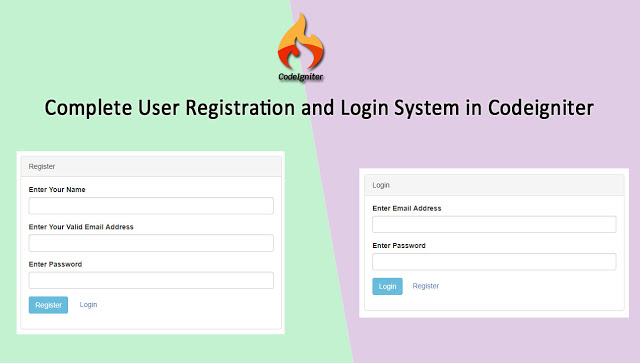
[**User Registration and Login System in Codeigniter 3**](https://www.webslesson.info/2018/10/user-registration-and-login-system-in-codeigniter-3.html)

User authentication is a required feature of any Web application. Based on user authentication user can access the system by providing their user authentication details. So, in this post we are login to learn Complete user registration and login system in Codeigniter framework. If you have use Codeigniter Framework to your web project, then in that project you want to add give rights to user to register into your system, after completed registration process, user can login into system by providing his login details which he has used in registration process. If login details are proper then user can access system. So here we will make complete user authentication system in Codeigniter in which user can registration into system and login into system features. Here you can learn this login and registration functionality in Codeigniter framework.  
  
If you are start learning Codeigniter framework for web devlopment, then you must have to learn User Register and Login system which is most required feature of any web based application, so you have to must learn this system in Codeigniter, so you can make web based application in Codeigniter in which you can add this User register and Login system. Codeigniter framework has many build-in libraries and helper class which will very helpful to make user authentication system. In this post we will use Codeigniter database library, session library, form validation library, email library for sending verification email address link, encrypt library for convert simple password into hash formatted password. We will also use Codeigniter url helper, form helper for make this user register and login system in Codeigniter framework.  
  
Here we will make user registration and login system using Session in Codeigniter framework. In this post you can find step by step process for make user registration and login system in Codeigniter with Session library with Mysql database. In this system you can find following features of User Registration and Login system using Codeigniter framework with Mysql database.

**See Also**

* [**Codeigniter - Detect User IP Browser Operating System Details**](https://www.webslesson.info/2018/09/codeigniter-detect-user-ip-browser-operating-system-details.html)
* [**Inline Table CRUD in Codeigniter using Ajax jquery**](https://www.webslesson.info/2018/09/inline-table-crud-in-codeigniter-using-ajax-jquery.html)
* [**Codeigniter - Load More Data on page scroll using Ajax**](https://www.webslesson.info/2018/08/codeigniter-load-more-data-on-page-scroll-using-ajax.html)
* [**Autocomplete Search Box using Typeahead in Codeigniter**](https://www.webslesson.info/2018/07/autocomplete-search-box-using-typeahead-in-codeigniter.html)
* [**Codeigniter Dynamic Dependent Select Box using Ajax**](https://www.webslesson.info/2018/06/codeigniter-dynamic-dependent-select-box-using-ajax.html)

* User Registration form to submit user details and Insert into Mysql Database
* Send email verification email to registered email address for email verification
* Convert user simple password to hash string for user authentication
* Register and Login form data validation
* Check email already registered or not at the time of user registration
* User login form for access into system
* Store user authentication details in Session
* User logout from system

[](https://1.bp.blogspot.com/-_lkMX7UAgkc/W9gZV_UerQI/AAAAAAAAA9o/ExKylU53hBYFm5AeDogXyu70HukgKqaSwCLcBGAs/s1600/complete-login-register-system-in-codeigniter.jpg)

**Database table creation**

For store user information, we have to required table in mysql table, following script will make table in mysql database.

--

-- Database: `testing`

--

-- --------------------------------------------------------

--

-- Table structure for table `codeigniter\_register`

--

CREATE TABLE `codeigniter\_register` (

`id` int(11) NOT NULL,

`name` varchar(250) NOT NULL,

`email` varchar(250) NOT NULL,

`password` text NOT NULL,

`verification\_key` varchar(250) NOT NULL,

`is\_email\_verified` enum('no','yes') NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Indexes for dumped tables

--

--

-- Indexes for table `codeigniter\_register`

--

ALTER TABLE `codeigniter\_register`

ADD PRIMARY KEY (`id`);

--

-- AUTO\_INCREMENT for dumped tables

--

--

-- AUTO\_INCREMENT for table `codeigniter\_register`

--

ALTER TABLE `codeigniter\_register`

MODIFY `id` int(11) NOT NULL AUTO\_INCREMENT;

**Autoload Libraries & Helper**

In User Register and Login system in Codeigniter, we have to load some libraries and helper at the time of first code execution. For this we have to open **application/cofig/autoload.php** file and load following files and helper.  
  
**autoload.php** 

$autoload['libraries'] = array('session','database');

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

[View Demo](http://demo.webslesson.info/codeigniter-demo/register)

**Database Connection**

After load of required library and helper, we want to make database connection, for this we have to open **application/config/database.php** file and write following database configuration.  
  
**database.php** 

<?php

$active\_group = 'default';

$query\_builder = TRUE;

$db['default'] = array(

'dsn' => '',

'hostname' => 'localhost',

'username' => 'root',

'password' => '',

'database' => 'testing',

'dbdriver' => 'mysqli',

'dbprefix' => '',

'pconnect' => FALSE,

'db\_debug' => (ENVIRONMENT !== 'production'),

'cache\_on' => FALSE,

'cachedir' => '',

'char\_set' => 'utf8',

'dbcollat' => 'utf8\_general\_ci',

'swap\_pre' => '',

'encrypt' => FALSE,

'compress' => FALSE,

'stricton' => FALSE,

'failover' => array(),

'save\_queries' => TRUE

);

?>

**Controllers (Register.php)**

This controller is used for handles all register replated functionality.  
**\_construct() -**This function is used for load form validation and encrypt library, register model at the time of new object of this class has been created.  
**index() -**This function will load register form in browser.  
**validation() -**This function has received register form request, in this function first we have validate form data using Codeigniter form validation library. After success validation of form data, it will proceed for insert data into mysql table. After successfully insert of register form data, this function will generate dynamic email verification email with dynamic email verification link and send email to registered email address using Codeigniter email library.  
**verify\_email() -**This function is used for verify email address, this function received request for email verification from email verification link. Once email has been verified user can login into system.

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Register extends CI\_Controller {

public function \_\_construct()

{

parent::\_\_construct();

if($this->session->userdata('id'))

{

redirect('private\_area');

}

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

$this->load->library('encrypt');

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

}

function index()

{

$this->load->view('register');

}

function validation()

{

$this->form\_validation->set\_rules('user\_name', 'Name', 'required|trim');

$this->form\_validation->set\_rules('user\_email', 'Email Address', 'required|trim|valid\_email|is\_unique[codeigniter\_register.email]');

$this->form\_validation->set\_rules('user\_password', 'Password', 'required');

if($this->form\_validation->run())

{

$verification\_key = md5(rand());

$encrypted\_password = $this->encrypt->encode($this->input->post('user\_password'));

$data = array(

'name' => $this->input->post('user\_name'),

'email' => $this->input->post('user\_email'),

'password' => $encrypted\_password,

'verification\_key' => $verification\_key

);

$id = $this->register\_model->insert($data);

if($id > 0)

{

$subject = "Please verify email for login";

$message = "

<p>Hi ".$this->input->post('user\_name')."</p>

<p>This is email verification mail from Codeigniter Login Register system. For complete registration process and login into system. First you want to verify you email by click this <a href='".base\_url()."register/verify\_email/".$verification\_key."'>link</a>.</p>

<p>Once you click this link your email will be verified and you can login into system.</p>

<p>Thanks,</p>

";

$config = array(

'protocol' => 'smtp',

'smtp\_host' => 'smtpout.secureserver.net',

'smtp\_port' => 80,

'smtp\_user' => 'xxxxxxx',

'smtp\_pass' => 'xxxxxxx',

'mailtype' => 'html',

'charset' => 'iso-8859-1',

'wordwrap' => TRUE

);

$this->load->library('email', $config);

$this->email->set\_newline("\r\n");

$this->email->from('info@webslesson.info');

$this->email->to($this->input->post('user\_email'));

$this->email->subject($subject);

$this->email->message($message);

if($this->email->send())

{

$this->session->set\_flashdata('message', 'Check in your email for email verification mail');

redirect('register');

}

}

}

else

{

$this->index();

}

}

function verify\_email()

{

if($this->uri->segment(3))

{

$verification\_key = $this->uri->segment(3);

if($this->register\_model->verify\_email($verification\_key))

{

$data['message'] = '<h1 align="center">Your Email has been successfully verified, now you can login from <a href="'.base\_url().'login">here</a></h1>';

}

else

{

$data['message'] = '<h1 align="center">Invalid Link</h1>';

}

$this->load->view('email\_verification', $data);

}

}

}

?>

**Controllers(Login.php)**

In this Login controller you can find following function for all Login operation like, load login form, login form data verification.  
  
**\_construct() -**This function is used for load form validation and encrypt library, login model at the time of new object of this class has been created.  
**index() -**This function will load Login form on web page.  
**validation() -**This function received request for validate login details. After successfully verify user login details, page will redirect to private\_area controller.

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Login extends CI\_Controller {

public function \_\_construct()

{

parent::\_\_construct();

if($this->session->userdata('id'))

{

redirect('private\_area');

}

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

$this->load->library('encrypt');

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

}

function index()

{

$this->load->view('login');

}

function validation()

{

$this->form\_validation->set\_rules('user\_email', 'Email Address', 'required|trim|valid\_email');

$this->form\_validation->set\_rules('user\_password', 'Password', 'required');

if($this->form\_validation->run())

{

$result = $this->login\_model->can\_login($this->input->post('user\_email'), $this->input->post('user\_password'));

if($result == '')

{

redirect('private\_area');

}

else

{

$this->session->set\_flashdata('message',$result);

redirect('login');

}

}

else

{

$this->index();

}

}

}

?>

**Controller(Private\_area.php)**

After successfully validate login details page has been redirect to this controller. In this there following function.  
**\_construct() -**This function is used validate used login or not into system, if user not login into system then this function will redirect page to login form.  
**index() -**This function will display Welcome user message with Logout page link.  
**logout() -**This function will remove all session value, this function will logout user from system and redirect to login form.

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Private\_area extends CI\_Controller {

public function \_\_construct()

{

parent::\_\_construct();

if(!$this->session->userdata('id'))

{

redirect('login');

}

}

function index()

{

echo '<br /><br /><br /><h1 align="center">Welcome User</h1>';

echo '<p align="center"><a href="'.base\_url().'private\_area/logout">Logout</a></p>';

}

function logout()

{

$data = $this->session->all\_userdata();

foreach($data as $row => $rows\_value)

{

$this->session->unset\_userdata($row);

}

redirect('login');

}

}

?>

**Models(Register\_model.php)**

**insert() -**Models function is used for database operation, So this function will insert data into Mysql table.  
**verify\_email($key) -**This model function has been used for database operation for verify email address.

<?php

class Register\_model extends CI\_Model

{

function insert($data)

{

$this->db->insert('codeigniter\_register', $data);

return $this->db->insert\_id();

}

function verify\_email($key)

{

$this->db->where('verification\_key', $key);

$this->db->where('is\_email\_verified', 'no');

$query = $this->db->get('codeigniter\_register');

if($query->num\_rows() > 0)

{

$data = array(

'is\_email\_verified' => 'yes'

);

$this->db->where('verification\_key', $key);

$this->db->update('codeigniter\_register', $data);

return true;

}

else

{

return false;

}

}

}

?>

**Models(Login\_model.php)**

In this model there is only one function can\_login($email, $password), this function is used to verify login form details with Mysql table details, If login details match then it will return true, otherwise it will return error message. This function convert hash password to simple string and match with Login form password.

<?php

class Login\_model extends CI\_Model

{

function can\_login($email, $password)

{

$this->db->where('email', $email);

$query = $this->db->get('codeigniter\_register');

if($query->num\_rows() > 0)

{

foreach($query->result() as $row)

{

if($row->is\_email\_verified == 'yes')

{

$store\_password = $this->encrypt->decode($row->password);

if($password == $store\_password)

{

$this->session->set\_userdata('id', $row->id);

}

else

{

return 'Wrong Password';

}

}

else

{

return 'First verified your email address';

}

}

}

else

{

return 'Wrong Email Address';

}

}

}

?>

**Views(register.php)**

This html output file, which output we view on web page in browser. It will load register form on web page. In this file we have make register form, in which we have use **set\_value()** form helper function, by using this function it can remember form data on form again load on validation error. Same way we have also use **form\_error()** form validation library function for display form validation error on web page. Below you can find complete source code of registration form.

<!DOCTYPE html>

<html>

<head>

<title>Complete User Registration and Login System in Codeigniter</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />

</head>

<body>

<div class="container">

<br />

<h3 align="center">Complete User Registration and Login System in Codeigniter</h3>

<br />

<div class="panel panel-default">

<div class="panel-heading">Register</div>

<div class="panel-body">

<form method="post" action="<?php echo base\_url(); ?>register/validation">

<div class="form-group">

<label>Enter Your Name</label>

<input type="text" name="user\_name" class="form-control" value="<?php echo set\_value('user\_name'); ?>" />

<span class="text-danger"><?php echo form\_error('user\_name'); ?></span>

</div>

<div class="form-group">

<label>Enter Your Valid Email Address</label>

<input type="text" name="user\_email" class="form-control" value="<?php echo set\_value('user\_email'); ?>" />

<span class="text-danger"><?php echo form\_error('user\_email'); ?></span>

</div>

<div class="form-group">

<label>Enter Password</label>

<input type="password" name="user\_password" class="form-control" value="<?php echo set\_value('user\_password'); ?>" />

<span class="text-danger"><?php echo form\_error('user\_password'); ?></span>

</div>

<div class="form-group">

<input type="submit" name="register" value="Register" class="btn btn-info" />

</div>

</form>

</div>

</div>

</div>

</body>

</html>

**Views(email\_verification.php)**

This view file is used for display success message of email verification with Login link.

<!DOCTYPE html>

<html>

<head>

<title>Complete Login Register system in Codeigniter</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />

</head>

<body>

<div class="container">

<br />

<h3 align="center">Complete Login Register system in Codeigniter</h3>

<br />

<?php

echo $message;

?>

</div>

</body>

</html>

**Views(login.php)**

This view file will load Login form on web page.

<!DOCTYPE html>

<html>

<head>

<title>Complete User Registration and Login System in Codeigniter</title>

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />

</head>

<body>

<div class="container">

<br />

<h3 align="center">Complete User Registration and Login System in Codeigniter</h3>

<br />

<div class="panel panel-default">

<div class="panel-heading">Login</div>

<div class="panel-body">

<?php

if($this->session->flashdata('message'))

{

echo '

<div class="alert alert-success">

'.$this->session->flashdata("message").'

</div>

';

}

?>

<form method="post" action="<?php echo base\_url(); ?>login/validation">

<div class="form-group">

<label>Enter Email Address</label>

<input type="text" name="user\_email" class="form-control" value="<?php echo set\_value('user\_email'); ?>" />

<span class="text-danger"><?php echo form\_error('user\_email'); ?></span>

</div>

<div class="form-group">

<label>Enter Password</label>

<input type="password" name="user\_password" class="form-control" value="<?php echo set\_value('user\_password'); ?>" />

<span class="text-danger"><?php echo form\_error('user\_password'); ?></span>

</div>

<div class="form-group">

<input type="submit" name="login" value="Login" class="btn btn-info" />&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href="<?php echo base\_url(); ?>register">Register</a>

</div>

</form>

</div>

</div>

</div>

</body>

</html>

So, this is complete source code of Codeigniter Login Register system, if you have any question regarding this tutorial, you can comment into comment box.