**Experiment No. 7**

**Title:** Demonstrate use of codeIgniter for application development.

**Batch:B1 RollNo.:1514033 ExperimentNo.:7**

### Aim: Write a codeigniter application to display “Welcome to codeIgniter” by loading view page in index() of controller and “Development Framework is Interesting subject” by loading view page in myfunction() of controller. Make sure your application should not bypass index.php.

**Resources needed:**CodeIgniter, notepad

### Theory:

### CodeIgniter is an application development framework, which can be used to develop websites, using PHP. It is an Open Source framework. It has a very rich set of functionality, which will increase the speed of website development work.

### Installing CodeIgniter and creating application

### 1. Download CodeIgniter from its official website.

### https://www.codeigniter.com

### 2. Unzip CodeIgniter package: Downloaded CodeIgniter will be in zip format. Unzip it

### 3. Create folder namedCodeIgniter in C>xampp>htdocs

### 4. Copy content of unzip folder insideCodeIgniter folder.

### 5. Open browser and type: localhost/CodeIgniter : It will display the following default page

### 

### 6. Set the base URL in application/config/config.php file with any text editor

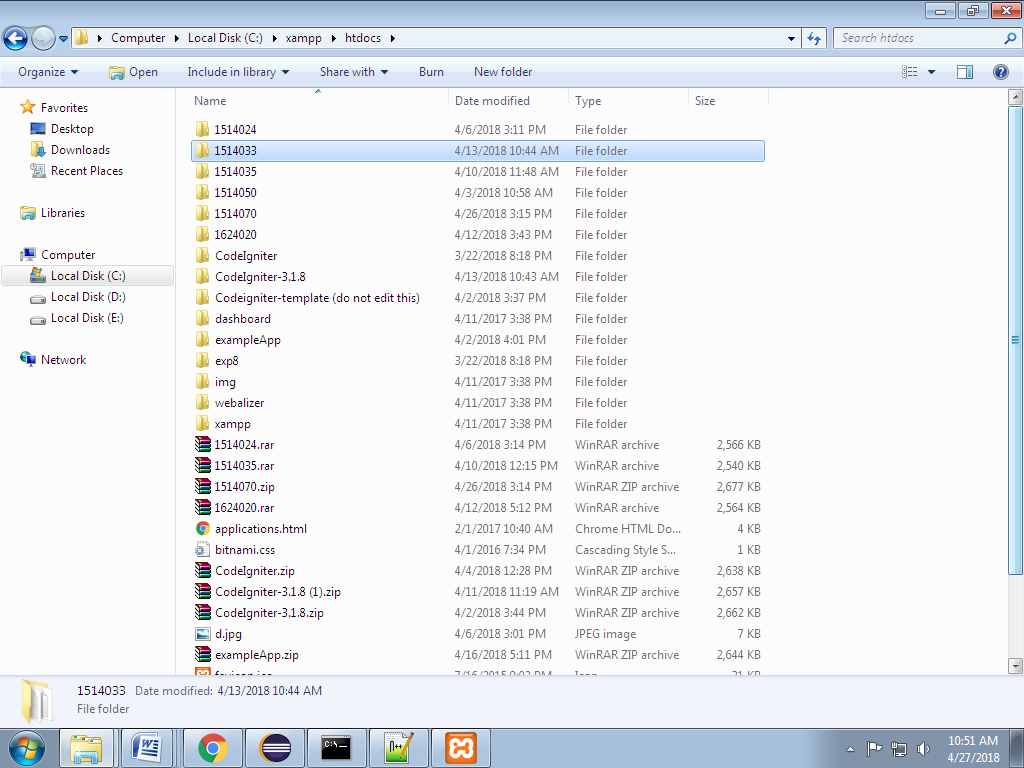
### Ex. $config['base\_url'] = 'http:localhost/CodeIgniter';

### 7. Create controller (.php) and save in C:\xampp\htdocs\CodeIgniter\application\controllers

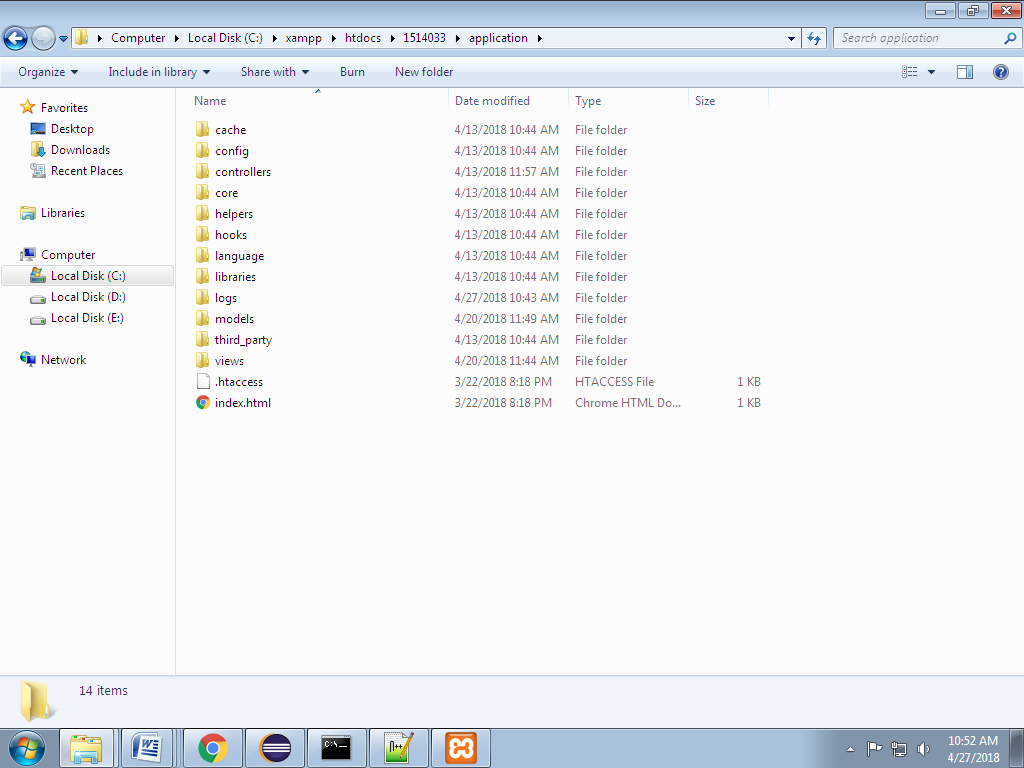
### 8. Create view (.php) and save in C:\xampp\htdocs\CodeIgniter\application\views

### Results: (Screen shots of application development steps, program code and web browser displaying the specified message.)

**XAMPP Folder:**

****

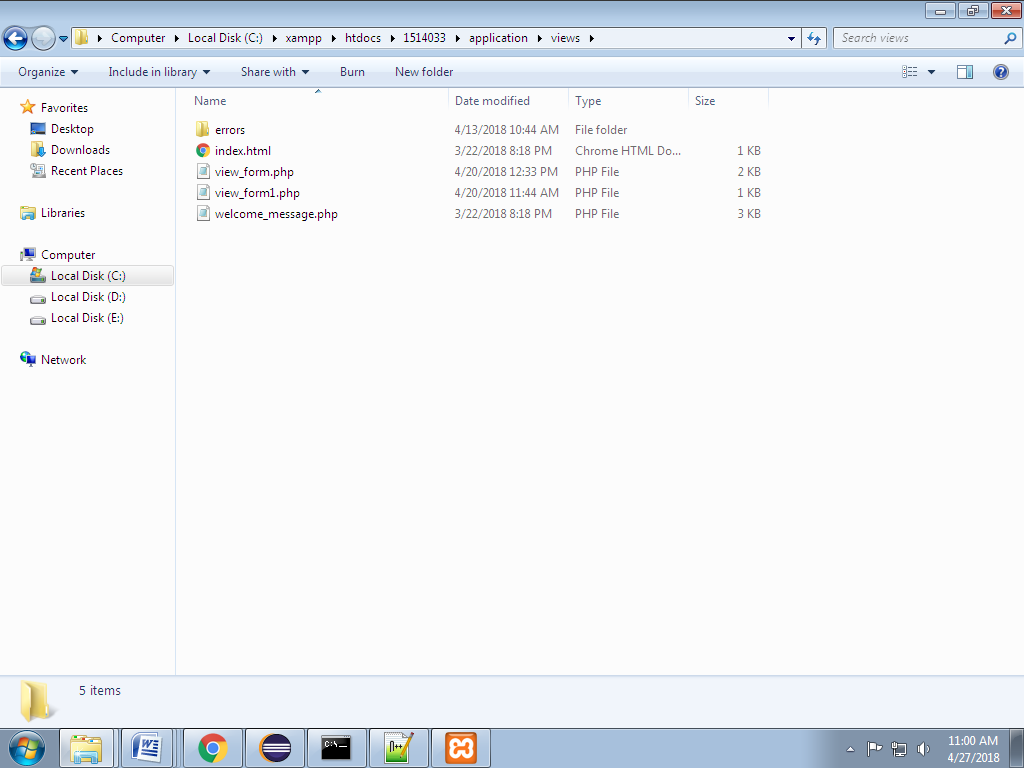
**Application folder :**

****

**Config:autoload.php**

$autoload['helper'] = array('form','html');

**View folder**

****

**View:view\_form.php**

**<html>**

**<head>**

**<title>Codeigniter Form Helper</title>**

**<meta name="robots" content="noindex, nofollow">**

**</head>**

**<body>**

**<?php**

**echo form\_open('register');**

**echo form\_label('Name :','emp\_name');**

**echo form\_error('emp\_name');**

**$data=array('id' => 'emp\_name', 'name' => 'emp\_name');**

**echo form\_input($data);**

**echo br();**

**echo form\_label('Email-ID','email\_id')."<br>";**

**$data\_email = array('type' => 'email','name' => 'email\_id','id' => 'email\_id','class' => 'input\_box','placeholder' => 'Please Enter Email');**

**echo form\_input($data\_email);**

**echo br();**

**echo form\_label('Password');**

**$data\_password = array('name' => 'password','id' => 'password','class' => 'input\_box','placeholder' => 'Please Enter Password');**

**echo form\_password($data\_password);**

**echo br();**

**echo form\_label('Gender');**

**$data\_gender = array('Male' => 'Male','Female' => 'Female');**

**echo form\_dropdown('select', $data\_gender, 'Male', 'class="dropdown\_box"');**

**echo br();**

**echo form\_reset('reset', 'Reset', "class='submit'");**

**echo form\_submit('submit', 'Submit', "class='submit'");**

**echo form\_close();**

**?>**

**</body>**

**</html>**

**View:view\_form1.php**

**<?php**

**echo "<div id='result\_show'>";**

**echo "<label class='label\_output'>Entered Employee Name : </label>" . $employee\_name."</br>";**

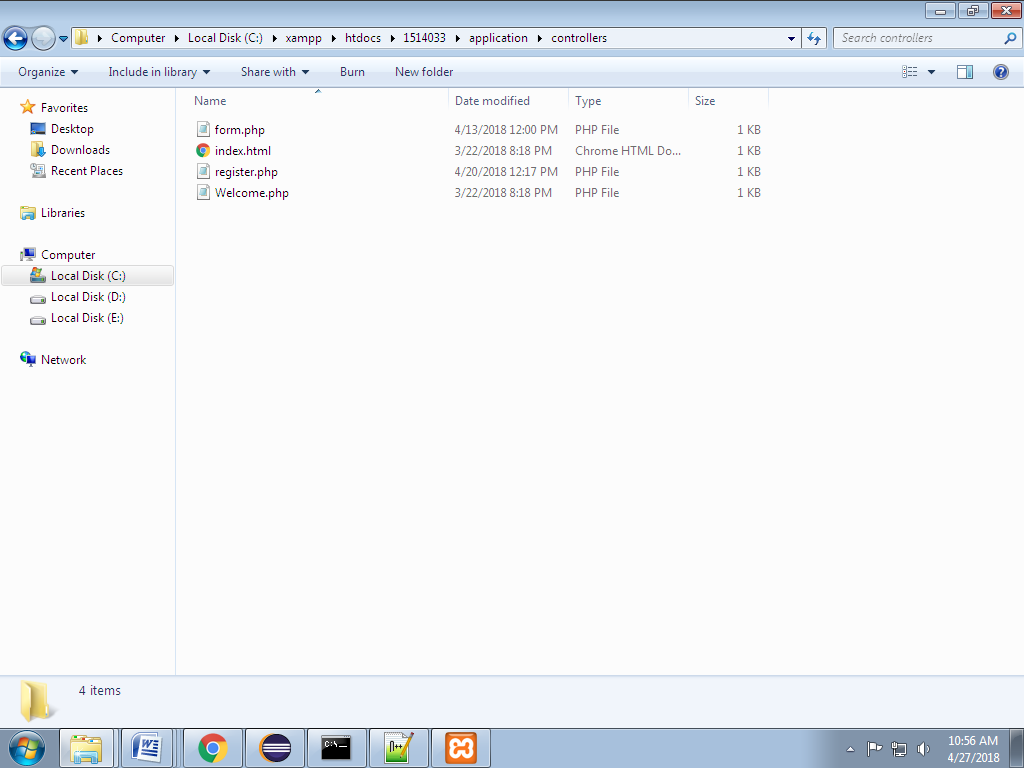
**echo "<label class='label\_output'>Entered Employee Email : </label>" . $employee\_email."<br>";**

**echo "<label class='label\_output'>Entered Password : </label>" . $employee\_password."<br>";**

**echo "<label class='label\_output'>Entered Gender : </label>" . $employee\_gender;**

**echo "</div>";**

**?>**

****

**controller:form.php**

**<?php**

**class form extends CI\_Controller {**

**public function index() {**

**// load view\_form.php present in views folder**

**$this->load->view("view\_form");**

**}**

**}**

**?>**

**Controller:register.php**

**<?php**

**class register extends CI\_Controller {**

**function \_\_construct() {**

**parent::\_\_construct();**

**$this->load->model('Insert\_model');**

**}**

**public function index() {**

**$this->load->library('form\_validation');**

**$this->form\_validation->set\_error\_delimiters('<div class="error">', '</div>');**

**$this->form\_validation->set\_rules('emp\_name', 'Name', 'required|min\_length[5]|max\_length[15]');**

**if ($this->form\_validation->run() == FALSE) {**

**$this->load->view('view\_form');**

**} else {**

**$data = array(**

**'employee\_name' => $this->input->post('emp\_name'),**

**'employee\_email' => $this->input->post('email\_id'),**

**'employee\_password' => $this->input->post('password'),**

**'employee\_gender' => $this->input->post('select')**

**);**

**$this->Insert\_model->form\_insert($data);**

**$data['message'] = 'Data Inserted Successfully';**

**$this->load->view("view\_form1", $data);**

**}**

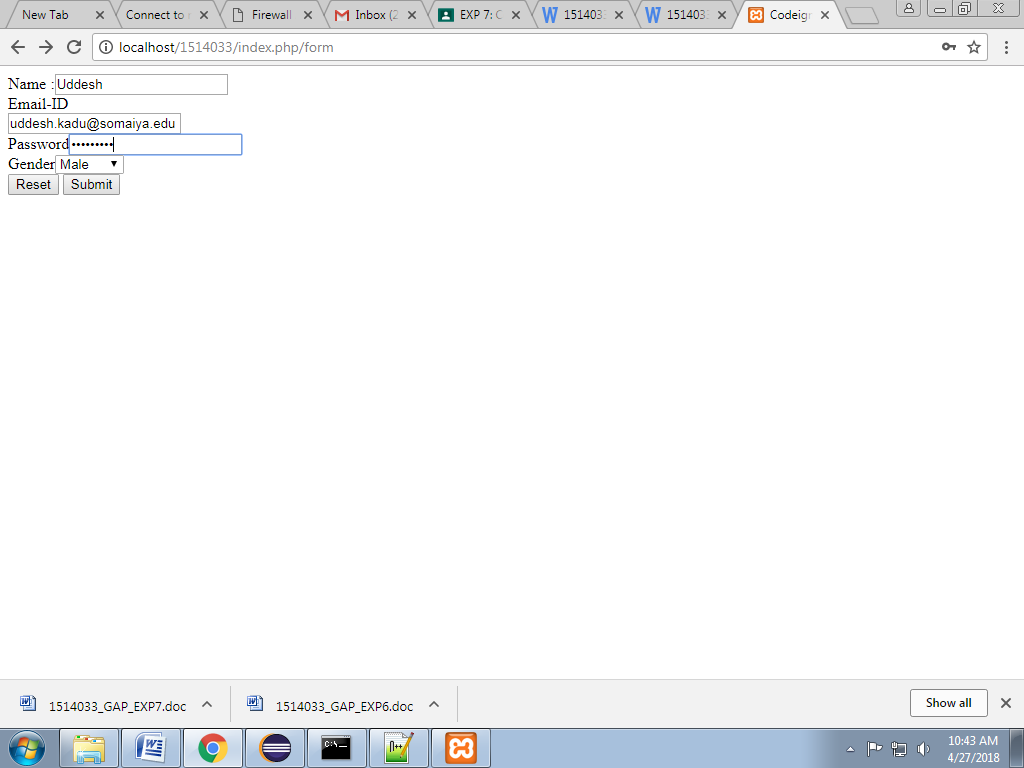
**}**

**}**

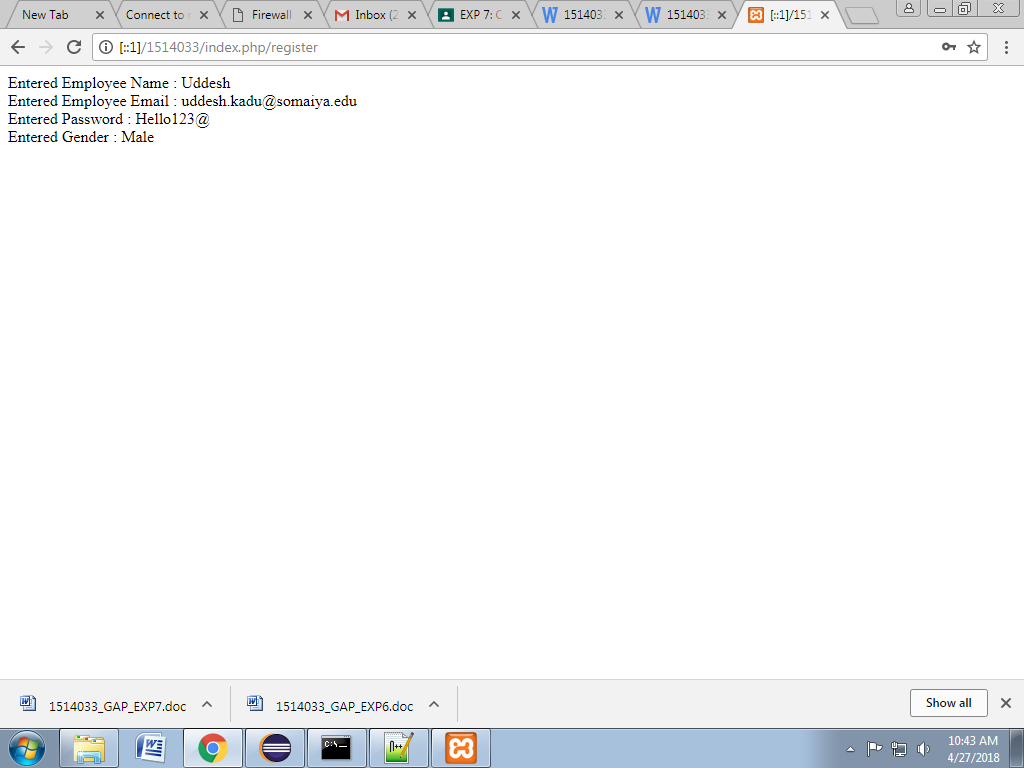
**?>**

**Output:-**

**form.php**

****

**On submit**

****

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Questions:**

1. **Explain features of CodeIgniter in detail.**

**Ans.**

# Features of CodeIgniter

There is a huge demand for CodeIgniter framework in PHP developers due to its versatile features and advantages. A web application developed on CodeIgniter performs effectively and rapidly. It provides an advanced set of aspects to write from scratch to build a dynamic web application.

## Important Features

* **Free to use**

It is licensed under MIT license, so it is free to use.

* **Follows MVC Pattern**

It uses Model-View-Controller which basically separates logic and presentation parts. Request comes to controller, database action is performed through model and output is displayed through views.

But in normal PHP scripting, every page represents MVC which increases complexity.

* **Light weight**

It is extremely light-weighted. CodeIgniter core system requires very small library, other libraries may be added upon dynamic request based upon your needs. That is why it is quite fast and light weighted.

* **Generate SEO friendly URLs**

URLs generated by CodeIgniter are search-engine friendly and clean. It uses a segment based approach rather than standard query based approach.

* **Built-in libraries**

It comes with full packet libraries that enable all the web needed tasks like database, form validation, sending email, manipulating images, sending emails, etc.

**Model-View-Controller Based System**

CodeIgniter make use of [**the MVC system**](http://customwebsitedevelopement.blogspot.com/2015/06/principles-of-mvc-for-php-developers.html) to find complex models and helps you to create application easily by using controllers and views. It assists the developer to build core libraries for the system and enables you to integrate your own existing scripts with database.

**Error Handling**

It offers simple user-friendly interfaces that help you to detect error functions throughout the application globally. In this way of approach, it provides instructions to error logging class that allows debugging messages to be saved as text files. It displays all PHP errors in your web application without missing inadequacy.

**Form Validation**

This feature helps you to write the code in a single line by using effective validation framework system. It generates codes without any errors and ensures various control structures to be placed within the HTML form.

**Migration**

Developers can manage database schema updates across various web application fields easily with helps of the migration aspect. Migration from one server to another server is simple and hassle-free in CodeIgniter.

**Configuration and Customizable**

This aspect helps the developers to create what they need to develop based on the web applications. You can edit the existing files in CodeIgniter easily and it does not confuse beginners to develop a new life. In addition, configuration and customizable of the files are easy in this frameworks.

**Benefits or Advantages of CodeIgniter**

It helps web developers to find out the errors in programming codes and fix-up the issues instantly in the web applications.

* It provides you user-friendly interface that help developers to create a dynamic, flexible, secure and large web applications effectively in a short span of time.
* Web developers can make use of this powerful framework to customize the configuration files easily and it provides hassle-free migration from source to destination service hosting.
* The framework helps the programmer to create both front-end and rear-end of a web application effectively and easily in a secured manner.
* Programmers can create web applications with additional features and high-end functionalities by using in-built resource and libraries of the CodeIgniter.

## Some other Features

* Security and XSS Filtering
* File uploading, session management, pagination, data encryption
* Flexible URI Routing
* Zip encoding class
* Error logging
* Full page caching
* Localization

### Outcomes:

### Demonstrate the use of frameworks in developing applications.

**Conclusion: (Conclusion to be based on the objectives and outcomesachieved)**

Thus the CodeIgnitior was successfully installed and basic program of printing form inputs was done successfully.

**Grade: AA / AB / BB / BC / CC / CD/DD**

**Signature of faculty in-charge withdate**

**References:**

1.By David Upton ;“Codeigniter for Rapid PHP Application Development”; Packt

Publishing, July 2007