

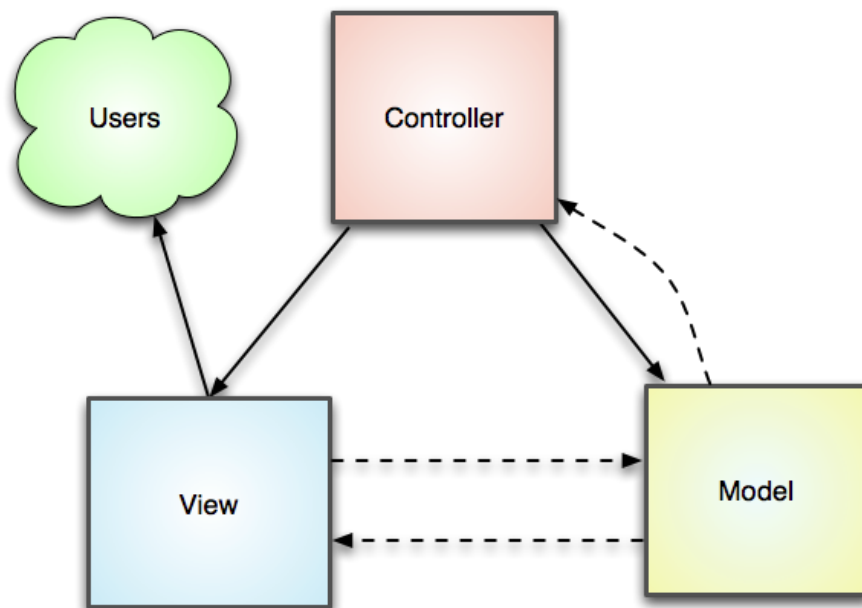
## Zend Framework Getting Started

### I. Overview of Zend Framework

Zend Framework is an open source framework for developing web applications and services with PHP 5. Zend Framework is implemented using 100% object-oriented code. The component structure of Zend Framework is somewhat unique; each component is designed with few dependencies on other components. This loosely coupled architecture allows developers to use components individually. We often call this a "use-at-will" design. (refers to zend manual).

#### a. Model – View – Controller

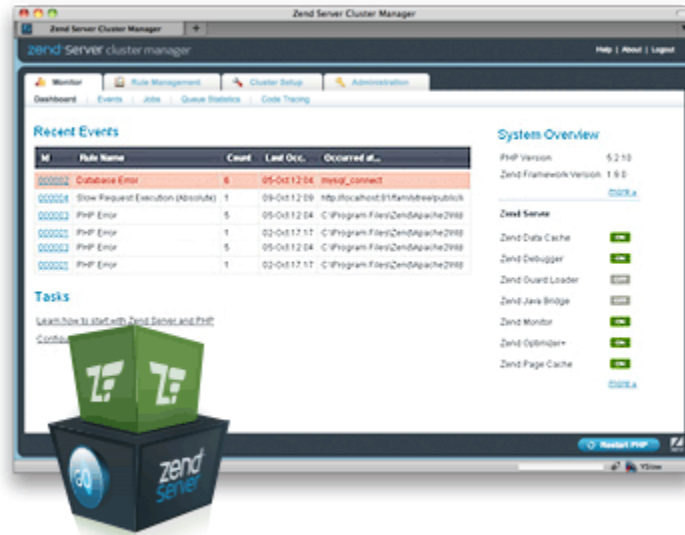
- *Model* - This is the part of your application that defines its basic functionality behind a set of abstractions. Data access routines and some business logic can be defined in the model.
- *View* - Views define exactly what is presented to the user. Usually controllers pass data to each view to render in some format. Views will often collect data from the user, as well. This is where you're likely to find HTML markup in your MVC applications.
- *Controller* - Controllers bind the whole pattern together. They manipulate models, decide which view to display based on the user's request and other factors, pass along the data that each view will need, or hand off control to another controller entirely.



## II. Environment Settings

### a. Zend Server CE (Community Edition)

<http://www.zend.com/en/products/server-ce/downloads>



Zend Server Community Edition (CE) is the free edition of Zend Server. The different between Zend Server and Zend Serve CE describe below.

Description	Zend Server	Zend Server CE
Complete PHP stack that includes ZF	Yes	Yes
Opcode acceleration (Zend Optimizer+)	Yes	Yes
PHP application deployment	Yes	
Page caching and job queue support	Yes	
PHP monitoring and code tracing	Yes	
<a href="#">Zend Server Cluster Manager</a> support	Yes	
technical support, updates and hotfixes	Yes	IBM only, for 1 year

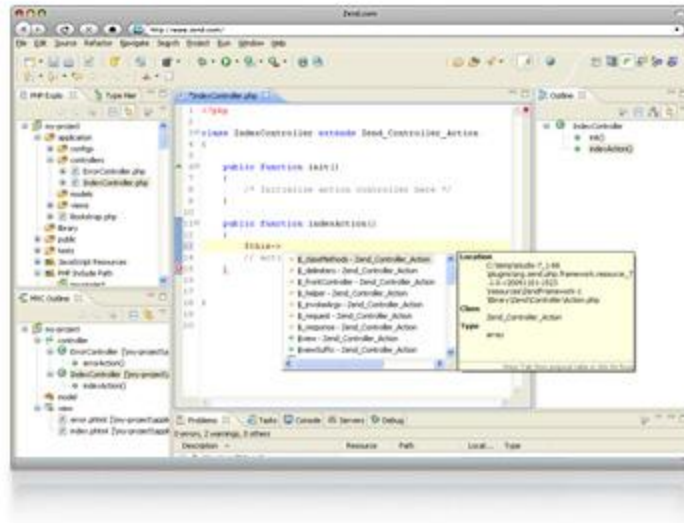
Packaging : Linux, Windows, Mac Os X, IBM i

Support PHP Version : 5.2 atau 5.3.

Notes: This Tutorial will be only showed using Zend Server CE Windows version. So, you advised for this specific OS.

## b. Zend Studio vs Eclipse PHP

Zend actually already provided tools for develop zend application and integrated with zend framework as well. It is Zend Studio. You can download it in <http://www.zend.com/products/studio/>. This tool is free trial for 60 days



Already bundled based on specific OS (Operating System) such as Linux, Windows, and Mac OS X.

Another tool for zend application development is Eclipse PHP which is could be downloaded at <http://eclipse.org/pdt/downloads/>. As far as this document being made, I just know the last version of PHP Eclipse is Helios. This tool is free and unlimited for PHP development, but since zend framework is actually one of many PHP Framework, we can use this tool.

Notes : you may be advised to use Eclipse PHP since I will use this tools as the mostly screenshot on every examples I made.

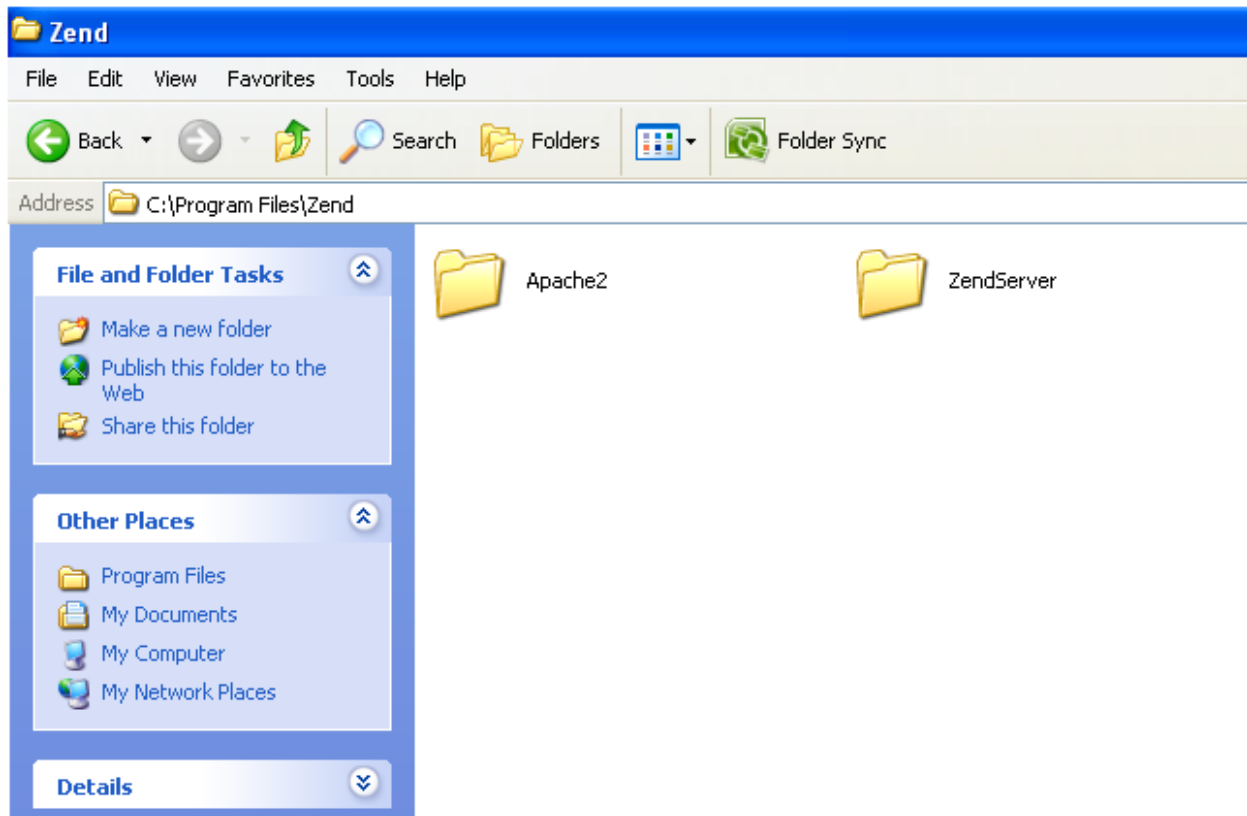
## c. Zend Framework

You can download at <http://framework.zend.com/download/current/>. Choose Zend Framework Full Version because it's already included with the documentation and manual.

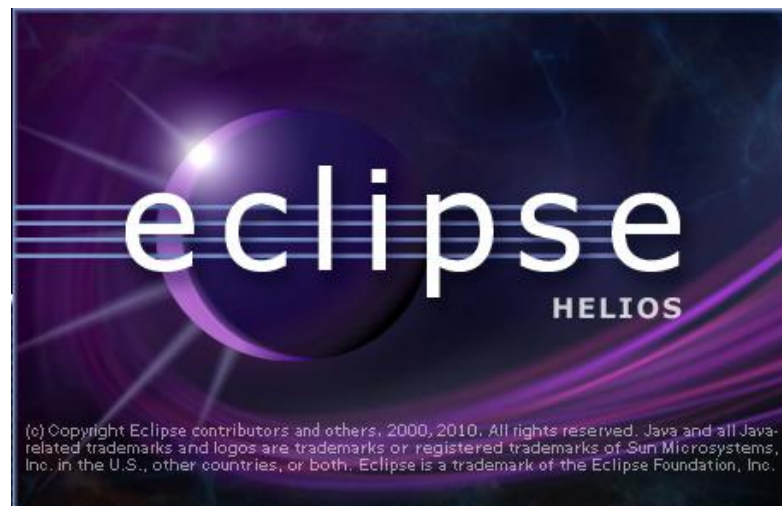
### III. Installation

#### a. Zend Server CE

For Installing Zend Server CE, simply, you just click the installer for windows. Follow every step to complete the installations. By default, zend will be located in **C:/Program Files/Zend/**.



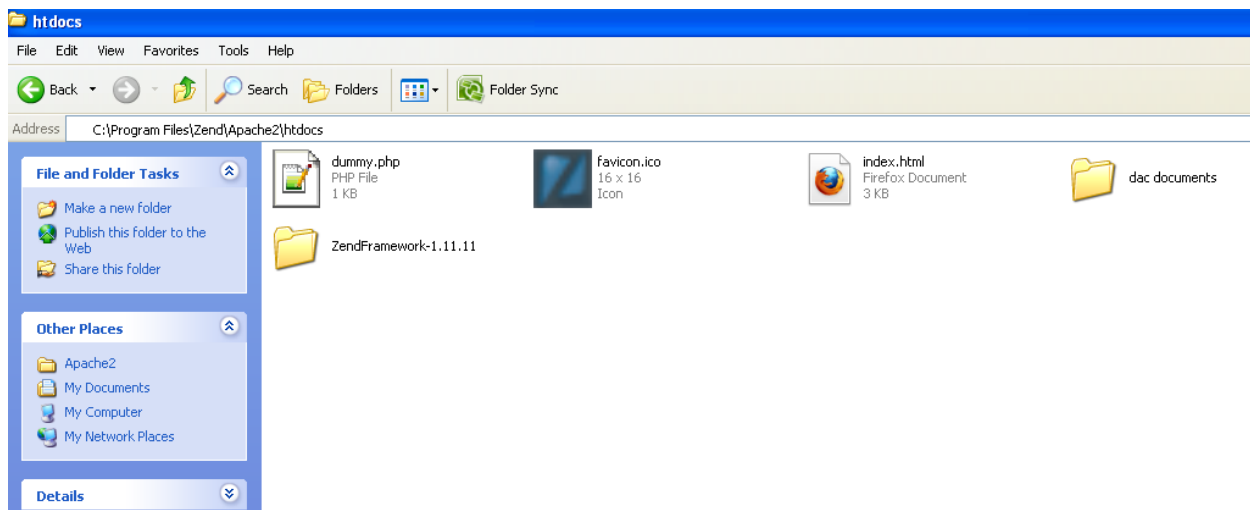
#### b. Eclipse PHP



Install Eclipse PHP isn't really hard, seriously. For windows, you can simply extract it, and run it by clicking on the executable file, usually, **eclipse.exe**. Or, if you downloaded installer version, you just need to follow the instructions, and run it. There is another dependency for using eclipse, you must have installed **Java JDK or JRE** on your computer.

c. Zend Framework

Extract downloaded Zend Framework on **C:\Program Files\Zend\Apache2\htdocs**. The following results :




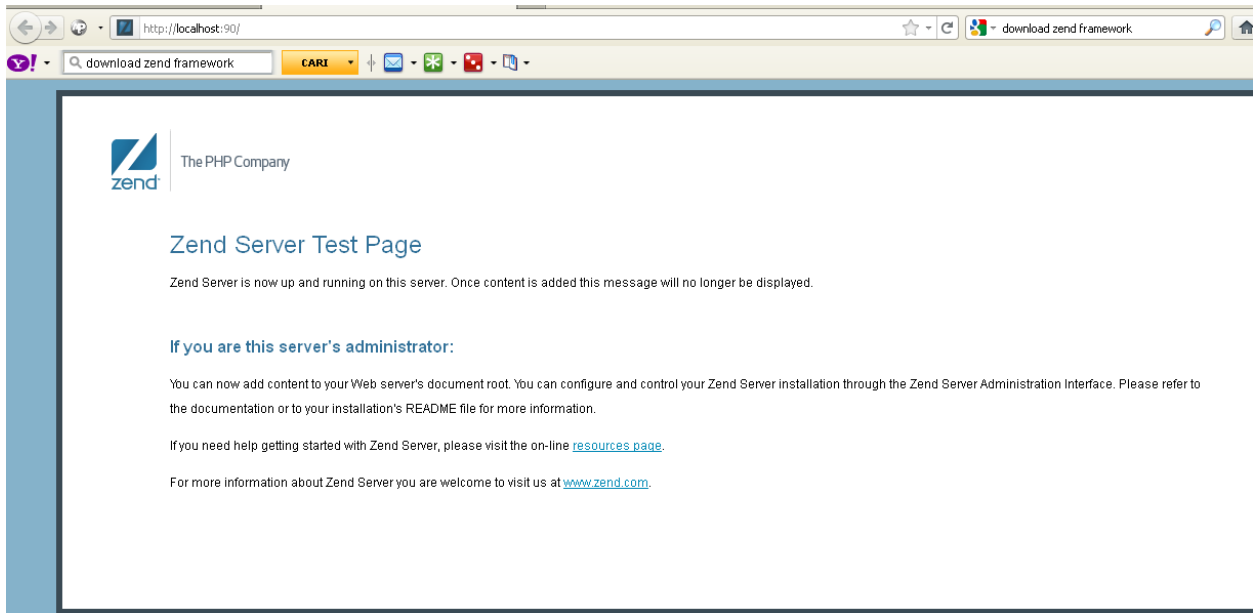
In this training, I use Zend Framework with current version which is version 1.11.11. You are advised to use the same version, too, to experience the same result.

IV. Development

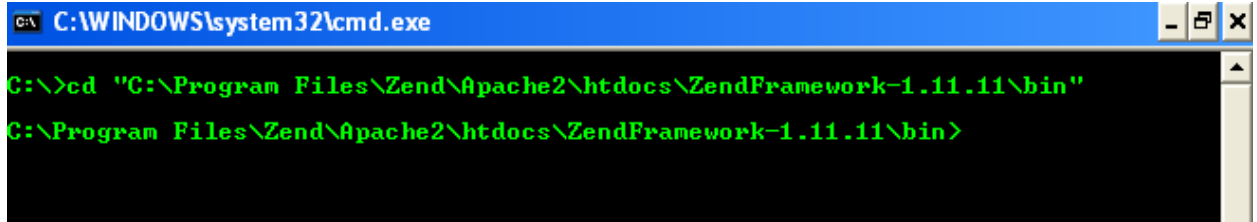
a. Installation Zend Application

Firstly, make sure your Zend Server already running on your computer.

Clicked on apache icon on the right side of your desktop window (  ) and start the apache zend server. Open your browser, and type [http://localhost:\[portnumber\]/](http://localhost:[portnumber]/), e.g. <http://localhost:80/> or <http://localhost:90/>. Port Number is set when you installed zend server before. In this session, I will use the second example (<http://localhost:90/>). If you succeed, the preview on your browser should be like the picture below.



Well, now, open your zend framework folder (already extracted) on **C:\Program Files\Zend\Apache2\htdocs\ (C:\Program Files\Zend\Apache2\htdocs\ZendFramework-1.11.11\bin)** using command prompt.



Next, we will create Zend Framework using **zf create project** command. See example below.

```
C:\WINDOWS\system32\cmd.exe

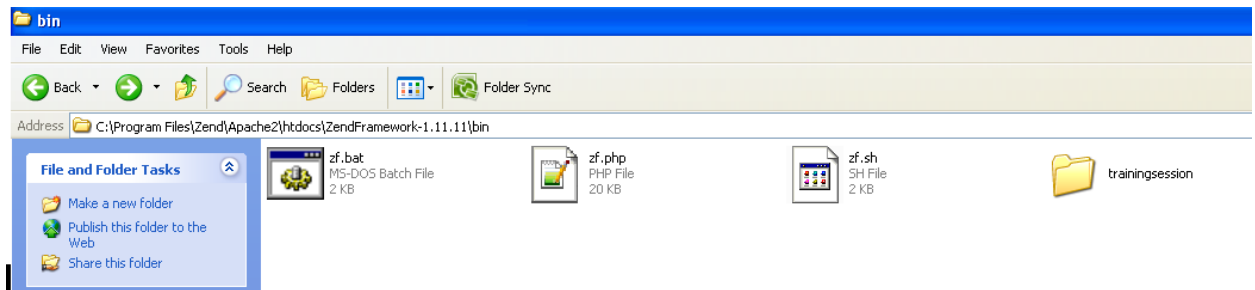
C:\Program Files\Zend\Apache2\htdocs\ZendFramework-1.11.11\bin>zf create project
trainingsession
Creating project at C:/Program Files/Zend/Apache2/htdocs/ZendFramework-1.11.11/b
in/trainingsession
Note: This command created a web project, for more information setting up your U
HOST, please see docs/README
Testing Note: PHPUnit was not found in your include_path, therefore no testing a
ctions will be created.

C:\Program Files\Zend\Apache2\htdocs\ZendFramework-1.11.11\bin>_
```

Type on your command window : **zf create project trainingsession**.  
trainingsession is the name of our project. Back to explorer and open directory :

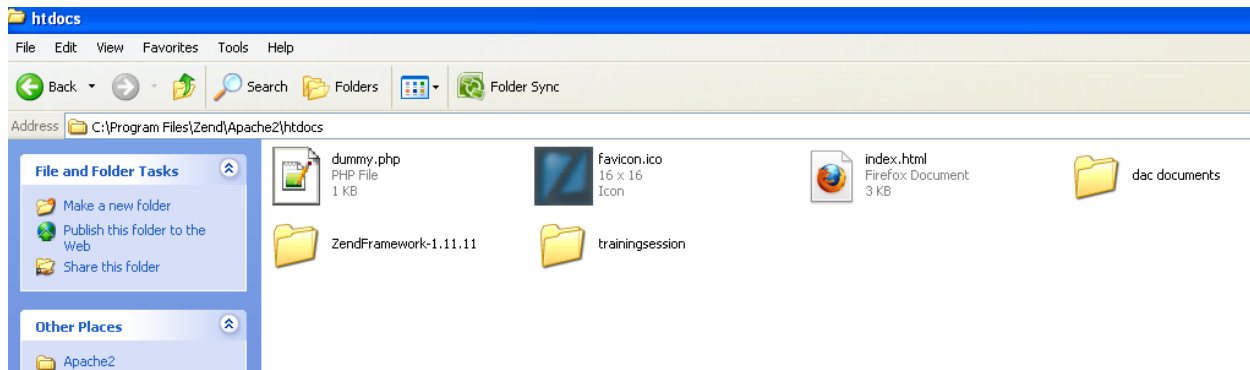
C:\Program Files\Zend\Apache2\htdocs\ZendFramework-1.11.11\bin

You'll find sharingsession folder like the picture below.



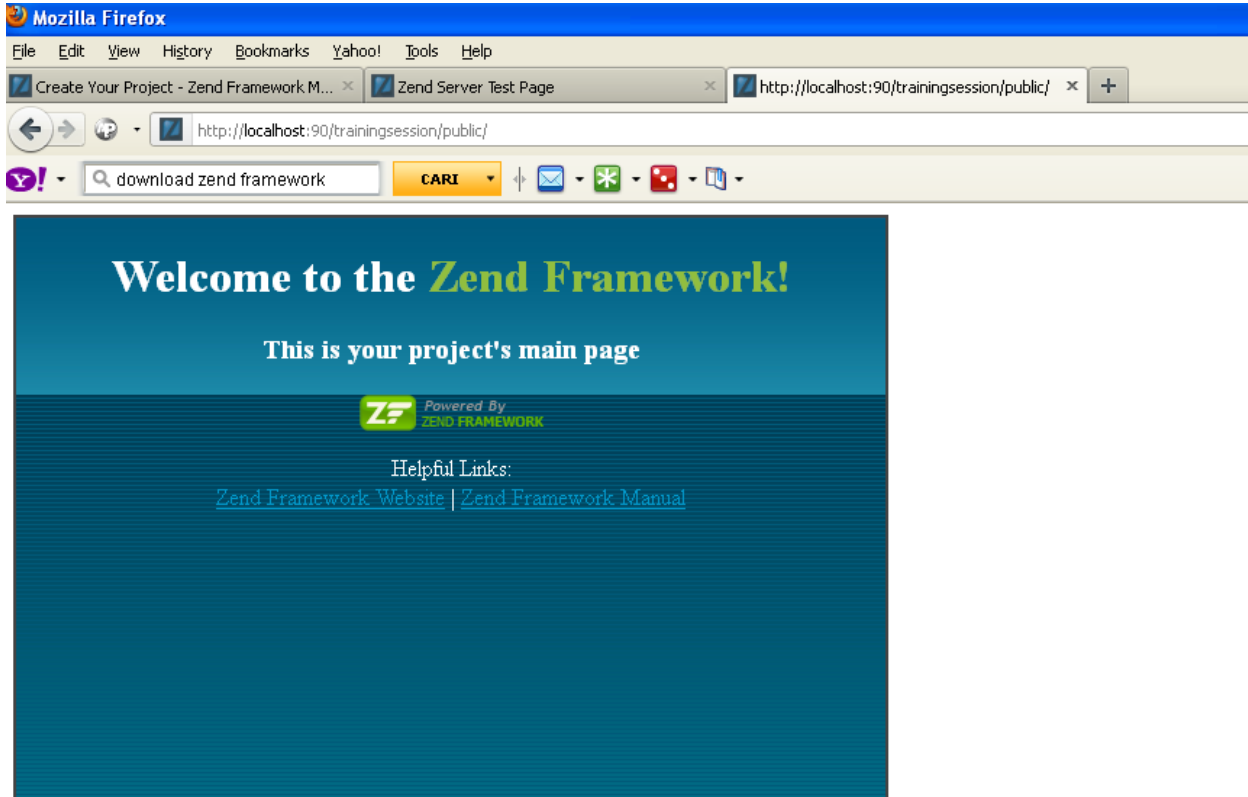
Cut and paste trainingsession folder into

C:\Program Files\Zend\Apache2\htdocs



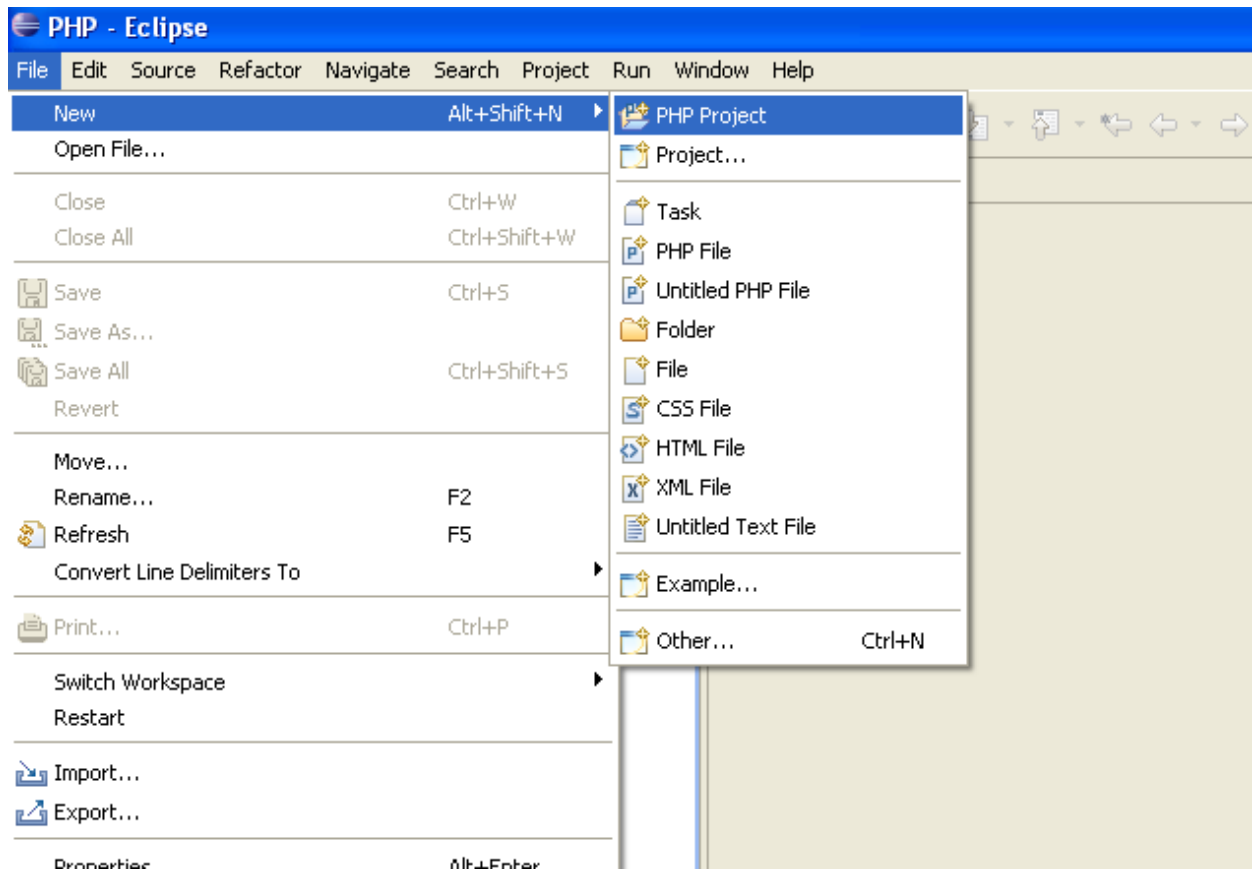
Well, you already made a simple project using zend framework. To access this application, back to your browser and browse for url :

<http://localhost:90/trainingession/public/>

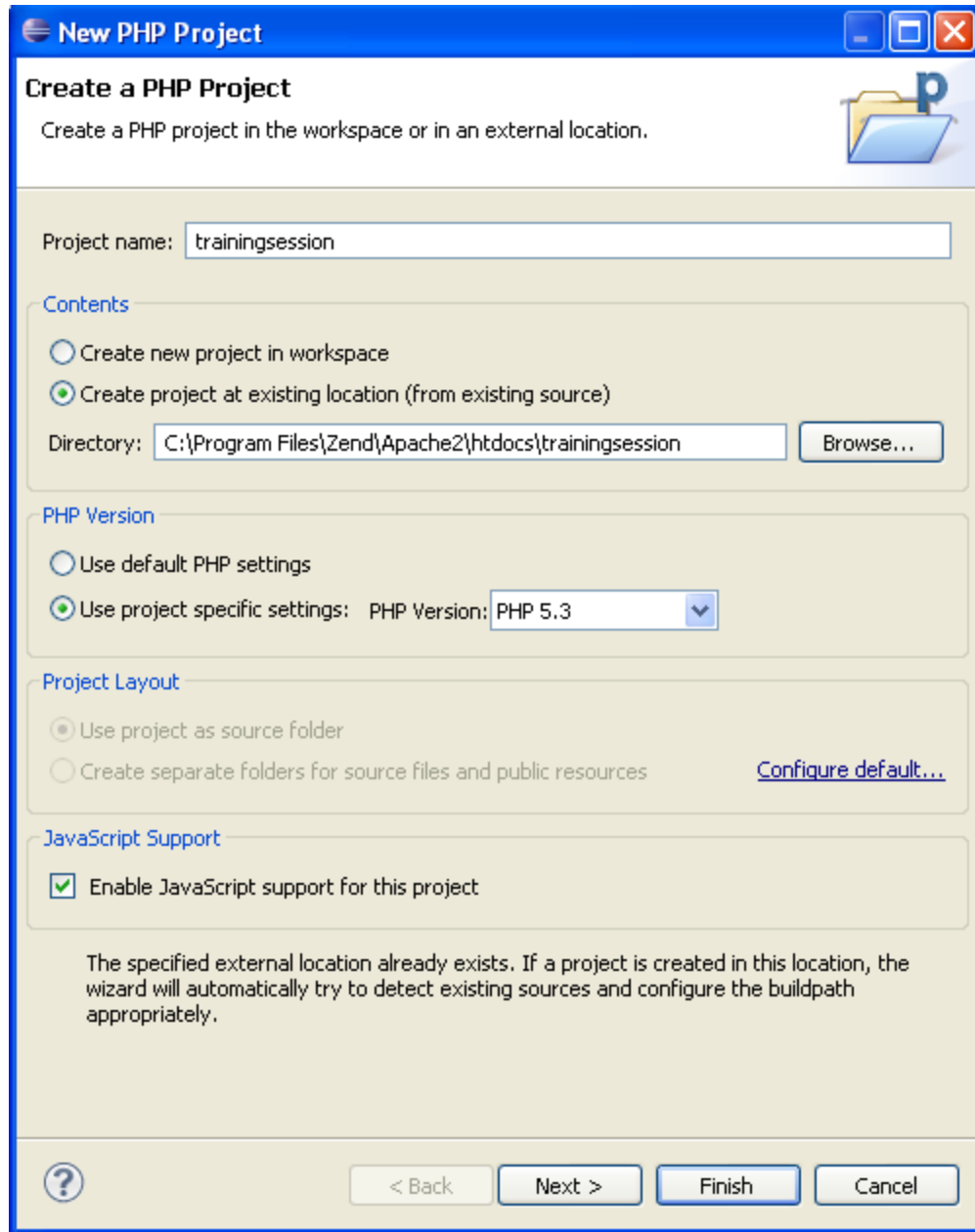


b. Develop Simple MVC Architecture

Now, open your IDE/Tool (I mean Eclipse) and create new **PHP Project** just like the picture below.



Next, fill the form just like this :



## Settings

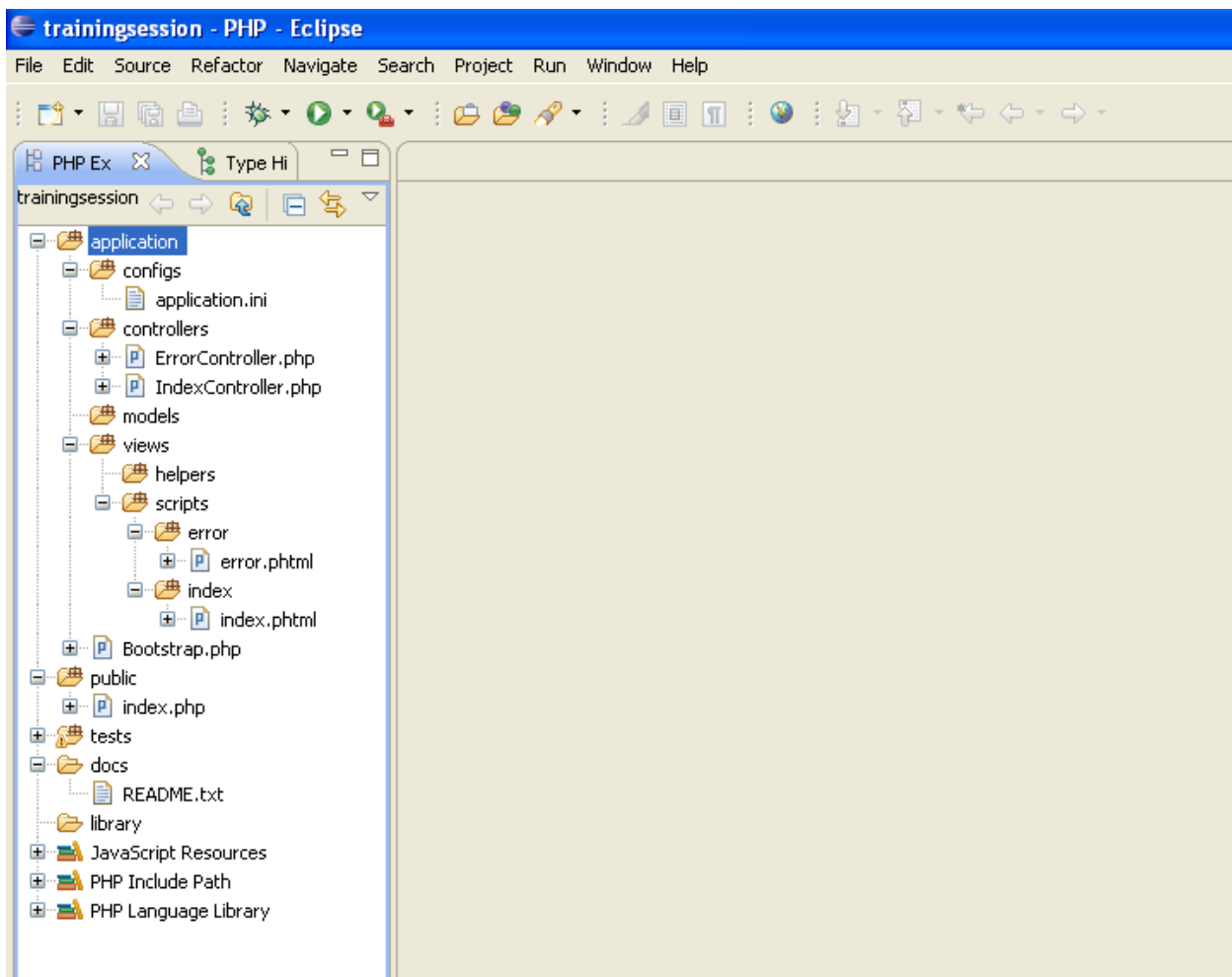
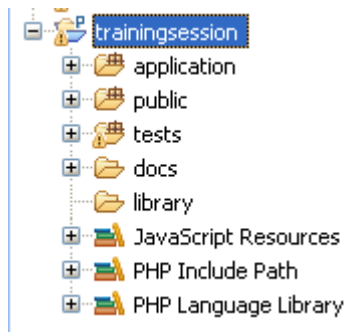
**Project name** : trainingsession

**Contents** : create project at existing location

(C:\Program Files\Zend\Apache2\htdocs\trainingsession)

**PHP Version** : PHP 5.3 (Based on the downloaded Zend Server CE)

The following is the result until this step.



First, I would like to explain the structure of Zend MVC Web Application showed the picture above. Our controllers will be placed on **application/controllers** directory. View will be place on **application/views/scripts/**. And Models will be placed on **application/models**.

## i. Create View Form

Usually, view form is html/css/javascript part of web development. In this section, I won't explain html, css or even javascript part. For more details you can just google it by yourself. I just want to show you how to create simple elegant form using form builder which is available on the internet. I'll use this form builder, <http://www.phpform.org/formbuilder/>. Just added some input form by simply clicked on form panel on the right side. See the picture below for the result.

### User Registration

This is user registration form

**Name**

First Last

**Birth Date**

/  /

MM DD YYYY

**Address**

Street Address

Address Line 2

City State / Province / Region

Zip / Postal Code Country

**additional**

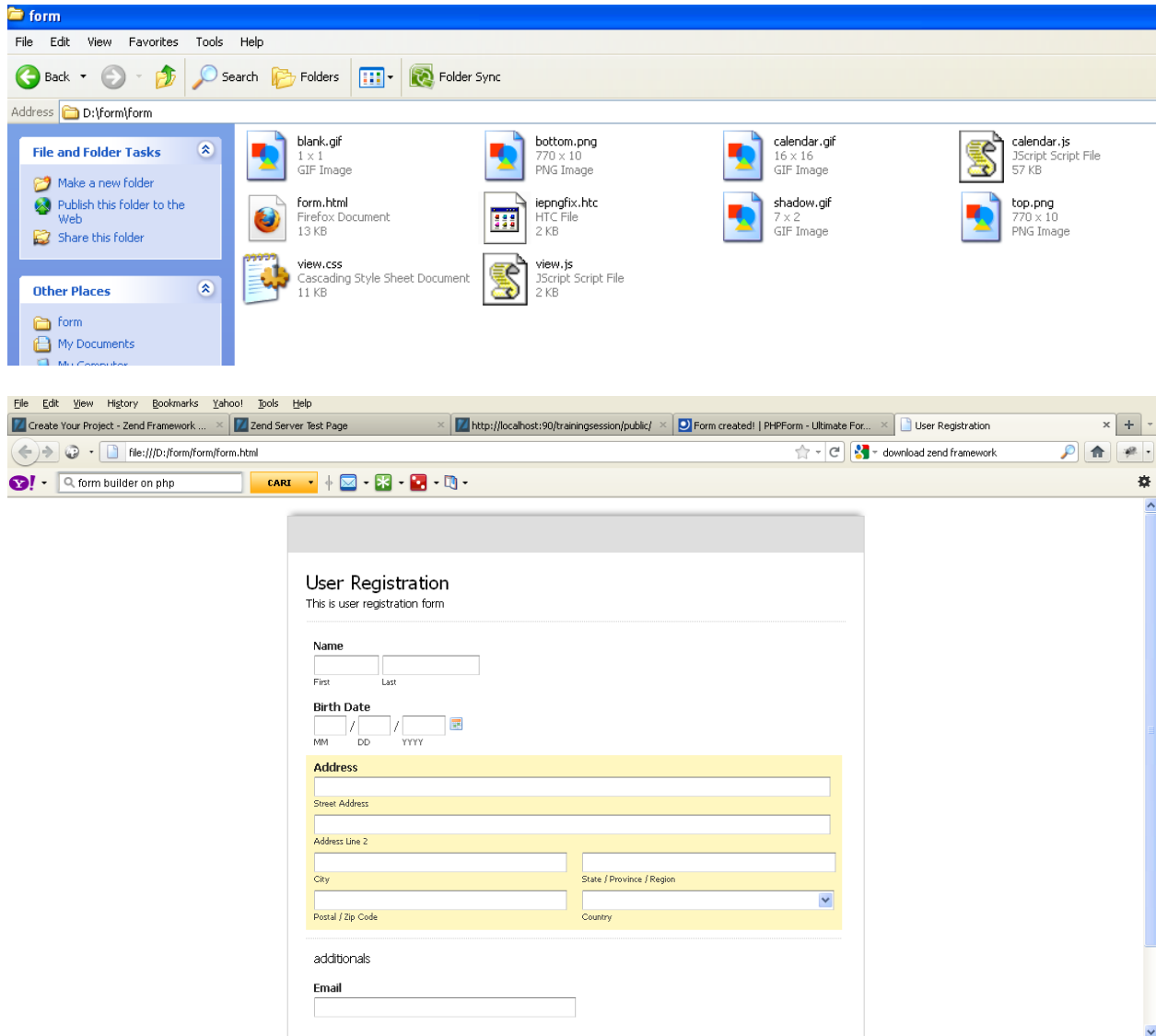
**Email**

**Add a Field** Field Properties Form Properties

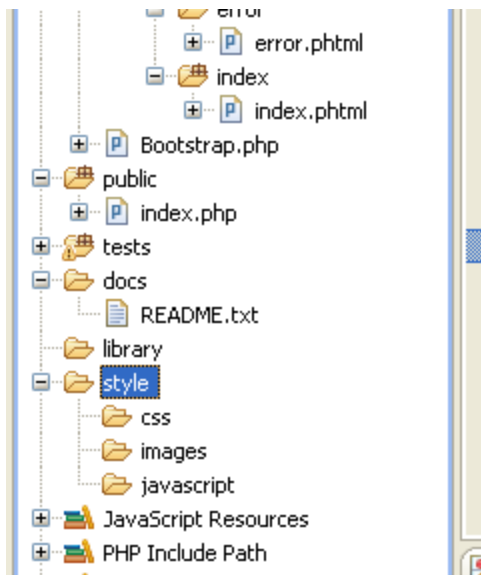
↓ Click to Add a Field

	Single Line Text		Number
	Paragraph Text		Checkboxes
	Multiple Choice		Drop Down
	Name		Date
	Time		Phone
	Address		Web Site
	Price		Email
	Section Break		File Upload

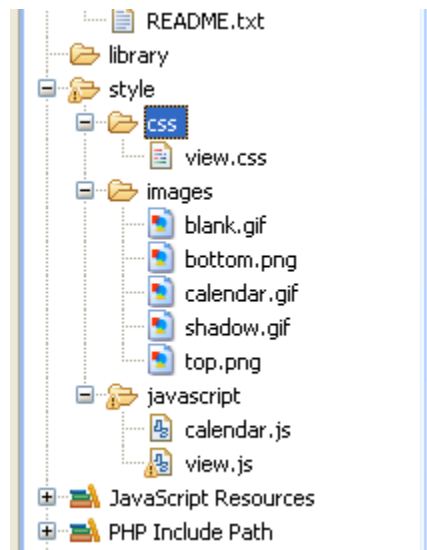
Then, you can click save and download the generated form files. Open folder, extract, and open .html file by clicked it twice.



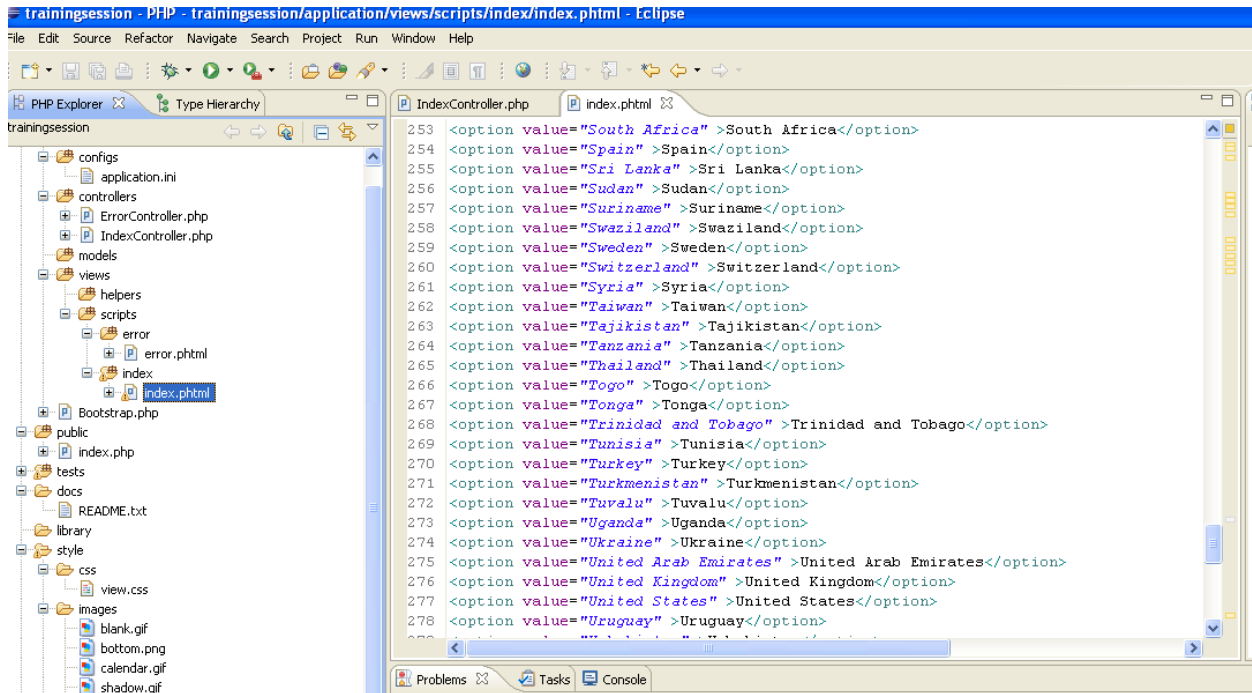
Now, open your eclipse again. Create folder style, javascript, images, and css as shown below.



Now, moved generated form code into associated folder. Move **.css** files into **css** folder, **.gif** and **.png** into **images** folder, and **.js** into **javascript** folder.



Now open, **form.html** (from generated code), select all, copy, and paste it on **index.phtml** in folder **application/views/scripts/index/**.



The screenshot shows the Eclipse IDE interface. On the left, the PHP Explorer shows a project named 'trainingsession' with a file tree including 'configs', 'controllers', 'models', 'views', 'public', 'tests', 'docs', 'library', and 'style'. The 'views/scripts/index' directory is expanded, showing 'error.phtml' and 'index.phtml'. The main editor displays the content of 'index.phtml', which is a list of country names enclosed in HTML option tags. The list includes: South Africa, Spain, Sri Lanka, Sudan, Suriname, Swaziland, Sweden, Switzerland, Syria, Taiwan, Tajikistan, Tanzania, Thailand, Togo, Tonga, Trinidad and Tobago, Tunisia, Turkey, Turkmenistan, Tuvalu, Uganda, Ukraine, United Arab Emirates, United Kingdom, United States, and Uruguay.

```
253 <option value="South Africa" >South Africa</option>
254 <option value="Spain" >Spain</option>
255 <option value="Sri Lanka" >Sri Lanka</option>
256 <option value="Sudan" >Sudan</option>
257 <option value="Suriname" >Suriname</option>
258 <option value="Swaziland" >Swaziland</option>
259 <option value="Sweden" >Sweden</option>
260 <option value="Switzerland" >Switzerland</option>
261 <option value="Syria" >Syria</option>
262 <option value="Taiwan" >Taiwan</option>
263 <option value="Tajikistan" >Tajikistan</option>
264 <option value="Tanzania" >Tanzania</option>
265 <option value="Thailand" >Thailand</option>
266 <option value="Togo" >Togo</option>
267 <option value="Tonga" >Tonga</option>
268 <option value="Trinidad and Tobago" >Trinidad and Tobago</option>
269 <option value="Tunisia" >Tunisia</option>
270 <option value="Turkey" >Turkey</option>
271 <option value="Turkmenistan" >Turkmenistan</option>
272 <option value="Tuvalu" >Tuvalu</option>
273 <option value="Uganda" >Uganda</option>
274 <option value="Ukraine" >Ukraine</option>
275 <option value="United Arab Emirates" >United Arab Emirates</option>
276 <option value="United Kingdom" >United Kingdom</option>
277 <option value="United States" >United States</option>
278 <option value="Uruguay" >Uruguay</option>
```

Now, refresh your browser on <http://localhost:90/trainingsession/public/>.



## User Registration

### User Registration

This is user registration form

- Name  First  Last
- Birth Date  / MM  / DD  YYYY Pick a date.
- Address
  - Street Address
  - Address Line 2
  - City
  - State / Province / Region
  - Postal / Zip Code
  - Country
- **additional**
- Email
- 

Generated by [pForm](#)

Well, now, we will configure the path of the css, javascript and the images. Open index.phtml on eclipse and change the following code.

### View.css

```
<link rel="stylesheet" type="text/css" href="/view.css" media="all">
```

into

```
<link rel="stylesheet" type="text/css" href="/trainingession/style/css/view.css" media="all">
```

### View.js

```
<script type="text/javascript" src="view.js"></script>
```

Into

```
<script type="text/javascript" src="/trainingession/style/javascript/view.js"></script>
```

## Calendar.js

```
<script type="text/javascript" src="calendar.js"></script>
```

Into

```
<script type="text/javascript"
src="/trainingsession/style/javascript/calendar.js"></script>
```

## Top.png

```

```

Into

```

```

## Calendar.gif

```

```

into

```

```

## Bottom.png

```

```

Into

```

```

Now, if you are done on changing the path, refresh your browser (<http://localhost:90/trainingsession/public/>).

The screenshot shows a web browser window with the title 'User Registration - Mozilla Firefox'. The address bar contains 'http://localhost:90/trainingession/public/'. The page content is a registration form with the following sections:

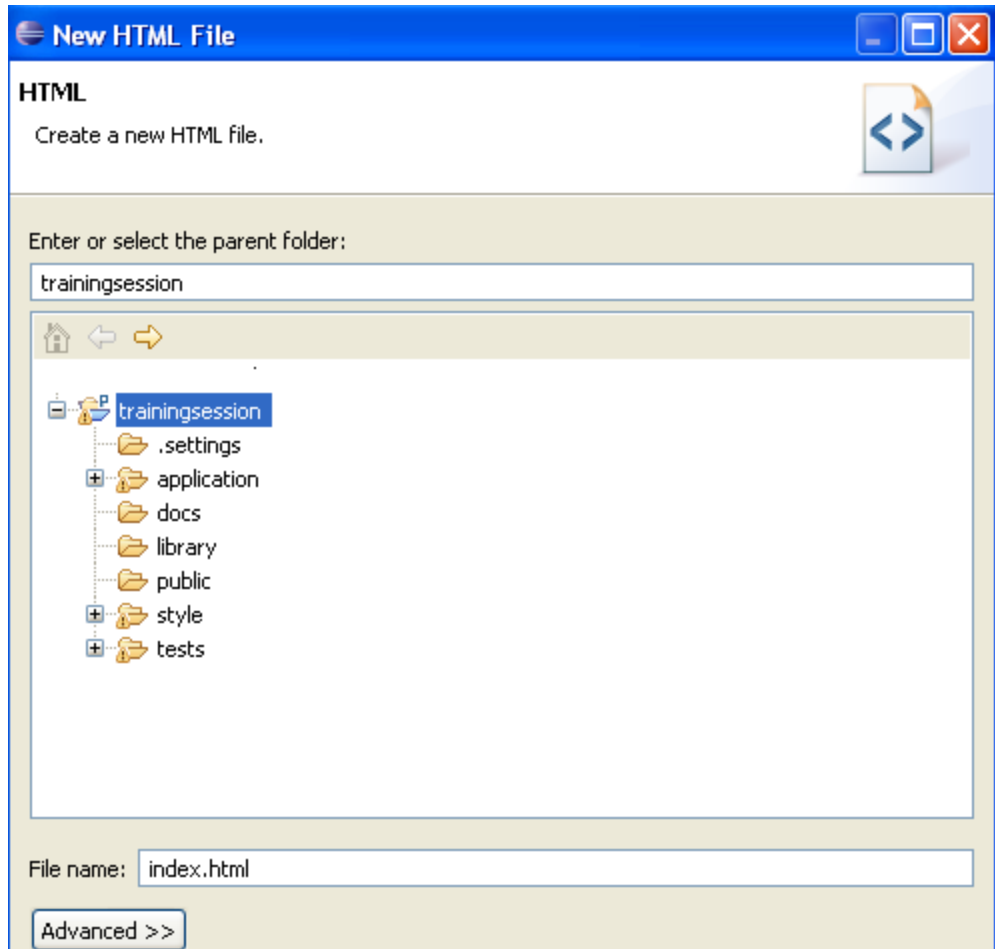
- User Registration**: This is user registration form
- Name**: Two input fields for 'First' and 'Last'.
- Birth Date**: Three input fields for 'MM', 'DD', and 'YYYY'.
- Address**: A series of input fields for 'Street Address', 'Address Line 2', 'City', 'State / Province / Region', 'Postal / Zip Code', and 'Country'.
- additional**: A section with a label 'additional' and an empty input field.
- Email**: A single input field for an email address.

Now, I would like to redirect to <http://localhost:90/trainingession/public> if users access <http://localhost:90/trainingession> since the main program is in there (/public). I just need to create index.html on <http://localhost:90/trainingession> and add the following code :

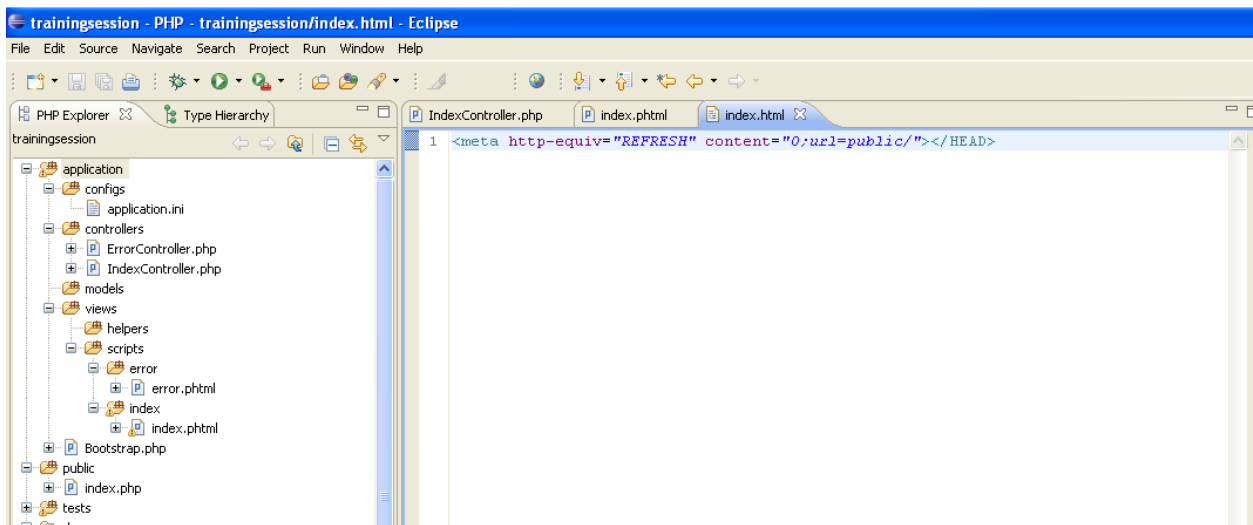
```
<meta http-equiv="REFRESH" content="0;url=public/"></HEAD>
```

The following steps are

1. Create index.html



2. Added the above code

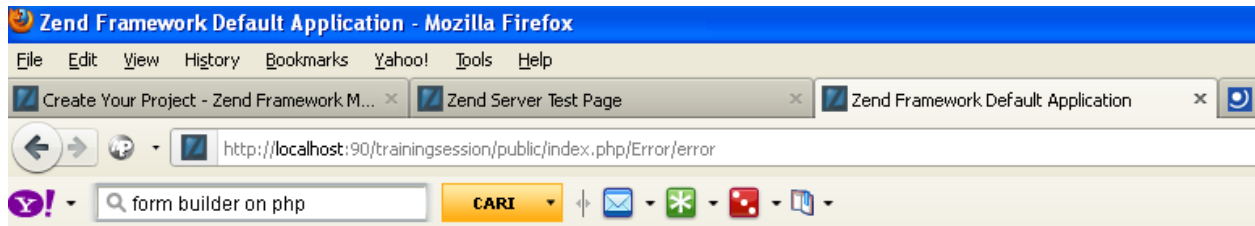


3. Try to refresh <http://localhost:90/trainingsession/>. Now it should be redirected into <http://localhost:90/trainingsession/public>.

i. Path configuration on Zend

Actually, while we browse <http://localhost:90/trainingession/public>, the zend first execute function `index()` on `IndexController`, and automatically, execute the view named `index.phtml` on folder `index` (same as the controller name, it's a must). This is why we can view directly without configured which controller and view to be rendered. This url is also same as <http://localhost:90/trainingession/public/index.php/Index/index>. the first "Index" is reference to `IndexController` and the second "index" is reference to `index` function.

Another example is <http://localhost:90/trainingession/public/index.php/Error/error> which is execute function `error()` on `ErrorController.php`.



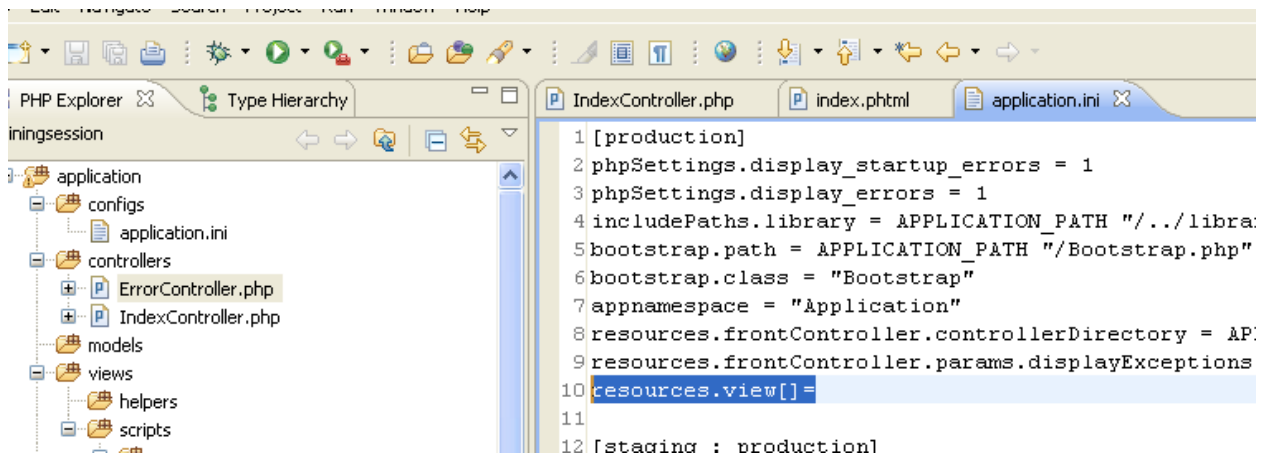
## An error occurred

You have reached the error page

ii. Clean Url Implementation

To remove **index.php** on path to accomplish clean url, just follow these steps :

1. On `config/application.ini`  
Add : `resources.view=[]`



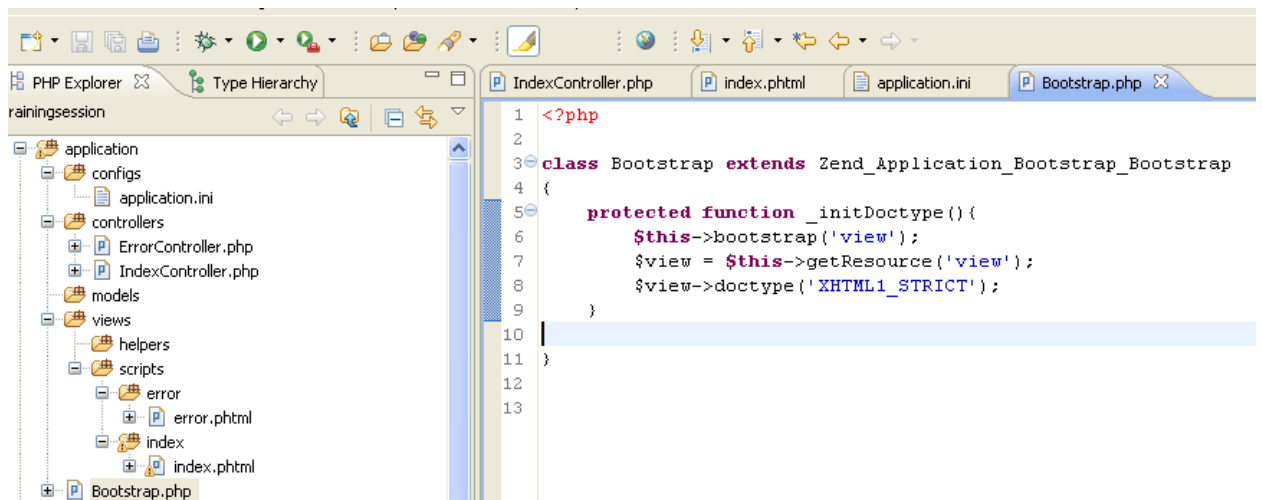
The screenshot shows the PHP Explorer interface with the 'application.ini' file open. The file content is as follows:

```

1 [production]
2 phpSettings.display_startup_errors = 1
3 phpSettings.display_errors = 1
4 includePaths.library = APPLICATION_PATH "/../libra:
5 bootstrap.path = APPLICATION_PATH "/Bootstrap.php"
6 bootstrap.class = "Bootstrap"
7 appnamespace = "Application"
8 resources.frontController.controllerDirectory = AP:
9 resources.frontController.params.displayExceptions
10 resources.view[] =
11
12 [staging : production]

```

2. In application/Bootstrap.php  
Add the following code :



The screenshot shows the PHP Explorer interface with the 'Bootstrap.php' file open. The code content is as follows:

```

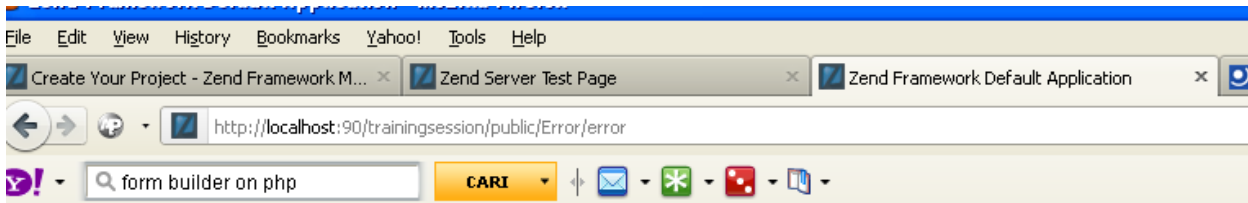
1 <?php
2
3 class Bootstrap extends Zend_Application_Bootstrap_Bootstrap
4 {
5     protected function _initDoctype() {
6         $this->bootstrap('view');
7         $view = $this->getResource('view');
8         $view->doctype('XHTML1_STRICT');
9     }
10
11 }
12
13

```

3. Edit httpd.conf

```
C:\Program Files\Zend\Apache2\conf\httpd.conf - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
form.js index.phtml SubmitController.php application.ini httpd.conf
157 # ServerName gives the name and port that the server uses to identify itself.
158 # This can often be determined automatically, but we recommend you specify
159 # it explicitly to prevent problems during startup.
160 #
161 # If your host doesn't have a registered DNS name, enter its IP address here.
162 #
163 #ServerName www.example.com: 90
164 #
165 # DocumentRoot: The directory out of which you will serve your
166 # documents. By default, all requests are taken from this directory, but
167 # symbolic links and aliases may be used to point to other locations.
168 #
169 DocumentRoot "C:\Program Files\Zend\Apache2\htdocs"
170
171 <VirtualHost *:90>
172     <Directory "C:\Program Files\Zend\Apache2\htdocs/trainingssession/public">
173         Options Indexes MultiViews FollowSymLinks
174         AllowOverride All
175         Order allow,deny
176         Allow from all
177     </Directory>
178     <Directory "C:\Program Files\Zend\Apache2\htdocs/trainingssessionextjs/public">
179         Options Indexes MultiViews FollowSymLinks
180         AllowOverride All
181         Order allow,deny
182         Allow from all
183     </Directory>
184 </VirtualHost>
```

4. Now, for access function error on ErrorController you can just type <http://localhost:90/trainingssession/public/Error/error>.



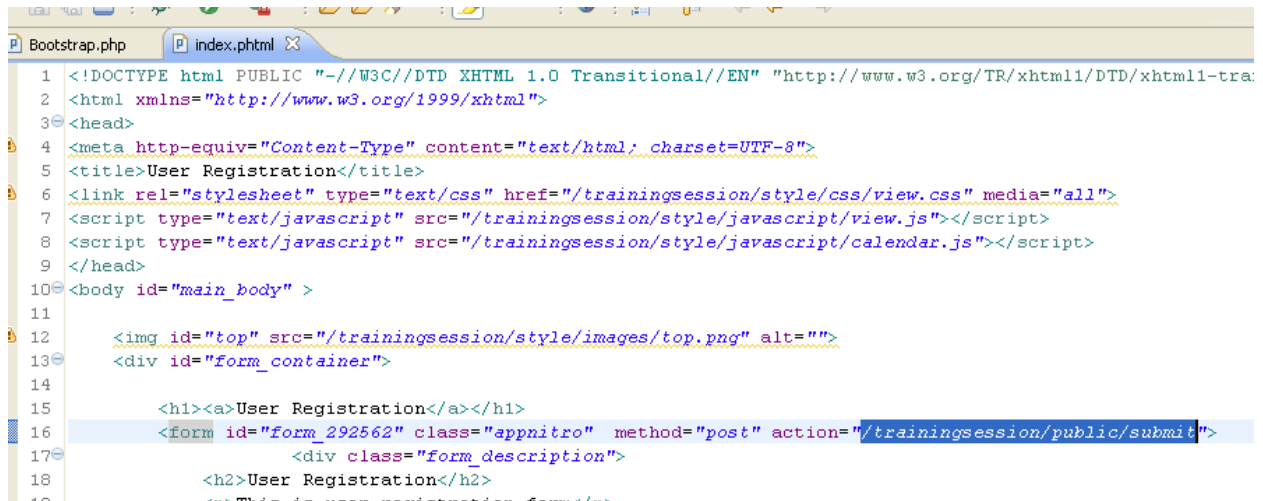
## An error occurred

You have reached the error page

Next, we will add **action** into the form when submit. So, we will need controller to handle the request.

These are the following steps :

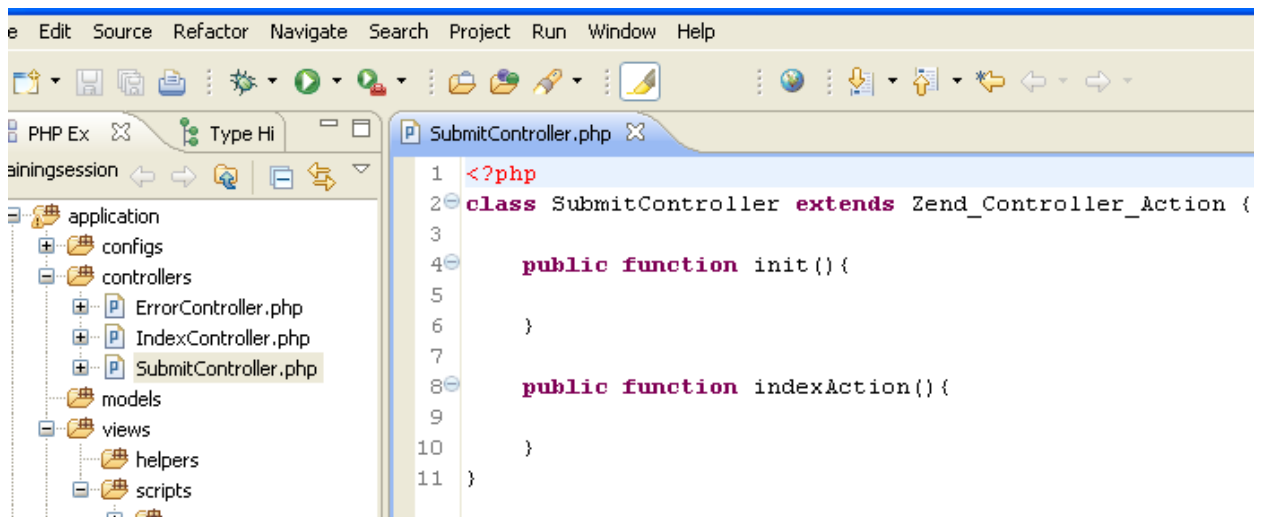
1. Add action url into the form. (/trainingsession/public/submit)



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-tra
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5 <title>User Registration</title>
6 <link rel="stylesheet" type="text/css" href="/trainingsession/style/css/view.css" media="all">
7 <script type="text/javascript" src="/trainingsession/style/javascript/view.js"></script>
8 <script type="text/javascript" src="/trainingsession/style/javascript/calendar.js"></script>
9 </head>
10 <body id="main_body" >
11
12 
13 <div id="form_container">
14
15 <h1><a>User Registration</a></h1>
16 <form id="form_292562" class="appnitro" method="post" action="/trainingsession/public/submit">
17 <div class="form_description">
18 <h2>User Registration</h2>
19 ...
```

This url means the submit button will post form data into SubmitController and default function which is function indexAction();. So, on controller folder, create new file which is named SubmitController

2. Create SubmitController.php and method/function indexAction()

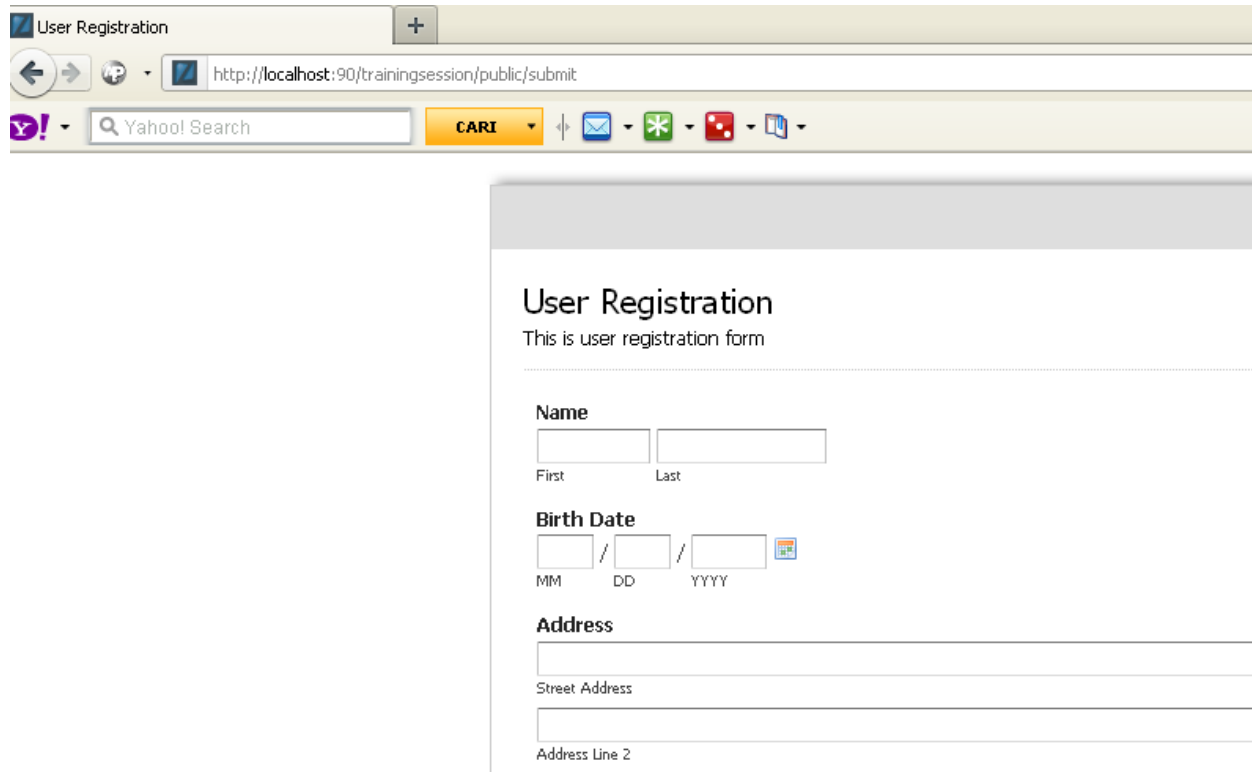


```
1 <?php
2 class SubmitController extends Zend_Controller_Action {
3
4     public function init() {
5
6     }
7
8     public function indexAction() {
9
10    }
11 }
```

3. Add a little source code like below.

```
SubmitController.php X
1 <?php
2 class SubmitController extends Zend_Controller_Action {
3
4     public function init(){
5
6     }
7
8     public function indexAction(){
9         //override default render path
10        $this->getHelper("viewRenderer")->setNoRender();
11        $view = new Zend_View();
12        $view->setScriptPath(APPLICATION_PATH."/views/scripts");
13
14
15
16        //render spesific file
17        echo $view->render("Index/index.phtml");
18    }
19 }
```

The first line of source code above shows that we wouldn't use default view render which is Submit/index.phtml. Instead, we will create new view (line 2) and setting the path of new View into /sharingsession/application/views/scripts/. On the last lines, we render View which is located in /sharingsession/application/views/scripts/Index/index.phtml.



The screenshot shows a web browser window with the title "User Registration". The address bar contains the URL "http://localhost:90/trainingession/public/submit". Below the browser window, a form titled "User Registration" is displayed. The form includes the following fields:

- Name:** Two input fields labeled "First" and "Last".
- Birth Date:** Three input fields labeled "MM", "DD", and "YYYY", with a calendar icon next to the "YYYY" field.
- Address:** Two input fields labeled "Street Address" and "Address Line 2".

Pay attentions that this is the same view as the form view before. Yes, because we simply tell zend to render `Index/index.phtml` which is actually the same file when we access it via url on `localhost:90/sharingession/public/Index/index`.

#### 4. Transfer data into View

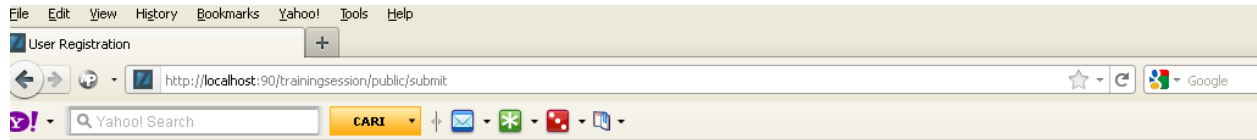
Now, we will create simple notification into the view. See the picture below :

```
SubmitController.php X index.phtml
1 <?php
2 class SubmitController extends Zend_Controller_Action {
3
4     public function init(){
5
6     }
7
8     public function indexAction(){
9         //override default renderer path
10        $this->getHelper("viewRenderer")->setNoRender();
11        $view = new Zend_View();
12        $view->setScriptPath(APPLICATION_PATH."/views/scripts");
13
14        $view->notification = "Data Successfully Saved on Database";
15
16        //render spesific file
17        echo $view->render("Index/index.phtml");
18    }
19 }
```

On the index.phtml, add the following code :

```
SubmitController.php X index.phtml X
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5 <title>User Registration</title>
6 <link rel="stylesheet" type="text/css" href="/trainingsession/style/css/view.css" media="all">
7 <script type="text/javascript" src="/trainingsession/style/javascript/view.js"></script>
8 <script type="text/javascript" src="/trainingsession/style/javascript/calendar.js"></script>
9 </head>
10 <body id="main_body" >
11
12     <?php
13         echo $this->notification;
14     ?>
15
16     
17     <div id="form_container">
18
19         <h1><a>User Registration</a></h1>
20         <form id="form_292562" class="appnitro" method="post" action="/trainingsession/public/
21             <div class="form_description">
22                 <h2>User Registration</h2>
```

Now, you can try to submit your form by click on submit button.



Data Successfully Saved on Database

**User Registration**  
This is user registration form

---

**Name**  
   
First Last

**Birth Date**  
 /  /    
MM DD YYYY

**Address**  
  
Street Address  
  
Address Line 2

## 5. Saving data into database

Well, I'll use MySQL Database to demonstrate this sub section. I'll assume you already knew about the definition of database and minimum knowledge about SQL. In MVC architecture, all the things which are directly connected with database should be put on model classes/model files. Now, try to create model file on folder `/models/` with filename "Register.php" like the following picture.

```

1 <?php
2 class Application_Model_Register{
3     public $db;
4
5     public function __construct(){
6         $config = new Zend_Config(
7             array(
8                 "database" => array(
9                     "adapter" => "Mysqli",
10                    "params" => array(
11                        "host" => "localhost",
12                        "dbname" => "training",
13                        "username" => "root",
14                        "password" => "",
15                        "port" => 3306
16                    )
17                )
18            );
19
20
21     $this->db = Zend_Db::factory($config->database);
22 }
23 }

```

The code above showed how to configure database connection. Notice that, you can manage more than one connection into some databases with simply create another model files. Now, add one function to submit data into table users. Here is the sample code.

```

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

6. After this, add the following code on the SubmitController class which is will be used for calling our model. Here is the following code :

```

}

public function indexAction(){
    //override default render path
    $this->getHelper("viewRenderer")->setNoRender();
    $view = new Zend_View();
    $view->setScriptPath(APPLICATION_PATH."/views/scripts");

    //insert data into database
    $name = $this->_request->getQuery("name");
    $address = $this->_request->getQuery("address");

    //create object, including creating connection into database
    $register = new Application_Model_Register();
    //call the createUser function
    $register->createUser($name, $address);

    //added variable into view
    $view->notification = "Data Successfully Saved on Database";

    //render specific file
    echo $view->render("Index/index.phtml");
}

```

7. Notice that, on the following code, we will get the request parameter such as name and address from form input. After get the values of this attributes, we'll simply create Application\_Model\_Register object, and calling createUser method with associated parameters. Now, in order to send the right data (such as name and address) into this controller, we'll change name attribute on the html tag (form view). Here is the picture which is described it.

a. Change method to get

```

<h1><a>User Registration</a></h1>
<form id="form_292562" class="appnitro" method="get" action="/trainingsession/public/submit">

```

b. Change attribute name into "name" from before "element\_1\_1".

```

<label class="description" for="element_1">Name </label>
<span>
    <input id="element_1_1" name="name" class="element text" maxlength="255" size="8" value=""/>
    <label>First</label>
</span>

```

c. Change attribute name into "address" from before "element\_2\_1"

```

<div>
<input id="element_2_1" name="address" class="element_text_large" value="" type="text">
<label for="element_2_1">Street Address</label>
</div>

```

- Now, you already create View, Controller, and Model to handle this form submit request. To be make sure, we need to test it, right ? So, firstly, I will create database in my MySQL RDBMS. I'll use MySQL which is bundled with XAMPP. You can download it if you want. This is the screenshot.

The screenshot shows the phpMyAdmin interface for a database named 'training'. The selected table is 'users'. The table structure is as follows:

#	Column	Type	Collation	Attributes	Null	Default	Extra	Action
1	name	varchar(20)	latin1_swedish_ci		No	None		Change Drop More
2	address	text	latin1_swedish_ci		No	None		Change Drop More

A message above the table indicates: "MySQL returned an empty result set (i.e. zero rows). ( Query took 0.0006 sec )". The SQL query shown is: "SELECT \* FROM `users` LIMIT 0, 30".

I will submit this form data.

## User Registration


This is user registration form

---

**Name**

First Last

**Birth Date**

/  / 

  
MM DD YYYY

**Address**

Street Address

And this following picture shows the result on my table.

The screenshot shows the phpMyAdmin interface for a database named 'training'. The 'users' table is selected. A query is executed: `SELECT * FROM `users` LIMIT 0, 30`. The result shows one row with the name 'Hafidh' and address 'Jalan Melati No 3'. The interface includes navigation buttons like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', and 'Import'. Below the query, there are options to show 30 rows starting from row # 0 in a horizontal view. A table with columns 'name' and 'address' is displayed below the query options.

	name	address
<input type="checkbox"/>	Hafidh	Jalan Melati No 3