ် group လုပ်ပြီး သိမ်းလို့ရပါတယ်။ table တစ်ခုချင်းစီထဲမှာလည်း မတူညီတဲ့ fields တွေ ထပ်ရှိလို့ရပါသေးတယ်။ Eg. table တစ်ခုထဲမှာ field တစ်ခုက firstname ဖြစ်ပြီးတော့ နောက် field တစ်ခုမှာ သူ့ရဲ့ last name ပေါ့။ ဒေတာတွေကို တစ်ခုချင်းစီ ကို လိုသလို ထုတ်သုံးနိုင်ဖို့ဆိုရာ Data structure ကို တည်ဆောက်တဲ့ပေါ်မှာလည်း မူတည်ပါတယ်။ Field တွေက text, date, numbers , and so on တွေကို ကိုင်တွယ် ထိန်းသိမ်းနိုင်ပါတယ်။

သင်တည်ဆောက်မယ့် MySQL tables တွေကတော့ သင် ဘယ်လို information မျိုးကို သိမ်းမလဲဆိုတဲ့ အပေါ် မူတည်ပြီး ကွဲပြား နိုင်ပါတယ်။ MySQL ဟာ tables တွေကို linked ချိတ်ပေးနိုင်ပါတယ်။ tables တွေထဲမှာ ပါတဲ့ တစ်ချို့ fields တွေရဲ့ values တွေဟာ တူတူ တွေဘဲဖြစ်ပါတယ်။

ဥပမာ .. table တစ်ခုမှာ Customer's\_ID, Customer's name, Customer's address တွေပါမယ်။ နောက် table တစ်ခုထဲမှာ Customer's\_ID, Customer's past order ပါမယ်။ ဒါဆိုရင် common အနေနဲ့ Customer's\_ID field ဖြစ်ပြီးတော့ ID number က တူတူ ဖြစ်တဲ့အတွက် table နှစ်ခုကို linked together လုပ်လို့ရပါတယ်။ ဒါက ဘာကို ပြပေးနိုင်သလဲဆိုရင် table ၂ ခုလုံးမှာ ရှိတဲ့ data တွေကို တစ်ခါတည်းနဲ့ သင့်ကို ပြပေးနိုင်ပါတယ်။ ဘာလို့ဆိုတော့ ဆက်စပ်နေတာကိုး ။

| MySQL field type | Description   | Example  |
|------------------|---|--|
| char(length)     | Any character - based data can be stored in this field, but the field will have a fixed length denoted by the value in the parentheses  | Customer's State field always has two characters and would use char(2) |
| varchar(length)  | Any character - based data can be in this field, and the data can vary in length from 0 up to 255 characters. The maximum length of the field is denoted in parentheses.  | Customer 's Address field has letters and numbers and varies in length |
| int(length)      | Integers that can range from 2,147,483,648 to +2,147,483,647 can be stored in this field. The length parameter limits the number of digits that can be shown, not the value. Mathematical functions can be performed on data in this field. | Quantity of a product on hand.   |



| MySQL field type                      | Description   | Example  |
|---------------------------------------|---|--|
| int(length) unsigned                  | Positive integers (and zero) up to 4,294,967,295 can be in this field. The length parameter limits the number of digits that can be displayed. Mathematical functions can be performed on data in this field. | Customer ID (if entirely numerical)  |
| text                                  | Any character - based data can be in this field, with a maximum size of 65,536 characters.  | Comments field that allows longer text to be stored without limiting the field to 255 characters         |
| decimal(length,dec)                   | Numeric field that can store decimals. The length parameter limits the number of digits that can be displayed, and the dec parameter limits the number of decimal places that can be stored.                  | Prices. For example, a price field that would store prices up to 999.99 would be defined as decimal(5,2) |
| enum( " option1 " , " option2 " , ... | Allows only certain values to be stored in this field, such as " true " and " false ," or a list of states. 65,535 different options are allowed.   | Gender field for your users will have a value of either " male " or " female. "                          |
| date                                  | Stores a date in YYYY - MM - DD format.   | Date of an order, a birthday, or the date a user joined as a registered user                             |
| time                                  | Stores time in hh:mm:ss format.   | Time a news article was added to the web site.   |
| datetime                              | Multipurpose field that stores both the date and time together as YYYY - MM - DD hh:mm:ss   | Last date and time a user visited your web page  |

အပေါ်ဇယားမှာ ပြခဲ့တာတွေက အဓိက အများဆုံး သုံးနေတာတွေဖြစ်ပါတယ်။ အောက်မှာ ပြထားတဲ့ဟာတွေကတော့ တစ်ခါတစ်ရံမှာ ကိုယ့်ရဲ့ လိုအပ်ချက်ပေါ်မူတည်ပြီး သုံးရတဲ့ဟာတွေဖြစ်ပါတယ်။

| MySQL field type  | Description  |
|-------------------|--|
| year(length)      | Stores a year. By default, the year is four digits, though it is possible to specify a two - digit format by using the length parameter.   |
| tinyint(length)   | Numeric field that stores integers from - 128 to 127. (Adding the unsigned parameter allows storage of 0 to 255.)  |
| smallint(length)  | Numeric field that stores integers from - 32,768 to 32,767. (Adding the unsigned parameter allows storage of 0 to 65,535.)   |
| mediumint(length) | Numeric field that stores integers from - 8,388,608 to 8,388,607. (Adding the unsigned parameter allows storage of 0 to 16,777,215.)   |
| bigint(length)    | Numeric field that stores integers from 9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. (Adding the unsigned parameter allows storage of 0 to 18,446,744,073,709,551,615.)                     |
| tinytext          | Allows storage of up to 255 characters.  |
| mediumtext        | Allows storage of up to 1,677,215 characters   |
| longtext          | Allows storage of up to 4,294,967,295 characters.  |
| blob              | Equal to a text field, except that it is case - sensitive when sorting and comparing. Stores up to 65,535 characters. blob and its derivatives (which follow) are generally used to store binary data. |
| tinyblob          | Equal to the tinytext field, except that it is case - sensitive when sorting and comparing; see blob , above.  |
| mediumblob        | Equal to the mediumtext field, except that it is case-sensitive when sorting and comparing; see blob , above.  |
| longblob          | Equal to the longtext field, except that it is case - sensitive when sorting and comparing; see blob , above.  |



Choosing the right field type

သင်ဟာ database တစ်ခုလုပ်တော့မယ်ဆိုရင် ဘယ် field type ကို ရွေးမယ်ဆိုတာ ကြိုတင်စဉ်းစားသင့်ပါတယ်။ ဒီလို စဉ်းစားနိုင်ဖို့အတွက် အောက်မှာ မေးခွန်းလေးတွေနဲ့ ပြထားပေးပါတယ်။ ကိုယ့်ကိုယ်ကို မေးခွန်းမေးကြည့်ပါ။

၁။ Field တစ်ခုမှာ numbers and letters တွေ နှစ်ခုလုံး ပါမှာလား။

အဖြေက Yes ဆိုရင် သင့်အနေနဲ့ char, varchar, text, tinytext, mediumtext, longtext, blob, tinyblob, longblob တွေကို စဉ်းစားရမှာပါ။ နောက်ပြီး ထပ်မေးပါ။ characters ဘယ်လောလောက် ပါနိုင်သလဲပေါ့။ တကယ်လို့ပေါ့

၁။ 0-255 characters, variable length ဆိုရင် varchar ကို သုံးသင့်တယ်။ space တွေကိုလည်း ထည့်တွက်မှာဖြစ်ပါတယ်။

၂။ 256-65536 characters ဆိုရင် text ကို သုံးသင့်တယ်။

၃။ 65537-1677215 characters ဆိုရင် mediumtext ကို သုံးသင့်တယ်။

၄။ 1677216-4294967295 characters ဆိုရင် longtext ကိုသုံးသင့်တယ်။

၅။ သင်အသုံးပြုမယ့်ဟာက letter or number ဖြစ်တဲ့အပြင် finite number တွေပါခဲ့မယ်ဆိုရင် enum ကို သုံးရပါမယ်။

၆။ သင်အသုံးပြုမယ့်ထဲမှာ date သို့မဟုတ် time only ဆိုရင် timestamp ကို အသုံးပြုရမယ်။ နှစ်ခုလုံးအတွက် တစ်ခါတည်း သိမ်းမယ်ဆိုရင် datetime ကိုသုံးရမယ်။ year တစ်ခုတည်းဆိုရင် year ကို အသုံးပြုလို့ရပါတယ်။



၇။ ကိုယ်အသုံးပြုမယ့် fields မှာ numbers and mathematical functions တွေဘဲပါမယ်ဆိုရင် အောက်ပါ တို့ကို အသုံးပြုမယ့် size ပေါ်မူတည်ပြီး သုံးနိုင်ပါတယ် -

- (ကက) -127 to 127 use tinyint
- (ခခ) -32768 to 32767 use smallint
- (ဂဂ) -8388608 to 8388607 use mediumint
- (ဃဃ) -2147483648 to 2147483647 use int
- (ငင) -9223372036854775808 to 9223372036854775807 use bigint
- (စစ) Integers from 0 to 255, use tinyint unsigned.
- (ဆဆ) Integers from 0 to 65,535, use smallint unsigned.
- (ဇဇ) Integers from 0 to 16,777,215, use mediumint unsigned.
- (ဈဈ) Integers from 0 to 4,294,967,295, use int unsigned.
- (ညည) Integers from 0 to 18,446,744,073,709,551,615, use bigint unsigned.
- (ဋဋ) Decimals with fixed decimal places, use dec

**null or not null**

MySQL server က database ထဲမှာ ရှိတဲ့ field တွေဟာ အလွတ် ဖြစ်နိုင်သလား မဖြစ်နိုင်ဘူးလားဆိုတာကိုလည်း သူက ဆန်းစစ်ပါတယ်။ ဖြစ်နိုင် မဖြစ်နိုင်ဆိုတာကို မိမိဘာသာ null , not null ဆိုတာနဲ့ သတ်မှတ်ပေးရတာဘဲဖြစ်ပါတယ်။ null လို့ သတ်မှတ်လိုက်ရင် အဲဒီ field မှာ ဘာမှမရှိလည်းဖြစ်ပါတယ်။ တကယ်လို့ not null လို့သာ သတ်မှတ် လိုက်မယ်ဆိုရင် field ထဲမှာ something or anything တစ်ခုခု ရှိရမှာဖြစ်ပါတယ်။ သူ့ည (0) ဟာဆိုရင် quantity အနေနဲ့ ဘာမှမရှိပေမယ့် data အနေနဲ့ကြည့်မယ်ဆိုရင်တော့ something ဖြစ်ပါတယ်။

not null သတ်မှတ်ထားတဲ့ Field တစ်ခုမှာ သင်က ဘာမှ မဖြည့်ခဲ့မိဘူးဆိုရင် MySQL က အဲဒီနေရာမှာ something အနေနဲ့ 0 (သုည) ကို default အနေနဲ့ ထည့်ပေးမှာဖြစ်ပါတယ်။ ဒါကြောင့်မို့လို့လည်း Data validation လုပ်မယ်ဆိုရင် MySQL data check ပေါ် မမှီခိုသင့်ပါဘူး။ PHP ပေါ်မှာသာ ပိုပြီးလုပ်သင့်ပါတယ်။ နောက်ပိုင်း စာအုပ်မှာ data validation တွေကိုပါ ရေးသားသွားဖို့ ရှိပါတယ်။

Indexes

MySQL ဟာ သူ့ထဲမှာ ရှိတဲ့ data တွေကို ရှာတဲ့အခါမှာ မြန်ဆန်စေဖို့အတွက် index ကို သုံးပါတယ်။ ဥပမာ အနေနဲ့ ပြောပြရရင် သင်ဟာ စတိုးဆိုင်တစ်ခုရဲ့ manager ဆိုပါစို့။ ဒီတော့ သင်က စာရင်းဇယားတွေကို ထိန်းသိမ်းရတော့မယ်။ receipt တွေကို ထိန်းရမယ်။ သင့်မှာ ဘောက်ချာတွေ အများရှိလာပြီ။ အဝတ်အထည်ဘောက်ချာတွေ၊ နို့ဆီဘောက်ချာတွေ၊ သယ်ယူပို့ဆောင်ခဘောက်ချာတွေ..စသဖြင့် အများကြီးရှိလာပြီ ။ သင်က ဒါတွေကို ဒီအတိုင်းဘဲ သိမ်းထားလိုက်မယ်။ နေ့စဉ်ဘောက်ချာတွေက ၅ နှစ်လောက်ကြာလာတဲ့အခါမှာ သင့်မှာ ထောင်နဲ့ သောင်းနဲ့ချီရှိလာမယ်လေ။ တစ်ရက်မှာ သူဌေးက လွန်ခဲ့တဲ့ ၄ နှစ်လောက်က Levis ဂျင်းဘောင်းဘီ ဘောက်ချာ တစ်ခုကို တောင်းမယ်ဆိုပါတော့။ သင် ရှာရပြီလေ။ တစ်ရွက်ချင်းတစ်ခုချင်း ရှာရပြီ။ သင့်မှာ ဒေတာ က ရှိတယ်။ ဒါပေမယ့် သေချာ စနစ်တကျ မစီထားခဲ့တော့ ရှာရခက်နေပြီ။ အစကတည်းက နှစ်အလိုက် ၊ နှစ်အလိုက်မှာမှ ပစ္စည်း အမျိုးအစားအလိုက် သေချာ စနစ်တကျ ခွဲထုတ်ထားပြီး သိမ်းထားခဲ့မယ်ဆိုရင် မြန်မြန်ဆန်ဆန် ရှာလိုရတာပေါ့။ ၂၀၀၆ ခုနှစ်ကဘောက်ချာကိုတောင်းရင် ၂၀၀၆ ထဲသွားရှာလိုက်ယုံဘဲ။ ဒီအထဲမှာမှ ဂျင်းဘောင်းဘီဆိုရင် ၂၀၀၆ ခုနှစ် ဂျင်းဘောင်းဘီ ဆိုတဲ့ အထဲသွားရှာလိုက်ရင် သင်လိုချင်တာ ချက်ချင်းရပြီလေ။ MySQL မှာလည်း ဒီလိုဘဲ ။ သူ့ အမျိုးအစားလိုက်လေးတွေ ခွဲထားပေးတာကို index လုပ်တယ်လို့ ခေါ်ပါတယ်။ MySQL table တွေမှာ index အနည်းဆုံး တစ်ခု လိုပါတယ်။ ဘာနဲ့ သတ်မှတ်သလဲဆိုရင် primary key ဆိုတာနဲ့သတ်မှတ်ပါတယ်။ ဒီ field ဟာဆိုရင် not null or UNIQUE ဖြစ်ရပါမယ်။ ဥပမာ Customer ID ပေါ့။ Customer တစ်ယောက်စီဟာ နာမည် တူတွေ ရှိနိုင်ပါတယ်။ ဥပမာ သင့်ဆီမှာ MgMg ဆိုတဲ့ Customer ဆိုတဲ့ အမည်တူ ၂ ယောက်ရှိတယ်ဆိုပါစို့။ သူတို့က နာမည် ဘဲ တူနိုင်ပါတယ်။ ကျွန်တော်တို့ Database ထဲမှာတော့ Customer ID ၂ ခုနဲ့ မှတ်ထားပြီလေ။ ၁ နဲ့ ၂ ပေါ့။ Customer ID 1 ရဲ့ receipt တွေကိုဘဲလိုချင်တယ်ဆိုရင်လည်း ရပြီလေ။ လွယ်လွယ်ကူကူ နဲ့ ထုတ်ပေးလို့ရပြီ။

Auto-Increment

field တစ်ခုမှာ data record တစ်ခု ထပ်ပေါင်းလာတိုင်း automatic ၁ ပေါင်းပေးချင်တယ်ဆိုရင် auto increment ကို သုံးရပါတယ်။ တစ်ချို့နေရာတွေမှာ ဒီကောင်က တော်တော်ကို အသုံးဝင်ပါတယ်။ ဥပမာ Customer ID နံပါတ် သတ်မှတ်မယ်ဆိုပါစို့။



ပထမတစ်ယောက်က နံပါတ် ၁ နောက် ၂၊ ၃၊ ၄၊ သူ့ဟာသူ အစဉ်အတိုင်း ပေါင်းပေးသွားတာပါ။ သူ့ရဲ့ command ကတော့ auto\_increment ဖြစ်ပါတယ်။

**Types of MySQL Tables and Storage Engine**

ခုဆိုရင် သင်ဟာ tables တွေရဲ့ general features တွေကို သိရှိပြီးတဲ့နောက်မှာ ထပ်သိရမှာက Tables type နှစ်မျိုးရှိတယ်ဆိုတာကိုပါ။ transaction-safe tables (TSTs) and Non-transaction-saffe tables (NTSTs) တွေဘဲဖြစ်ပါတယ်။ Transaction-safe table မှာဆိုရင် lost ဖြစ်သွားတဲ့ data တွေကို recoverd လုပ်လို့ရပါတယ်။ ချက်ချင်းလုပ်လိုက်တဲ့ data changes တွေကိုလည်း roll-back ပြန်လုပ်လို့ရပါတယ်။ Non-transaction-safe table ကတော့ memory usage ကို သုံးရာမှာ ပိုနည်းပြီး ပိုလည်း မြန်မြန်ဆန်ဆန်လုပ်ပါတယ်။ ဒါပေမယ့် သူ့မှာ data တွေ lost ဖြစ်သွားရင် recovery လုပ်လို့မရပါဘူး။ permanent delete ဖြစ်သွားတာပါ။ တစ်ခုခုမှားသွားရင် rollback ပြန်လုပ်လို့မရပါဘူး။

MySQL မှာ storage engine တွေ အများကြီးရှိပါတယ်။ အဲဒီထဲက most common ကတော့ အောက်ပါအတိုင်း ဖြစ်ပါတယ်။

- MyISAM
- MERGE
- MEMORY
- InnoDB
- BDB

**MyISAM**

သူကတော့ MySQL ရဲ့ default enging ဖြစ်ပြီးတော့ Non-transaction-safe table ကို support လုပ်ပါတယ်။ MySQL ရဲ့ field type အားလုံး၊ parameters and functions အားလုံးကို support လုပ်ပါတယ်။

**MERGE**

သူကတော့ ထပ်တူကျတဲ့ MyISAM tables တွေကို တစ်ခုတည်းအဖြစ် manipulate လုပ်ပေးနိုင်ပါတယ်။ သူလည်း Non-transaction-safe tables ကို support လုပ်ပါတယ်။



**MEMORY**

Temporary tables တွေမှာ အများဆုံး အသုံးပြုပါတယ် ဘာဖြစ်လို့လဲဆိုတော့ သူက အရမ်းကို မြန်ဆန်စွာ အလုပ်လုပ်နိုင်တဲ့အတွက်ကြောင့်ဘဲဖြစ်ပါတယ်။ ဒါပေမယ့် MyISAM table တွေရဲ့ features တော်တော်များများကို support မလုပ်နိုင်ပါဘူး ။ ဥပမာ auto\_increment or blob/text columns ။ သူ့ကို Log file တွေနဲ့ သုံးတဲ့ နေရာမှာ သုံးတာများပါတယ်။ Memory ထဲမှာ မှတ်ထားတာဖြစ်တဲ့အတွက် စက်ကို shutdown ချလိုက်ရင် ပျောက်သွားမှာပါ။ NTSTs ကိုဘဲ support လုပ်ပါတယ်။

**InnoDB**

သူကတော့ transaction-safe table ကို support လုပ်ပါတယ်။ အရမ်းကိုကြီးတဲ့ database တွေမှာ သုံးပါတယ်။

**BDB**

BDB or BerkeleyDB လို့လည်းခေါ်ပြီးတော့ transaction-safe table ကို support လုပ်ပါတယ်။ သူလည်း အရမ်းကြီးတဲ့ applications တွေကို InnoDB လိုဘဲ သုံးနိုင်ပါတယ်။

**MySQL syntax and commands**

MySQL command တွေကို shell command prompt ကနေ တိုက်ရိုက် ပြုလုပ်နိုင်ပေမယ့် ကျွန်တော်တို့က PHP ကိုလေ့လာနေတာဆိုတော့ PHP နဲ့ဘဲလုပ်တာကိုဘဲ ပြောသွားပါမယ်။

MySQL command တွေကို ရေးတဲ့အခါမှာ CAPITAL LETTERS တွေနဲ့ဘဲ ရေးလေ့ရေးထရှိပါတယ်။ မရေးလည်းရပါတယ်။ ဒီလိုရေးချင်းအားဖြင့် MySQL command တွေနဲ့ database , variables, tables တွေနဲ့ ကွဲကွဲပြားပြား မြင်ရနိုင်ပါတယ်။

ခု သင်ခန်းစာမှာ အသုံးပြုသွားမယ့် MySQL commands တွေက အောက်ပါအတိုင်းဖြစ်ပါတယ်။

- CREATE : Creates new databases and tables
- ALTER : Modifies existing tables
- SELECT : Chooses the data you want
- DELETE : Erases the data from your table

- DESCRIBE : Lets you know the structure and specifics of the table
- INSERT INTO tablename VALUES : Puts values into the table
- UPDATE : Lets you modify data already in a table
- DROP : Deletes an entire table or database

PHP နဲ့ MySQL database ကို ကောင်းကောင်း ချိတ်ဆက်နိုင်ဖို့ သင့် အနေနဲ့ steps အနည်းငယ်ကို သိထားဖို့ လိုပါတယ်။ အသုံးများတဲ့ functions တွေထဲက most commonly ဖြစ်တဲ့ functions တစ်ချို့ကို ဖော်ပြပေးလိုက်ပါတယ်။

*mysql\_connect([\$host[, \$username[, \$password]])* : Connects to the MySQL server and returns a resource which is used to reference the connection.

*mysql\_select\_db(\$database[, \$resource])* : Equivalent to the MySQL command USE and sets the active database.

*mysql\_query(\$query[, \$resource])* : Used to send any MySQL command to the database server. In the case of SELECT queries, a reference to the result set will be returned.

*mysql\_fetch\_array(\$result)* : Return a row of data from the query 's result set as an associative array, numeric array or both.

*mysql\_fetch\_assoc(\$result)* : Return a row of data from the query 's result set as an associative array.

*mysql\_error([\$resource])* : Shows the error message generated by the previous query

အခုချိန်မှာ သင့်အနေနဲ့ MySQL commands တွေက အစိမ်းသက်သက်ဖြစ်နေမှာပါ။ စိတ်မပူပါနဲ့ ဆက်ပြီး ဖတ်သွားပါ။ ရေးရင်းရေးရင်း နဲ့ ဒါတွေဟာ used to ဖြစ်လာမှာပါ။

MySQL commands တွေကို mysql\_query သုံးပြီး PHP ကနေ ပို့လိုရပါတယ်။ ဒီလိုပို့တဲ့နေရာမှာ variable သုံးပြီးပို့တာနဲ့ တိုက်ရိုက်ပို့တာဆိုပြီး ရှိပါတယ်။ အောက်က ဥပမာကတော့ တိုက်ရိုက် ပို့တာပါ။



```
$results = mysql_query('SELECT * FROM TABLE');
```

အပေါ်ကလိုရေးတာထက်စာရင် အောက်ပါအတိုင်း အဆင့် ၂ ဆင့်နဲ့ ရေးတာက ပိုကောင်းပါတယ်။

```
$query = 'SELECT * FROM TABLE';
$result = mysql_query($query);
```

ဒီလိုရေးခြင်းအားဖြင့် သင်ရေးလိုက်တဲ့ command ဟာ error တစ်ခုခုရှိခဲ့ရင်သို့မဟုတ် bug တစ်ခုခု ရှိခဲ့ရင် သင့်အနေနဲ့ \$query မှာ ရှာပြင်ရတာ လွယ်ကူပါတယ်။ result အနေနဲ့ကတော့ အပေါ်က အတိုင်းဘဲဖြစ်ဖြစ် အောက်ကအတိုင်းဘဲဖြစ်ဖြစ် အတူတူဘဲရမှာပါဘဲ။ ဒီ စာအုပ်မှာတော့ ဒုတိယနည်းလမ်းကိုဘဲ အများဆုံး အသုံးပြုပြီး ဖော်ပြသွားမှာဖြစ်ပါတယ်။

### Connecting to the MySQL server

ပထမဆုံး အနေနဲ့ MySQL ကို သုံးဖို့ဆိုရင် MySQL server နဲ့ ချိတ်ဆက်ဖို့လိုပါတယ်။ ဆာဗာနဲ့ချိတ်ဆက်ဖို့ အတွက် ဆာဗာမှာ သင် အစကတည်းက သတ်မှတ်ထားတဲ့ Host name, username and password တွေလိုပါတယ်။

Host name: Host name ဆိုတာ သင် MySQL server တင်ထားတဲ့နေရာဘဲဖြစ်ပါတယ်။ ခု သင်ခန်းစာမှာတော့ localhost ဘဲဖြစ်ပါတယ်။ ဘာဖြစ်လို့လဲဆိုတော့ သင် install လုပ်သမျှအားလုံးသည် locally အတွင်းမှာဘဲဖြစ်နေတဲ့အတွက်ကြောင့်ဘဲဖြစ်ပါတယ်။ တစ်ခြား ဆာဗာကဟာကို သုံးမယ်ဆိုရင်တော့ အဲဒီ server ကို ထည့်ပေးရမှာပါ။ Username and password: Server ရဲ့ username နဲ့ password တွေကိုပြောတာဘဲဖြစ်ပါတယ်။ ခု သင်ခန်းစာမှာ username က root ဖြစ်ပြီးတော့ password ကတော့ blank (ပါတစ်ဝေ့မထားဘူး) ဖြစ်ပါတယ်။

MySQL ကို connect လုပ်ရာမှာ mysql\_connect() ဆိုတဲ့ function နဲ့ လုပ်မှာဖြစ်ပါတယ်။ PHP/MySQL နဲ့ ပါတ်သတ်သမျှ statements အားလုံးကို variables နဲ့ သုံးလို့ရသလို Plain text အနေနဲ့လည်း သုံးလို့ရပါတယ်။ နမူနာ အနေနဲ့

```
$host= 'localhost';
$user= 'root';
```



```

$pass= "";
$db = mysql_connect($host, $user, $pass);

```

အောက်မှာ ထပ်ကြည့်ပါဦး။ ဒါက plain text အနေနဲ့ တစ်ခါတည်း ပြတာပါ။

```

$db = mysql_connect('localhost', 'root', "");

```

အလုပ်လုပ်ပုံခြင်းကတော့ အတူတူဘဲဖြစ်ပါတယ်။

### Looking at a Ready-made Database

ကျွန်တော်တို့ ရှေ့က သင်ခန်းစာမှာ moviesite နဲ့ပါတ်သတ်တဲ့ PHP code တွေကို ရေးခဲ့တယ်။ ခု တစ်ခါ အဲဒီရေးပြီးသား site ကို Database တစ်ခုနဲ့ တွဲပေးလိုက်မယ်။ Database ထဲမှာ ဘာတွေပါမလဲဆိုရင် -

movie table : ထဲမှာ movie နဲ့ပါတ်သတ်တဲ့ name and movie information ပါမယ်။

movietype table: ထဲမှာ မတူညီတဲ့ movie တွေရဲ့ category တွေပါမယ်။

people table : ထဲမှာတော့ movie တွေရဲ့ actor name and directors နာမည်တွေပါမှာပါ။ ဟုတ်ပြီ။ Table တစ်ခုကို ဘယ်လို Create လုပ်မလဲ။ သူ့ရဲ့ common ပုံစံလေးက ဒီလိုပါ-

```

CREATE TABLE [IF NOT EXISTS] tablename (
    fieldname definition,
    ...
    [key definitions]
)
table options

```

Insert command ကို ရေးတဲ့ ပုံစံကတော့

INSERT INTO tablename ( field names ...) VALUES (field values...)

ဒီ command နှစ်ခုအတွက် အခြေခံအနေနဲ့ ခုချိန်မှာ ဒီလောက် သိထားရင်ရပါပြီ။ ကျွန်တော်တို့ လက်တွေ့ တစ်ခုလောက် ရေးကြည့်မယ်လေ။ ဒါကို db1.php ဆိုပြီး နာမည်ပေးပါမယ်။

<?php

```

$db = mysql_connect('localhost','root','') or
    die('Cannot connect to mysql database');
mysql_select_db('moviesite',$db) or die(mysql_error($db));
//Create the movie table
$query = 'CREATE TABLE IF NOT EXISTS movie (
            movie_id                INT UNSIGNED NOT NULL
                                     AUTO_INCREMENT,
            movie_name              VARCHAR(255) NOT NULL,
            movie_type              TINYINT NOT NULL DEFAULT
            0,
            movie_year              SMALLINT UNSIGNED NOT
            NULL DEFAULT 0,
            movie_leadactor        INTEGER UNSIGNED NOT
            NULL DEFAULT 0,
            movie_director         INTEGER UNSIGNED NOT
            NULL DEFAULT 0,
            PRIMARY KEY (movie_id),
            KEY (movie_type, movie_year)
        ) ENGINE MYISAM' ;
mysql_query($query,$db) or die(mysql_error($db));
    
```

```

$query = 'CREATE TABLE IF NOT EXISTS movietype (
            movietype_id TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,
            movietype_label VARCHAR(100) NOT NULL,
            PRIMARY KEY (movietype_id)
        )ENGINE MYISAM';
$query = 'CREATE TABLE IF NOT EXISTS movietype (
            movietype_id TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,
            movietype_label VARCHAR(100) NOT NULL,
            PRIMARY KEY (movietype_id)
        )ENGINE MYISAM';
mysql_query($query,$db) or die (mysql_error($db));
//Create the people table
$query = 'CREATE TABLE IF NOT EXISTS people(
            people_id          INTEGER UNSIGNED NOT
            NULL AUTO_INCREMENT,
            people_fullname    VARCHAR(255) NOT NULL,
            people_isactor     TINYINT(1) UNSIGNED NOT
            NULL DEFAULT 0,
            people_isdirector  TINYINT(1) UNSIGNED NOT NULL
            DEFAULT 0,
            PRIMARY KEY (people_id)
        )ENGINE MYISAM';
mysql_query($query,$db) or die (mysql_error($db));

echo ' Movie database successfully created!';

?>

```



နောက်တစ်ခု ကို ထပ်ရေးပါမယ်။ ခုနက ဟာက Database ထဲမှာ Table တွေ တည်ဆောက်လိုက်တာပါ။ ခုဟာက အဲဒီ တည်ဆောက်ပြီးလိုက်တဲ့ သက်ဆိုင်ရာ Tables တွေထဲကို သက်ဆိုင်ရာ data တွေ ထည့်ပေးတာပါ။ သေချာလေး လေ့လာစေချင်ပါတယ်။ ခုဟာကိုတော့ db2.php ဆိုပြီး နာမည်ပေးထားပါတယ်။

```
<?php
$db = mysql_connect('localhost','root','') or die ('Unable to connect to database');
mysql_select_db('moviesite',$db) or die ('Unable to connect to database');

$query = 'INSERT INTO movie (movie_id, movie_name, movie_type, movie_year, movie_
leadactor, movie_director) VALUES
(1, "Bruce Almighty", 5, 2003, 1, 2),
(2, "Office Space", 5, 1999, 5, 6),
(3, "Grand canyon", 2, 1991, 4, 3)';
mysql_query($query,$db) or die (mysql_error($db));
```

```
$query = 'INSERT INTO movietype (movietype_id, movietype_label) VALUES
(1, "SCI FI"),
(2, "Drama"),
(3, "Adventure"),
(4, "War"),
(5, "Comedy"),
(6, "Horror"),
(7, "Action"),
(8, "Kids")';
mysql_query($query,$db) or die (mysql_error($db));
```

```
$query = 'INSERT INTO people (people_id, people_fullname, people_isactor, people_isdi-
rector) VALUES
(1,"Jim Carrey", 1, 0),
(2,"Tom shadyac", 0, 1),
(3,"Lawrence Kasdan", 0, 1),
(4,"Kevin Klein", 1, 0),
```

```

(5,"Ron livingston", 1, 0),
(6,"Mike judge", 0, 1);

mysql_query($query,$db) or die (mysql_error($db));
echo " DATA INSERTED SUCCESSFULLY";

```

?>

ပထမဆုံး အနေနဲ့ db1.php ကို RUN ပါမယ်။ ဘယ်လို Run ရမလဲဆိုရင် http://localhost/db1.php ဆိုပြီး Browser ကနေ ခေါ်လိုက်ပါ။ ဒါဆိုရင် Movie database successfully created! ဆိုပြီးပေါ်လာပါမယ်။ နောက်ပြီးတော့ db2.php ကို run ပါ။ ဒါဆိုရင် သင်ဖန်တီးထားတဲ့ moviesite ဆိုတဲ့ database ထဲမှာ table ၃ ခုနဲ့ သူနဲ့ သက်ဆိုင်ရာ data တွေ ရောက်ရှိသွားပါပြီ။

### How it works

ခုနက ရေးထားတဲ့ code တွေ ဘယ်လို အလုပ်လုပ်သလဲဆိုတာ ကျွန်တော် ဆက်ရှင်းပြပါမယ်။ ပထမဆုံး အနေနဲ့ MySQL server နဲ့ connect လုပ်ပါတယ် ။ ဒီလိုလုပ်ရမှာ Error တစ်ခုခု ရှိခဲ့မယ်ဆိုရင် (ဥပမာ host name, username, password တွေမှားရင်) script ကို ချက်ခြင်း Stop လုပ်ဖို့ပါ ထည့်သွင်းရေးလိုက်ပါတယ်။

```

// connect to MySQL
$db = mysql_connect('localhost', 'root', 'p@55w0Rd!') or
die ('Unable to connect. Check your connection parameters.');
```

Connect လုပ်လို့ရသွားပြီဆိုရင် ကိုယ်က ဘယ် database ကို သုံးမှာလဲ ဒါမှမဟုတ် ကိုသုံးမယ့် database မရှိရင် လုပ်ပေးပါ ဆိုတဲ့ Command လေးကိုပါ ထည့်ရေးပေးလိုက်တယ်။ ကျွန်တော်တို့က moviesite ဆိုတဲ့ database ကို သုံးမှာလေ။ ဒါကြောင့်မို့လို့

```

//create the main database if it doesn't already exist
$query = 'CREATE DATABASE IF NOT EXISTS moviesite';
mysql_query($query, $db) or die(mysql_error($db));
```

ဟုတ်ပြီ။ moviesite ဆိုတဲ့ Database တစ်ခုကို တည်ဆောက်ပြီးသွားပြီဆိုရင် အဲဒီထဲကို ကိုယ်အသုံးပြုမယ့် table တွေ ထည့်မယ်လေ။ အောက်ပါအတိုင်း Table တစ်ခုခြင်းစီကို Create လုပ်ပါတယ်။ Table ထဲက field တစ်ခုခြင်းစီမှာ လိုအပ်မယ့် parameters တွေပါ တစ်ခါတည်း ထည့်ပါမယ်။ ခု သုံးပြုထားတာတွေကို နားမလည်ရင် ရှေ့ကရှင်းပြထား တာတွေနဲ့ ပြန်တိုက်ပြီး ဖတ်ကြည့်စေချင်ပါတယ်။

```
//Create the movie table
$query = 'CREATE TABLE IF NOT EXISTS movie (
    movie_id          INT UNSIGNED NOT NULL AUTO_INCREMENT,
    movie_name        VARCHAR(255) NOT NULL,
    movie_type        TINYINT NOT NULL DEFAULT 0,
    movie_year        SMALLINT UNSIGNED NOT NULL DEFAULT 0,
    movie_leadactor   INTEGER UNSIGNED NOT NULL DEFAULT 0,
    movie_director    INTEGER UNSIGNED NOT NULL DEFAULT 0,
    PRIMARY KEY (movie_id),
    KEY (movie_type, movie_year)
) ENGINE MYISAM' ;
```

ဒါက ကျွန်တော်တို့ SQL statement ရေးလိုက်တာပါ။ ဒီ Statement ကို \$query ဆိုတဲ့ variable ဆောက်ပြီး ရေးလိုက်တာပါ။ ဒါကို MySQL server ဆီကို mysql\_query() ဆိုတဲ့ function ကို သုံးပြီးတော့ send လုပ်ပေးရမှာပါ။ အဲဒီလို လုပ်ရင်းနဲ့ ကိုယ်ရေးလိုက်တဲ့ statement မှာ Error ပါရင် ပြန်ရယ် code တွေကို execute မလုပ်ဘဲ ရပ်လိုက်ဖို့ error handling control code လေးပါ တစ်ခါတည်း ထည့်ရေးလိုက်တာပါ။

```
mysql_query($query, $db) or die (mysql_error($db));

mysql_query($statements,$source) ဆိုတာလေးတော့ မှတ်ထားလိုက်ပါဦး။
```



ဒီလိုနည်းနဲ့ Table ၃ ခုလုံးကို Create လုပ်ပြီးသွားရင် အမှား တစ်ခုမှမရှိရင် echo 'Movie database successfully created!'; ဆိုပြီး နောက်ဆုံးမှာ ရေးခဲ့တော့ အားလုံးပြီးသွားရင် Movie database successfully created! ဆိုပြီး ပေါ်လာရပါမယ်။ ဒီလောက်ဆိုရင် ရေးရေးလေးတော့ သဘောပေါက်လိုက်ပြီထင်ပါတယ်။ ဖတ်ဖတ်ခြင်းတော့ အခက်အခဲနဲ့ကြုံရမှာပါ။ များများရေးမှ များများ မှတ်မိပြီး မြန်မြန်သဘောပေါက်လာမယ်လေ။

ဆက်ပြီးတော့ db2.php ကိုလည်း ရှင်းပါဦးမယ်။

၁။ MySQL server နဲ့ချိတ်ပါတယ်။ အပေါ်ကအတိုင်းပါဘဲ။

၂။ အသုံးပြုမယ့် database ကိုရွေးပေးရပါတယ်။ ဆာဗာထဲမှာ တစ်ခုထက်မကတဲ့ database တွေ ရှိနေမယ်လေ။ ရှိရှိမရှိရှိ အသုံးပြုမယ့် database ကိုတော့ရွေးပေးရမှာဘဲ။

//make sure you're using the correct database

mysql\_select\_db('moviesite',\$db) or die ('Unable to connect to database');

ဒီမှာဆိုရင် mysql\_select\_db() ဆိုတာကိုတွေ့မှာပါ။ အဲဒါက database ကို ရွေးပေးတဲ့ဟာပါ။

mysql\_select\_db(databaseName, Source) ဆိုပြီး မှတ်ထားဖို့လိုပါတယ်။

၃။ Data တွေကို Insert လုပ်ပါတယ်။ ပထမဆုံး အနေနဲ့ movie ဆိုတဲ့ Table ထဲကို insert လုပ်ပါတယ်။ Table ထဲမှာ ရှိတဲ့ Field အစဉ်အတိုင်း လုပ်ပါတယ်။

INSERT INTO movie ဆိုတာ movie ဆိုတဲ့ table ထဲကို insert လုပ်မှာပါလို့ ပြောလိုက်တာပါ။

(movie\_id, movie\_name, movie\_type, movie\_year, movie\_leadactor, movie\_director) ဆိုတာတွေက movie Table ထဲမှာရှိတဲ့ Field name တွေပါ။

VALUES

- (1, "Bruce Almighty", 5, 2003, 1, 2),
- (2, "Office Space", 5, 1999, 5, 6),
- (3, "Grand Canyon", 2, 1991, 4, 3)

ဆိုတာတွေက ခုနက field တစ်ခုခြင်းစီမှာ ထည့်မယ့်ဟာတွေကိုပြောတာပါ။ Data တွေကို row by row ထည့်ရတာပါ။

Querying the database

ဟုတ်ပြီ။ ခုဆို သင်ဖန်တီးထားတဲ့ moviesite ဆိုတဲ့ database ထဲမှာ data တွေ ရောက်နေပြီ။ ဒီတော့ သင်က ဒီ data တွေကို အသုံးပြုခြင်းမှာဘဲဖြစ်ပါတယ်။ ကျွန်တော်တို့တွေက database ထဲမှာရှိတဲ့ data တွေကို ကိုယ်ဖန်တီး ထားတဲ့ website ပေါ်ကို တင်ချင်တယ်ဆိုရင် SELECT ဆိုတဲ့ statement ကို အသုံးပြုရမှာဘဲဖြစ်ပါတယ်။ ကိုယ် ပြုချင်တာလေးတွေကို select လုပ်ပြီး ပြတဲ့သဘောပါ။ သူ့ပုံစံလေးကတော့ အောက်ပါအတိုင်းလေးဘဲဖြစ်ပါတယ်။

```
SELECT [field names]
AS [alias]
FROM [tablename]
WHERE [criteria]
ORDER BY [fieldname to sort on ] [ASC|DESC]
LIMIT [offset , maxrows]
```

သင့်အနေနဲ့ နောက်ပိုင်းမှာ ဒီထက်မကတဲ့ parameters တွေ ထပ်ပေါင်းနိုင်ပါတယ်။ ဒါပေမယ့် ခု ပုံစံလေးက လည်း အများဆုံး အသုံးပြုနေတဲ့ ပုံစံလေးတစ်ခုပါ။ တစ်ခုချင်းစီ ထပ်ရှင်းပြပါမယ်။

SELECT[field names] : သင် အသုံးပြုချင်တဲ့ Field ကို ရွေးတာပါ။ Table တစ်ခုထဲက field အကုန်လုံးကို ရွေးချင်တယ်ဆိုရင်တော့ name တစ်ခုချင်းစီ ရေးရင် ရသလို SELECT \* FROM ဆိုပြီးတော့ shortcut သုံးပြီး ရေးလို့ရပါတယ်။ SELECT \* FROM movie ဆိုရင် movie Table ထဲက Fields အားလုံးကို ရွေးလိုက်တာပါ။

AS : alias name ဖြစ်ပါတယ်။ နောက်ထပ် မွေးစားနာမည်ပေးလိုက်တဲ့သဘောပါ။ ဒီလို ပေးထားခြင်းအားဖြင့် နောက်ထပ် အဲဒီ ဒေတာကို တစ်ခြားနာမည်ဖြင့် (ကိုယ်ပေးထားတဲ့ Alias name ) reference လုပ်လို့ရပါတယ်။

```
SELECT movie_name, movie_year AS release_year FROM movie
```



FROM: သင် ယူသုံးမယ့် Field ရဲ့ table နာမည်ဘဲဖြစ်ပါတယ်။ တစ်ခုထဲဖြစ်နိုင်သလို တစ်ခုထက်မကလည်း ဖြစ်နိုင်ပါတယ်။

WHERE: where ကို အောက်မှာ သပ်သပ် ထပ်ရှင်းပြပါမယ်။

ORDER BY: default အနေနဲ့ result တွေကို ascending စီပေးပါတယ်။ ASC ပါ။ သင်က De- scending စီချင်တယ်ဆိုရင်တော့ DESC ပါ။

LIMIT: limit ဆိုတာကတော့ ထွက်လာမယ့် Data တွေကို တစ်ခါထွက်ရင် ဘယ်လောက်ဘဲ ထွက်ပါ ဆိုပြီး သတ်မှတ်ပေးလိုက်တာပါ။ ဥပမာ- site တစ်ခုမှာ users တွေ အများကြီးရှိနိုင်ပါတယ်။ အဲဒီထဲကမှ ကိုယ်က ဘာကို ပြချင်သလဲဆိုရင် နောက်ဆုံး မန်ဘာဝင်တဲ့ မန်ဘာတွေကို ပြချင်တယ်။ ဒါပေမယ့် နောက်ဆုံး ၁၀ ယောက်ကိုဘဲပြချင်တယ်ဆိုရင် limit 10 ဖြစ်သွားမှာပါ။

**WHERE**

Where clause က တော်တော်လေး အသုံးဝင်ပါတယ်။ အများထဲကမှ သတ်မှတ်ထားတဲ့ အရည်အချင်း ရှိတဲ့ ဟာတွေကို ပြန်ရွေးထုတ်ပေးတာပါ။ ဥပမာလေး အနေနဲ့ ပြကြည့်မယ်။

```
// retrieves all information about all customers
SELECT * FROM customers;
// retrieves all information about male customers
SELECT * FROM customers WHERE gender = "Male"
```

ဒီ ၂ ခုကို နှိုင်းယှဉ်လိုက်မယ်ဆိုရင် အပေါ်တစ်ကြောင်းက Customers အားလုံးကို ရွေးလိုက်တာပါ။ အောက် တစ်ကြောင်းကတော့ Customers တွေကို ရွေးတော့ရွေးတယ် ဒါပေမယ့် အရွေးခံရတဲ့သူတွေ အားလုံးက ယောက်ျား တွေဖြစ်ဖို့လိုတယ်။ ကဲ ခုလောက်ဆို where clause ကို နည်းနည်း သဘောပေါက်မယ်ထင်ပါတယ်။

ခု where clause ကို နည်းနည်းလေး detail ကျကျ ဆက်ကြည့်ကြရအောင်-

where clause တစ်ခုကို သုံးပြီးဆိုရင် comparison တွေ ထည့်သုံးမှ ပိုပြီး ပြည့်စုံပါတယ်။ where clause က heart of WHERE clause လို့ ခေါ်လည်း မမှားပါဘူး။ comparison operators တွေက အောက်ပါအတိုင်းဖြစ်ပါတယ်။



- = is used to test if two values are equal
- != is used to test if two values are not equal
- < is used to test if one value is less than the second
- <= is used to test if one value is less than or equal to the second
- > is used to test if one value is greater than the second
- >= is used to test if one value is greater than or equal to the second

LIKE like ကိုတော့ ဘယ်နေရာမှာသုံးလဲဆိုရင် text တွေကို compare လုပ်တဲ့နေရာမှာသုံးပါတယ်။ % နဲ့ ဖော်ပြပြီး wildcards အနေနဲ့ သုံးပါတယ်။ wildcard ကို ဘယ်လိုနေရာမှာ သုံးသလဲဆိုတော့ သင်က field တစ်ခုထဲက data ကို လိုချင်တယ် အကုန်လုံးလည်း မမှတ်မိဘူး။ နည်းနည်းတော့ မှတ်မိတယ်ဆိုပါတော့ ။ ဥပမာ သင်က shoe ဆိုတာဘဲ မှတ်မိတယ်။ အဲဒီဒေတာတွေကို လိုချင်ရင်

```
SELECT * FROM products WHERE description LIKE "%shoes%"
```

ဒါဆိုရင် description ဆိုတဲ့ field ထဲက shoe ပါတဲ့ text pattern တွေအားလုံးကို record လုပ်လိုက်တာပါ။

ဥပမာ -

"blue shoes", "red shoes", "no shirts, no shoes, no hat"

နောက်ဆုံး တစ်ခုမှာလည်း shoes ဆိုတဲ့ စကားလုံးပါလို့ record ထဲမှာ ပါသွားတာပါ။

**Logical operators**

where clause မှာ logical operators တွေကိုလည်း ထည့်သုံးလို့ရပါတယ်။ ဘာတွေလဲဆိုတော့ AND , OR, NOT , XOR တွေပါ။

```
SELECT * FROM products WHERE description LIKE "%shoes%" AND price <= 2500
```

ဒီမှာဆိုရင် AND ဆိုတဲ့ logical operator တစ်ခု ပိုလာပါတယ်။ text pattern မှာ shoes ပါရမှာ အပြင် သူ့ တန်ဖိုးကလည်း 2500 နဲ့ ညီရင် ညီ မညီရင် နည်းရမှာဖြစ်ပါတယ်။ အဲဒီ ဟာတွေနဲ့ ကိုက်ညီတာတွေကိုဘဲ record လုပ်ပြီး ထုတ်ပြသွားမှာဖြစ်ပါတယ်။

ဒီလိုဘဲ တစ်ဆင့်ခြင်း တစ်ခုခြင်း step by step လုပ်သွားရင် သင့် အနေနဲ့ နောက်ပိုင်းမှာ ပိုပြီး မြင်သာထင်သာ ရှိလာပြီး သဘောတရားတွေကိုလည်း ပိုပြီးနားလည်လာမယ်။ အဲဒီအခါကြရင် ကိုယ်စိတ်ကူးရှိသလို ကွန်မြူပြီး လုပ် လုပ်သွားနိုင်မှာဘဲဖြစ်ပါတယ်။ ခု SELECT query ကို သဘောပေါက်ပြီဆိုရင် လေ့ကျင့်ခန်းလေး နည်းနည်းလောက် လုပ်ကြည့်လိုက်ရအောင်။

```

<?php
    $db = mysql_connect('localhost','root','') or
    die ('Unable to connect. Check your connection parameters.');
```

mysql\_select\_db('moviesite1',\$db) or die(mysql\_error(\$db));

```

    $query = 'SELECT movie_name, movie_type
              FROM movie
              Where movie_year >1990
              ORDER BY movie_type';

    $result = mysql_query($query, $db) or die (mysql_error($db));
    // Show the result
    while ($row = mysql_fetch_array($result)){
        extract($row);
        echo $movie_name. ' - ' . $movie_type.<br />;
    }
?>
```

ဒီ script ကို select1.php လို့ နာမည်ပေးပြီး Run လိုက်ရင် အောက်ပါအတိုင်း browser က ပြပေးမှာပါ။

- Grand canyon - 2
- ThaGyanMoe - 5
- Office Space - 5



ကျွန်တော် while block ကနေ စရှင်းပြပါမယ်။ အပေါ် ၂ ခုက ခဏခဏ ရှင်းခဲ့ပြီးပြီ။ MySQL query တစ်ခုကို \$result ဆိုတဲ့ variable ထဲ ထည့်ပြီး server ကို ပို့လိုက်တယ်။ ဆာဗာကနေ ပြန်ထုတ်ပေးဖို့ while ကို သုံးတယ်။ data တွေက row by row ရှိနေတာနော်။ ပြန်ထုတ်ပေးဖို့အတွက် mysql\_fetch\_array() ဆိုတဲ့ function တစ်ခုကို သုံးပါတယ်။ result အနေနဲ့ ဖြစ်လာမယ့် အဖြေ တစ်ခုချင်းစီကို \$row ဆိုတဲ့ variable ထဲ ထည့်လိုက်ပါတယ်။ နောက်ပြီးတော့ အဲဒီ row ကို extract လုပ်လိုက်ပါတယ်။ အဲဒီ row ထဲမှာ ဘာပါလဲဆိုတော့ movie name and movie type တွေပါတယ်လေ။ သူတို့တွေက variables တွေ အနေနဲ့ ပါလာတာပါ။ အော်တို မက်တစ်တော့ မပြဘူး။ echo လေးနဲ့ output ထုတ်ပေးလိုက်တယ်။ တကယ်တော့ ကျွန်တော် ရှင်းပြနေတာ ကြာတယ်။ ကိုယ့်ဟာကို သေချာကြည့်လိုက်ရင် နားလည်ပါတယ်။ လွယ်လွယ်လေးပါ။ ကျွန်တော် ပုံလေးနဲ့ ထပ်ရှင်းပြပါမယ်။

Query (\$result)

SERVER

| movie_name   | movie_type | movie_year |
|--------------|------------|------------|
| ThaGyanMoe   | 5          | 2003       |
| Office Space | 5          | 1991       |
| Grand canyon | 2          | 1999       |
|              |            |            |

Out put

while (\$row = mysql\_fetch\_array(\$result) က \$result နဲ့ တောင်းလိုက်တဲ့ query အတွက် data တွေကို mysql\_fetch\_array() ဆိုတဲ့ function ကိုသုံးပြီး ခေါ်လိုက်တာပါ။ သူ့ထဲမှာ ဘာပါလာမလဲဆိုတော့ \$movie\_name and \$movie\_type ဆိုတဲ့ variable ၂ ခုပါလာပါမယ်။ ၂ ခုပေါင်း pack တစ်ခု လုပ်လိုက်ပြီး \$row ဆိုပြီး နာမည်ပေးလိုက်ပါတယ်။ ပြီးရင် အဲဒီ pack (\$row) ဆိုတာကို extract function သုံးပြီး ဖြေချလိုက်ပါတယ်။ ဖြေချလိုက်တော့ ဘာထွက်လာမလဲ ။ ခုနက variable ၂ ခု ထွက်လာမှာပေါ့။ အဲဒီ variables တွေရဲ့ တန်ဖိုးတွေက ပထမအကြိမ်မှာတော့ ThaGyanMoe နဲ့ သူ့ရဲ့ movie type ဖြစ်တဲ့ 5 တွေပေါ့။ နောက် ဒုတိယအကြိမ် while loop အတွက်က ဒုတိယ တန်ဖိုးပေါ့ဗျာ။



ဘယ်အချိန်ထိ loop ပတ်မှာလဲဆိုတော့ ၁၉၉၀ နောက်ပိုင်းထွက်တဲ့ movie အကုန်လုံးကို Out put ထုတ်ပေးပြီးတဲ့ အချိန်ထိပတ်မှာပါ။ ဘာဖြစ်လို့လဲဆိုတော့ server ကိုပို့တဲ့ query ထဲမှာ where clause မှာ >1990 လို့ သတ်မှတ်လိုက်တယ်လေ။ နောက်ပြီးတော့ while loop မှာကြတော့လည်း သူ့ရဲ့ condition က \$result ပေါ်မှာမှီခိုသွားတယ်။

**Working with PHP and arrays of Data (foreach)**

foreach loop ဟာ ဘာနဲ့ ဆင်တူသလဲဆိုရင် while loop နဲ့ဆင်တူပါတယ်။ အဓိက ရည်ရွယ်ချက်ကတော့ every result row တွေကို block of statements တစ်ခု အနေနဲ့ ဖော်ပြနိုင်ဖို့ဘဲဖြစ်ပါတယ်။ နောက်ပိုင်း သင်ခန်းစာတွေမှာ ပိုပြီး ရှင်းလင်းသွားမှာပါ။ သူ့ရဲ့ ပုံစံလေးက

```
foreach ($row as $value) {
    echo $value;
    echo '< br >';
}
```

လက်တွေ့ သင်ခန်းစာလေး တစ်ခုလောက်လုပ်ကြည့်ရအောင်။ select1.php ကို ဘဲ နည်းနည်း ပြင်ရေးကြည့်မယ်။ ပြင်ထားတဲ့နေရာက highlight ပြထားတဲ့နေရာနော်-

```
<?php
$db = mysql_connect('localhost','root','') or
die ('Unable to connect. Check your connection parameters. ');
mysql_select_db('moviesite1',$db) or die(mysql_error($db));

$query = 'SELECT movie_name, movie_type
FROM movie
Where movie_year >1990
ORDER BY movie_type';
```

```

$result = mysql_query($query, $db) or die (mysql_error($db));
/*
// Show the result
while ($row = mysql_fetch_array($result)){
    extract($row);
    echo $movie_name. ' - '. $movie_type.<br />';
}
*/
while ($row = mysql_fetch_assoc($result)) {
    foreach ($row as $value) {
        echo $value . ' ';
    }
    echo '<br/> ';
}
?>

```

ဘယ်လို အလုပ်သွားသလဲဆိုတော့ ကျွန်တော် ရှင်းပြပါမယ်။ highlight ပြထားသလိုပြင်ပြီး script ကို run ရင်လည်း အရင် အဖြေအတိုင်းဘဲရတယ်ဆိုတာ သင်သတိပြုမိပါလိမ့်မယ်။ ခုဒီနေရာမှာ mysql\_fetch\_array အစား mysql\_fetch\_assoc ကို သုံးထားတာ သတိပြုမိပါလိမ့်မယ်။ တကယ်တမ်းတော့ mysql\_fetch\_array မှာ value က ၂ ခုံ return ဖြစ်ပါတယ်။ associative and numerical တို့ဖြစ်ပါတယ်။ foreach ကို သုံးပြီးဆိုရင် သူက array ထဲက value တိုင်းကို တစ်ခုခြင်းစီ ထုတ်ပေးမှာဖြစ်တဲ့အတွက် mysql\_fetch\_array(\$result) ဆိုရင် သင့် အဖြေတွေဟာ ၂ ခုံစီဖြစ်နေမှာပါ။ duplicate value တွေဖြစ်နေမှာပေါ့။ အောက်ပါအတိုင်းထွက်နေလိမ့်မယ်။

```

Grand canyon Grand canyon 2 2
ThaGyanMoe ThaGyanMoe 5 5
Office Space Office Space 5 5

```

အဲဒီတော့ ဒီလို မဖြစ်ရအောင် mysql\_fetch\_array(\$result, MYSQL\_ASSOC) ဆိုပြီး associative တစ်ခုဘဲ return ပြန်အောင် ရေးရပါမယ်။ ခုလိုမျိုးရေးရတာ ရှည်တယ်ဆိုရင် mysql\_fetch\_assoc(\$result) ဆိုပြီး တန်း ရေးလို့ရပါတယ်။ ဒါဆိုရင်တော့ result တွေက duplicate မဖြစ်တော့ပါဘူး။

## Using foreach to create table

ဒါကတော့ HTML table ထဲမှာ ထွက်လာမယ့် result တွေကို table နဲ့ပြတာပါ။ မရှင်းပြတော့ပါဘူး။ လေ့လာကြည့်ပါ။

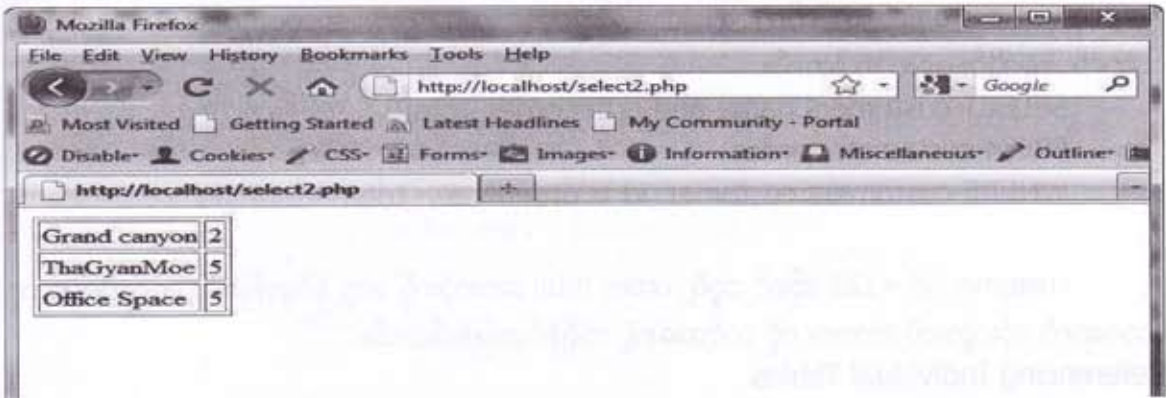
```
<?php
$db = mysql_connect('localhost', 'root', '') or
    die ('Unable to connect. Check your connection parameters. ');
mysql_select_db('moviesite1', $db) or die(mysql_error($db));

// select the movie titles and their genre after 1990
$query = 'SELECT movie_name, movie_type
        FROM movie
        WHERE movie_year > 1990
        ORDER BY movie_type';
$result = mysql_query($query, $db) or die(mysql_error($db));

// show the results
echo '<table border="1">';
while ($row = mysql_fetch_assoc($result)) {
    echo '<tr>';
    foreach ($row as $value) {
        echo '<td>' . $value . '</td>';
    }
    echo '</tr>';
}
echo '</table>';
?>
```

အထက်ပါ script ကို select2.php လို့ နာမည်ပေးပြီး browser ကနေ ခေါ်တင်ကြည့်ပါ။ အောက်ပါအတိုင်း မြင်ရပါမယ်။





### Working with two tables

ကျွန်တော်တို့ဟာ ခုဆိုရင် MySQL commands တွေ functions တွေသုံးပြီးတော့ table တစ်ခုထဲက data တွေကို output ထုတ်ပြန်နိုင်ပါပြီ။ ဒါပေမယ့် အထက်က ပုံကိုကြည့်ပါ။ movie name ကတော့ ဟုတ်ပါပြီ။ ဘေးက 2 တွေ 5 တွေက ဘာတွေလဲဗျ။ အဲဒါ ကျွန်တော်တို့က တော့ သိတယ်လေ။ movie type တွေဆိုတာ။ ဘေးလူက ဘယ်သိမလဲ။ သူနဲ့ သက်ဆိုင်တဲ့ drama တို့ action တို့က တစ်ခြား table ထဲမှာ သိမ်းထားတယ်။ ဒီလိုဆိုရင် အကုန်လုံး နားလည်အောင် အဲဒါတွေကို ထုတ်ပြန်ရမယ်။ ဒါဆိုရင် Table ၂ ခုကို သုံးဖို့လိုလာပြီပေါ့။ ဟုတ်တယ်နော်။ Table ၁ ခုနဲ့ အထက် အတူသုံးဖို့လိုတဲ့အခါမှာ နည်း ၂ နည်းနဲ့ သုံးလို့ရပါတယ်။

နည်းလမ်း(၁)။ သင်ပြုလုပ်ထားတဲ့ query ထဲမှာ table တစ်ခုခြင်းစီကို reference လုပ်မယ်။ ပြီးရင် အဲဒါကို link ပြန်ချိတ်မယ်။

နည်းလမ်း(၂)။ Table တွေကို JOIN ဆိုတဲ့ function သုံးပြီး ဆက်ပါမယ်။

ကျွန်တော်တို့ နည်းလမ်း ၂ ခုလုံးကို အသုံးပြုကြည့်ရအောင်။ ပြီးမှ တစ်ခုခြင်းစီကို အသေးစိတ် ထပ်ဆွေးနွေးတာပေါ့။

### Referencing Two Tables

ကျွန်တော်တို့ SELECT ကိုသုံးပြီး table ၂ ခုကဟာကို ထုတ်သုံးကြမယ် ဆိုပါစို့။ Table နာမည်တွေက customers , order ဆိုပြီးထားလိုက်မယ်။ customers table ထဲကနေ name ဆိုတဲ့ field နဲ့ order table ထဲက order\_total ဆိုတဲ့ဟာကို ရွေးမယ်။ အဲဒီ Table

၂ ခုလုံးမှာ customer\_id ဆိုတဲ့ Field ပါပြီးတော့ value က အတူတူဘဲဖြစ်မယ်။ ဒါဆိုရင် အောက်ပါအတိုင်း ရေးလို့ရပါတယ်။

```
SELECT customers.name and order.order_total
FROM customers, order
WHERE customers.customer_id = order.order_total
```

customer\_id =123 ဆိုရင် သူ့ရဲ့ order total အားလုံးကို တွေ့နိုင်မှာဖြစ်ပြီး သင့်ဆိုင်မှာ သူ လာရောက် သုံးသွားတဲ့ money ကို သင့်အနေနဲ့ သိရှိနိုင်မှာဖြစ်ပါတယ်။

### Referencing Individual Tables

ခု multiple tables တွေကို ဘယ်လို reference လုပ်ရသလဲဆိုတာ ပြသွားပါမယ်။

```
<?php
```

```
$db = mysql_connect('localhost', 'root', '') or
    die ('Unable to connect. Check your connection parameters.');
```

```
mysql_select_db('moviesite1', $db) or die(mysql_error($db));
```

```
/*
```

```
// select the movie titles and their genre after 1990
```

```
$query = 'SELECT movie_name, movie_type
FROM movie
WHERE movie_year > 1990
ORDER BY movie_type';
```

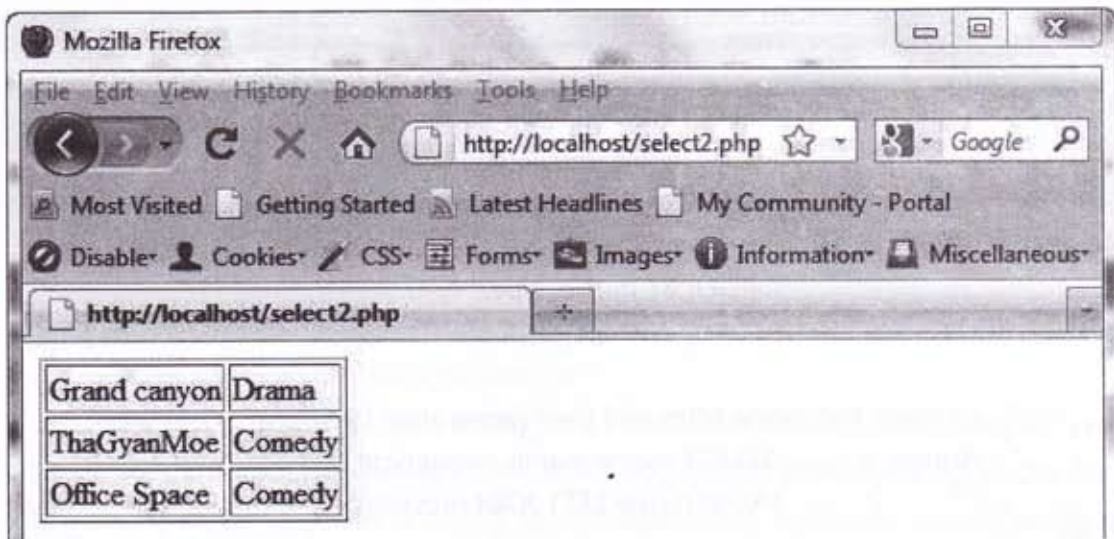
```
*/
```

```
$query = 'SELECT movie.movie_name, movietype.movietype_label
FROM movie, movietype
WHERE movie.movie_type = movietype.movietype_id AND movie_
year > 1990
ORDER BY movie_type';
```

```
$result = mysql_query($query, $db) or die(mysql_error($db));
```

```
// show the results
echo '<table border="1">';
while ($row = mysql_fetch_assoc($result)) {
    echo '<tr>';
    foreach ($row as $value) {
        echo '<td>' . $value . '</td>';
    }
    echo '</tr>';
}
echo '</table>';
```

?>



ခုဆိုရင် movie name ရယ် movie type တွေကို တွေ့ရပါပြီ။ ကြည့်တဲ့သူတွေလည်း နားလည်သွားပါပြီ။ ဒီမှာက movie table ထဲက movie\_type field ရယ် movietype table ထဲက movietype\_id field တွေက same value တွေကို ကိုယ်ကူးပြုထားတဲ့အတွက် အဲဒီ field နှစ်ခုကို link ချိတ်ပြီး movie\_name, movietype\_label ကို output ထုတ်လိုက်တာပါ။



## Joining Two Tables

ဒီသင်ခန်းစာမှာတော့ Table နှစ်ခုကို JOIN ဆိုတဲ့ဟာနဲ့ ဆက်ပြီးသုံးမှာပါ။

```
<?php
```

```
    $db = mysql_connect('localhost', 'root', '') or
        die ('Unable to connect. Check your connection parameters.');
```

```
    mysql_select_db('moviesite1', $db) or die(mysql_error($db));
```

```
    /*
```

```
    // select the movie titles and their genre after 1990
```

```
    $query = 'SELECT movie_name, movie_type
              FROM movie
              WHERE movie_year > 1990
              ORDER BY movie_type';
```

```
    */
```

```
    /*
```

```
    $query = 'SELECT movie.movie_name, movietype.movietype_label
              FROM movie, movietype
              WHERE movie.movie_type = movietype.movietype_id AND
movie_year > 1990
              ORDER BY movie_type';
```

```
    */
```

```
    // select the movie titles and their genre after 1990
```

```
    $query = 'SELECT movie_name, movietype_label
              FROM movie LEFT JOIN movietype ON movie_type = movi-
etype_id
              WHERE movie.movie_type = movietype.movietype_id AND
movie_year > 1990
              ORDER BY movie_type';
```

```
    $result = mysql_query($query, $db) or die(mysql_error($db));
```

```
// show the results
echo '<table border="1">';
while ($row = mysql_fetch_assoc($result)) {
    echo '<tr>';
    foreach ($row as $value) {
        echo '<td>' . $value . '</td>';
    }
    echo '</tr>';
}
echo '</table>';
?>
```

ဒီscript ကို run လိုက်မယ်ဆိုရင်လည်း ရှေ့မှာ ပြခဲ့တဲ့ result တွေအတိုင်းဘဲ ထွက်လာမှာပါ။ join ကနေ စရှင်းပြပါမယ်။ အရင်ဆုံးကတော့ ဘယ် fields တွေသုံးမလဲဆိုတာကို SELECT statement သုံးပြီး ပြောပါတယ်။

```
SELECT movie_name, movietype_label
```

နောက်ပြီးတော့ LEFT join statement ကို သုံးသွားတာ တွေ့ပါလိမ့်မယ်။

```
FROM movie LEFT JOIN movietype
```

အရှင်းဆုံး ရှင်းပြရရင် second table ဖြစ်တဲ့ movietype ဟာ ဆိုရင် first table ဖြစ်တဲ့ movie table ပေါ် depend လုပ်ပါတယ်။ main information တွေကို First table ကနေ ယူပြီးတော့ တစ်ခြား အနည်းငယ်လိုအပ်တဲ့ information တွေကို second table ကနေ ယူပါတယ်။ နောက်ပြီးတော့ ဘယ် field နှစ်ခုကို join လုပ်မှာလဲဆိုတာကို ဆာဗာကို အသိပေးရပါတယ်။

```
ON movie_type = movietype_id
```

ကျန်တဲ့ Code တွေကတော့ အတူတူဘဲဖြစ်ပါတယ်။ ဒီလောက်ဆိုရင် join ဘယ်လို အလုပ်သလဲဆိုတာ သိသွားပြီ။



## Exercises

## For chapter (3)

1. Create a PHP program that prints the lead actor and director for each movie in the database.
2. Pick only comedies from the movie table, and show the movie name and the year it was produced. Sort the list alphabetically.
3. Show each movie in the database on its own page, and give the user links in a " page 1, page 2, page 3 " – type navigation system. Hint: Use LIMIT to control which movie is on which page.



## Chapter (4) Using Tables to Display Data

ခုဆိုရင် သင်ဟာ PHP နဲ့ MySQL ကို တွဲ ပြီးတော့ သုံးတတ်နေပြီး dynamic pages ကို ဖန်တီး နေနိုင်ပါပြီ။ တကယ်လို့ သင် ပြချင်တဲ့ Data တွေက row တွေအများကြီး ဖြစ်နေမယ်။ ဒီလိုဆိုရင် ကြည့်ရှုရမယ့်သူအဖို့ ရှင်းရှင်းလင်း လင်း ကြည့်နိုင်ဖို့ သင် ဖန်တီးပေးဖို့လိုလာပြီမဟုတ်လား။ ဒီလို ဖန်တီးဖို့ရာ ရှင်းလင်းလွယ်ကူရမယ်၊ organized ဖြစ်ရမယ်။ neat ဖြစ်ရမယ်။ ဒီလိုတွေ ဖြစ်ဖို့ အတွက် အကောင်းဆုံးကတော့ table ကို အသုံးပြုရမှာပါ။

ဒီသင်ခန်းစာမှာ အောက်ပါ အကြောင်းအရာများပါဝင်ပါတယ် -

- Creating a table to hold the data from the database .
- Creating column headings automatically .
- Populating the table with the results of a basic MySQL query .
- Populating the table with the results of more complex MySQL queries
- Making the output user - friendly

### Creating a table

database ထဲက data တွေကို Table ထဲ မထည့်ခင်မှာ Table layout ကို အရင်ဆုံး စဉ်းစားသင့်ပါတယ်။ ဘာတွေပါမယ်။ heading ကို ဘယ်လိုဟာတွေထားမယ် စသဖြင့်ပေါ့။ ကျွန်တော်တို့ အောက်က ပုံစံအတိုင်းလေး လုပ်ကြည့်မယ်လေ။ table1.php လို့ နာမည်ပေးလိုက်ပါ။

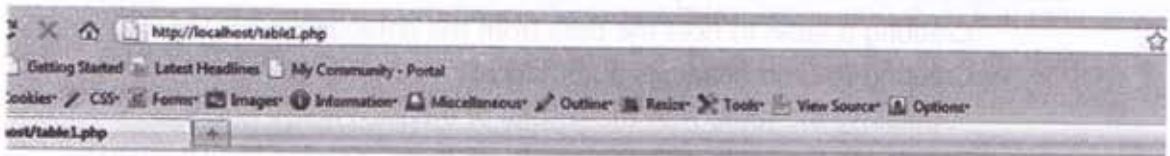
```

<div style="text-align: center;">
  <h2> Movie Review Database </h2>
  <table border="1" cellpadding="2" cellspacing="2" style="width: 70%;
margin-left: auto; margin-right: auto;">
    <tr>
      <th> Movie Title </th>

```

```

<th> Year of Release </th>
<th> Movie Director </th>
<th> Movie Lead Actor </th>
<th> Movie Type </th>
</tr>
</table>
</div>
    
```



**Movie Review Database**

| Movie Title | Year of Release | Movie Director | Movie Lead Actor | Mon |
|-------------|-----------------|----------------|------------------|-----|
|-------------|-----------------|----------------|------------------|-----|

အိုကေ။ ကျွန်တော်တို့ က table တစ်ခု တော့ တည်ဆောက်ပြီးသွားပြီ။ ဒီ Table ထဲကို data တွေ insert လုပ်ဖို့အတွက် ပထမဦးဆုံး database နဲ့ connect လုပ်ရမယ်လေ။ ဒီတော့ အရင် သင်ခန်းစာတွေအတိုင်း

```

<?php
    $db = mysql_connect('localhost', 'root', '') or
        die ("Unable to connect. Check your connection parameters.");
    mysql_select_db('moviesite1', $db) or die(mysql_error($db));
?>
<div style="text-align: center;">
    <h2> Movie Review Database </h2>
    <table border="1" cellpadding="2" cellspacing="2" style="width: 70%;
margin-left: auto; margin-right: auto;">
        <tr>
            <th> Movie Title </th>
            <th> Year of Release </th>
            <th> Movie Director </th>
            <th> Movie Lead Actor </th>
            <th> Movie Type </th>
        </tr>
    </table>
</div>

```

နောက်တစ်ဆင့်အနေနဲ့ dataတွေကို ပြဖို့အတွက် ကျွန်တော်တို့က MySQL server ကနေ data တောင်းခံရပါတယ်။ ဒီလို တောင်းခံဖို့အတွက် query တစ်ခု တည်ဆောက်လိုက်ပါမယ်။ ပြီးရင် Server ကနေ data row ဘယ်နှစ်ခု ထုတ်ပေးလိုက်လဲဆိုတာကို mysql\_num\_row() function သုံးပြီး ကြည့်လိုက်ပါဦးမယ်။

```

<?php
    $db = mysql_connect('localhost', 'root', '') or
        die ("Unable to connect. Check your connection parameters.");
    mysql_select_db('moviesite1', $db) or die(mysql_error($db));
    // retrieve information
    $query = 'SELECT movie_name, movie_year, movie_director, movie_leadac-
tor, movie_type

```



```

FROM movie
ORDER BY movie_name ASC, movie_year DESC';
$result = mysql_query($query, $db) or die(mysql_error($db));

// determine number of rows in returned result
$num_movies = mysql_num_rows($result);
?>
<div style="text-align: center;">
  <h2> Movie Review Database </h2>
  <table border="1" cellpadding="2" cellspacing="2" style="width: 70%;
margin-left: auto; margin-right: auto;">
    <tr>
      <th> Movie Title </th>
      <th> Year of Release </th>
      <th> Movie Director </th>
      <th> Movie Lead Actor </th>
      <th> Movie Type </th>
    </tr>
  </table>
</div>

```

အခုချိန်ထိ လုပ်ငန်းစဉ်တွေအားလုံးဟာ browser ရဲ့ နောက်ကွယ်မှာဘဲ အလုပ်လုပ်တုန်းရှိပြီးတော့ result ကို မပြနိုင်သေးပါဘူး။ result ကို ပြဖို့အတွက် data တွေကို output ထုတ်ပေးဖို့လိုပါတယ်။ ပထမဆုံး အနေနဲ့ ကျွန်တော်တို့ ရှေ့သင်ခန်းစာမှာ လေ့လာထားခဲ့တဲ့ while loop ကို သုံးပြီး result တစ်ခု ထုတ်ကြည့်ပါမယ်။ အပေါ်မှာ ပြထားတဲ့ script ရဲ့ </tr> tag အဆုံးနဲ့ </table> tag အဆုံးကြားမှာ while loop ကို ထားလိုက်ပါမယ်။ တစ်ခါတည်းဘဲ ထွက်လာမယ့် data တွေကို HTML နဲ့ တွဲထုတ်ဖို့ ရေးပါမယ်။

```

<?php
$db = mysql_connect('localhost', 'root', '') or
die ('Unable to connect. Check your connection parameters.');
```

```

// retrieve information
$query = 'SELECT movie_name, movie_year, movie_director, movie_leadac-
tor, movie_type
FROM movie
ORDER BY movie_name ASC, movie_year DESC';
$result = mysql_query($query, $db) or die(mysql_error($db));

// determine number of rows in returned result
$num_movies = mysql_num_rows($result);
?>
<div style="text-align: center;">
  <h2> Movie Review Database </h2>
  <table border="1" cellpadding="2" cellspacing="2" style="width: 70%;
margin-left: auto; margin-right: auto;">
    <tr>
      <th> Movie Title </th>
      <th> Year of Release </th>
      <th> Movie Director </th>
      <th> Movie Lead Actor </th>
      <th> Movie Type </th>
    </tr>
  <?php
    // loop through the results
    while ($row = mysql_fetch_assoc($result)) {
      extract($row);
      echo ' <tr> ';
      echo ' <td> ' . $movie_name . ' </td> ';
      echo ' <td> ' . $movie_year . ' </td> ';
      echo ' <td> ' . $movie_director . ' </td> ';
      echo ' <td> ' . $movie_leadactor . ' </td> ';
      echo ' <td> ' . $movie_type . ' </td> ';
      echo ' </tr> ';
    }
  </?php
  </table>
</div>

```

?>

</table>

<p> <?php echo \$num\_movies; ?> Movies </p>

</div>



**Movie Review Database**

| Movie Title  | Year of Release | Movie Director | Movie Lead Actor | Movie Type |
|--------------|-----------------|----------------|------------------|------------|
| Grand canyon | 1991            | 3              | 4                | 2          |
| Office Space | 1999            | 6              | 5                | 5          |
| TheGyanHoe   | 2003            | 2              | 1                | 5          |

3 Movies

### How it works

code တော်တော်များများ အလုပ်လုပ်ပုံကတော့ ရှေ့ကကသင်ခန်းစာတွေအတိုင်းပါဘဲ။ while statement မှာဆိုရင် loop ပတ်ပြီးတော့ record တွေဟာ return ဖြစ်ပါတယ်။ {} ထဲက code block တွေကို record တစ်ခုခြင်းစီ ထုတ်ပေးတာပါ။ တစ်ခါပြီးတိုင်းတစ်ခါ loop ပတ်ပါတယ်။ PHP မှာဆိုရင် loop ပတ်ဖို့ရာ record ဘယ်လောက်ရှိမလဲ၊ လက်ရှိ ဘယ် အဆင့်ထိရောက်နေပြီလဲဆိုတာ သူ့ဟာသူလည်း သိပါတယ်။

while loop ရဲ့ first line မှာရှိတဲ့ extract() function ကို field name နဲ့ နာမည်တူ variable ကို create လုပ်ပြီးတော့ variable တွေရဲ့ value တွေကို populate လုပ်ဖို့အတွက် သုံးလိုက်တာပါ။ current record တွေကို လုပ်တာနော်။ ကျန်တဲ့ line တွေကတော့ HTML ထဲ ထည့်သုံးလိုက်တာပါ။



Using heredoc syntax

ကျွန်တော်တို့ဟာ PHP mode မှာ echo statement နဲ့ Output တွေထုတ်ခဲ့ကြတယ်နော်။ <?php ?> ဆိုတဲ့ tags တွေကြားမှာရှိတာတွေပေါ့။ အဲဒီ tag နှစ်ခုရဲ့ အပြင်ဘက်မှာ HTML code တွေများလာတဲ့အခါမှာ တစ်ချို့တွေက မကြိုက်ကြဘူး။ PHP parse မလုပ်ဘဲ output မထုတ်ချင်ကြဘူး။ တစ်ချို့ကြတော့လည်း ဒီအတိုင်း(အထက်က ရေးခဲ့တဲ့အတိုင်း ) ရေးခြင်ကြတယ်။ PHP parse လုပ်ပြီးမှ ရေးမယ်ဆိုတဲ့သူတွေအတွက် heredoc ဆိုတဲ့ syntax လေးကို မိတ်ဆက်ပေးလိုက်ပါမယ်။

အရင်ဆုံး table1.php ထဲကဟာတွေကို table2.php ထဲကို ပြောင်းလိုက်ပါ။ ပြီးရင် အောက်ပါအတိုင်း ပြင်ရေးပါ။

၁။ table column's header ကို heredoc နဲ့ replace လုပ်ပါမယ်။ အောက်ပါအတိုင်းလုပ်ပါမယ်။

```
$table = <<<ENDHTML
<div style="text-align: center;" >
<h2> Movie Review Database </h2>
<table border="1" cellpadding="2" cellspacing="2" style="width: 70%;
margin-left: auto; margin-right: auto;">
<tr>
<th> Movie Title </th>
<th> Year of Release </th>
<th> Movie Director </th>
<th> Movie Lead Actor </th>
<th> Movie Type </th>
</tr>
ENDHTML;
```

၂။ ပြီးရင် while loop က echo statement တွေကိုလည်း ပြင်ရေးပါမယ်။

```
$table .= <<<ENDHTML
<tr>
```

```

<td> $movie_name </td>
<td> $movie_year </td>
<td> $movie_director </td>
<td> $movie_leadactor </td>
<td> $movie_type </td>
</tr>

```

ENDHTML;

```

၃။ ပြီးရင် အောက်ဆုံးက table ရဲ့ close tag နဲ့ number of movies အတွက် ပြင်ရေးပါမယ်။
$table .= <<<ENDHTML
</table>
<p> $num_movies Movies</p>
</div>
ENDHTML;

```

နံပါတ် ၂ နဲ့ နံပါတ် ၃ မှာ \$table .= ဆိုတာကို သတိထားမိမှာပါ။ နံပါတ် ၁ မှာက \$table = ဘဲလေ။ (dot) မပါဘူး။ အဲဒါက ဘာသဘောလဲဆိုတော့ .= လို့ရေးခြင်းအားဖြင့် ထိပ်ဆုံးက \$table ထဲမှာ save လုပ်ထားတယ်ပေါ့။ ဒါမှမဟုတ်လည်း depend ဖြစ်တယ်လို့ ပြောတာပါ။ = လို့ဘဲ ရေးလိုက်မယ်ဆိုရင် replace လုပ်သွားပါလိမ့်မယ်။ ခုက table တစ်ခုထဲမှာဘဲ ရှိနေတာလေ။ အဲဒီတော့ ကိုယ့်ဟာကို (dot) လေးတွေ ဖြုတ်ပြီး စမ်းကြည့်ပါ။ နောက်တစ်ခုက ENDHTML; ဆိုတာရေးတဲ့အခါမှာ သူ့ရဲ့ ရှေ့မှာ space မရှိရပါဘူး။

```

အဲဒီတော့ သင်ရေးလိုက်တဲ့ code တွေက အောက်ပါအတိုင်းဖြစ်ပါမယ်။
<?php
$db = mysql_connect('localhost', 'root', '') or
die ('Unable to connect. Check your connection parameters. ');
mysql_select_db('moviesite1', $db) or die(mysql_error($db));

```

```

// retrieve information
$query = 'SELECT movie_name, movie_year, movie_director, movie_
leadactor, movie_type
                FROM movie
                ORDER BY movie_name ASC, movie_year DESC';
$result = mysql_query($query, $db) or die(mysql_error($db));

// determine number of rows in returned result
$num_movies = mysql_num_rows($result);

$table = <<<ENDHTML
<div style="text-align: center;" >
<h2> Movie Review Database </h2>
<table border="1" cellpadding="2" cellspacing="2" style="width: 70%; margin-
left: auto; margin-right: auto;">
<tr>
<th> Movie Title </th>
<th> Year of Release </th>
<th> Movie Director </th>
<th> Movie Lead Actor </th>
<th> Movie Type </th>
</tr>
ENDHTML;
// loop through the results
while ($row = mysql_fetch_assoc($result)) {
    extract($row);
    $table .= <<<ENDHTML
        <tr>
        <td> $movie_name </td>
        <td> $movie_year </td>
        <td> $movie_director </td>

```



```

        <td> $movie_leadactor </td>
        <td> $movie_type </td>
    </tr>
ENDHTML;
}
$table .= <<<ENDHTML
    </table>
    <p> $num_movies Movies</p>
</div>
ENDHTML;
?>

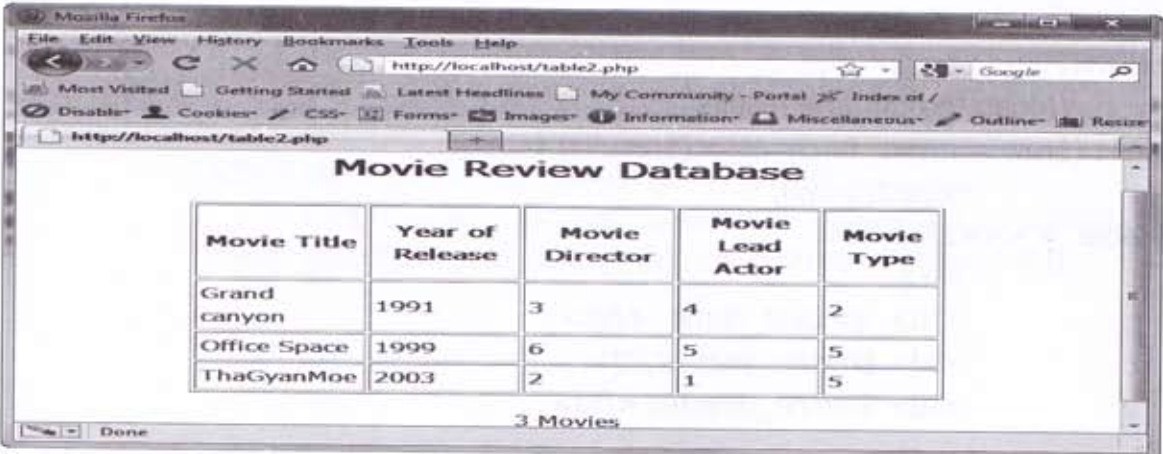
```

ဟုတ်ပြီ။ အဲဒီ script ကို run ကြည့်ပါ။ ဘာ result မှ တွေ့ရမှာမဟုတ်ဘူးနော်။ ဘာဖြစ်လို့လဲဆိုတော့ ကျွန်တော်တို့က table ဆိုတာကို variable လုပ်ခဲ့တယ်လေ။ ဒါကြောင့် အဲဒါကို echo နဲ့ output ပြန်ထုတ်ပေးမှရမယ်။ ဒီတော့ အောက်ဆုံးမှာ echo \$table; ဆိုတာလေးကို ထပ်ပေါင်းလိုက်ပါ။ ဒီလိုလေးပေါ့။ ဒါဆိုရင် output ထွက်လာမှာပါ။ ပုံ ကိုကြည့်ပါ။

```

ENDHTML;
echo $table;
?>

```



ENDHTML ဆိုတဲ့နေရာမှာ ကြိုက်တာ ရေးလို့ရတယ်နော်။ အစနဲ့ အဆုံးနဲ့ တူရင်ပြီ။

```
<<< HAHAHA
//code block
HAHAHA;
```

အောက်ကလိုဆိုရင်တော့ အလုပ်လုပ်မှာမဟုတ်ဘူး။

```
<<< HAHAHA
//code blcok
ENDHTML;
```

ခု ဇယားကွက် ကို တွေ့လာပြီဆိုတော့ movie director တွေ movie name တွေထည့်မယ်လေ။ ဒါမှ ကြည့်တဲ့သူ နားလည်မှာပေါ့။ ခု လေ့ကျင့်ခန်းမှာတော့ function လေးတွေ တည်ဆောက်ပြီး လုပ်ကြမယ်။

```
<?php
$db = mysql_connect('localhost','root','') or die (mysql_error($db));
mysql_select_db('moviesite',$db) or die (mysql_error($db));
// Get director
function get_director($director_id){
    global $db;
    $query = 'SELECT people_fullname
              FROM people
              WHERE people_id ='.$director_id;
    $result = mysql_query($query) or die (mysql_error($db));
    $row = mysql_fetch_assoc($result);
    extract($row);
    return $people_fullname;
}
// Get actor
function get_leadactor($leadactor_id){
    global $db;
    $query = 'SELECT people_fullname
              FROM people
              WHERE people_id ='.$leadactor_id;
    $result = mysql_query($query) or die (mysql_error($db));
    $row = mysql_fetch_assoc($result);
    extract($row);
```

```

        return $people_fullname;
    }
    // get movie
    function get_movietype($movietype_id){
        global $db;
        $query = 'SELECT movietype_label
                  FROM movietype
                  WHERE movietype_id = '.$movietype_id;
        $result = mysql_query($query) or die (mysql_error($db));
        $row = mysql_fetch_assoc($result);
        extract($row);
        return $movietype_label;
    }
    // retrieve information

    $query = 'SELECT movie_id, movie_name, movie_year, movie_director, movie_lead-
actor, movie_type
            FROM movie
            ORDER BY
            movie_name ASC,
            movie_year DESC';
    $result = mysql_query($query) or die (mysql_error($db));
    //$row = mysql_fetch_assoc($result);

    //determine the number of rows in results
    $num_movies = mysql_num_rows($result);
    $table = <<<ENDHTML
    <div style="text-align:center;">
        <h2>Movie review database</h2>
        <table border="1" cellspacing="2" cellpadding="2" style="width:70%;margin-
left:auto;margin-right:auto;">
            <tr>
                <th>Movie name</th>
                <th>Year of release</th>
                <th>Movie director</th>
                <th>Movie lead actor</th>
                <th>Movie type</th>
            </tr>
    ENDHTML;

```



```

// Loop through the results
while($row = mysql_fetch_assoc($result)){
    extract($row);
    $director = get_director($movie_director);
    $leadactor = get_leadactor($movie_leadactor);
    $movietype = get_movietype($movie_type);
    $table .= <<<ENDHTML
    <tr>
        <td><a href="movie_details.php?movie_
id=$movie_id" title="Click here more about details of $movie_name">$movie_name</
a></td>
        <td>$movie_year</td>
        <td>$director</td>
        <td>$leadactor</td>
        <td>$movietype</td>
    </tr>
ENDHTML;
}

$table .= <<<ENDHTML
</table>
<p> $num_movies</p>
</div>
ENDHTML;
echo $table;
?>

```

ဟာ.. code တွေ များလှချည်လား ..စိတ်ဓာတ်ကျမသွားနဲ့ဦးနော်။ ပြန်လည်း ကြည့်ကြည့်ဦး ။  
 သင် လေ့ကျင့်ခဲ့တဲ့ဟာတွေချည်းဘဲ။ ခု function တစ်ခုကို ရှင်းပြဦးမယ်။

```
// Get director
function get_director($director_id){
    global $db;
    $query = 'SELECT people_fullname
            FROM people
            WHERE people_id = '.$director_id;
    $result = mysql_query($query) or die (mysql_error($db));
    $row = mysql_fetch_assoc($result);
    extract($row);
    return $people_fullname;
}
```

အရင်ဆုံး get\_director ဆိုတဲ့ function name ကို ပေးလိုက်ပါတယ်။ statement အနေနဲ့ ကတော့ \$director\_id ပေါ့။ ပြီးရင် global variable ကို declare လုပ်ပါတယ်။ ဘာဖြစ်လို့လဲဆိုတော့ အောက်မှာ ဆက်သုံးမယ့် query အတွက် connection လုပ်ရမယ်လေ။ \$db က function code block ရဲ့ အပြင်ဖက်မှာလေ။ အပြင်က variable ကို ယူသုံးမယ်ဆိုရင် global လုပ်ရမယ်လို့ မှတ်ထားလို့ရပါတယ်။ ပြီးတော့ director name ရရှိအတွက် query တစ်ခု တည်ဆောက်လိုက်ပါတယ်။ full name တွေက people ဆိုတဲ့ Table ထဲက people\_fullname ဆိုတဲ့ field ထဲမှာ ရှိတာလေ။ အဲဒီတော့

```
SELECT people_fullname
FROM people
```

အဲဒီကမှ ဘယ်သူ့ နာမည်ကိုယူမှာလဲ။ အရင်ဆုံး people id ကို ရှာရမယ်။ ဘာကို reference လုပ်ရမလဲဆိုတော့ director id ကို လုပ်ရမယ်လေ။ အဲဒီတော့ people\_id = director\_id ပေါ့။ director\_id ဆိုတာ တကယ်မရှိဘူးနော်။ ကျွန်တော်တို့က နောက်ပိုင်းမှာ အဲဒီ \$director\_id ဆိုတဲ့နေရာမှာ တကယ်ရှိမယ့် \$movie\_director နဲ့ ယူသုံးလိုက်မယ့် သဘောအတွက် ရေးထားတာ။

```
'SELECT people_fullname
FROM people
WHERE people_id = '.$director_id;
```

အလုပ်လုပ်ပုံကို ရှင်းပြရရင် \$director\_id = 3 ဆိုရင်

```
'SELECT people_fullname
FROM people
```

WHERE people\_id = 3'; ဖြစ်သွားတာပေါ့။ ဒါဆိုရင် people\_id = 3 ဖြစ်တဲ့ နာမည်ကို ထုတ်ပေးမှာပါ။

နောက်ဆုံး while loop နေရာမှာ movie director, movie actor, movie type တွေအတွက် ( Function 3 ခု) ခေါ်ရေးလို့ရပြီလေ။

```
while($row = mysql_fetch_assoc($result)){
    extract($row);
    $director = get_director($movie_director);
    $leadactor = get_leadactor($movie_leadactor);
    $movietype = get_movietype($movie_type);
```

ပြီးရင် table ထဲက သက်ဆိုင်ရာနေရာတွေမှာ

```
<td>$director</td>
<td>$leadactor</td>
<td>$movietype</td>
```

ဆိုပြီး variable ၃ ခုကို ခေါ်ယူလိုက်ယုံပါဘဲ။

ဒီနေရာမှာ အပြန်ပြန်အလှန်လှန် စဉ်းစားစေချင်ပါတယ်။ ဘာဖြစ်လို့လဲဆိုတော့ တစ်ခုနဲ့ တစ်ခု ချိတ်ဆက်ပြီး အလုပ်လုပ်သွားတဲ့အတွက်ကြောင့် ကွင်းဆက်မိဖို့လိုပါတယ်။

**Adding links to table**

movie name နေရာမှာ သူနဲ့သက်ဆိုင်ရာ details data တွေကို ထည့်သွင်းနိုင်ဖို့အတွက် link ကိုထည့်ချင်တာပါ။ ဒါဆိုရင် movie name တစ်ခုကို click လုပ်လိုက်ရင်သက်ဆိုင်ရာ detail data တွေ ထွက်လာမှာပါ။

```
<td>
    <a href="movie_details.php?movie_id=$movie_id" title="Click here more about
details of
    $movie_name">$movie_name</a>
</td>
```

table2.php မှာ ဒီလိုလေး ပြင်ရေးပေးပါ။ ဒါကတော့ အထူးတလည်ရှင်းစရာမလိုတော့ဘူးထင်ပါတယ်။ ပြင်ပြီးသွားရင် table3.php လို့ နာမည်ပေးလိုက်ပါ။ ပြီးရင် Browser ကနေ



Movie review database

| Movie name   | Year of release | Movie director  | Movie lead actor | Movie type |
|--------------|-----------------|-----------------|------------------|------------|
| Grand canyon | 1991            | Lawrence Kasdan | Kevin Klein      | Drama      |
| Office Space | 1999            | Mike judge      | Ron livingston   | Comedy     |
| ThaGyanMoe   | 2003            | ZinWaing        | NayAung          | Drama      |

3 Movies

ဆိုပြီးတွေ့ရမှာပါ။ ဟုတ်ကဲ့ link တော့ လုပ်ပြီးသွားပါပြီ။ ဒါပေမယ့် နှိပ်လိုက်ရင် မရှိဘူးဘဲပြနေမှာပါ။ ကျွန်တော်တို့က link လေးဘဲလုပ်ရသေးတာကိုး။ ခု ကျွန်တော်တို့က movie detail တွေ ပြနိုင်ဖို့အတွက် နောက်ထပ် ဒေတာတွေလိုပါတယ်။ database ထဲကို ထပ်ပေါင်းပေးပါမယ်။ column တွေထပ်ဖြည့်မယ်လေ။

```
<?php
$db = mysql_connect('localhost','root','') or die (mysql_error($db));
mysql_select_db('moviesite1',$db) or die (mysql_error($db));
//Alter the movie table to add movie length, cost and taking fields
$query = 'ALTER TABLE movie ADD COLUMN (
movie_runningtime TINYINT UNSIGNED NULL,
movie_cost DECIMAL(4,1) NULL,
```

```

        movie_taking DECIMAL(4,1) NULL
    );
    mysql_query($query,$db) or die (mysql_error($db));

// Adding data to each movie
$query = 'UPDATE movie SET
        movie_runningtime =101,
        movie_cost = 81,
        movie_taking = 242.6
        WHERE movie_id = 1';
    mysql_query($query,$db) or die (mysql_error($db));

$query = 'UPDATE movie SET
        movie_runningtime =89,
        movie_cost = 10,
        movie_taking = 10.8
        WHERE movie_id = 2';
    mysql_query($query,$db) or die (mysql_error($db));

$query = 'UPDATE movie SET
        movie_runningtime = 134,
        movie_cost = NULL,
        movie_taking = 33.2
        WHERE movie_id= 3';
    mysql_query($query,$db) or die (mysql_error($db));

echo ' Successfully added data to the database.';

?>
```

db3.php လို့ နာမည်ပေးပြီး run လိုက်ပါ။ Successfully added data to the database. ဆိုတာ ပေါ်လာပါမယ်။ ဘာတွေ ပါလာသလဲဆိုရင် movie တစ်ခုခြင်းစီရဲ့ ကုန်ကျစားရိတ်(movie\_cost)၊ (ကြာမြင့်ချိန်)movie length, ပြန်လည်ရရှိငွေ(movie\_taking) ဆိုတာတွေပါလာမယ်။ ဒါတွေအပြင် ကျွန်တော်တို့က နောက်ထပ် နည်းနည်းလေး ထပ်ကွန့်ပြီးလုပ်ရရင် movie cost ရယ် movie taking ရယ်ကို နှိုင်းယှဉ်ပြီး ဒီ ဇာတ်ကားဟာ မြတ်သလား ရုံးသလား ဆိုတာကို အရောင်နဲ့ ပြပါမယ်။ သူ့ပုံစံလေးက အောက်ပါအတိုင်းလေးပါ။

// Function to calculate if a movie made a profit, loss or just broke even

```
function calculate_differences($takings, $cost) {
    $difference = $takings - $cost;

    if ($difference < 0) {
        $color = 'red';
        $difference = '$' . abs($difference) . 'million';
    } elseif ($difference > 0) {
        $color = 'green';
        $difference = '$' . $difference . 'million';
    } else {
        $color = 'blue';
        $difference = 'broke even';
    }

    return '<span style="color:' . $color . ';">' . $difference . '</span >';
}
```

ရုံးရင် အနီနဲ့ပြခိုင်းပြီး မြတ်ရင် အစိမ်းနဲ့ ပြမှာပါ။ ကုန်ကျစားရိတ်မသိတဲ့ဟာကိုတော့ အပြာ နဲ့ ပြခိုင်းထားပါတယ်။

သူ့ကို အားလုံးနဲ့ပေါင်းစပ်ရေးလိုက်ရင် အောက်ပါအတိုင်းရမှာ။ movie\_details.php လို့ နာမည်ပေးလိုက်ပါ။



```

<?php
// Connect to mysql database
$db = mysql_connect('localhost','root','') or die ('Unable to connect');
mysql_select_db('moviesite1',$db) or die (mysql_error($db));

function get_director($director_id){
    global $db;
    $query = 'SELECT people_fullname
              FROM people
              WHERE people_id = '.$director_id;
    $result = mysql_query($query, $db) or die(mysql_error($db));
    $row = mysql_fetch_assoc($result) or die (mysql_error($db));
    extract($row);
    return $people_fullname;
}

function get_leadactor($leadactor_id){
    global $db;
    $query = 'SELECT people_fullname
              FROM people
              WHERE people_id = '.$leadactor_id;
    $result = mysql_query($query, $db) or die (mysql_error($db));
    $row = mysql_fetch_assoc($result) or die (mysql_error($db));
    extract ($row);
    return $people_fullname;
}

function get_movietype($type_id){
    global $db;
    $query = 'SELECT movietype_label
              FROM movietype
              WHERE movietype_id = '.$type_id;

```

```

$result = mysql_query($query,$db) or die (mysql_error($db));
$row = mysql_fetch_assoc($result) or die (mysql_error($db));
extract ($row);
return $movietype_label;
}
// Function to calculate if a movie made a profit, loss or just broke even

function calculate_differences($takings, $cost) {

    $difference = $takings - $cost;

    if ($difference < 0) {
        $color = 'red';
        $difference = '$' . abs($difference) . 'million';
    } elseif ($difference > 0) {
        $color = 'green';
        $difference = '$' . $difference . 'million';
    } else {
        $color = 'blue';
        $difference = 'broke even';
    }

    return '<span style="color:' . $color . ';">' . $difference . '</span >';
}
// Retrieve information
$query = 'SELECT movie_name, movie_year, movie_director, movie_lead-
actor, movie_type,movie_type, movie_runningtime, movie_cost, movie_taking
FROM movie
WHERE

```

```

        movie_id = '._GET['movie_id'];
$result = mysql_query($query,$db);
$row = mysql_fetch_assoc($result);
$movie_name = $row['movie_name'];
$movie_director = get_director($row['movie_director']);
$movie_leadactor = get_leadactor($row['movie_leadactor']);
//$movie_type = get_movietype($row['movie_type']);
$movie_year = $row['movie_year'];
$movie_runningtime = $row['movie_runningtime'];
$movie_taking = $row['movie_taking'];
$movie_cost = $row['movie_cost'];
$movie_health = calculate_differences($row['movie_taking'], $row['movie_
cost']);
// Display the information.
echo <<<ENDHTML
    <html>
        <head>
            <title>Details and reviews for $movie_name.</title>
            <body>
                <div style="text-align:center";>
                    <h2>$movie_name</h2>
                    <h3><em>Get details</em></h3>
                    <table    cellpadding="2"    cellspacing="2"
style="width:70%;margin-left:auto;margin-right:auto;">
                        <tr>
                            <td><strong>Title</strong></td>
                            <td>$movie_name</td>
                            <td><strong>Releaseyear</strong><td>
                            <td><strong>$movie_year</strong><td>
                        </tr>

```



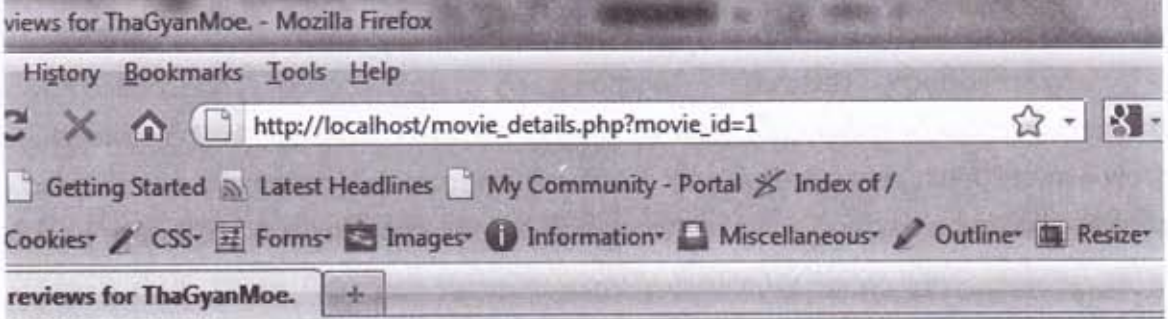


```

        <tr>
            <td><strong>Movie director</strong></td>
            <td>$movie_director</td>
            <td><strong>Cost</strong></td>
            <td><strong>$movie_cost</strong></td>
        </tr>
        <tr>
            <td><strong>Lead actor</strong></td>
            <td>$movie_leadactor</td>
            <td><strong>Takings</strong></td>
            <td><strong>$movie_taking</strong></td>
        </tr>
        <tr>
            <td><strong>Running Time</strong></td>
            <td>$movie_runningtime</td>
            <td><strong>Health</strong></td>
            <td><strong>$movie_health</strong></td>
        </tr>
    </table>
ENDHTML;
?>

```

table3.php ကို browser ကနေ ခေါ်လိုက်ပါ။ ပြီးရင် movie name တစ်ခုကို ကလစ်လုပ်လိုက်ပါ။ ဒါဆိုရင် သူနဲ့ ပါတ်သတ်တဲ့ movie detail တစ်ခု ထပ်ထွက်လာမှာပါ။



## ThaGyanMoe

*Get details*

|                       |            |                     |                |
|-----------------------|------------|---------------------|----------------|
| <b>Title</b>          | ThaGyanMoe | <b>Release year</b> | 2003           |
| <b>Movie director</b> | ZinWaing   | <b>Cost</b>         | 81.0           |
| <b>Lead actor</b>     | NayAung    | <b>Takings</b>      | 242.6          |
| <b>Running Time</b>   | 101        | <b>Health</b>       | \$161.6million |

### How it works

function ငှ ခုကို ရှင်းပြပြီးဖြစ်လို့ ကျော်ပြီး ရှင်းလိုက်တော့မယ်။

```
$query = 'SELECT movie_name, movie_year, movie_director, movie_lead-actor, movie_type,movie_type, movie_runningtime, movie_cost, movie_taking
FROM movie
WHERE
movie_id = '.$_GET['movie_id'];
```

အဲဒီ script မှာဆိုရင် movie\_id = '\$\_GET['movie\_id']; ဆိုတာ table3.php ကနေ request လုပ်လာတဲ့ movie id ကို GET method သုံးပြီး movie\_details.php မှာ compare လုပ်တာဖြစ်ပါတယ်။

နောက်တစ်ခုက extract() function ကို မသုံးတော့တာ သင်သတိထားမိရင် သတိထားမိပါလိမ့်မယ်။ တစ်ခါတည်း \$row ဆိုတဲ့ array ထဲကနေ တိုက်ရိုက်တန်းယူသုံးထားပါတယ်။ \$row = mysql\_fetch\_assoc(\$result); ဆိုတဲ့အထဲမှာ movie အားလုံးရဲ့ data က ပါလာတာမဟုတ်ဘဲ ကိုယ်သိချင်တဲ့ movie တစ်ခုရဲ့ data ဘဲပါတာဖြစ်တဲ့ အတွက် တစ်ခါတည်း \$row['\$value'] ဆိုတဲ့ array function သုံးပြီး ယူသုံးလိုက်တာပါ။ မှတ်မိလားမသိဘူး။ မမှတ်မိရင် ရှေ့က array အခန်းကို ပြန်ဖတ်စေချင်တယ်။

**Creating movie review table**

ခု movie တစ်ခုခြင်းစီမှာ သင့်ရဲ့ မှတ်ချက်လေးတွေ ထည့်ပါဦးမယ်။ တစ်ခါတည်း movie rating လေးတွေပါ ထည့်ကြည့်မယ်လေ။ သိပ်မခက်ပါဘူး။ အရင်ဆုံး database ထဲကို လိုတာလေးတွေ ထည့်ဦးမယ်လေ။

```
<?php
$db = mysql_connect('localhost','root','') or die ('Unable to connect to data-
base. Check your database parameters');
mysql_select_db('moviesite1',$db) or die (mysql_error($db));

$query = 'CREATE TABLE IF NOT EXISTS reviews(
    review_movie_id INT UNSIGNED NOT NULL,
    review_date DATE NOT NULL,
    reviewer_name VARCHAR(255) NOT NULL,
    reviewer_comment VARCHAR(255) NOT NULL,
    review_rating TINYINT UNSIGNED NOT NULL DEFAULT
0,
    KEY (review_movie_id)
) ENGINE=MYISAM DEFAULT CHARSET=utf8';
mysql_query($query,$db) or die (mysql_error($db));
```



```
// Insert data to table
$query =<<<<ENDSQL
    INSERT INTO reviews (review_movie_id, review_date, reviewer_
name, reviewer_comment, review_rating)
    VALUES
    (1,"2011-8-4","KoKo","I thought this was a great movie Even though
my girlriend made me see it against my will.",4),
    (1,"2011-9-4","MaMa","I like eraserhead better",3 ),
    (1,"2011-8-3","Shwe Zin","Very very Good. ",5),
    (2,"2011-8-25","Billy Bob","This is my favourite movie",5),
    (3,"2011-9-6","George B.,"I liked this movie , even though I thought
it was an informational movie from my travel agent.",4)
ENDSQL;
mysql_query($query,$db);
echo ' Movie database successfully updated';
?>
```

ဒါကို db4.php လို့ နာမည်ပေးပြီး Run လိုက်ပါ။ Movie database successfully updated ဆိုပြီး ပေါ်လာရင် ရပါပြီ။

ဒီသင်ခန်းစာမှာဆိုရင် movies and review ဆိုတဲ့ table နှစ်ခုကို link လုပ်ပြီးသက်ဆိုင်ရာ movie တွေ အတွက် review တွေကို ထုတ်ပေးမှာဖြစ်ပါတယ်။

ဒီသင်ခန်းစာအတွက် movie\_details.php ကို a lot of changes လုပ်ရမှာဖြစ်တဲ့အတွက် သင့်အနေနဲ့ movie\_details.php ကို back up လုပ်ထားသင့်ပါတယ်။ back up လုပ်ထားခြင်းအားဖြင့် သင့်အနေနဲ့ အမှားတွေ အများကြီး လုပ်မိတဲ့အခါ ဒါမှမဟုတ် original version ကို ပြန်လို့ခြင်တဲ့အခါမှာ အချိန်မရွေး ပြန်သွားလို့ရပါတယ်။

ခု ကျွန်တော်တို့ movie rating အတွက် function လေးတစ်ခု လုပ်ကြည့်ရအောင်။

```
function generate_ratings($rating){
    $movie_rating = ""; // Not double quote.
    for ($i=0;$i<$rating;$i++){
        $movie_rating .= '';
    }
    return $movie_rating;
}
```

ပြီးရင် အောက်ဆုံးနေရာမှာ movie information table ကို အပိုင်း ၂ ပိုင်းဖြစ်အောင် ခွဲထုတ်လိုက်ပါမယ်။ ဒါဆိုရင် ဘာ code မှမရှိသေးခင်မှာ အောက်ပါအတိုင်းဖြစ်နေမှာပေါ့။

```
</table>
ENDHTML;
```

```
echo <<< ENDHTML
    </div>
    </body>
</html>
ENDHTML;
?>
```

ပြီးရင် အောက်ပါ code တွေကို ခုနက heredoc ၂ ခုကြားမှာ ထည့်လိုက်ပါ။

```
// Retrieve the information reviews for this movie
$query = 'SELECT review_movie_id,review_date, reviewer_name, reviewer_com-
ment, review_rating
        FROM reviews
        WHERE review_movie_id= '.$_GET['movie_id'].'
        ORDER BY review_date DESC';
$result = mysql_query($query,$db) or die (mysql_error($db));
```

```
// Display reviews
echo <<<ENDHTML
    <h3><em>Reviews</em></h3>
    <table cellpadding="2" cellspacing="2" style="width:90%; margin-
left:auto;margin-right:auto;">
        <tr>
            <th style="width: 7em;">DATE</th>
            <th style="width: 10em;">Reviewer</th>
            <th>Comments</th>
            <th style="width:5em;">Rating</th>
        </tr>

ENDHTML;
while($row = mysql_fetch_assoc($result)){
    $date = $row['review_date'];
    $name= $row['reviewer_name'];
    $comment = $row['reviewer_comment'];
    $rating = generate_ratings($row['review_rating']);

    echo <<<ENDHTML
    <tr>
        <td style="vertical-align:top;text-align:center;">$date</td>
        <td style="vertical-align:top;">$name</td>
        <td style="vertical-align:top;">$comment</td>
        <td style="vertical-align:top;">$rating</td>
    </tr>

ENDHTML;
}
```



```
// Connect to mysql database
$db = mysql_connect('localhost','root','') or die ('Unable to connect');
mysql_select_db('moviesite1',$db) or die (mysql_error($db));
function get_director($director_id){
    global $db;
    $query = 'SELECT people_fullname
              FROM people
              WHERE people_id = '.$director_id;
    $result = mysql_query($query, $db) or die(mysql_error($db));
    $row = mysql_fetch_assoc($result) or die (mysql_error($db));
    extract($row);
    return $people_fullname;
}
function get_leadactor($leadactor_id){
    global $db;
    $query = 'SELECT people_fullname
              FROM people
              WHERE people_id = '.$leadactor_id;
    $result = mysql_query($query, $db) or die (mysql_error($db));
    $row = mysql_fetch_assoc($result) or die (mysql_error($db));
    extract ($row);
    return $people_fullname;
}
function get_movietype($type_id){
    global $db;
    $query = 'SELECT movietype_label
              FROM movietype
              WHERE movietype_id = '.$type_id;
    $result = mysql_query($query,$db) or die (mysql_error($db));
    $row = mysql_fetch_assoc($result) or die (mysql_error($db));
    extract ($row);
```

```

        return $movietype_label;
    }
    // Function to calculate if a movie made a profit, loss or just broke even
    function calculate_differences($takings, $cost) {

        $difference = $takings - $cost;

        if ($difference < 0) {
            $color = 'red';
            $difference = '$' . abs($difference) . 'million';
        } elseif ($difference > 0) {
            $color = 'green';
            $difference = '$' . $difference . 'million';
        } else {
            $color = 'blue';
            $difference = 'broke even';
        }

        return '<span style="color:' . $color . ';">' . $difference . '</span >';
    }
    // Generate ratings
    function generate_ratings($rating){
        $movie_rating = ""; // Not double quote.
        for ($i=0;$i<$rating;$i++){
            $movie_rating .= '';
        }
        return $movie_rating;
    }
}

```

```

// Retrieve information
$query = 'SELECT movie_name, movie_year, movie_director, movie_leadactor, mov-
ie_type,movie_type, movie_runningtime, movie_cost, movie_taking
        FROM movie
        WHERE
        movie_id = '.$_GET['movie_id'];

$result = mysql_query($query,$db);
$row = mysql_fetch_assoc($result);
$movie_name = $row['movie_name'];
$movie_director = get_director($row['movie_director']);
$movie_leadactor = get_leadactor($row['movie_leadactor']);
//$movie_type = get_movietype($row['movie_type']);
$movie_year = $row['movie_year'];
$movie_runningtime = $row['movie_runningtime'];
$movie_taking = $row['movie_taking'];
$movie_cost = $row['movie_cost'];
$movie_health = calculate_differences($row['movie_taking'], $row['movie_cost']);

// Display the information.
echo <<<ENDHTML
    <html>
        <head>
            <title>Details and reviews for $movie_name.</title>
        <body>
            <div style="text-align:center;">
                <h2>$movie_name</h2>
                <h3><em>Get details</em></h3>
                <table        cellpadding="2"        cellspacing="2"
style="width:70%;margin-left:auto;margin-right:auto;">

```



```

<tr>
  <td> <strong>Title</strong> </td>
  <td> $movie_name </td>
  <td> <strong>Release year</strong> </td>
  <td> <strong>$movie_year</strong> </td>
</tr>
<tr>
  <td> <strong>Movie director</strong> </td>
  <td> $movie_director </td>
  <td> <strong>Cost</strong> </td>
  <td> <strong>$movie_cost</strong> </td>
</tr>
<tr>
  <td> <strong>Lead actor</strong> </td>
  <td> $movie_leadactor </td>
  <td> <strong>Takings</strong> <td>
  <td> <strong>$movie_taking</strong> <td>
</tr>
<tr>
  <td> <strong>Running Time</strong> </td>
  <td> $movie_runningtime </td>
  <td> <strong>Health</strong> <td>
  <td> <strong>$movie_health</strong> <td>
</tr>
</table>

```

ENDHTML;

// Retrieve the information reviews for this movie

```
$query = 'SELECT review_movie_id,review_date, reviewer_name, reviewer_
comment, review_rating
```

```

FROM reviews
WHERE review_movie_id= '$_GET['movie_id'].'
ORDER BY review_date DESC';
$result = mysql_query($query,$db) or die (mysql_error($db));
// Display reviews
echo <<<ENDHTML
    <h3><em>Reviews</em></h3>
        <table cellpadding="2" cellspacing="2" style="width:90%; margin-
left:auto;margin-right:auto;">
            <tr>
                <th style="width: 7em;">DATE</th>
                <th style="width: 10em;">Reviewer</th>
                <th>Comments</th>
                <th style="width:5em;">Rating</th>
            </tr>

ENDHTML;
while($row = mysql_fetch_assoc($result)){
    $date = $row['review_date'];
    $name= $row['reviewer_name'];
    $comment = $row['reviewer_comment'];
    $rating = generate_ratings($row['review_rating']);

    echo <<<ENDHTML
        <tr>
            <td style="vertical-align:top;text-align:center;">$date</td>
            <td style="vertical-align:top;">$name</td>
            <td style="vertical-align:top;">$comment</td>
            <td style="vertical-align:top;">$rating</td>
        </tr>

ENDHTML;

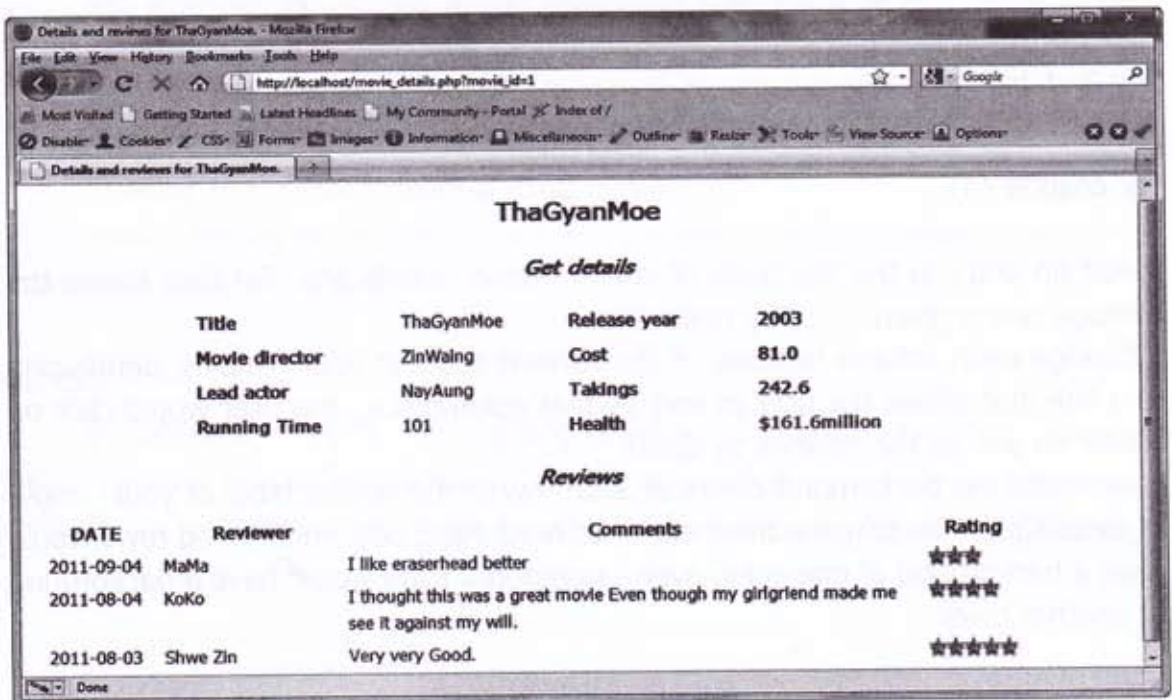
```

```

}
echo <<< ENDHTML
  </div>
  </body>
</html>
ENDHTML;
?>

```

table3.php ကို run ပြီးတော့ movie name တစ်ခုကို click လုပ်ကြည့်ပါ။ ဒါဆိုရင် သူနဲ့သက်ဆိုင်တဲ့ details တွေရော review တွေရော ထွက်လာမှာပါ။





generate rating function က နားလည်မယ်လို့ယူဆပါတယ်။ တစ်ခု သတိထားမိလား := ကို သုံးထားတာလေ။ အဲဒါက ဘာလဲဆိုတော့ movie rate 1 ထက်ကျော်ရင် star image ကိုလည်း ၁ ထပ်ပေါင်းပေးပြီး ပုံရိပ်အရေအတွက်ကိုပါ ထပ်တိုးခိုင်းတာပါ။

ခု အခြေအနေထိ သင်ရောက်လာပြီဆိုရင် သင်ဟာ code တွေကို တော်တော်လေး ကိုင်တွယ်နိုင်ခဲ့ပါပြီ။ changes မျိုးစုံလုပ်လာခဲ့ရတယ်။ ဒါပေမယ့် နောက်ဆုံးမှာတော့ အဆင်ပြေပြေ ဖြစ်ခဲ့တာဘဲမလား။ အရာအားလုံးကို တစ်ခါတည်းနဲ့ ဖြစ်ဖို့ဆိုတာ တော်တော်ခက်ပါတယ်။ ကိုယ်နဲ့ အကျွမ်းတဝင် မဖြစ်တဲ့ဟာဆို ပိုဆိုးတယ်လေ။ ခုလည်း ဒီလိုပါဘဲ။ ထပ်ခါတလဲလဲ လေ့ကျင့်ပါ။ အဓိကက ကိုယ်တိုင်ရေးဖို့လိုပါတယ်။ ရေးမှ သိမှာ။ မှတ်မိမှာ။ နားလည် လွယ်မှာပေါ့။

ဒီသင်ခန်းစာမှာဆိုရင် MySQL ကို table တွေနဲ့ ဘယ်လို ချိတ်ဆက် အသုံးပြုရမယ် ၊ movie\_ id ကို depend လုပ်ပြီး review table ထဲက reviews တွေအားလုံးကို ဘယ်လို အသုံးပြုရတာ၊ \$\_GET ဆိုတဲ့ superglobal သုံးပြီး page တစ်ခုကနေ နောက်တစ်ခုကို access လုပ်ရတာတွေကို သင့်အနေနဲ့ သိရှုသွားမှာဖြစ်ပါတယ်။

Exercises

For chapter (4)

- 1. Add an entry in the top table of your `movie_details.php` file that shows the average rating given by reviewers.
- 2. Change each column heading of the reviews table in your `movie_details.php` to a link that allows the user to sort by that column (i.e., the user would click on "Date" to sort all the reviews by date).
- 3. Alternate the background colors of each row in the review table of your `movie_details.php` file to make them easier to read. Hint: odd - numbered rows would have a background of one color, even - numbered rows would have a background of another color.

## Chapter (5) FORM Elements: Letting the User Work with Data

Interactive website တွေမှာ user input တွေ လိုအပ်ပါတယ်။ ယျေဘူယျအားဖြင့် form ပုံစံတွေနဲ့ သွားတာများပါတယ်။ HTML မှာတော့ form ပုံစံလေးတွေက ရိုးရိုးလေးတွေပါဘဲ။ အဲဒီထဲမှာ ကျွန်တော်တို့ submit လုပ်လိုက်တဲ့ data တွေကို ဘယ်နေရာ ကို ဘယ်လို ပို့မယ်ဆိုတာတွေ ဖော်ပြထားပါတယ်။ ပြီးတော့မှ PHP နဲ့ တွဲပြီး သုံးပါတယ်။ PHP script ဟာ HTML form က ပို့လိုက်တဲ့ data တွေကို ရယူပြီး action တစ်ခုခု ပြုလုပ်ပါတယ်။ ဘာ action တွေလဲဆိုတော့ Database ကို update လုပ်တာတို့၊ Sending an email, testing the data format and so on ပေါ့ဗျာ။ ကိုယ်ဘာလုပ်ခြင်သလဲဆိုတာတွေကို ဆက်လုပ်တာပေါ့။

အခု ဒီသင်ခန်းစမှာ သင်ဟာ simple application တစ်ခု ပြုလုပ်ပြီး edit, delete or add members of a data set တွေ ကို ပြုလုပ်ရမှာပါ။ PHP and MySQL interaction တွေ ပြုလုပ်ရမှာပေါ့ဗျာ။

ဒီ သင်ခန်းစမှာ အောက်ပါ အကြောင်းအရာများ ပါဝင်မှာဖြစ်ပါတယ် -

Creating forms using buttons, text boxes, and other form elements .

Creating PHP scripts to process HTML forms .

Passing hidden information to the form - processing script via hidden form controls and a URL query string

### First form

မိုင်တစ်ထောင် ခရီးကို စတင်ဖို့အတွက်ဆိုရင်တောင် ကျွန်တော်တို့ဟာ ပထမဆုံး ခြေတစ်လှမ်းကနေ စ ရတယ်မဟုတ်လားဗျာ။ ခုလည်း ကျွန်တော်တို့က very simple form ကနေ စတင်မယ်လေ။ ခု exercise မှာ HTML form ကနေ ရေးသွင်းလိုက်တဲ့ နာမည်ကို PHP နဲ့ ပြန်ပြမှာပါ။

ပထမဆုံး Form ပုံစံလေးတစ်ခုရေးပါမယ်။ form1.html လို့နာမည်ပေးပါ။

```

<html>
  <head>
    <title>Say my name</title>
  </head>
  <body>
    <form action="formprocess1.php" method="post">
      Username: <input type="text" name="username" />
      <input type="submit" name="submit" value="Submit" />
    </form>
  </body>
</html>

```

ခု ဒုတိယအနေနဲ့ Form ကနေ ပို့လိုက်တဲ့ data တွေကို PHP နဲ့ ပြန်ဖမ်း ပြီး output ထုတ်ပေးပါမယ်။

```

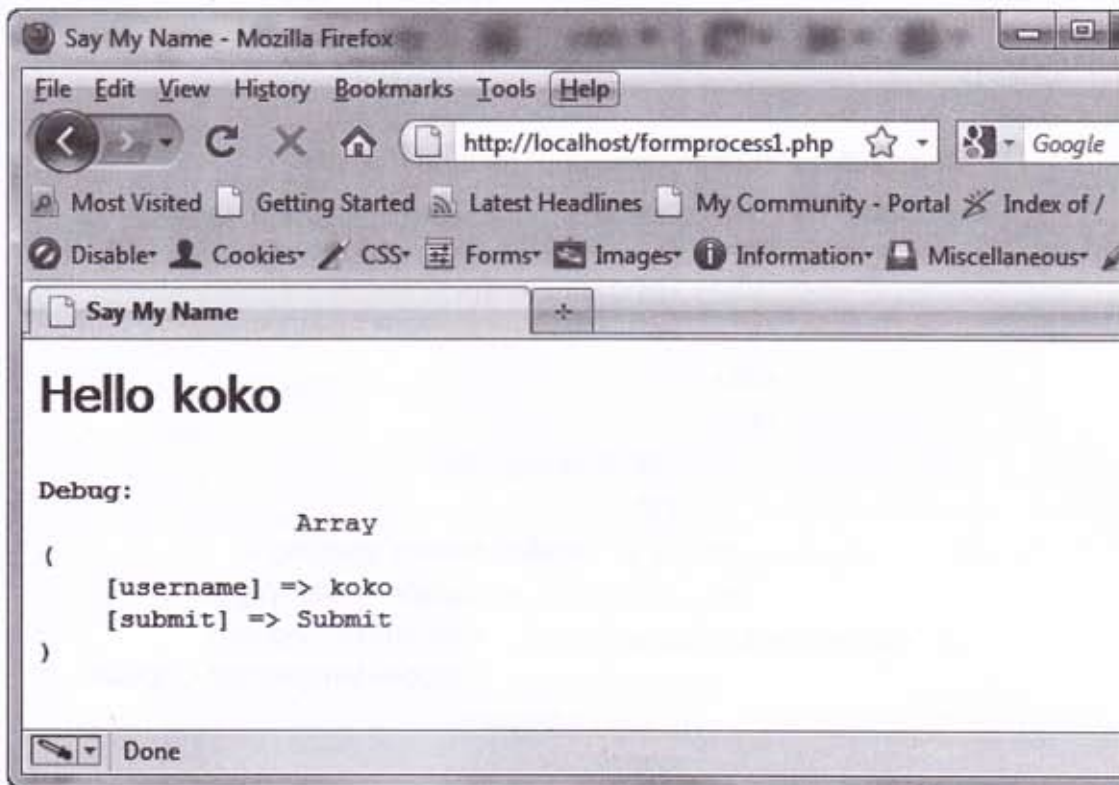
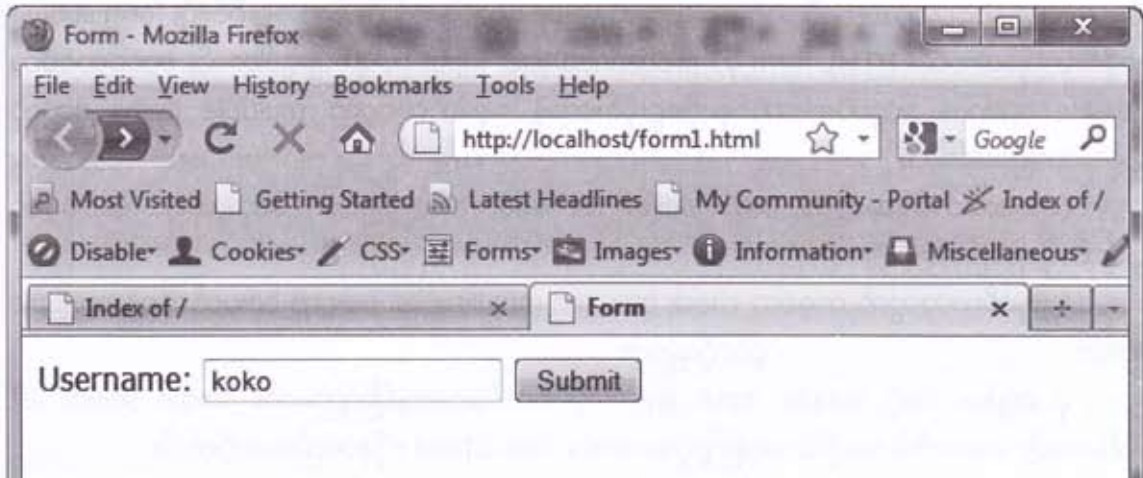
<html>
  <head>
    <title>Say My Name</title>
  </head>
  <body>
    <h2><?php echo 'Hello ' . $_POST['username']; ?></h2>
    <pre><strong>Debug:</strong>
    <?php print_r($_POST)?></pre>
  </body>
</html>

```

ဒီ script မှာဆိုရင် HTML form ကနေ post method နဲ့ပို့လိုက်တဲ့ data အားလုံးကို super global variable ဖြစ်တဲ့ \$\_POST နဲ့ ပြန်ယူပါတယ်။ array index name ကတော့ HTML form မှာ သုံးခဲ့တဲ့ name=" " ဆိုတဲ့ အထဲက name ဘဲဖြစ်ပါတယ်။ ကျွန်တော်တို့က name="username" လို့ သုံးခဲ့တော့ \$\_POST['username'] ဆိုပြီးဖြစ်သွားတာပေါ့။

print\_r ကတော့ the whole content of super global (\$\_POST array) ထဲကဟာတွေကို အကုန်ထုတ်ပြနိုင်တာဖြစ်ပါတယ်။ ပုံမှာကြည့်ပါ။





ခု ဒီ exercise မှာတော့ user ကို သင် ပေးထားတဲ့ value တွေထဲက ကြိုက်ရာ ရွေးခိုင်းပြီး form submit လုပ်ခိုင်းမှာဖြစ်ပါတယ်။ HTML form မှာ predefined user input က နှစ်မျိုးရှိပါတယ်။ ပထမတစ်ခုက available options တွေထဲက တစ်ခုကိုရွေးခိုင်းတာနဲ့ နောက်တစ်ခုက multiple items တွေကို ရွေးခိုင်းတာဖြစ်ပါတယ်။

(၁)ပထမတစ်ခုအတွက် Drop down list and radio button တွေနဲ့ဆိုရင် တစ်ခု ဘဲ ရွေးချယ်လို့ရမှာဖြစ်ပြီး

(၂)ဒုတိယအတွက် ကတော့ check box and multiline list box ကို သုံးမယ်ဆိုရင် multiple choices လုပ်လို့ရမှာပါ။

ခု ကျွန်တော်တို့ simple input type ကို စတင်လေ့ကျင့်ကြည့်မယ်။ single select list နဲ့လုပ်မှာပေါ့။ အောက်ပါ အတိုင်း ရေးကြည့်ရအောင်။ form2.html လို့နာမည်ပေးလိုက်ပါ။

```

<html>
  <head>
    <title>form2</title>
  </head>
  <body>
    <form action="formprocess2.php" method="post">
      <table>
        <tr>
          <td>Name:</td>
          <td><input type="text" name="name" /></td>
        </tr>
        <tr>
          <td>Greeting</td>
          <td>
            <select name="greeting">
              <option>Hello</option>
              <option>Hi</option>
              <option>Mingalarpar</option>
            </select>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>

```

```

        <tr>
            <td> </td>
            <td>
                <input type="checkbox" name="debug"
checked="checked" />
                Display debug info
            </td>
        </tr>
        <tr>
            <td colspan="2" style="text-align:center;">
                <input type="submit" name="submit"
value="submit" />
            </td>
        </tr>
    </table>
</form>
</body>
</html>

```

ပြီးသွားရင် form2.html အတွက် formprocess2.php ကို ဆက်ရေးပါမယ်။

```

<html>
  <head>
    <title>Greeting2</title>
  </head>
  <body>
    <?php
      echo $_POST['name'].' '.$_POST['greeting'];
      if(isset($_POST['debug'])){
        echo '<pre> <h2>Debug</h2>'. "\n";
        print_r($_POST);
        echo '</pre>';
      }
    ?>

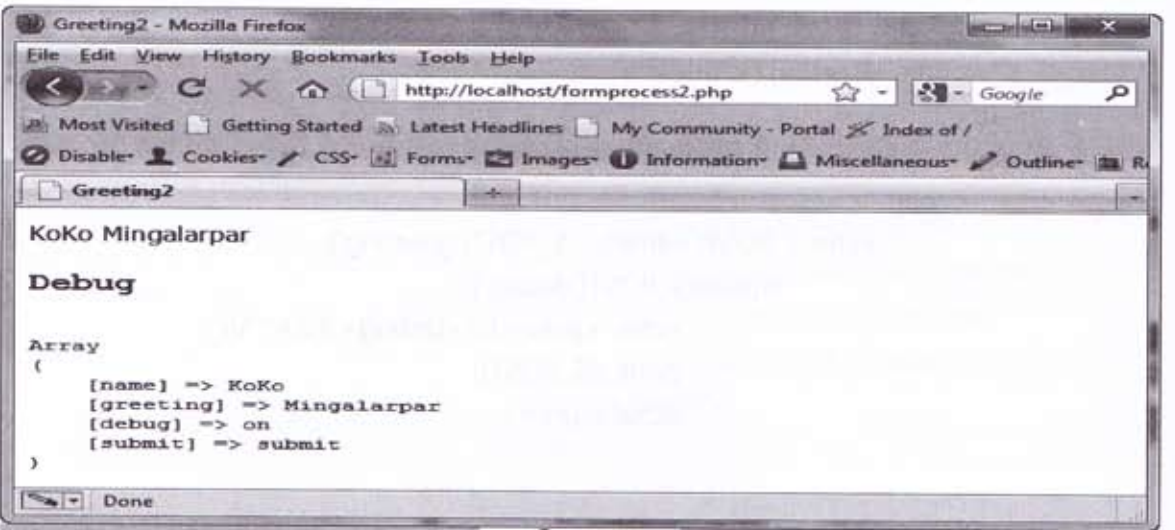
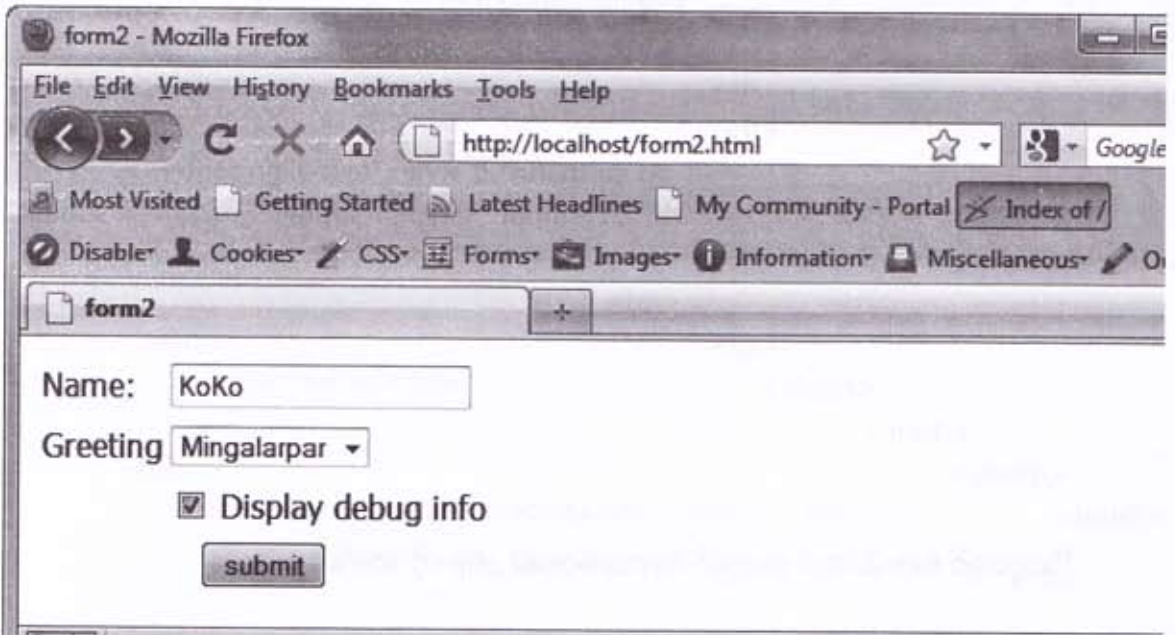
```



```
</body>
```

```
</html>
```

ခုထိ နားလည်ဦးမယ်ထင်ပါတယ်။ HTML တတ်ထားသူတွေ ဆိုတော့ သိပ် အခက်အခဲမဖြစ်နိုင်ပါဘူး။



Form တွေဟာဆိုရင် user တွေက ပို့လိုက်တဲ့ data တွေကို ကြိုတင်သတ်မှတ်ထားတဲ့ နည်းလမ်းတွေပေါ်မှာ အခြေခံပြီးတော့ system ဆီကိုပို့ပါတယ်။ သင်ဟာ form တစ်ခုထဲမှာ action တစ်ခုထက်ပိုတဲ့ ကြိုတင်သတ်မှတ်ချက်တွေကို တစ်ခြား submit နည်းလမ်းတွေသုံးပြီး လုပ်လို့ရပါတယ်။

Radio buttons , Multiline list box

ခု ဒီ သင်ခန်းစမှာ radio button , Multiline list boxes တွေသုံးပြီးတော့ movie, actor , director တွေကို search or create လုပ်မှာပါ။

ပထမဆုံး form3.php ကို ရေးပါမယ်။ အောက်ပါအတိုင်းပါ။

```
<html>
<head>
<title> Add/Search Entry </title>
<style type="text/css">
<!--
td {vertical-align: top;}
-->
</style>
</head>
<body>
<form action="formprocess3.php" method="post">
<table>
<tr>
<td> Name </td>
<td> <input type="text" name="name"/> </td>
</tr>
<tr>
<td> Movie Type </td>
<td>
<select name="movie_type">
<option value=""> Select a movie type... </option>
<option value="Action"> Action </option>
<option value="Drama"> Drama </option>
```

```

    <option value="Comedy"> Comedy </option>
    <option value="Sci-Fi"> Sci-Fi </option>
    <option value="War"> War </option>
    <option value="Other"> Other... </option>
  </select>
</td>
</tr> <tr>
<td> Item Type </td>
<td>
    <input type="radio" name="type" value="movie" checked="checked" />
Movie <br/>
    <input type="radio" name="type" value="actor"/> Actor <br/>
    <input type="radio" name="type" value="director"/> Director <br/>
  </td>
</tr> <tr>
<td></td>
<td> <input type="checkbox" name="debug" checked="checked" />
Display Debug info
</td>
</tr> <tr>
<td colspan="2" style="text-align: center;">
  <input type="submit" name="submit" value="Search" />
  <input type="submit" name="submit" value="Add" />
</td>
</tr>
</table>
</form>
</body>
</html>

```



formprocess3.php ကိုဆက်ပြီးရေးပါမယ်။

```

<?php
    if ($_POST['type'] == 'movie' && $_POST['movie_type'] == "") {
        header('Location: form3.php');
    }
?>
<html>
<head>
<title> <?php echo $_POST['submit'] . ' ' . $_POST['type'] . ' : ' . $_POST['name'];
?> </title>
</head>
<body>
<?php
    if (isset($_POST['debug'])) {
        echo ' <pre> ';
        print_r($_POST);
        echo ' </pre> ';
    }

    $name = ucfirst($_POST['name']);
    if ($_POST['type'] == 'movie')
    {
        $foo = $_POST['movie_type'] . ' ' . $_POST['type'];
    } else {
        $foo = $_POST['type'];
    }

    echo ' <p> You are ' . $_POST['submit'] . 'ing ' ;
    echo ($_POST['submit'] == 'Search') ? 'for ' : " ";
    echo 'a ' . $foo . ' named ' . $name . ' </p> ' ;
?>
</body>
</html>

```

OK... အခု browser ကနေ form3.php ကို ဖွင့်လိုက်ပါ။ အောက်ပါပုံအတိုင်းမြင်ရပါမယ်။

The screenshot shows a web browser window with the address bar displaying `http://localhost/form3.php`. The browser's address bar includes navigation buttons (back, forward, refresh, home) and a search icon. Below the address bar, there are several tabs: "Most Visited", "Getting Started", "Latest Headlines", "My Community - Portal", and "Index of /". A toolbar contains icons for "Disable\*", "Cookies\*", "CSS\*", "Forms\*", "Images\*", "Information\*", and "Miscellaneous\*". A search bar labeled "Add/Search Entry" is present. The main content area contains a form with the following fields and options:

- Name:** A text input field containing "ThaGyanMoe".
- Movie Type:** A dropdown menu currently showing "Drama".
- Item Type:** Three radio button options: "Movie" (selected), "Actor", and "Director".
- Display Debug info:** A checked checkbox.
- Buttons:** "Search" and "Add" buttons.

The screenshot shows a web browser window with the address bar displaying `http://localhost/formprocess3.php`. The browser's address bar includes navigation buttons (back, forward, refresh, home) and a search icon. Below the address bar, there are several tabs: "Most Visited", "Getting Started", "Latest Headlines", "My Community - Portal", and "Index of /". A toolbar contains icons for "Disable\*", "Cookies\*", "CSS\*", "Forms\*", "Images\*", "Information\*", and "Miscellaneous\*". A search bar labeled "Add movie: ThaGyanMoe" is present. The main content area displays the output of a PHP script:

```
Array
(
    [name] => ThaGyanMoe
    [movie_type] => Drama
    [type] => movie
    [debug] => on
    [submit] => Add
)
```

Below the array output, the text "You are Adding a Drama movie named ThaGyanMoe" is displayed.

### Multiple submit buttons

အောက်မှာပြထားတဲ့ button တွေကို ကြည့်ကြည့်ပါ။ same name ဖြစ်နေပါတယ်။ ဒါပေမယ့် different values တွေပါ။ အဲဒီ button တွေထဲက တစ်ခုခုကို ကလစ်လုပ်လိုက်ရင် submit form ဖြစ်သွားမှာပါ။ Debug block မှာ ကြည့်လို့ရပါတယ်။ button တွေဟာ own information/value တွေကိုဘဲပို့ပေးပါတယ်။ script က ဒီ value တွေကို ဖတ်ပြီးတော့ သင့်အား form တစ်ခုထဲမှာဘဲ တစ်ခြား process တွေကိုပါ ထည့်သွင်းနိုင်စေမှာဖြစ်ပါတယ်။

```
<input type="submit" name="submit" value="Search" />
<input type="submit" name="submit" value="Add" />
```

### Basic input testing

အောက်မှာပြထားတဲ့ code တွေဟာဆိုရင် item type ကို movie ဟုတ်မဟုတ်စစ်ဆေးပါတယ်။ ပြီးတော့ တစ်ခါတည်း user က movie type ကို ရွေးချယ်ထားမထားဆိုတာ check လုပ်ပါတယ်။ တကယ်လို့ မရွေးထားခဲ့ဘူးဆိုရင် form pagae ကို ပြန်သွားအောင် header() function နဲ့ redirect လုပ်ထားတာပါ။

```
if ($_POST['type'] == 'movie' && $_POST['movie_type'] == "") {
    header('Location: form3.php');
}
```

\$\_POST['movie\_type'] == " မှာ " က double quote မဟုတ်ဘူးနော်။ single quote နှစ်ခုပါ။ PHP will automatically send some HTTP headers before it sends any output to the browser. This is important to keep in mind, because once sent, HTTP headers cannot be sent again. If you use the header() function after output has already been sent (either by an echo or print statement, HTML, whitespace, etc.), then you will receive a warning.



Ternary operator

Ternary operator ဆိုတာ php ရဲ့ specific operator မဟုတ်ပါဘူး။ language တော်တော်များများမှာလည်း ပါပါတယ်။ သူ့ကို ဘယ်လိုနေရာတွေမှာသုံးလဲဆိုရင် အဓိကအားဖြင့် True / False စစ်တဲ့နေရာမှာသုံးတာများပါတယ်။ သူ့ကိုသုံးခြင်းအားဖြင့် ရရှိနိုင်တဲ့အကျိုးကျေးဇူးတွေက clean code/short ဖြစ်တယ်။ user validating တွေကိုစစ်တဲ့ နေရာမှာအများဆုံးအသုံးချနိုင်ပါတယ်။ if - else အစားထိုးသုံးတာပေါ့။ သူ့ရဲ့ပုံစံလေးက အောက်ပါအတိုင်းဖြစ်ပါတယ်။

[expression] ? [execute if true] : [execute if false];  
နမူနာလေးတစ်ခုနဲ့ပြပါမယ်။ formprocess3.php ကဟာကိုဘဲယူပြလိုက်မယ်။

```
echo $_POST['submit'] == 'Search' ? 'for ' : '';
```

ဒါကိုရှင်းပြရရင် user ဟာ search ကို ကလစ် ခဲ့မယ်ဆိုရင် for ဆိုတာကို ထုတ်ပေးမှာဖြစ်ပြီး Add ကို ကလစ်ခဲ့တယ်ဆိုရင် empty string ကိုဘဲထုတ်ပေးမှာပါ။ ဒါကို if - else နဲ့ရေးမယ်ဆိုရင်

```
if ($_POST['submit'] == 'Search'){
    echo 'for ';
} else{
    echo '';
}
```

if - else နဲ့သုံးမလား ternary operator နဲ့သုံးမလားဆိုတာတော့ကိုယ့်ဟာကို ဆုံးဖြတ်ဖို့လိုပါတယ်။ နေရာတိုင်းမှာ if -else အစား ternary operator ကို သုံးလို့ မသင့်တော်ပါဘူး။ တစ်ခါကလစ်လေမှာ maintain လုပ်ရတာ ခက်ပါတယ်။ အောက်က ဥပမာတစ်ခုကို ထပ်လေ့လာပါဦး။

```

$num = 0;
if ($num < 0) {
    $value = 'The value is negative.';
    echo $value;
} else {
    if ($num > 0) {
        $value = 'The value is positive.';
        echo $value;
    } else {
        $value = 'The value is zero.';
        echo $value;
    }
}
?>

```

ဒါက ternary နဲ့ရေးတာ

```

<?php
    $num = 42;
    echo 'The value is ';
    echo ($num < 0) ? 'negative.' : ($num > 0) ? 'positive.' : 'zero.';
?>

```

### Linking forms together

ခုဆိုရင်သင်ဟာ form တွေကို တော်တော်အသုံးပြုတတ်နေပြီလို့ယူဆလို့ရပါတယ်။ form တွေကို တစ်ခုခြင်းစီ သုံးပြီးတဲ့နောက်မှာ အခုတော့ တစ်ခုနဲ့တစ်ခု ချိတ်ဆက်အသုံးပြုပါမယ်။ ခု exercise မှာတော့ database ထဲကို new item add လုပ်ခြင်းနဲ့ ရှိပြီးသားထဲကဟာကို Search လုပ်ခြင်းတွေကို ပြုလုပ်သွားမှာဖြစ်ပါတယ်။ ပထမဆုံး အနေနဲ့တော့ database နဲ့ interaction လုပ်တာကို မလုပ်ဘဲ form တွေကိုဘဲ create လုပ်ပြီး link လုပ်ကာ browser ပေါ်မှာ ပြနိုင်အောင် အရင်ဆုံး ပြုလုပ်သွားကြမှာဖြစ်ပါတယ်။

ပထမဆုံး form4.php ကို ရေးပါမယ်။



```

<html>
  <head>
    <title>Multipurpose form</title>
    <style type="text/css">
      <!--
      td{
        vertical-align:top;
      }
      -->
    </style>
  </head>
  <body>
    <form action="form4a.php" method="post">
      <table>
        <tr>
          <td>Name</td>
          <td><input type="text" name="name" /></td>
        </tr>
        <tr>
          <td>Item Type</td>
          <td>
            <input type="radio" name="type"
            value="movie" checked="checked" />Movie<br />
            <input type="radio" name="type"
            value="actor" />Actor<br />
            <input type="radio" name="type"
            value="director" />Director<br />
          </td>
        </tr>
        <tr>
          <td>Movie type</td>

```



```

        <td>
            <select name="movie_type" >
                <option value="">Select a movie
type...</option>
                <option value="Action">Action</
option>
                <option value="Drama">Drama</
option>
                <option
value="Comedy">Comedy</option>
                <option value="Sci-Fi">Sci-Fi</op-
tion>
                <option value="War">War</op-
tion>
                <option value="other">Other</op-
tion>
            </select>
        </td>
    </tr>
    <tr>
        <td>
            <input type="checkbox" name="debug"
checked="checked" />Display debug info
        </td>
    </tr>
    <tr>
        <td colspan="2" style="text-align:center;">
            <input type="submit" name="submit"
value="Search" />
            <input type="submit" name="submit"
value="Add" />

```

```

        </td>
      </tr>
    </table>
  </form>
</body>
</html>

```

ဒုတိယအနေနဲ့ form4a.php ကိုရေးမှာဖြစ်ပါတယ်။

```

<?php
// Make sure the user selected a movie type if they are adding a movie.
// If not, send them back to the first form
if($_POST['submit'] == 'Add'){
    if($_POST['type'] == 'movie' && $_POST['movie_type'] == ""){
        header('location: form4.php');
    }
}
?>
<html>
  <head>
    <title>Multipurpose form</title>
    <style type="text/css">
      <!--
      td{
        vertical-align:center;
      }
      -->
    </style>
  </head>
  <body>
    <?php
      // Show a form to collect more information if the user is adding
something

```

```

if($_POST['submit']=='Add'){
    echo '<h2> Add '.ucfirst($_POST['type']).'</h2>';
}

?>
<form action="form4b.php" method="post">
    <input type="hidden" name="type" value="<?php echo $_
POST['type']; ?>" />
    <table>
        <tr>
            <td>Name</td>
            <td><?php echo $_POST['name']; ?>
                <input type="hidden" name="name"
value="<?php echo $_POST['name']; ?>" />
            </td>
        </tr>
        <?php
        if($_POST['type'] == 'movie'){
            ?>
            <tr>
                <td>
                    Movie Type
                </td>
                <td>
                    <?php echo $_POST['movie_type']; ?>
                    <input type="hidden" name="movie_type"
value="<?php echo $_POST['movie_type']; ?>" />
                </td>
            </tr>

```



```

        <tr>
            <td>Year</td>
            <td>
                <input type="text" name="year" />
            </td>
        </tr>
        <tr>
            <td>Movie description</td>
            <?php
            }else{
                echo '<tr> <td>Biography</td>';
            }
            ?>
            <td> <textarea          name="extra"          rows="5"
cols="60"> </textarea> </td>
        </tr>
        <tr>
            <td colspan="2" style="text-align:center;">
                <?php
                    if(isset($_POST['debug'])){
                        echo '<input type="hidden"
name="debug" value="on" />';
                    }
                ?>
                <input  type="submit"  value="Submit"
value="Add" />
            </td>
        </tr>
    </table>
</form>

```

```

<?php
    // If user is just searching for something
    )else if($_POST['submit'] == 'Search'){
        echo '<h1> Search for '.ucfirst($_POST['type']).'</h1>';
        echo '<p> Searching for '.$_POST['name'].'...</p>';
    }
    if(isset($_POST['debug'])){
        echo '<h2>Debug:</h2><pre>';
        print_r($_POST);
        echo '</pre>';
    }
?>
</body>
</html>

```

တတိယအဆင့်အနေနဲ့ form4b.php ကို ရေးပါမယ်။

```

<html>
    <head>
        <title>From 4b</title>
    </head>
    <body>
        <?php
            if($_POST['type'] == 'movie'){
                echo '<h1>NEW'. ucfirst($_POST['movie_type']).' : :';
            }else{
                echo '<h1>NEW'. ucfirst($_POST['type']).' : :';
            }
            echo $_POST['name'].'</h1>';
            echo '<table>';
            if($_POST['type'] == 'movie'){
                echo '<tr>';
                echo '<td>Year: </td>';
                echo '<td>'.$_POST['year'].'</td>';
                echo '</tr><tr>';
                echo '<td>Movie description: </td>';
            }
        }
    </body>
</html>

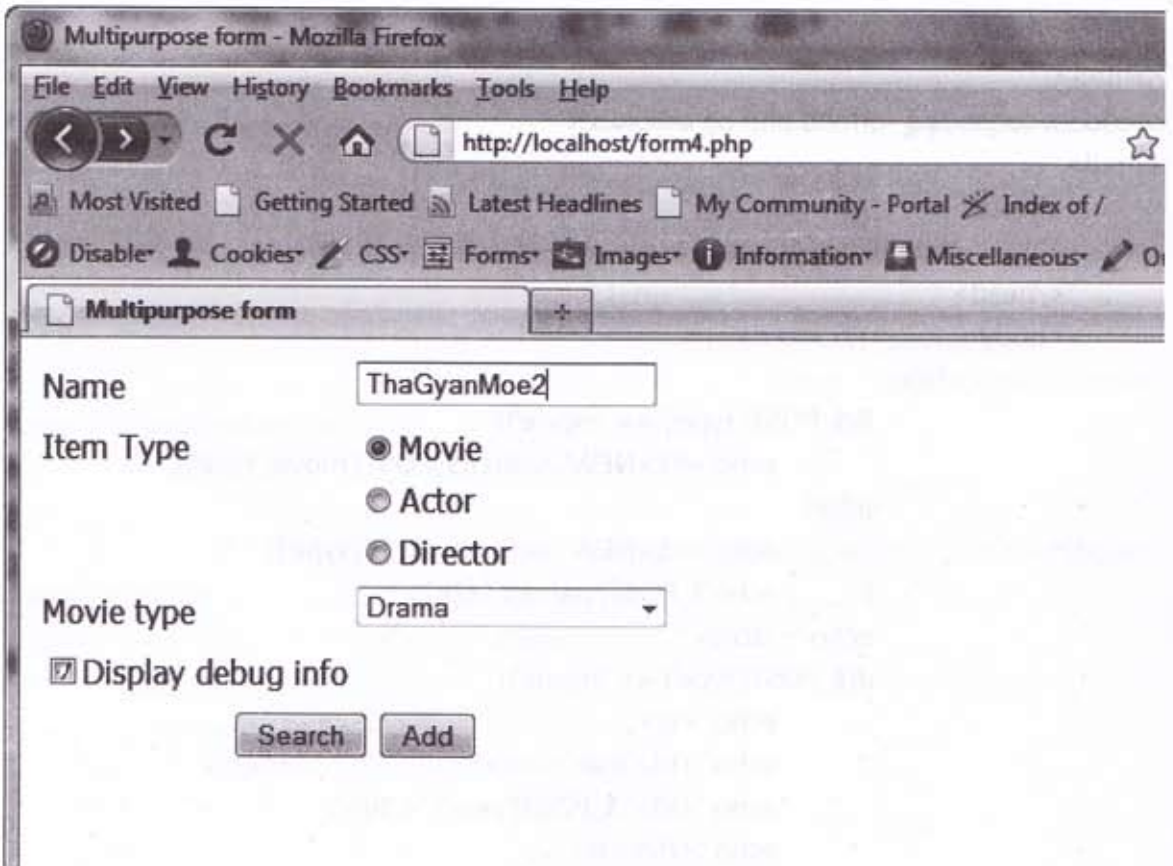
```

```

    }else{
        echo '<tr><td>Biography: </td>';
    }
    echo '<td>' .$_POST['extra'].'</td>';
    echo '</tr>';
    echo '</table>';
?>
</body>
</html>

```

Browser ကနေ form4.php ကိုဖွင့်လိုက်ပါ။ အောက်ပါအတိုင်းမြင်ရပါမယ်။





Browser: <http://localhost/form4a.php>

Navigation: Most Visited, Getting Started, Latest Headlines, My Community - Portal, Index of /

Tools: Disable, Cookies, CSS, Forms, Images, Information, Miscellaneous

Tab: Multipurpose form

## Add Movie

Name: ThaGyanMoe2

Movie Type: Drama

Year:

Movie description:

Browser: <http://localhost/form4b.php> ☆ Google

Navigation: Most Visited, Getting Started, Latest Headlines, My Community - Portal, Index of /

Tools: Disable, Cookies, CSS, Forms, Images, Information, Miscellaneous

Tab: From 4b

## NEWDrama : ThaGyanMoe2

Year: 2001

Movie description: very good movie. I like very much.



## Exercises

## For chapter (5)

1. Create a form and a processing page that let you choose a rating (stars, thumbs up, number from 1 to 5, whatever), and provide comments for a movie.
2. Create a form with several text input boxes that allow you to populate the options of a select field on a subsequent page.
3. Create a calculator form that takes two numbers and calculates their sum.

## Chapter (6) Letting the user Edit the Database

database တွေဟာ သူတို့ထဲမှာရှိတဲ့ data တွေကို self generate မလုပ်နိုင်ပါဘူး။ system တစ်ခုရဲ့ အကူညီကိုတော့ အနည်းနဲ့အများလိုအပ်ပါတယ်။ integrated system လိုဟာမျိုးပေါ့။ ခုဆိုရင် သင်တာ basic SQL syntax တွေကို သိရှိနားလည်ပြီးခဲ့တဲ့နောက်မှာ Table ထဲက data တွေကို user တွေကို မြင်နိုင်အောင် ထုတ်ပြနိုင်ခဲ့ပြီး ဖြစ်ပါတယ်။ ခုတစ်ဆင့်တက်ပြီးတော့ user တွေကို Edit , delete, update တွေလုပ်လို့ရအောင် ပြုလုပ်ပေးကြပါမယ်။ ဒီသင်ခန်းစာမှာ အောက်ပါအကြောင်းအရာမ ပြားပါဝင်ပါတယ် -

- Adding entries
- Deleting entries
- Modifying entries

ကျွန်တော်တို့က ခုလက်ရှိသုံးနေတဲ့ moviesite1 database ကိုဘဲ ဆက်သုံးသွားမှာဖြစ်ပါတယ်။ ပထမဆုံး database ကို ပြင်ဖို့အတွက် logic ကျကျစဉ်းစားကြည့်မယ်ဆိုရင် admin ကသာ ပြင်သင့်တယ်မဟုတ်လား။ အဲဒီတော့ ကျွန်တော်တို့က admin page တစ်ခုကို create လုပ်ရမယ်။ database မှာ အဓိက ဘာတွေပါလဲလို့ကြည့်ရင် movie and people ဘဲပါတယ်လေ။ ဟုတ်ဘူးလား။ အဲဒီတော့ movie ဆိုတဲ့ table ရယ် people table တွေကို add or delete လုပ်လို့ရအောင် ရေးမယ်။

### Creating admin page

admin.php ကို အောက်ပါအတိုင်းလေးရေးပါမယ်။

```
<?php
$db = mysql_connect('localhost','root','') or die ('Unable to connect. Check your
connection parameters. ');
mysql_select_db('moviesite1',$db) or die (mysql_error($db));
?>
```



```

<html>
  <head>
    <title>Movie database</title>
    <style type="text/css">
      th {
        background-color: #999;
      }
      .odd_row{
        background-color: #EEE;
      }
      .even_row{
        background-color: #FFF;
      }
    </style>
  </head>
  <body>
    <table style="width:100%;">
      <tr>
        <th colspan="2">
          Movie <a href="movie.php?action=add">Add</a>
        </th>
      </tr>
      <?php
        $query = 'SELECT * FROM movie';
        $result = mysql_query($query,$db);
        $odd = true;
        while ( $row = mysql_fetch_assoc($result)){
          echo ($odd == true )? '<tr class="odd_row">' : '<tr
class="even_row">';
          $odd = !$odd;
          echo '<td style="width:25%">';

```

```

        echo $row['movie_name'];
        echo '</td><td>';
        echo '<a href="movie.php?action=edit&id='.$row['
movie_id'].'">[Edit]</a>';
        echo '<a href="delete.
php?type=movie&id='.$row['movie_id'].'">[Delete]</a>';
        echo '</td></tr>';
    }
?>
<tr>
<th colspan="2">People<a href="people.php?action=add">
Add</a></th>
</tr>
<?php
    $query = 'SELECT * FROM people';
    $result = mysql_query($query,$db);
    $odd = true;
    while ( $row = mysql_fetch_assoc($result)){
        echo ($odd ==true) ? '<tr class="odd_row">' : '<tr
class="even_row">';
        $odd = !$odd;
        echo '<td style="width:25%">';
        echo $row['people_fullname'];
        echo '</td><td>';
        echo '<a href="people.php?action=edit&id='.$row['
people_id'].'">[Edit]</a>';
        echo '<a href="delete.php?type=people&id='.$row[
'
        echo '</td></tr>';
    }

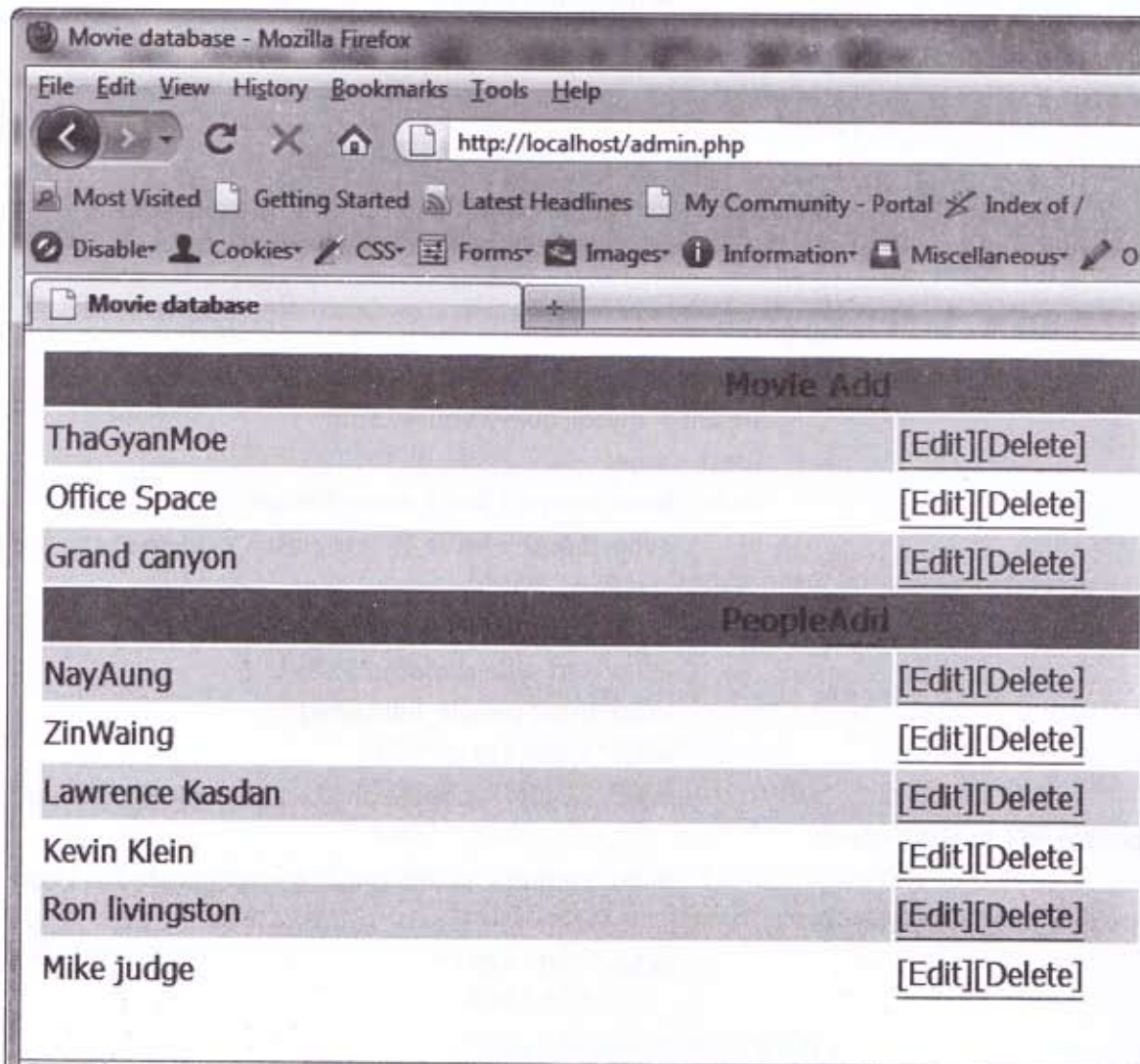
```

```

?>
</table>
<body>
</html>

```

browser ကနေ admin.php ကို ကြည့်ရင်အောက်ပါအတိုင်းတွေ့ရမှာပါ။





admin.php ထဲကဟာတွေကို ခုချိန်မှာတော့ပြင်လို့မရသေးဘူးနော်။ ဘာလို့ဆိုတော့ ကျွန်တော်တို့တွေ create မလုပ်ရသေးဘူး။

How it works

Odd and even rows တွေကို different color နဲ့ ဖော်ပြပေးထားတာပါ။ while loop တွေ မစခင်မှာ \$odd = true လို့ပေးခဲ့ပါတယ်။ while loop ထဲမှာတော့ == သုံးပြီး ရေးထားတာကို သတိပြုပါ။ = က တန်ဖိုး တစ်ခု သတ်မှတ်တာဖြစ်ပြီး == ကတော့ compare လုပ်တာဖြစ်ပါတယ်။ အစဦးဆုံးကို true လုပ်လိုက်တော့ tr တန်ဖိုးက ternary operator အရ true ဖြစ်သွားတယ်လေ။ ဒီတော့ tr = odd\_row ဖြစ်သွားတာပေါ့။ odd\_row ဆိုတော့ သူက background-color: #EEE; လို့ သတ်မှတ်ခဲ့တဲ့အတွက် မီးခိုးရောင်လေး ဖြစ်သွားမှာပါ။ အဲဒီကနေ တစ်ဆင့်ထပ်ဆင်းတော့ \$odd != \$odd ဖြစ်သွားတယ်။ ဆိုလိုတာက false ပြန်ဖြစ်ပြီပေါ့။ အဲဒီ info လေးကို နောက်ထပ်ပတ်မယ့် Loop မှာ ထည့်ပေးလိုက်တာလေ။ ဒီတော့ false ဖြစ်သွားတော့ FFF အရောင်(အဖြူ)နဲ့ ပြပေးတာပါ။

Inserting a Record in a Relational Database

Database တွေမှာ များသောအားဖြင့် table တစ်ခုထက်ပိုပြီး ပါလေ့ရှိပါတယ်။ အဲဒီ table တွေအားလုံးဟာ တစ်ခုနဲ့တစ်ခု လုံးဝ independent ဖြစ်နေလို့လည်းဖြစ်ပါတယ်။ ဒါပေမယ့် အဲဒါက ဘာနဲ့တူလဲဆိုတော့ သင်္ဃာတန်း ကားကို ဝယ်ထားပြီး မမောင်းဘဲ ဂိုထောင်ထဲ သိမ်းထားတာနဲ့တူပါတယ်။ သာမန်အားဖြင့် database ထဲက table တွေဟာ တစ်ခုခုနဲ့ ဆက်စပ်နေလေ့ရှိပါတယ်။

ဟိုးအရင်တစ်ချိန်တုန်းကတော့ အခုသုံးနေတဲ့ relational database လို့ခေါ်တဲ့ အမျိုးအစားမျိုးမရှိခဲ့ပါဘူး။ database ထဲက row တိုင်းဟာ information အားလုံးကို ထိန်းသိမ်းရပါတယ်။ ဥပမာ ပေါ့.. ခု movie database ထဲက movie table ဆိုပါစို့။ အဲဒီ table ထဲမှာ movie တွေရော actor , director, movietype, အစရှိသဖြင့် အကုန် ထည့်ထားရမှာပါ။ တကယ်လို့ သင်္ဃာတန်း Movie category တစ်ခုခုကို ပြင်ချင်တယ်ဆိုပါစို့။ သင့်အနေနဲ့ row တိုင်းကို လိုက်ပြင်ဖို့လိုပါတယ်။ ဒါ ရှေးရှေးတုန်းကဟာ။ ခုခေတ် relational database management system (RDBMS) မှာတော့ ဒီလို ပြင်ဖို့ မလိုတော့ပါဘူး။ movietype table တစ်ခု လုပ်လိုက်မယ်။ ပြီးရင် တစ်ခုခုနဲ့ link လုပ်လိုက်မယ်။ movietype\_id လိုဟာမျိုးပေါ့။ movietype\_id = 5 က drama ဆိုပါတော့။

တကယ်လို့ အဲဒီ drama ကို action လို့ ပြောင်းချင်တယ်ဆိုရင် movietype\_id = 5 ကို action လို့ပြောင်းလိုက်ရင် ကျန်တဲ့ movie တွေအကုန်လုံး ( movietype\_id =5 ဖြစ်တဲ့ဟာတွေ) action ဖြစ်သွားမှာပါ။ ဒါကို relational database လို့ခေါ်ပါတယ်။ table တစ်ခုနဲ့တစ်ခု link လုပ်ဖို့ကို primary key / foreign key pair ဆိုတာနဲ့ လုပ်ရပါတယ်။

primary key ဆိုတာက set value or value ဖြစ်ပြီးတော့ သူ့ကို Table ထဲက တစ်ခုထဲရှိတဲ့ ဟာမျိုးကို သတ်မှတ်ပါတယ်။ ဥပမာ movie id ဆိုပါစို့။ သင့် database ထဲမှာ movie တွေအများကြီးရှိပါမယ်။ အဲဒီထဲမှာ နာမည်တူတွေလည်း ရှိနိုင်တယ်လေ။ ဒါကြောင့် movie name ကို primary သတ်မှတ်ရင် မတိကျတာတွေရလာနိုင်တယ်။ ဒါပေမယ့် movie id ဆိုတာကတော့ တစ်ခုတည်းရှိတာ။ 1, 2, 3, 4 and so on သွားတာလေ။ movie နံပါတ် ၁ လို့ ညွှန်းလိုက်တာနဲ့ အဲဒီ movie နံပါတ် ၁ ကိုဘဲ တန်းသွားတာ။ အဲဒီတော့ primary key တွေကို ခုလိုမျိုး uniquely ရှိတဲ့ တန်ဖိုးတွေကိုဘဲ သတ်မှတ်သင့်ပါတယ်။ foreign key ဆိုတာကတော့ တစ်ခြားtable ထဲမှာ ခုနက primary key ကို ပြန်ပြီး reference လုပ်တာပါ။

အောက်က exercise ကတော့ PHP and SQL ကို သုံးပြီးတော့ movie database ထဲကို insert လုပ်မှာပါ။ သွင်းမယ့် movie type က known file type ဖြစ်ပါတယ်။ movietype table ထဲမှာ ရှိနေတဲ့ဟာပေါ့။

```
<?php
$db = mysql_connect('localhost','root','') or die ('Unable to connect, check your
database parameters. ');
mysql_select_db('moviesite1',$db) or die(mysql_error($db));
?>
<html>
  <head>
    <title>Add movie</title>
  </head>
  <body>
    <table>
      <tr>
        <td><form method="post" action="commit.php?action=add&type=movie"
        <tr>
        <td>Movie name:<input type="text" name="movie_
name" /> </td> </tr>
    </table>
  </body>
</html>
```

```

<tr><td>Movie type:<select name="movie_type">
  <?php
    $query = 'SELECT movietype_id, movietype_label
              FROM movietype
              ORDER BY movietype_label';
    $result = mysql_query($query,$db);
    // Populate the select options with the result
    while ($row = mysql_fetch_assoc($result)){
      foreach($row as $value){
        echo '<option value="'. $row['movietype_id']. ">';
        echo $row['movietype_label']. '</option>';
      }
    }
  ?>
</select>
<tr>
  <td>
    Movie year: <select name="movie_year">
      <?php
        for($yr=Date("Y"); $yr >=1970; $yr--){
          echo '<option value="'. $yr. ">'. $yr. '</option>';
        }
      ?>
    </select>
  </td>
</tr>
<tr><td>Movie actor:<select name="movie_leadactor">
  <?php

```



```

$query = 'SELECT people_id, people_fullname
FROM people
WHERE people_isactor=1
ORDER BY people_fullname';
$result = mysql_query($query,$db) or
die(mysql_error($db));

// Populate
while ($row = mysql_fetch_assoc($result)){
  foreach ($row as $value){
    echo '<option value="'. $row['people_id'] .'">';
    echo $row['people_fullname'].'</option>';

      }
    }

?>
</select>
<tr><td>Movie director:<select name="movie_di-
rector">

<?php
$query = 'SELECT people_id, people_fullname
FROM people
WHERE people_isdirector=1
ORDER BY people_fullname';

```

```
die(mysql_error($db));
```

```
$result = mysql_query($query,$db) or
```

```
// Populate
```

```
while ($row = mysql_fetch_assoc($result)){
```

```
    foreach($row as $value){
```

```
        echo '<option value="".'.$row['people_id'].''>';
```

```
        echo $row['people_fullname'].'</option>';
```

```
    }
```

```
}
```

```
?>
```

```
</select>
```

```
</td></tr>
```

```
<tr colspan="2">
```

```
    <td><input type="submit" name="submit"
```

```
value="Add" /></td>
```

```
</tr>
```

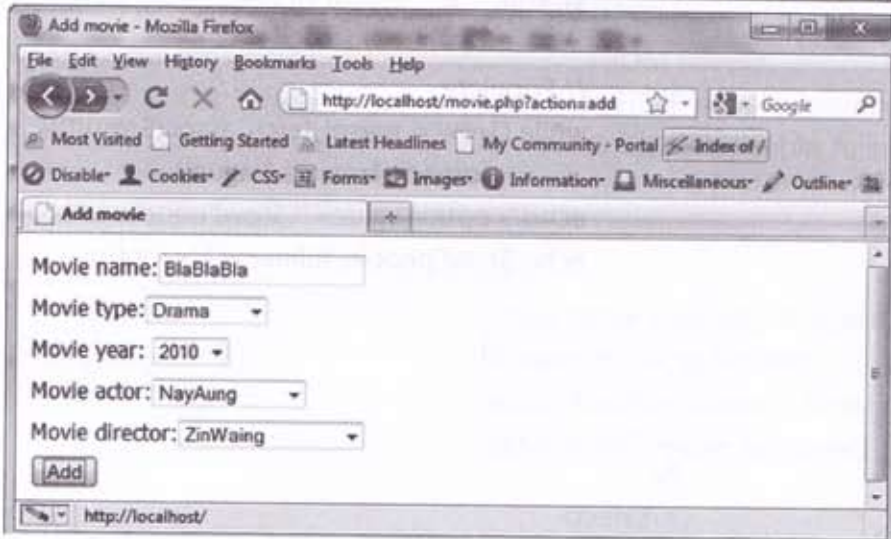
```
</form>
```

```
</table>
```

```
</body>
```

```
</html>
```

ဒါကို movie.php လို့ နာမည်ပေးပြီးတော့ browser ကနေ ကြည့်ကြည့်ပါ။



ဒီလိုပုံစံလေးမြင်ရမှာပါ။ ခုချိန်မှာ Add လုပ်လိုက်ရင် commit.php မရှိသေးတဲ့အတွက် လုပ်လို့ရမှာမဟုတ်ပါဘူး။ ခု commit.php file လေးကို ထပ်ရေးပါမယ်။

```
<?php
    $db = mysql_connect('localhost','root','') or die ('Unable to connect, check your
database parameters.');
```

```
    mysql_select_db('moviesite1',$db) or die(mysql_error($db));
?>
```

```
<html>
    <head>
        <title> Add Movie database</title>
    </head>
    <body>
        <?php
            switch($_GET['action']){
                case 'add':
                    switch ($_GET['type']){
                        case 'movie':
```

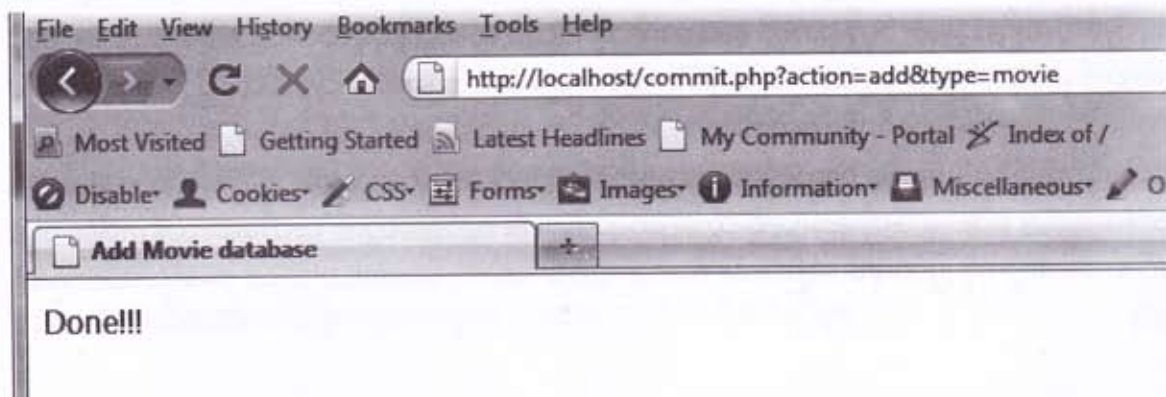


```

$query = 'INSERT INTO movie (movie_name, movie_
year, movie_type, movie_leadactor, movie_director) VALUES (
    "$_POST['movie_name'].",
    '$_POST['movie_year'].',
    '$_POST['movie_type'].',
    '$_POST['movie_leadactor'].',
    '$_POST['movie_director].'
);
break;
}
break;
}
if (isset($query)){
    $result= mysql_query($query, $db) or die (mysql_error($db));
}
?>
<p>Done!!!</p>
</body>
</html>

```

Add လုပ်လိုက်ရင် အောက်ပါအတိုင်းပေါ်လာမှာပါ။ data တွေ insert လုပ်တာ ပြီးသွားကြောင်းပေါ်လာမှာပါ။



```

ရှေ့သင်ခန်းစာမှာတုန်းက for loop ကို ရှင်းပြခဲ့တယ်နော်။ ခု year ကို တွက်ထုတ်တဲ့နေရာမှာလည်း
for loop ကို သုံးပါတယ်။ သူ့ရဲ့ ပုံစံက ဒီလိုလေးလေး
for (initialize; condition; increment) {
    // code
}

```

ခုက initialize အနေနဲ့ Current date ကို Date function သုံးပြီး ခေါ်လိုက်တယ်။ နောက်ပြီးတော့ condition အနေနဲ့ 1970 ကို သတ်မှတ်လိုက်တယ်။ increment ကိုတော့ decrement လုပ်လိုက်တယ်။ တစ်နှစ်စီလျော့သွားမှာပေါ့။

ကျန်တဲ့ movie actor and movie director တို့ကိုလည်း movie type ရှာတုန်းက အတိုင်းဘဲလုပ်လိုက်ပါတယ်။

ခုဆိုရင် သင်ဟာ Add လုပ်ဖို့ အဆင်သင့်ဖြစ်နေပါပြီ။ Add လုပ်လိုက်တာနဲ့ action မှာ ပြထားတဲ့အတိုင်း commit.php ကို ကူးပြောင်းသွားပြီးတော့ အဲဒီထဲက ခိုင်းစေထားတဲ့အတိုင်း ဆက် run ပါတယ်။ commit.php မှာ ဘာရေးထားတာလဲဆိုတော့ movie.php မှာ ရေးထားခဲ့တဲ့ <form action="commit.php?action=add&type=movie" code တွေကို GET method နဲ့ လှမ်းယူပါတယ်။ switch statement နဲ့ ယူတာပါ။ ခုက add တစ်ခုထဲဖြစ်တာဖြစ်တဲ့အတွက် switch statement က သိပ်မလိုဘူးလို့ယူဆလိုရပါတယ်။ ဘာဖြစ်လို့လဲဆိုတော့ action=add တစ်ခုထဲဖြစ်တဲ့အတွက်ဖြစ်ပါတယ်။ တကယ်လို့ delete တွေ edit တွေပါလာမယ်ဆိုရင် action=edit, action=delete , type=movie ကလည်း type=actor, type=director စသဖြင့်တွေဖြစ်လာရင် switch statement ရဲ့ အသုံးဝင်ပုံကို ပိုသိလာမှာပါ။ ခုကတော့ Switch ကနေ \$\_GET ['action'] ကိုယူတယ်ပြီးရင် \$\_GET['type'] ကို ယူတယ်။ action ကို ယူပြီးတာနဲ့ သူ့ရဲ့ case ကိုကြည့်တယ်။ add လုပ်မှာလား delete လုပ်မှာလားပေါ့။ အဲဒီတော့ case က လောလောဆယ် add ပေါ့။ add လုပ်မှာကိုသိပြီးတဲ့နောက်မှာ ဘာကို add မှာလည်းဆိုတာ ဆက်ကြည့်ဖို့လိုပါတယ်။ ဒီတော့ case add အောက်မှာနောက်ထပ် switch statement ထပ်တည်ဆောက်ပါတယ်။ type ကိုကြည့်ပြီးတော့ add လုပ်လို့ရအောင်ပါ။ ခုသင်ခန်းစာမှာတော့ type က movie တစ်ခုတည်းဖြစ်တဲ့အတွက် case 'movie': ပေါ့။ ဟုတ်ပြီ အဲဒီ movie အောက်မှာမှ သူ့အတွက် query တစ်ခု တည်ဆောက်ပေးလိုက်တယ်။ INSERT query ပေါ့။ ခု ဒါကို နားလည် သွားပြီဆိုရင် delete လုပ်ဖို့အတွက် ဆက်ရေးမယ်။ movie.php ကို code တွေ ကို ပြင်ရေးပေးရဦးမယ်။ အရေးကြီးဆုံးကတော့ commit.php က switch statement ဘဲ။



Deleting record

delete လုပ်တာကတော့ very easy ပါဘဲ။ delete လုပ်မယ်ဆိုရင်တော့ သေချာဖို့လိုပါတယ်။ ဘာလို့လဲဆိုတော့ delete လုပ်ပြီးသွားရင် data ကို ပြန် မရနိုင်တော့ပါဘူး။

DELETE statement ကိုသုံးရတာ SELECT statement ကို သုံးရတာနဲ့ဆင်ပါတယ်။ အောက်ပါအတိုင်းလေးပါဘဲ။

```
SELECT * FROM movie where movie_id = 4;
```

delete statement က အောက်ပါအတိုင်းဖြစ်ပါတယ်။

DELETE \* FROM movie where movie\_id = 4; ဖြစ်ပါတယ်။ ဒီမှာအရေးကြီးတာက where clause ကို သေချာအောင် ရွေးပေးဖို့လိုပါတယ် ။ MySQL database က relational database type ဆိုတာ သိထားခဲ့ပြီးပြီနော်။

ဒါကြောင့် Table တော်တော်များများဟာ တစ်ခုနဲ့တစ်ခု ဆက်စပ်နေလေ့ရှိပါတယ်။ table တစ်ခုထဲကဟာကို delete လုပ်လိုက်ရင် နောက် Table တစ်ခုက အဲဒီ delete လုပ်လိုက်တာကို reference လုပ်ဖို့ လိုတယ်ဆိုရင် သင်ဟာ တောတော်များများကို လိုက်ပြင်ရမှာဖြစ်ပါတယ်။ MySQL မှာ relation ဖြစ်တဲ့ဟာတွေကို auto delete လုပ်မပေးပါဘူး။ ဒီတော့ သင့်အနေနဲ့ delete လုပ်တော့မယ်ဆိုရင်တော့ သေချာစဉ်းစားပေးပါ။ ဒီလိုမျိုး ပြဿနာကို ကျော်လွှားနိုင်ဖို့အတွက် cascade delete လုပ်ပြီးကျော်လွှားနိုင်ပါတယ်။

ခု သင်ဟာ ဘယ်လို delete လုပ်ရမယ်ဆိုတာ သိပြီနော်။ သိပြီးပြီဆိုရင် people table ထဲက actor တစ်ယောက်ကို delete လုပ်ကြည့်မယ်။ ပထမဆုံး movie table ကို update လုပ်ဖို့လိုပါတယ်။ ဘာလို့လဲဆိုတော့ movie table ထဲမှာ movie\_leadactor ဆိုတာပါတယ်လေ။ သင်က people ထဲက NayAung ကို ဖျက်ယုံဘဲဖျက်လိုက်မယ်ဆိုရင် movie table ထဲက movie\_leadactor ထဲမှာ NayAung ဆိုတဲ့နာမည်ကျန်ခဲ့မယ်လေ။ ဒီတော့ အဲဒီနာမည်နေရာကို တစ်ခုခု အစားထိုးပေးရမယ်။ သုည 0 ပေါ့။ တကယ်လို့ NayAung ဆိုတဲ့နာမည် တစ်ခြား table မှာပါသေးတယ်ဆိုလည်း update လုပ်ပေးဖို့လိုပါတယ်။ ခုက ၂ နေရာမှာဘဲပါလို့ပါ။ တကယ်လို့ movie table ထဲကဟာကို delete လုပ်မယ်ဆိုရင်တော့ တစ်ခြားနေရာတွေမှာ update လုပ်ဖို့ မလိုအပ်ပါဘူး။



ခု delete.phpကို ရေးရအောင်။

```

<?php
    $db = mysql_connect('localhost','root','') or die ('Unable to connect, check your
database parameters. ');
    mysql_select_db('moviesite1',$db) or die(mysql_error($db));

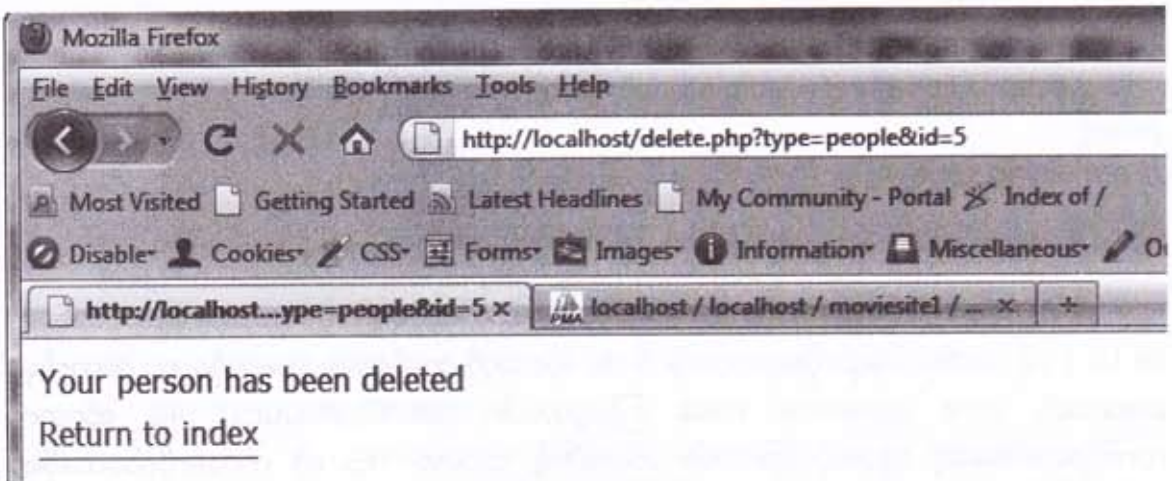
    switch($_GET['type']){
        case('people'):
            $query = 'UPDATE movie SET movie_leadactor = 0
                WHERE movie_leadactor = '.$_
GET['id'];
            $result = mysql_query($query,$db) or die (mysql_error($db));

            $query = 'DELETE FROM people
                WHERE people_id='.$_GET['id'];
            $result = mysql_query($query,$db) or die (mysql_error($db));
            echo 'Your person has been deleted <br />';
            echo '<a href="movie_index.php">Return to index</a>';
            break;
        case('movie'):
            $query = 'DELETE FROM movie
                WHERE movie_id='.$_GET['id'];
            $result = mysql_query($query,$db) or die (mysql_error($db));
            echo ' Your movie has been deleted.<br />';
            echo '<a href="movie_index.php">Return to index</a>';
            break;
    }
?>

```

ခု admin.php ကို browser ကနေ run ပြီးတော့ People အောက်က Ron livingston ကို delete လုပ်ကြည့်ရအောင်။ Delete ကို click လိုက်တာနဲ့ ချက်ခြင်း delete ဖြစ်သွားပြီး အောက်ပါအတိုင်း တွေ့ရမှာပါ။

| People.php      |   |
|-----------------|---|
| NayAung         | <a href="#">[Edit]</a> <a href="#">[Delete]</a> |
| ZinWaing        | <a href="#">[Edit]</a> <a href="#">[Delete]</a> |
| Lawrence Kasdan | <a href="#">[Edit]</a> <a href="#">[Delete]</a> |
| Kevin Klein     | <a href="#">[Edit]</a> <a href="#">[Delete]</a> |
| Ron Livingston  | <a href="#">[Edit]</a> <a href="#">[Delete]</a> |
| Mike judge      | <a href="#">[Edit]</a> <a href="#">[Delete]</a> |



```
အောက်ပါအတိုင်း ရေးလို့ရပါတယ်။
if - else လေးသုံးပြီးရေးလို့ရပါတယ်။
if {
    //code no execute
    //return to index page
}else{
    //process delete query
}
```

ဒီတော့ ဒီလိုလေးရေးကြည့်မယ်။

```

if(!isset($_GET['do']) || ($_GET['do'] != 1)){
    switch ($_GET['type']){
        case ('movie'):
            echo 'Are you sure you want to delete this movie? <br
/>';
            break;
        case('people'):
            echo'Are you sure you want to delete this people?<br
/>';
            break;
    }
    echo '<a href=".'.$_SERVER['REQUEST_URI'].'&do=1">Yes</a>';
    echo ' Or <a href="admin.php">Return to index</a>';
}else{
// remaining code
}

```

ဒါက ဘာကို ပြသလဲဆိုတော့ တစ်ခုခုကို မဖျက်ခင်မှာ confirm တစ်ခု တောင်းတာပါ။  
isset(\$\_GET['do']) လို့ရေးရတာက do လုပ်ရမယ်လို့ မသတ်မှတ်ခဲ့ရင် လို့ရေးတာပါ။ နောက်တစ်ခုက  
do != 1 လို့ သတ်မှတ်ခဲ့ရင်လို့ရေးတာပေါ့။ ဒီ do ဆိုတာကို ဘယ်ကနေ ရမလဲဆိုတော့ အောက်မှာ  
ရေးထားတဲ့ echo statement ကနေ ပြန်ယူတာပါ။ SERVER\_REQUEST\_URI ဆိုတာက  
လက်ရှိရောက်နေတဲ့ နေရာကိုပြောတာပါ။ တကယ်လို့ သင်ဟာ YES ကို ကလစ်လိုက်မယ်ဆိုရင်  
do=1 ဖြစ်သွားပြီ။ ဒီနေရာမှာ စဉ်းစားစရာလေးတစ်ခုကိုပြောပြမယ်နော်။ if statement က  
အလုပ်လုပ်နေပြီဆိုရင်။ ဘာလို့ အောက်က else statement က အလုပ်လုပ်သွားတာလဲလို့။ အမှန်ကတော့  
အခု အခြေအနေမှာ else statement က အလုပ်ကို လုပ်ရမှာ။ ဒါမှ delete ဖြစ်မှာလေ။ if statement  
မှာက သူ့ရဲ့ statement မှာ (!isset(\$\_GET['do']) || (\$\_GET['do'] != 1)) ဒီလိုရေးခဲ့တဲ့အတွက်  
do မလုပ်ရဘူးလို့ သတ်မှတ်ခဲ့တာပေါ့။ ဒါပေမယ့် သူ့ရဲ့ အောက်က YES ကို ကလစ်လိုက်မယ်ဆိုရင်  
do =1 ဖြစ်သွားပြီလေ။ ဒီတော့ အဲဒီ statement က မှားသွားပြီလေ။ ဒီတော့ else statement က  
ဆက်ပြီး အလုပ်လုပ်သွားတာပေါ့။ သေချာနားလည်အောင် ကြည့်ပေးပါ။ တစ်ခါတည်းမရရင် ရတဲ့အထိ  
နားလည်အောင်ကြိုးစားစေချင်ပါတယ်။ Programming ရေးတာ thinking အများကြီးပါပါတယ်။



အဲဒီတော့ သူ့ရဲ့ အစအဆုံး code က အောက်ပါအတိုင်းဖြစ်သွားမှာပေါ့။

```

<?php
    $db = mysql_connect('localhost','root','') or die ('Unable to connect, check
your database parameters.');
```

mysql\_select\_db('moviesite1',\$db) or die(mysql\_error(\$db));

```

    if(!isset($_GET['do']) || ($_GET['do'] != 1)){
        switch ($_GET['type']){
            case ('movie'):
                echo' Are you sure you want to delete this movie? <br />';
                break;
            case('people'):
                echo' Are you sure you want to delete this people?<br />';
                break;
        }
        echo '<a href="'. $_SERVER['REQUEST_URI'].'&do=1">Yes</a>';
        echo ' Or <a href="admin.php">Return to index</a>';
    }else{
        switch($_GET['type']){
            case('people'):
                $query = 'UPDATE movie SET movie_leadactor = 0
                WHERE movie_leadactor ='. $_GET['id'];
                $result = mysql_query($query,$db) or die (mysql_
error($db));

                $query = 'DELETE FROM people
                WHERE people_id='. $_GET['id'];
                $result = mysql_query($query,$db) or die (mysql_
error($db));
```

```

echo 'Your person has been deleted<br />';
echo '<a href="movie_index.php">Return to index</a>';
break;
case('movie'):
    $query = 'DELETE FROM movie
              WHERE movie_id='.$_GET['id'];
    $result = mysql_query($query,$db) or die (mysql_
error($db));

    echo ' Your movie has been deleted.<br />';
    echo '<a href="movie_index.php">Return to index</
a>';

    break;
    }
}
?>

```

ခု people table ထဲမှာ Ron livingston ဆိုတာ မရှိတော့ပါဘူး။ သူ သရုပ်ဆောင်ထားတဲ့ ရုပ်ရှင်ဖြစ်တဲ့ office space မှာလည်း movie\_leadactor နေရာမှာလည်း သူညာ 0 က အစားထိုးသွားပါပြီ။ ခုလောက်ဆို delete လုပ်တာကို အတော်များများ နားလည် သဘောပေါက်လောက်ပြီထင်ပါတယ်။

### Editing Data in a Record

ဒီ exercise မှာတော့ သင့်အနေနဲ့ database ကို ပြင်ဆင်ခွင့်ရှိအောင် ပြုလုပ်သွားမှာဖြစ်ပါတယ်။ ရေးပြီးသား movie.php ကိုဘဲ အနည်းငယ်ပြင်ရေးယုံပါဘဲ။

```

<?php
$db = mysql_connect('localhost','root','') or die ('Unable to connect, check your
database parameters.');
```

```
mysql_select_db('moviesite',$db) or die(mysql_error($db));
```

```

if($_GET['action'] == 'edit'){
    $query = 'SELECT movie_name, movie_type, movie_year, movie_lead-
actor, movie_director
                FROM movie
                WHERE movie_id='.$_GET['id'];
    $result = mysql_query($query,$db);
    extract(mysql_fetch_assoc($result));
}else{
    $movie_name="";
    $movie_type=0;
    $movie_year=date("Y");
    $movie_leadactor=0;
    $movie_director=0;
}
?>
<html>
  <head>
    <title><?php echo ucfirst($_GET['action']); ?> movie</title>
  </head>
  <body>
    <table>
      <form action="commit.php?action=<?php echo $_
GET['action'];?>&type=movie" method="post">
        <tr><td>Movie name:<input type="text" name="movie_
name" value="<?php echo $movie_name; ?>" /></td></tr>
        <tr><td>Movie type:<select name="movie_type">
          <?php
            $query = 'SELECT movietype_id, movietype_label
                      FROM movietype
                      ORDER BY movietype_label';
            $result = mysql_query($query,$db);

```



```

// Populate
while ($row = mysql_fetch_assoc($result)){
    foreach($row as $value){ // NEED TO

        if($row['movietype_
            id']== $movie_type){
                echo '<option
                    value="'. $row['movietype_id']. "' selected="selected">';
            }else{
                echo '<option
                    value="'. $row['movietype_id']. "'>';
            }
            echo
            $row['movietype_label']. '</option>';
        }
    }
}
?>
</select>
</td>
</tr>
<tr> <td> Movie actor: <select name="movie_leadactor">
<?php
    $query = 'SELECT people_id, people_fullname
                FROM people
                WHERE people_isactor=1
                ORDER BY people_fullname';
    $result = mysql_query($query,$db) or die(mysql_
error($db));

```

```

// Populate
while ($row = mysql_fetch_assoc($result)){
    foreach ($row as $value){
        if($row['people_
id'] == $movie_leadactor){
            echo ' <option
value="" . $row['people_id'] . "" selected="selected" >';
        }else{
            echo ' <option val-
ue="" . $row['people_id'] . "" >';
        }
        echo $row['people_
fullname'] . ' </option >';
    }
}
?>
</select >
<tr > <td >Movie director:<select name="movie_di-
rector">
<?php
$query = 'SELECT people_id, people_fullname
FROM people
WHERE people_isdirector=1
ORDER BY people_fullname';
$result = mysql_query($query,$db) or die(mysql_error($db));
// Populate
while ($row = mysql_fetch_assoc($result)){
    foreach($row as $value){
        if($row['people_id'] == $movie_director){
            echo ' <option value="" . $row['people_
id'] . "" selected="selected" >';

```

```

        }else{
            echo '<option
value="'. $row['people_id']. "'>';
        }
    }
    echo
$row['people_fullname'].'</option>';
}
}
?>
</select>
</td></tr>
<tr colspan="2">
    <?php
        if($_GET['action']=='edit'){
            echo '<input type="hidden" val-
ue="'. $_GET['id']. "' name="movie_id" />';
        }
    ?>
    <td><input type="submit" name="submit"
value="<?php echo ucfirst($_GET['action']); ?>" /></td>
    </tr>
</form>
</table>
</body>
</html>

```

ပြီးရင် commit.php မှာ edit အတွက် ပြင်ရေးပေးဖို့လိုပိသေးတယ်။



```

case 'edit':
    switch ($_GET['type']){
    case 'movie':
        $query = 'UPDATE movie SET
            movie_name="'.$_POST['movie_
name'].'",
            movie_year='.$_POST['movie_
year'].',
            movie_type='.$_POST['movie_
type'].',
            movie_leadactor='.$_
POST['movie_leadactor'].',
            movie_director='.$_POST['movie_
director'].',
            WHERE movie_id='.$_
POST['movie_id'];
        break;
    }
    break;

```

ဒီ code လေးတွေ ထပ်ပေါင်းပေးရမှာပါ။ အစအဆုံးဆိုရင် အောက်ပါအတိုင်း မြင်ရမှာဖြစ်ပါတယ်။

```

<?php
    $db = mysql_connect('localhost','root','') or die ('Unable to connect, check
your database parameters. ');
    mysql_select_db('moviesite1',$db) or die(mysql_error($db));
?>
<html>
    <head>
        <title> Add Movie database</title>
    </head>
    <body>

```

```

<?php
switch($_GET['action']){
case 'add':
    switch ($_GET['type']){
    case 'movie':
        $query = 'INSERT INTO movie (movie_name,
movie_year, movie_type, movie_leadactor, movie_director) VALUES (
            "$_POST['movie_name'].",
            '$_POST['movie_year'].',
            '$_POST['movie_type'].',
            '$_POST['movie_leadactor'].',
            '$_POST['movie_director'].'
        );
        break;
    }
    break;
case 'edit':
    switch ($_GET['type']){
    case 'movie':
        $query = 'UPDATE movie SET
movie_name="$_POST['movie_name'].",
movie_year='$_POST['movie_year'].',
movie_type='$_POST['movie_type'].',
movie_leadactor='$_POST['movie_leadactor'].',
movie_director='$_POST['movie_director'].'
        WHERE movie_id='$_POST['movie_id'];
        break;
    }
    break;
}
}

```

```

if (isset($query)){
    $result= mysql_query($query, $db) or die (mysql_
error($db));
}
?>
<p>Done!!!</p>
</body>
</html>

```

**How it works**

commit.php ကိုတော့ အကျယ်တဝန်မရှင်းတော့ပါဘူး။ ရှေ့မှာ ရှင်းခဲ့တဲ့ဟာကို နားလည်ရင် ခုထပ်ပေါင်း ရေးလိုက်တာကိုလည်း နားလည်မှာ ဖြစ်ပါတယ်။

ပထမဆုံး If statement ကို ကြည့်ပါ။ query string parameter name ဖြစ်တဲ့ action ကို defined လုပ်ခဲ့တယ်။ action = edit ဖြစ်ခဲ့ရင် corresponding database record နဲ့ပါတ်သတ်တဲ့ id တွေရဲ့ query string parameter တွေကို မြင်တွေ့ရမှာဖြစ်ပါတယ်။ ပြီးတော့ some variables တွေကိုလည်း set လုပ်နိုင်မှာဖြစ်ပါတယ်။

```

<td><input type="text" name="movie_name" value="< ?php echo $movie_name; ? >
"/></td>

```

ဒီ ဥပမာလေးကိုတစ်ချက်ကြည့်ကြည့်ပါ။ movie\_name ဆိုတဲ့ field က \$movie\_name ဆိုတဲ့ variable ထဲက content တွေကို default အနေနဲ့ယူပါတယ်။ ဒီလိုလုပ်ခြင်းအားဖြင့် record ကို edit လုပ်ရန် form ကို အဲဒီ record data နဲ့ဘဲ reload ဖြစ်စေပြီး user ကို ပြန်ပြပါတယ်။ ဒါကြောင့် ThaGyanMoe ကို Edit လုပ်လိုက်ရင် Movie name နေရာမှာ ThaGyanMoe ဆိုတဲ့ ရှိနှင့်ပြီးသား record data ကို ပြတာပါ။

အထက်ကလိုမျိုး text field ကို ပြင်ရတာ ရှင်းလင်းလွယ်ကူပါတယ်။ list field ကို ပြင်ရတာကြတော့ တစ်မျိုးဖြစ်သွားပြီ။ သင့်အနေနဲ့ User ကို dropdown list လေးဘဲထုတ်ပြလို့မရဘူး။ user အနေနဲ့ original value ကို မပြင်ဘဲ ထားရင်ရော ဘယ်လိုလုပ်မလဲ။ သင့်အနေနဲ့ original value ကို default အနေနဲ့ ပြကိုပြထားရမယ်။ ဒါမှ user က မပြင်ခြင်ရင် skip လုပ်သွားနိုင်မှာပါ။ အောက်က code လေးကို လေ့လာကြည့်ပါ။



<?php

```

$query = 'SELECT movietype_id, movietype_label
FROM movietype
ORDER BY movietype_label';
$result = mysql_query($query,$db);
// Populate
while ($row = mysql_fetch_assoc($result)){
    foreach($row as $value){
        if($row['movietype_id']==$movie_
type){
            echo '<option value="'. $row['movietype_id']. "'
selected="selected">';
        }else{
            echo '<option value="'. $row['movietype_id']. "'>';
        }
        echo $row['movietype_label']. '</option>';
    }
}
}

```

?>

ဒီ script မှာဆိုရင် list တစ်ခုလုံးကို reload လုပ်ပါတယ်။ ဒါ့အပြင် filed အတွင်းက current value နဲ့ default value ကို compare လုပ်ပါတယ်။ တကယ်လို့ identical ဖြစ်ခဲ့မယ်ဆိုရင် selected လုပ်လိုက်ပါတယ်။ ဆိုလိုတာက သင့် အနေနဲ့ list ထဲကနေ ဘာကိုဘဲရွေးရွေး identical ဖြစ်မှာဘဲလေ။ သင့်အနေနဲ့ list ထဲမှာ predefined လုပ်ထားတာတွေက default value တွေဖြစ်သွားမှာပေါ့။ action တို့ drama တို့ ပေါ့။ တခြား director တို့ ဘာတို့လည်း ဒီ concept တွေချည်းပါဘဲ။

Exercise

For chapter (6)

1. Create the edit/delete code for the people table. Use the movie code as an example.

## Answers For Exercises

## For chapter (2)

## 1. Ans

```

<html>
  <head>
    <title> How many days in this month? </title>
  </head>
  <body>
    <?php
date_default_timezone_set('Asia/Yangon');
$month_name = date('F');
echo ' <p> The month is ' . $month_name . ' . </p> ';
echo ' <p> There are ';
$month = date('n');
if ($month == 1) { echo '31'; }
if ($month == 2) { echo '28 (unless it\'s a leap year)'; }
if ($month == 3) { echo '31'; }
if ($month == 4) { echo '30'; }
if ($month == 5) { echo '31'; }
if ($month == 6) { echo '30'; }
if ($month == 7) { echo '31'; }
if ($month == 8) { echo '31'; }
if ($month == 9) { echo '30'; }
if ($month == 10) { echo '31'; }
if ($month == 11) { echo '30'; }
if ($month == 12) { echo '31'; }
echo ' days in this month. </p> ';
$months_left = 12 - $month;
echo ' <p> There are ' . $months_left . ' months left in the year. </p> ';
?>
  </body>
</html>

```



**2. Ans**

```
<?php
include 'footer.php';
?>
```

Then you need to create the file footer.php , which consists of these lines

```
<div style="text-align: center">
This site developed by: <a href="mailto:admin@example.com"> Admin </a>
</div>
```

**3. Ans greeting\_for\_chap2.php**

```
<html>
<head>
<title> Greeting </title>
</head>
<body>
<?php
date_default_timezone_set('Yangon/Asia');

if (date('G') >= 5 && date('G') <= 11) {
    echo ' <h1> Good Morning! </h1> ';
}
else if (date('G') >= 12 && date('G') <= 18) {
    echo ' <h1> Good Afternoon! </h1> ';
}
else if (date('G') >= 19 && date('G') <= 4) {
    echo ' <h1> Good Evening! </h1> ';
}
?>
</body>
</html>
```

## 4. Ans

First you would display a form to your users, possibly on the login page, such as this:

```
<table>
<tr>
<td> <label for="font"> Select Font: </label> </td>
<td> <select id="font" name="font">
<option value="Verdana"> Verdana </option>
<option value="Arial"> Arial </option>
<option value="Times New Roman"> Times New Roman </option>
</select>
</td>
</tr>
<tr>
<td> <label for="size"> Select Size: </label> </td>
<td> <select id="size" name="size">
<option value="10px"> 10px </option>
<option value="12px"> 12px </option>
<option value="16px"> 16px </option>
<option value="20px"> 20px </option>
</select>
</td>
</tr>
<tr>
<td> <label for="color"> Select Color: </label> </td>
<td> <select id="size" name="size">
<option value="black"> black </option>
<option value="green"> green </option>
<option value="purple"> purple </option>
<option value="red"> red </option>
</select>
</td>
</tr>
</form>
```

Store the received values in your session variables. You would add something like

မိမိ၏ ဓာပေ

this to whatever the first script is that processes the form

```
$_SESSION['font']=$_POST['font'];
$_SESSION['size']=$_POST['size'];
$_SESSION['color']=$_POST['color'];
```

Then, every time you had text that needed to be styled, you could output your session variables ' value, like this:

```
<p <?php
echo ' style="font-family: ' . $_SESSION['font'] . '";
echo 'font-size: ' . $_SESSION['size'] . '";
echo 'color: ' . $_SESSION['color'] . '";
?>> Text to display </p>
```

**5. Ans**

You would add a line like this to the end of your font preference form:

```
<input type="checkbox" id="save_prefs" name="save_prefs"/>
<label for="save_prefs"> Save these preferences for the next time you login. </
label>
```

Then at the very beginning of the script that processes the incoming form data, you would add a statement that looks something like this:

```
if (isset($_POST['save_prefs'])) {
    setcookie('font', $_POST['font'], time() + 60);
    setcookie('size', $_POST['size'], time() + 60);
    setcookie('color', $_POST['color'], time () + 60);
}
```

Instead of accessing those variables through the session later when you need them, you would access them through the cookie like this:

```
echo $_COOKIE['font'];
```



**6. Ans**

```
<?php
function display_times($num) {
    echo ' <h1> You have viewed this page ' . $num . ' time(s). </h1> ';
}

// get the cookie value and add 1 to it for this visit
$num_times = 1;
if (isset($_COOKIE['num_times'])) {
    $num_times = $_COOKIE['num_times'] + 1;
}

// set the value back to the cookie for the next time
setcookie('num_times', $num_times, time() + 60);
?>
<html>
<head>
<title> Viewed Times </title>
</head>
<body>
<?php display_times($num_times); ?>
</body>
</html>
```

**For chapter (3)**
**1. Ans**

```

<?php
function get_people_fullname($db, $people_id) {
    $query = 'SELECT
        people_fullname
    FROM
        people
    WHERE
        people_id = ' . $people_id;
    $result = mysql_query($query, $db) or die(mysql_error($db));
    $row = mysql_fetch_assoc($result);
    return $row['people_fullname'];
}

$db = mysql_connect('localhost', 'root', '') or
    die ('Unable to connect. Check your connection parameters.');
```

```

mysql_select_db('moviesite1', $db) or die(mysql_error($db));
?>
<html>
<head>
<title> Movie Info </title>
</head>
<body>
<table border="1">
<tr>
<th> Movie Name </th>
<th> Lead Actor </th>
<th> Director </th>
</tr>

```

```

<?php
//get the movies
$query = 'SELECT
    movie_name, movie_leadactor, movie_director
FROM
    movie';
$result = mysql_query($query, $db) or die(mysql_error($db));

while ($row = mysql_fetch_assoc($result)) {
    //call our functions to get specific info
    $actor_name = get_people_fullname($db, $row['movie_leadactor']);
    $director_name = get_people_fullname($db, $row['movie_director']);

    //show table row
    echo ' <tr> ';
    echo ' <td> ' . $row['movie_name'] . ' </td> ';
    echo ' <td> ' . $actor_name . ' </td> ';
    echo ' <td> ' . $director_name . ' </td> ';
    echo ' </tr> ';
}
?>
</table>
</body>
</html>

```

## 2. Ans

```

<?php
$db = mysql_connect('localhost', 'root', '') or
    die ('Unable to connect. Check your connection parameters.');
```

mysql\_select\_db('moviesite1', \$db) or die(mysql\_error(\$db));

```

?>

```



```

<html>
<head>
<title> Comedy Movies </title>
</head>
<body>
<table border="1">
<tr>
<th> Movie Name </th>
<th> Year </th>
</tr>
<?php
// get the movie_type for comedies
$query = 'SELECT
    movietype_id
FROM
    movietype
WHERE
    movietype_label = "Comedy";
$result = mysql_query($query, $db) or die(mysql_error($db));
$row = mysql_fetch_assoc($result);

//get the movies
$query = 'SELECT
    movie_name, movie_year
FROM
    movie
WHERE
    movie_type = ' . $row['movietype_id'] . '
ORDER BY
    movie_name';
$result = mysql_query($query, $db) or die(mysql_error($db));

while ($row = mysql_fetch_assoc($result)) {

```

```

//show table row
echo ' <tr> ';
echo ' <td> ' . $row['movie_name'] . ' </td> ';
echo ' <td> ' . $row['movie_year'] . ' </td> ';
echo ' </tr> ';
}
?>
</table >
</body >
</html>

```

### 3. Ans

```

<?php
$db = mysql_connect('localhost', 'root', '') or
    die ('Unable to connect. Check your connection parameters.');
```

mysql\_select\_db('moviesite1', \$db) or die(mysql\_error(\$db));

```

//get our starting point for the query from the URL
if (isset($_GET['offset'])) {
    $offset = $_GET['offset'];
} else {
    $offset = 0;
}
//get the movie
$query = 'SELECT
    movie_name, movie_year
FROM
    movie
ORDER BY

```

```

movie_name
LIMIT ' . $offset . ', 1';
$result = mysql_query($query, $db) or die(mysql_error($db));
$row = mysql_fetch_assoc($result);
?>
<html>
<head>
<title> <?php echo $row['movie_name']; ?> </title>
</head>
<body>
<table border="1" >
<tr>
<th> Movie Name </th>
<th> Year </th>
</tr> <tr>
<td> <?php echo $row['movie_name']; ?> </td>
<td> <?php echo $row['movie_year']; ?> </td>
</tr>
</table>
<p>
<a href="page.php?offset=0" > Page 1 </a> ,
<a href="page.php?offset=1" > Page 2 </a> ,
<a href="page.php?offset=2" > Page 3 </a>
</p>
</body>
</html>

```



## For chapter (4)

## 1. Ans

First you need to add the display in your `movie_details.php` file, as follows:

```
// display the information
echo <<<ENDHTML
<html>
<head>
  <title> Details and Reviews for: $movie_name </title>
</head>
<body>
  <div style="text-align: center;">
    <h2> $movie_name </h2>
    <h3> <em> Details </em> </h3>
    <table cellpadding="2" cellspacing="2" style="width: 70%; margin-left: auto; margin-
right: auto;">
      <tr>
        <td> <strong> Title </strong> </strong> </td>
        <td> $movie_name </td>
        <td> <strong> Release Year </strong> </strong> </td>
        <td> $movie_year </td>
      </tr> <tr>
        <td> <strong> Movie Director </strong> </td>
        <td> $movie_director </td>
        <td> <strong> Cost </strong> </td>
        <td> $$movie_cost < td/ >
      </tr> <tr>
        <td> <strong> Lead Actor </strong> </td>
        <td> $movie_leadactor </td>
        <td> <strong> Takings </strong> </td>
        <td> $$movie_takings < td/ >
      </tr> <tr>
```

```

<td> <strong> Running Time </strong> </td>
<td> $movie_running_time </td>
<td> <strong> Health </strong> </td>
<td> $movie_health < td/ >
</tr> <tr>
<td> </td>
<td> </td>
<td> <strong> Average Review </strong> </td>
<td> $average_review < td/ >
</tr>
</table>

```

ENDHTML;

Then you can create a function to determine the number of total reviews and calculate the average. These lines would go near the top of your `movie_details.php` file where the rest of the functions are defined.

```

function get_avg_review($db, $movie_id) {
    $query = 'SELECT
        review_rating
    FROM
        reviews
    WHERE
        review_movie_id = ' . $movie_id;
    $result = mysql_query($query, $db);
    $total_reviews = mysql_num_rows($result);

    $current = 0;
    while ($row = mysql_fetch_assoc($result)) {
        $current = $current + $row['review_rating'];
    }

    return $current / $total_reviews;
}

```

Then call the function when you get the main information for the movie by adding the following lines to

movie\_details.php :

```
$movie_health = calculate_differences($row['movie_takings'],
    $row['movie_cost']);
$average_review = get_avg_review($db, $_GET['movie_id']);
$average_review = round($average_review, 2);
```

## 2. Ans

You need to change your movie\_details.php file as follows:

```
if (isset($_GET['sort'])) {
    $sort = $_GET['sort'];
} else {
    $sort = 'review_date';
}
```

// retrieve reviews for this movie

```
$query = 'SELECT
    review_movie_id, review_date, reviewer_name, review_comment,
    review_rating
FROM
    reviews
WHERE
    review_movie_id = ' . $_GET['movie_id'] . '
ORDER BY
    ' . $sort . ' ASC';
```

```
$result = mysql_query($query, $db) or die(mysql_error($db));
```

```
// display the reviews
```

```
$mid = $_GET['movie_id'];
```

```
echo <<<ENDHTML
```

```
<h3> <em> Reviews </em> </h3>
```

```
<table cellpadding="2" cellspacing="2" style="width: 90%; margin-left: auto; margin-
```



```
right: auto;" >
    <tr>
        <th style="width: 7em;">
            <a href="movie_details.php?movie_id=$mid&ort=review_date">
Date </a>
        </th>
        <th style="width: 10em;">
            <a href="movie_details.php?movie_id=$mid&sort=reviewer_
name"> Reviewer</a>
        </th>
        <th>
            <a href="movie_details.php?movie_id=$mid&sort=review_
comment">Comments</a>
        </th>
        <th style="width: 5em;">
            <a href="movie_details.php?movie_id=$mid&sort=review_
rating">Rating</a>
        </th>
    </tr>
ENDHTML;
```

3. Ans

Your movie\_details.php file will need these lines added in or changed:

```
$odd = true;
while ($row = mysql_fetch_assoc($result))
{
    $date = $row['review_date'];
    $name = $row['reviewer_name'];
    $comment = $row['review_comment'];
    $rating = generate_ratings($row['review_rating']);

    if ($odd) {
        echo '<tr style="background-color: #EEEEEE;">';
    } else {
```

```
echo "<tr style='background-color: #FFFFFF;'>";
}
echo <<<ENDHTML
  <td style="vertical-align:top; text-align: center;"> $date</td>
  <td style="vertical-align:top;"> $name </td>
  <td style="vertical-align:top;"> $comment </td>
  <td style="vertical-align:top;"> $rating </td>
</tr>
ENDHTML;
$odd = !$odd;
}
```

For chapter (5)

1. Ans

```
<form method="post" action="showratings.php">
<table>
<tr>
<td> Movie title: </td>
<td> <input type="text" name="movie"> </td>
</tr> <tr>
<td> Rating: </td>
<td>
<?php
for ($i = 1; $i <= 5; $i++) {
echo '<input type="radio" name="rating" value="' . $i . '">'. $i . ' <br/> ';
}
?>
</td>
</tr> <tr>
<td> Comments: </td>
<td> <textarea name="comments" cols="40" rows="10"> </textarea> </td>
</tr>
</table>
</form>
```



## 2. Ans

This exercise takes the form of two pages. The first provides the form to enter the options, and the latter shows the result. First, the form on the input page:

```
<form method="post" action="showratings.php">
  <table>
    <tr>
      <td> Select options: </td>
      <td>
        <?php
        for ($i = 0; $i < 5; $i++) {
          echo ' <input type="text" name="sval[" . $i . "]" /> <br/> ';
        }
        ?>
      </td>
    </tr>
  </table>
</form>
```

And then, on the second page:

```
<select>
  <?php
  for ($i = 0; $i < 5; $i++) {
    echo ' <option value="" . $_POST['sval'][$i] . "' > ' . $_POST['sval'][$i] .
      ' </option> ';
  }
  ?>
</select>
```

3. Ans

This exercise is easily handled using a few form fields and some basic arithmetic:

```
<?php
$num1 = (isset($_POST['num1'])) ? $_POST['num1'] : null;
$num2 = (isset($_POST['num2'])) ? $_POST['num2'] : null;
$operator = (isset($_POST['operator'])) ? $_POST['operator'] : '+';
?>
<form method="post" action="#" >
<div>
<input type="text" name="num1" size="3" value=" <?php echo $num1; ?>" />
<select name="operator">
<option value="+"
<?php if ($operator == '+') { echo 'selected="selected"'; } ?>>+ </option>
<option value="-"
<?php if ($operator == '-') { echo 'selected="selected"'; } ?>>- </option>
<option value="*"
<?php if ($operator == '*') { echo 'selected="selected"'; } ?>>& times; </option>
<option value="/"
<?php if ($operator == '/') { echo 'selected="selected"'; } ?>>& divide;</option>
</select>
<input type="text" name="num2" size="3" value=" <?php echo $num2; ?>" />
<input type="submit" value="=" />
<strong>
<?php
if (!is_null($num1) && !is_null($num2)) {
    if ($operator == '+') {
        echo $num1 + $num2;
    } else if ($operator == '-') {
        echo $num1 - $num2;
```



```
} else if ($operator == '*') {  
    echo $num1 * $num2;  
} else if ($operator == '/') {  
    echo $num1 / $num2;  
} else {  
    echo 'UNKNOWN';  
}  
}  
?>  
</strong>  
</div>  
</form>
```



For chapter (6)

1. Ans

One possible solution is as follows. Change commit.php as highlighted:

```
<?php
//connect to MySQL
$db = mysql_connect('localhost', 'root', '') or
    die ('Unable to connect. Check your connection parameters.');
```

```
//make sure our recently created database is the active one
mysql_select_db('moviesite1', $db) or die(mysql_error($db));
?>
<html>
<head>
<title> Commit </title>
</head>
<body>
<?php
switch ($_GET['action']) {
case 'add':
    switch ($_GET['type']) {
    case 'movie':
        $query = 'INSERT INTO movie
            (movie_name, movie_year, movie_type, movie_leadactor,
            movie_director)
        VALUES
            (" . $_POST['movie_name'] . "',
            ' . $_POST['movie_year'] . ',
            ' . $_POST['movie_type'] . ',
            ' . $_POST['movie_leadactor'] . ',
            ' . $_POST['movie_director'] . ')';
        break;
```

```

case 'people':
    $query = 'INSERT INTO people
              (people_fullname, people_isactor, people_isdirector)
              VALUES
              ('" . $_POST['people_name'] . "',
              ' . $_POST['people_isactor'] . ',
              ' . $_POST['people_isdirector'] . ')';
    break;
}
break;
case 'edit':
    switch ($_GET['type']) {
    case 'movie':
        $query = 'UPDATE movie SET
                  movie_name = '" . $_POST['movie_name'] . "',
                  movie_year = ' . $_POST['movie_year'] . ',
                  movie_type = ' . $_POST['movie_type'] . ',
                  movie_leadactor = ' . $_POST['movie_leadactor'] . ',
                  movie_director = ' . $_POST['movie_director'] . '
                  WHERE
                  movie_id = ' . $_POST['movie_id'];
        break;
    case 'people':
        $query = 'UPDATE people SET
                  people_fullname = '" . $_POST['people_fullname'] . "',
                  people_isactor = '" . $_POST['people_isactor'] . "',
                  people_isdirector = '" . $_POST['people_isdirector'] . "'
                  WHERE
                  people_id = ' . $_GET['id'];
        break;
    }
    break;

```

```
if (isset($query)) {  
    $result = mysql_query($query, $db) or die(mysql_error($db));  
}  
?>  
<p> Done! </p>  
</body>  
</html>
```



# PHP

လေ့ကျင့်စာအုပ်များ

ကိုကျော်