

# CodeIgniter Rocks

CodeIgniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications.

<https://codeigniter.com/>

# Download CodeIgniter

---

- **wget** <https://github.com/bcit-ci/CodeIgniter/archive/3.1.6.zip>
- **git clone** <https://github.com/bcit-ci/CodeIgniter.git>

# Configure CodeIgniter

---

- Change the base URL in application/config/config.php

```
$config['base_url'] = '~/a999999/LAB9_10';
```

- Setup the database in application/config/database.php

```
$db['default'] = array(
    'dsn'      => '',
    'hostname' => '10.10.23.183',
    'username' => 'a999999',
    'password' => 'MyMySQLPassword',
    'database' => 'db_a999999',
    'dbdriver' => 'mysqli',
```

# Welcome to CodeIgniter !

---

- Point your browser to

[http://all.deei.fct.ualg.pt/~a999999/LAB9\\_10/index.php](http://all.deei.fct.ualg.pt/~a999999/LAB9_10/index.php)

Welcome to CodeIgniter!

The page you are looking at is being generated dynamically by CodeIgniter.

If you would like to edit this page you'll find it located at:

```
application/views/welcome_message.php
```

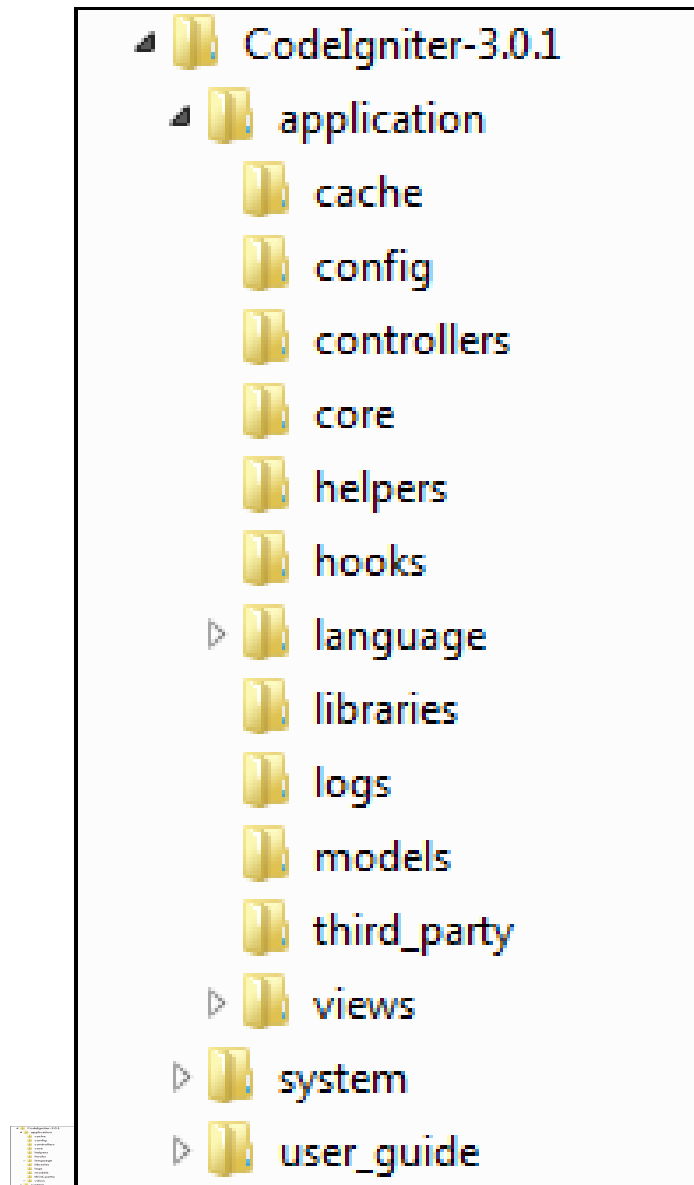
The corresponding controller for this page is found at:

```
application/controllers/Welcome.php
```

If you are exploring CodeIgniter for the very first time, you should start by reading the [User Guide](#).

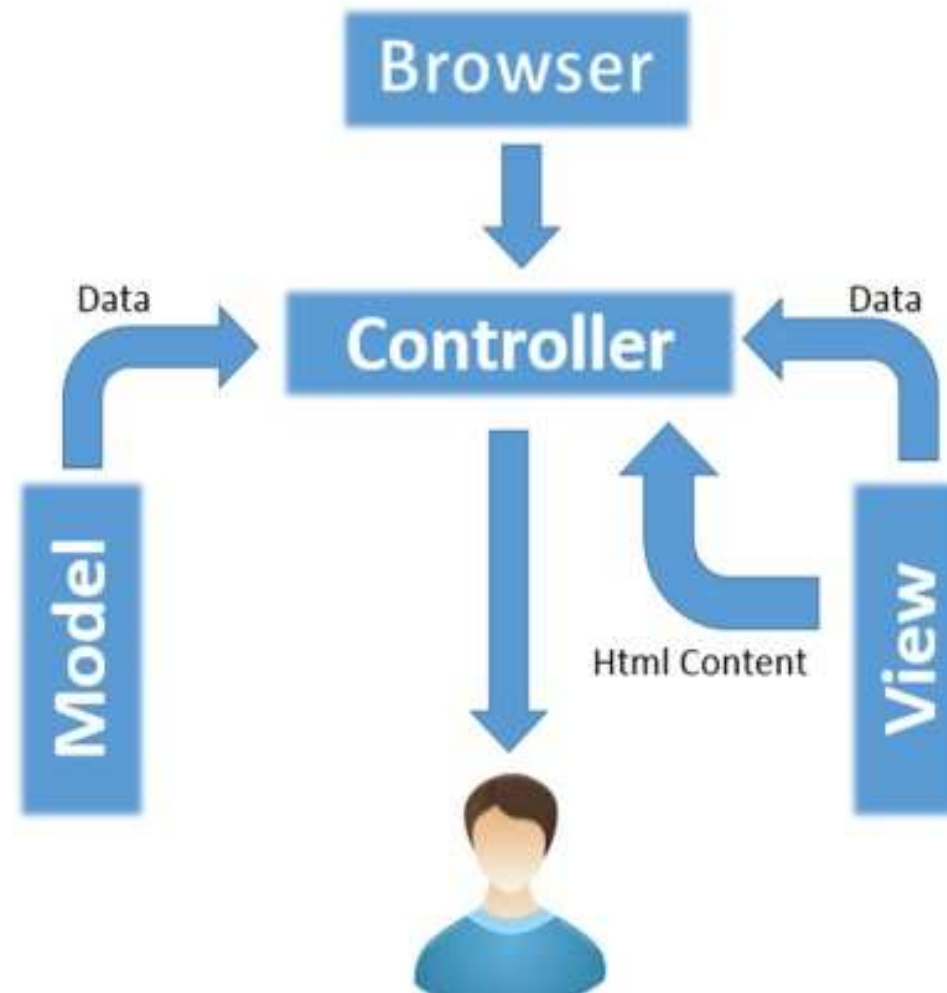
# Directory Structure

---



# MVC Paradigm

---



# MVC Paradigm ...

---

- **controllers** – This folder holds the controllers of the application.
- **models** – The database model will be placed in this folder.
- **views** – Application's HTML template files will be placed in this folder.

# The controller class

---

```
<?php
class Stud extends CI_Controller {

    function __construct() {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('Stud_model');
    }

    public function index() {

        $data['records'] = $this->Stud_model->show();
        $this->smarty->view('stud_view.tpl', $data ); //SMARTY
    }

    public function delete_student() {
        $roll_no = $this->uri->segment('3');
        $this->Stud_model->delete($roll_no);

        redirect('stud/index');
    }
}
?>
```



# The controller class ...

---

- **Points to Remember**

1. The PHP file name must start with an uppercase letter.
2. The name of the controller class must start with an uppercase letter.
3. The controller must be called with lowercase letter

- **The general syntax for calling the controller is**

**`http://www. domain.com/index.php/controller/method/arg`**

# URL Rewriting

---

**The segments in a URL normally follow this pattern**

```
www.domain.com/class/method/arg1/arg2/...
```

- **The first segment represents the controller class**
- **The second segment represents the class function (method)**
- **The third segment, and any additional segments, represent any variables that will be passed to the controller.**

# URL Rewriting ...

---

- URL `/~a9999/LAB9_10/index.php/stud`

gets method `index()` in file `Stud.php`

```
/users/a9999/public_html/LAB9_10/application/controllers/Stud.php
```

- URL `/~a9999/LAB9_10/index.php/stud/delete_student/89067`

gets method `“delete_student()”` in file `Stud.php`

```
/users/a9999/public_html/LAB9_10/application/controllers/Stud.php
```

... and passes value `“89067”` to the `delete_student()` method

# The model class

---

```
<?php
class Stud_model extends CI_Model {

    function __construct() {
        $this->load->database();
    }

    public function show()
    {
        $query = $this->db->get("stud");
        return $query->result(); //returns an object
    }

    public function get($roll_no)
    {
        $query = $this->db->get_where("stud",array("roll_no"=>$roll_no));
        return $query->row_array();
    }
}
?>
```

# The database db class ...

---

- `$query = $this->db->get("stud");`  
`return $query->result(); //returns an object !`  
`return $query->result_array(); //returns an array !`
- `$query = $this->db->get_where("stud",array("roll_no"=>$roll_no));`  
`return $query->row();`  
`return $query->row_array();`
- `return $this->db->insert("stud", $data)`
- `return $this->db->delete("stud", "roll_no = ".$roll_no)`
- `$this->db->set($data);`  
`$this->db->where("roll_no", $old_roll_no);`  
`return $this->db->update("stud", $data);`
- `$sql = "SELECT * FROM stud";`  
`$query = $this->db->query($sql);`  
`return $query->result_array();`

# The views (with SMARTY template)

---

```
<!DOCTYPE html>
<html lang = "en">
  <head>
    <meta charset = "utf-8">
    <title>Students Example</title>
  </head>
  <body>
    <a href = "{$base_url}studSmarty/add_student">Add</a>
    <table border = "1">
      <tr>
        <td>Sr#</td><td>Roll No.</td><td>Name</td> <td>Edit</td> <td>Delete</td>
      <tr>
        <td>
          {foreach $records as $r}
            <tr>
              <td>{$r.Name}</td>
              <td>{$r.roll_no}</td>
              <td>{$r.Name}</td>
              <td><a href =
                "{$base_url}index.php/stud/update_student_view/{$r.roll_no}">Edit</a></t
            <td><a href =
              "{$base_url}index.php/stud/delete_student/{$r.roll_no}">Delete</a></td>
            <tr>
          {/foreach}
        </td>
      </tr>
    </table>
  </body>
</html>
```

# The library classes

---

- [Calendaring Class](#)
- [Shopping Cart Class](#)
- [Config Class](#)
- [Email Class](#)
- [Encrypt Class](#)
- [Encryption Library](#)
- [File Uploading Class](#)
- [Form Validation](#)
- [FTP Class](#)
- [Image Manipulation Class](#)

- [Input Class](#)
- [Javascript Class](#)
- [Language Class](#)
- [Loader Class](#)
- [Output Class](#)
- [Pagination Class](#)
- [Template Parser Class](#)
- [Session Library](#)
- [HTML Table Class](#)
- [URI Class](#)
- [Zip Encoding Class](#)

# Form validation

---

## METHOD:

```
public function create()
{
    $this->load->library('form_validation');

    $this->form_validation->set_rules('title', 'Title', 'required');
    $this->form_validation->set_rules('text', 'Text', 'required');

    if ($this->form_validation->run() === FALSE)
    {
        $data['validation_errors']=validation_errors();
        $data['title']=set_value('title');
        $data['text']=set_value('text');
        $this->smarty->view('news/create', $data);
    }
    else
    {
        $this->news_model->set_news();
        $this->smarty->view('news/success');
    }
}
```



# Form validation ...

---

## TEMPLATE:

```
<h3>{$validation_errors}</h3>
```

```
<form action="news/create" method="post" >
```

```
  <label for="title">Title</label>
```

```
  <input type="input" name="title"
```

```
    value="{$title}"/>
```

```
<br />
```

```
  <label for="text">Text</label>
```

```
  <textarea name="text">
```

```
    {$text}
```

```
</textarea><br />
```

```
  <input type="submit" name="submit" value="Create news  
item" />
```

```
</form>
```

# Sessions

---

- **To initialize the Session class**

```
$this->load->library('session');
```

- **Retrieving session data**

```
$name = $_SESSION['item'] OR
```

```
$name = $this->session->item OR
```

```
$name = $this->session->userdata('item');
```

- **Assigning session data**

- 

```
$_SESSION['item'] = $name
```

```
$this->session->set_userdata('item', $name);
```

- **Unset session data**

```
unset($_SESSION['item']);
```

```
$this->session->unset_userdata('item');
```

- **Destroy session**

```
session_destroy();
```

```
$this->session->sess_destroy();
```

# The helper classes

---

- [Array Helper](#)
- [CAPTCHA Helper](#)
- [Cookie Helper](#)
- [Date Helper](#)
- [Directory Helper](#)
- [Download Helper](#)
- [File Helper](#)
- [Form Helper](#)
- [HTML Helper](#)
- [Inflector Helper](#)
- [Language Helper](#)
- [Number Helper](#)
- [Path Helper](#)
- [Security Helper](#)
- [Smiley Helper](#)
- [String Helper](#)
- [Text Helper](#)
- [Typography Helper](#)
- [URL Helper](#)
- [XML Helper](#)