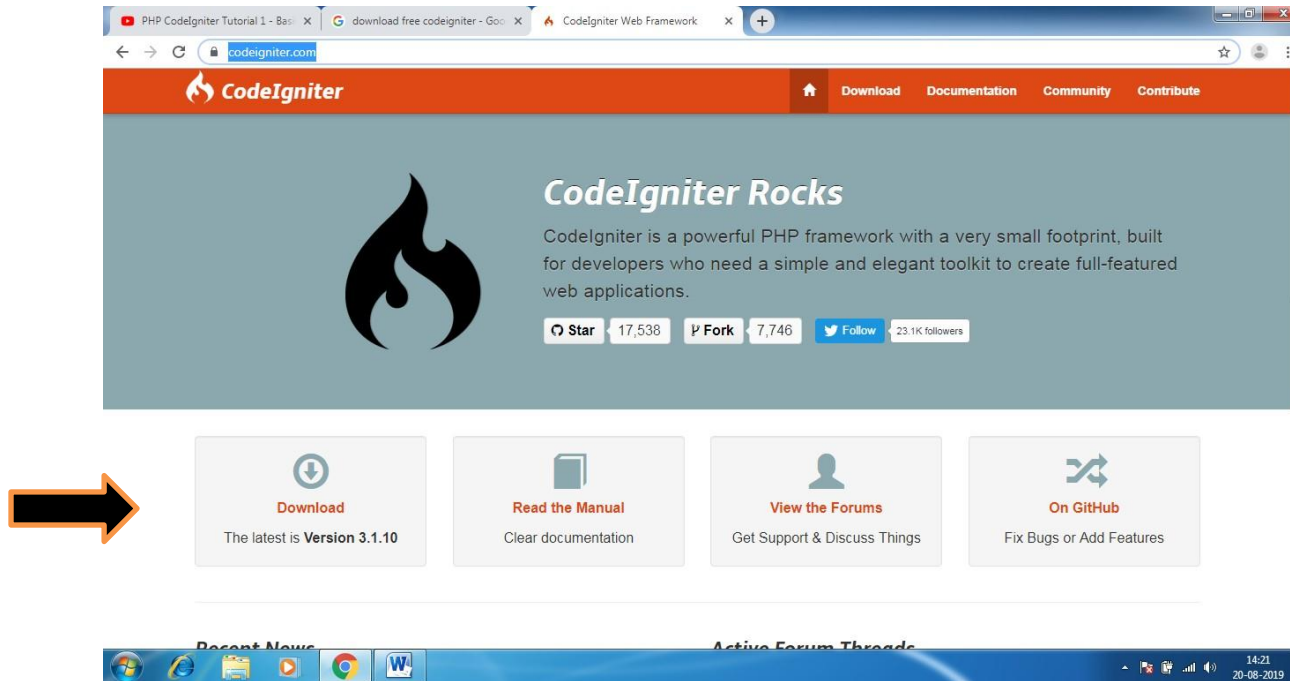


## Web development using CodeIgniter

### Installing CodeIgniter

Visit <https://www.codeigniter.com/> Download the latest version of CodeIgniter



Extract the zip file -> Open the folder -> Select application folder, system folder, index.php -> Copy them -> Create a folder (Say Test) in your www directory -> Paste them in this folder.

Start your web server -> Open browser and run this index.php (<http://localhost/Test/index.php>)

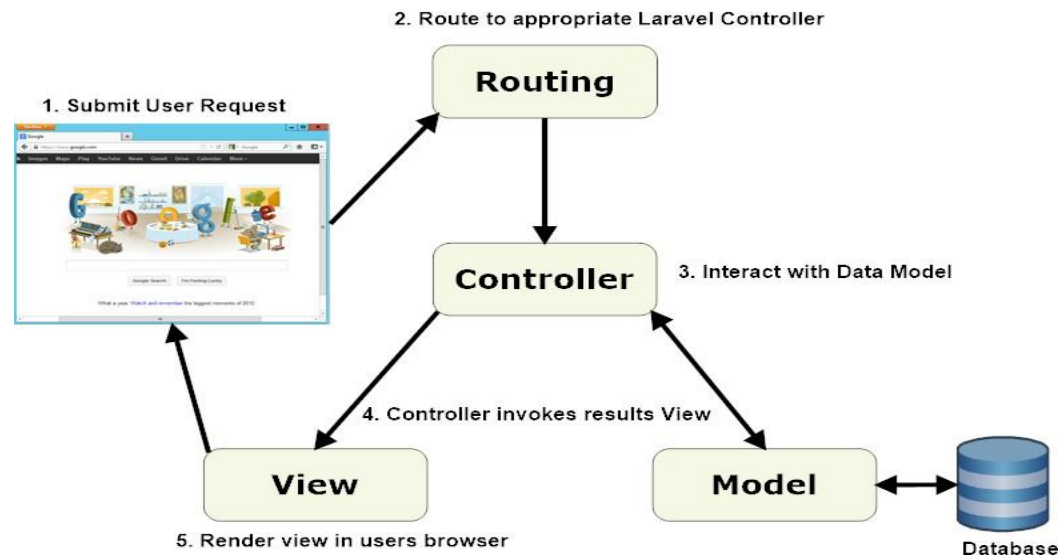
You will find **Welcome.php** under `C:\wamp\www\Test\application\controllers`

So just type the url <http://localhost/Test/index.php/Welcome> in your browser to run this file.

### CodeIgniter Overview

PHP MVC is an application design pattern that separates the application data and business logic (model) from the presentation (view). MVC stands for Model, View & Controller. The controller mediates between the models and views.

PHP MVC Frameworks increase the productivity in project development, this is because the base implementation of activities such as connecting to the database, sanitizing user input etc. are already partially implemented. It also hides all the complex implementation details.



1. **Model:** The Model deals with application related data. But the model does not deal with any logic about how to present the data.
2. **View:** The View element is used for presenting the data of the model to the user. This element deals with how to link up with the model's data but doesn't provide any logic regarding what this data all about or how users can use these data.
3. **Controller:** The Controller is in between the model and the view element. It listens to all the incident and actions triggered in the view and performs an appropriate response back to the events.

### Test your first Code

Open any IDE->Create a PHP file called MyController.php and save at <C:\wamp\www\Test\application\controllers> to test your code.

```
<?php
class MyController extends CI_Controller //MyController is the name of your controller
{
    public function index()
    {
        echo "Hello World.. This is my first controller";
    }
}
?>
```

To run the code, just type <http://localhost/Test/index.php/MyController> in your browser

Each controller class will extends CI\_Controller. Each time controller is called in the browser it always calls its `index()` function that is why we create it public, **public function index()**.

You may create as many functions you want but index will be called from browser by default.

## Web development using CodeIgniter

```
<?php
class MyController extends CI_Controller
{
    public function index()
    {
        echo "Hello World.. This is my first controller";
    }
    public function test()
    {
        echo "This is test function.";
    }
}
?>
```

Here, to call test function, simply type <http://localhost/Test/index.php/MyController/test>

The default function is index, you need not have to call index, it will be called itself each time you call the controller.

From the controller we do not send any information directly to the web pages. We call webpages from controller.

Now create a PHP (MyView.php) file in your **views** folder.

```
<?php
echo "This is our first view";
?>
```

## Web development using CodeIgniter

How to display this file in browser? So we are going to call MyView.php from controller.

```
<?php
class MyController extends CI_Controller
{
    public function index()
    {
        $this->load->view('MyView'); //call MyView.php without extension
    }
}
?>
```

We can do it from any other function but here we are calling it from index().

Now run in browser, type <http://localhost/Test/index.php/MyController>

The controller will load the file MyView.php

## Webdevelopment using CodeIgniter

### Simple Example of CodeIgniter-I

You can pass the data from the **controller** to the **view**. How do we do that?

#### *First\_Controller.php*

```
<?php
class First_Controller extends CI_Controller
{
    public function index()
    {
        $message['msg']="Hello Phoenix"; //Declaring a string variable with key-msg

        $this->load->view('First_View',$message); //Calling View with name of the variable not the key
    }
}
?>
```

#### *First\_View.php*

```
<!DOCTYPE html>
<head><title>Phoenix Training</title></head>
<body>
    <?php
        echo $msg;
    ?>
</body>
</html>
```

#### *Using route*

Open **config/routes.php** and make a new route for your url at the end of this file like this:

```
$route['default_controller']='First_Controller';
```

Save **routes.php**

Run in browser: [http://localhost/CI\\_1](http://localhost/CI_1)

**CI\_1** is your project folder name

Now refresh the browser and see the result. [http://localhost/CI\\_1/index.php/First\\_Controller](http://localhost/CI_1/index.php/First_Controller)

We have seen how to work with controller and view. Now we will see how to create model.

### *First\_Controller.php*

```
<?php
class First_Controller extends CI_Controller
{
    public function index()
    {
        $this->load->Model('First_Model'); //Loading the model first
        $message['msg']=$this->First_Model->Show("Hello Phoenix"); //Then calling the method of that Model
        $this->load->view('First_View',$message); //Calling View with an argument
    }
}
?>
```

### *First\_Model.php*

```
<?php
class First_Model extends CI_Model
{
    public function Show($str)
    {
        return $str." ". "We are testing CodeIgniter";
    }
}
?>
```

## Web development using CodeIgniter

### First\_View.php

```
<!DOCTYPE html>
<head><title>Phoenix Training</title></head>
<body>
    <?php
        echo $msg;
    ?>
</body>
</html>
```

In the above example we have simply used variable `$message` and key '`msg`'. While loading the view we are passing data variable using `$message`. We can access this variable `$message` in our view by the **key** of the variable, not by the name.

Now refresh the browser and see the result. [http://localhost/CI\\_1/index.php/First\\_Controller](http://localhost/CI_1/index.php/First_Controller)