8, it is time to focus on Localization and internationalization. When we use Different language for purpose of using every one is called internationalization and if we using our own language in a system is called localization.

8.1 Language Class

The Language Class provides functions to retrieve language files and lines of text for purposes of internationalization.

In your CodeIgniter **system** folder, you will find a **language** sub-directory containing a set of language files for the **english** idiom. The files in this directory (**system/language/english/**) define the regular messages, error messages, and other generally output terms or expressions, for the different parts of the CodeIgniter framework.

You can create or incorporate your own language files, as needed, in order to provide application-specific error and other messages, or to provide translations of the core messages into other languages. These translations or additional messages would go inside your **application/language/** directory, with separate sub-directories for each idiom (for instance, " Dari"and "Pashto").

The CodeIgniter framework comes with a set of language files for the "english" idiom. Additional approved translations for different idioms may be found in the CodeIgniter 3 Translations repositories. Each repository deals with a single idiom.

When CodeIgniter loads language files, it will load the one in **system/language/** first and will then look for an override in your **application/language/** directory.

8.2 For instance if we want to use different languages for example("Dari" , "English" and "Pashto") we need to take below steps.

8.2.1 create folders for every language you want to put in your application, go to "C/wamp/www/CI/application/languages/ and create for every language a folder by the name of that language.(create three folder"dari", "pashto", "english")

8.2.2 Than we place different files on it. The one we need to use for now is the main language file. so create in every folder the same file called main_lang.php.

8.3 we need to put contents on theses files, actually we need to put those variables that need to translate in deferent languages in these files.

8.4 We have for language helper and library, we need to load this helper. Go to C/wamp/www/CI/application/config/autoload.php find below line and add language helper as below if you not did it before.

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

8.5 To enhance our project we need to put some aesthetics to our project. For this we use bootstrap. We only need to down load the bootstrap and past it in the root of our project and than link it with our project.

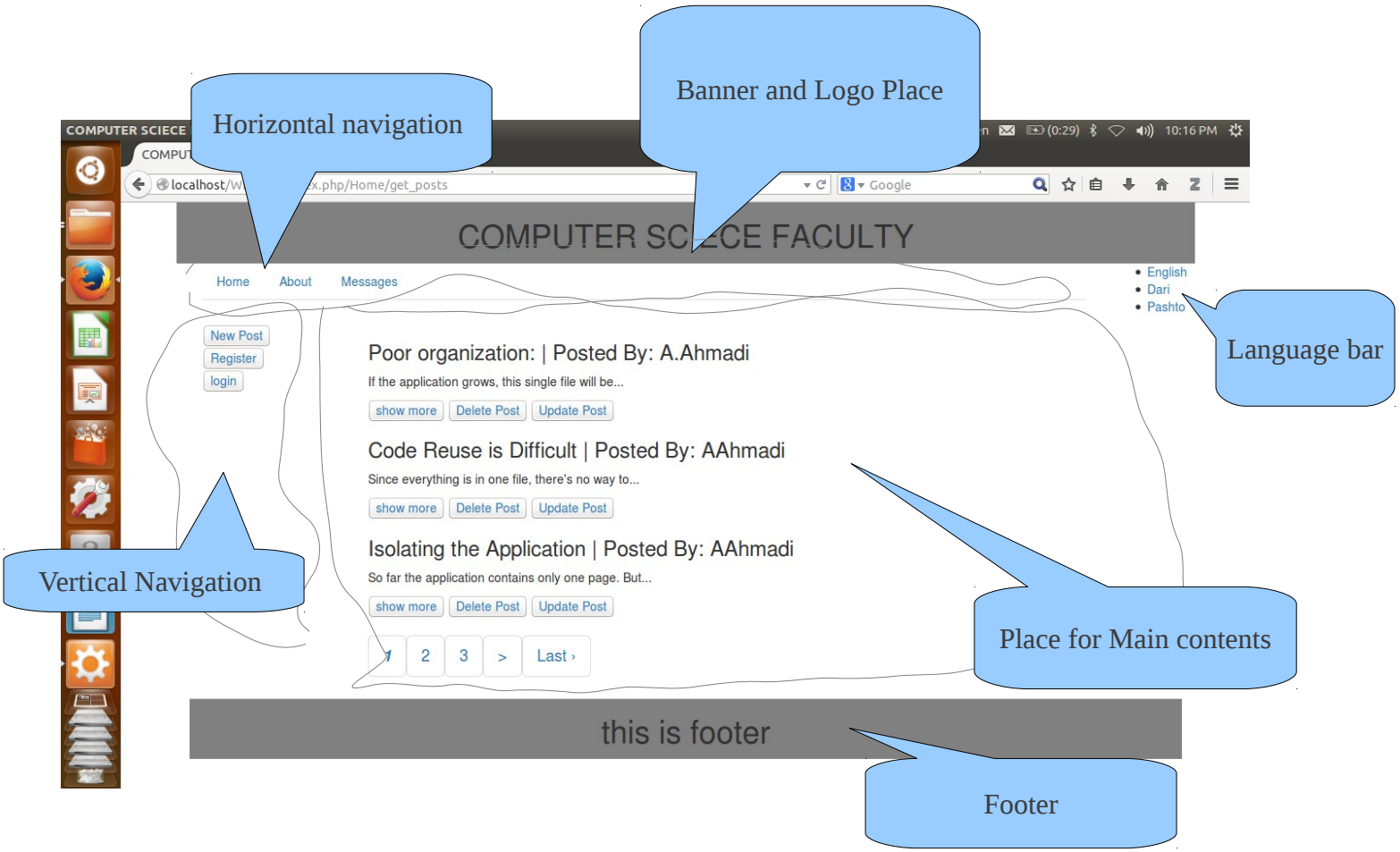
8.5.1 Download the bootstrap, unzip, rename it to bootstrap. Than copy this folder and go to C/wamp/www/CI/ and past it.

8.5.2 Now we need to use it in our project. For this go to C/wamp/www/CI/application/views/ and create new folder called includes than create to files in this folder header.php file and footer.php file.

Place only links in these two files as below.

```
//header.php

<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="">
<meta name="author" content="">

<title><?php echo $title?></title>
<!-- Bootstrap core CSS -->
<link href="<?php echo base_url('bootstrap/css/bootstrap.min.css'); ?>"
rel="stylesheet">
<!-- Bootstrap theme -->
<link href="<?php echo base_url('bootstrap/css/bootstrap-theme.min.css'); ?
>"rel="stylesheet">
<!-- Custom styles for this template -->
<link href="<?php echo base_url('bootstrap/css/theme.css');?>" rel="stylesheet">

</head>


//footer.php

<link href="<?php echo base_url('bootstrap/css/bootstrap.min.css'); ?>"
rel="stylesheet">
<!-- Bootstrap core JavaScript
================================================== -->
<!-- Placed at the end of the document so the pages
load faster -->
<h1> <?php echo mb_convert_case(lang('footer'), MB_CASE_TITLE) ?></h1>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"
type="text/javascript"> </script>
<script src="<?php echo base_url('bootstrap/js/bootstrap.min.js');?>"
type="text/javascript" ></script>
<script src="<?php echo base_url('bootstrap/js/docs.min.js');?>"
type="text/javascript" ></script>
</div> <!-- /container -->
</body>
</html>
```

8.5.3 we need a main view to load all content in it. So go to C/wamp/www/CI/application/views/ and create new file named main_view.php and load two previews views(heade and footer) and make it as main layout. For this we use bootstrap to structure our view in context of our prototype. In this project we have this prototype as shown below.



For above prototype we need below code for structuring it.

```
//main_view.php

<!DOCTYPE  html>
      <?php $this->load->view('includes/header')?>
      <body role="document">
<!-- END header.php -->
      <div class="container theme-showcase" role="main">

    <div class="row">
      <div class="col-md-12 text-center" style=" background:gray">
         <h1> <?php echo $main_title?></h1></div>
            <div class="col-md-12">
            <div class="col-md-11">
            <ul class="nav nav-tabs" role="tablist">
            <li ><a href="<?php echo base_url('index.php/Home/get_posts')?>">
                  Home</a></li>
            <li><a href="<?php echo base_url('index.php/User/view/about')?>">
            About</a></li>
            <li><a href="#">Messages</a></li>
             </ul>
            </div>
                        <div class="col-md-1">
                        <ul>  <?php
                        echo '<li>'.anchor('User/lang/english',"English")."</li>";
                        echo '<li>'.anchor('User/lang/dari',"Dari")."</li>";
                        echo '<li>'.anchor('User/lang/pashto',"Pashto")."</li>";
                               ?> </ul>
                        </div>
```

```
        <div class="col-md-2 ">
        <?php
        echo anchor('Home/add_post/',"<button> New Post</button> "." <br/>");
             $user= $this->session->userdata('user');
             echo anchor('User/register',"<button> Register </button>")." <br/>";
             if ($this->session->userdata('login')) {
                   echo "You loged in as ".$user->user_name." " ;
                   echo anchor('User/logout',"<button> logout </button>")."  ";

             }else{
                    echo anchor('User/user_login',"<button> login </button>")."";
             }
             ?></div>
        <div class="col-md-10"><?php $this->load->view($contents)?></div>
        <div class="col-md-12 text-center" style=" background:gray" >
       <?php $this->load->view('includes/footer')?></div>
        </div>
```

Note: Highlighted part is cut from something view.
The red highlighted part is to load dynamic view in this area. Every view that is loaded should assigned to the index of array called contents.

8.5.4 we need to change our controller to load all contents in this new layout.
Go to C/wamp/www/CI/application/controllers/home.php and edit it as below.

```
<?php

class Home extends CI_Controller {

      function __construct() {
            parent::__construct();
            $this->load->model('Posts_model');
            if (!$this -> session -> userdata('language'))
                  $this -> lang -> load('main');
            else
      $this -> lang -> load('main', $this -> session -> userdata('language'));
      }

      function index(){

            echo "Hello CodeIgniter";
      }
      //-------------------------------------------------------------------
      function dynamic_view(){

            $data['title'] = 'Dynamic View';
            $data['body'] = 'A view for Dynamic Contents!';
            $this->load->view('some_view',$data);
      }
      //-------------------------------------------------------------------
      function get_posts(){
            $this -> session -> set_userdata('last_visited', current_url());
            $this->load->library('pagination');
            $offset=$this->uri->segment(3);
            $config['base_url'] = base_url().'index.php/home/get_posts';
            $config['total_rows'] =$this->Posts_model->count_posts();
            $config['per_page'] = 3;
            $config['uri_segment'] = 3;
```

```
// to enhance pagination we need for this settings
            $config['cur_tag_open'] = '<span><b><i>';
            $config['cur_tag_close'] = '</i></b></span>';
            $config['full_tag_open'] = '<li>';
            $config['full_tag_close'] = '</li>';
            $this->pagination->initialize($config);
            $data['posts']=$this->Posts_model->get_posts($config['per_page'],
$offset);
            $data['title']=lang('title_main');
            $data['main_title']=lang('title_main');
            //$data['posts']=$this->Posts_model->get_posts();

            $data['contents']= 'something';
            $this->load->view('main_view',$data);
      }
      //---------------------------------------------------------------------
      function details($id){
            $data['post']=$this->Posts_model->get_post($id);
            $data['contents']='show_d';
            $data['main_title'] = mb_convert_case(lang('title_main'),
MB_CASE_TITLE);
            $this->load->view('main_view',$data);
      }
      //---------------------------------------------------------------------
      function add_post(){
            $data['contents']='new_post';
            $data['main_title'] = mb_convert_case(lang('title_main'),
MB_CASE_TITLE);
            $this->load->view('main_view',$data);
      }
      //---------------------------------------------------------------------

      function add(){
            $user = $this->session->userdata('user');
            $user_id=2;
            if ($this->session->userdata('login')) {
            $user_id=$user->user_id;
            }else{
                  $user_id=1;
            }
            $data = array(
                  'title' =>$this->input->post('title') ,
                  'body' =>$this->input->post('body'),
                  'user_id'=> $user_id
                  );
            $this->Posts_model->add_post($data);
            redirect('home/get_posts');

      }
      //-----------------------------------------
      function delete($id){
            $this->Posts_model->delete_post($id);
          redirect('home/get_posts');
      }
      //-----------------------------------------------
      function update_post($id){
            $data['post']=$this->Posts_model->get_post($id);
            $data['contents']='update_post';
```

```
            $data['main_title'] = mb_convert_case(lang('title_main'),
MB_CASE_TITLE);
            $this->load->view('main_view',$data);

      }
      //---------------------------------------------------------------------
      function update($id){
            $data = array(
                  'title' =>$this->input->post('title') ,
                  'body' =>$this->input->post('body')
            );
            $this->Posts_model->update_post($id,$data);
            redirect('home/get_posts');

      }
}
```

Note: the pink highlighted area means that we send our data to main view with data placed in different index.(for example:$data['contents']='update_post'; means we assign update_post view to contents index of array data that we extract it in main_view.php, line where stated **<?php** $this->load->view($contents)**?>)**.
The yellow part is depend to language that we will discuss it in future part.

8.6 Now it is time to work on internationalization and localization. To load the language and content of it we need for a function in a controller to take a language as a parameter and place it in the session. To create this function go to C/wamp/www/CI/application/controllers/User.php and create a function as highlighted.

```
<?php

class User extends CI_Controller{
      function __construct(){
            parent::__construct();
            $this->load->model('User_model');
// if a language is placed in the session use it else use main language in our
context it is english
      if (!$this -> session -> userdata('language'))
                  $this -> lang -> load('main');
            else
      $this -> lang -> load('main', $this -> session -> userdata('language'));
      }
      function index(){
            $this->load->view('includes/langbar');
// set current url as last visited
            $this -> session -> set_userdata('last_visited', current_url());
// go to language that is in session and load a variable called par_about.
          echo lang('par_about');


      }
      /*
       * this function used to load a view called rigister located in Form
directory of views
       */

      function register(){
```

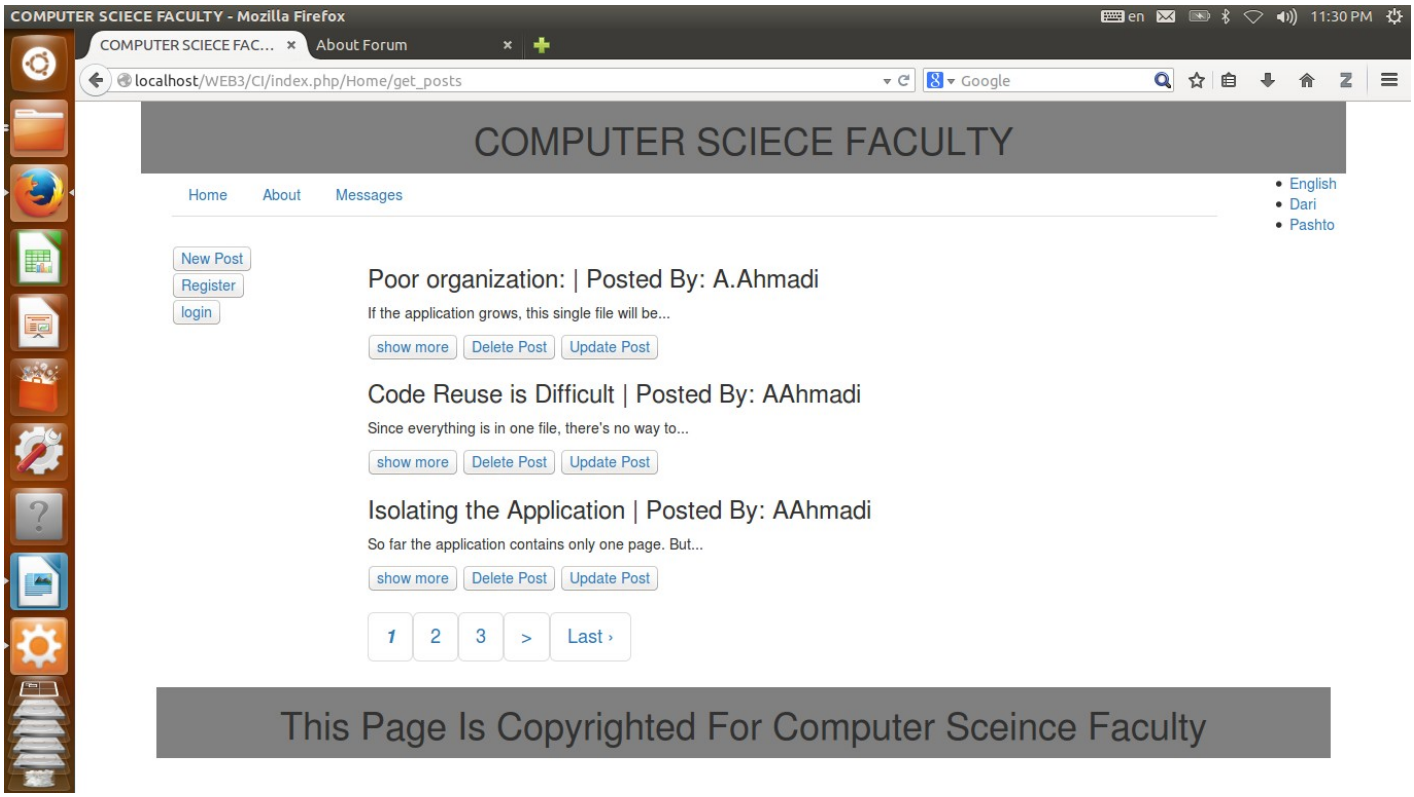
```
//take contents of a variable called title_main from language and assign it to
index of main_title of data array, than we can get it in our view
            $data['main_title'] = mb_convert_case(lang('title_main'),
MB_CASE_TITLE);
            $data['contents']= 'Form/register';
            $this->load->view('main_view',$data);

      }
      //-------------------------------------------------------------------------
-------
      /*
       * this function is to create new user and the form data is sent to this
function from register.php file
       * and send to user table of blog_data database
       */
      function register_user(){
            $data = array(
            'name'=>$this->input->post('name'),
            'lname'=>$this->input->post('lname'),
            'user_name'=>$this->input->post('uname'),
            'password'=>$this->input->post('password'),
            'email'=>$this->input->post('email')

            );
            //echo "<pre>"; print_r($data);echo "</pre>";
            $this->User_model->add_user($data);
            redirect('home/get_posts');

      }
      /*
       * this function load a file called login from the Form Directory.
       */
      function user_login(){
//take contents of a variable called title_main from language and assign it to
index of main_title of data array, than we can get it in our view
$data['main_title'] = mb_convert_case(lang('title_main'), MB_CASE_TITLE);
            $data['contents']= 'Form/login';
            $this->load->view('main_view',$data);
      }
      //-------------------------------------------------------------------------
-------
      /*
       * this function take login data and send to database and than retrive the
desired field and copmare
       * with these data
       */

      public function login() {
            $username = $this -> input -> post('username');
            $password = $this -> input -> post('password');

            if ($this -> input -> post('remember') == 'yes') {
                  // set a cookie with the username. It will expire in 269200
seconds(3 days).
                  $cookie = array('name' => 'username', 'value' => $username,
'expire' => 259200);
                  $this -> input -> set_cookie($cookie);
            } else {
```

```
                  // delete the cookie
                  $cookie = array('name' => 'username', 'value' => NULL, 'expire'
=> NULL);
                  $this -> input -> set_cookie($cookie);
            }
            $uid = ($this ->User_model->login($username, $password));
            if (!$uid){
                  redirect(site_url('home/get_posts'));
            }

            // if login successful than send the all data related to the user to
session array
            $this -> session -> set_userdata('uid', $uid);
            $this -> session -> set_userdata('user', $this->
User_model->get_userdata($uid));
            $this -> session -> set_userdata('login', true);
            //print_r($this->session->all_userdata());
            redirect($this -> session -> userdata('last_visited'));
      }
      //-------------------------------------------------------------------------
-------
      // this function is called by log out button and delete the session array
      public function logout() {
            $this -> session -> unset_userdata('login');
            $this -> session -> unset_userdata('user');
            $this -> session -> unset_userdata('uid');

            redirect($this -> session -> userdata('last_visited'));
      }
      //-------------------------------------------------------------------------
--------
//this function take a language as parameter and put it in the session
      public function lang($language) {
// here we send it to session
            $this -> session -> set_userdata('language',$language);
// redirect to last visited page.
            redirect($this -> session -> userdata('last_visited'));

      }
      //-------------------------------------------------------------------------
--------
// this function is for loading different views it take view as parameter and load
it in the main view
      public function view($page) {
            $this -> session -> set_userdata('last_visited', current_url());
                        // try to translate the title, then capitalize it.
            $data['title'] = mb_convert_case(lang('title_' . $page),
MB_CASE_TITLE);
// translate and capitalize the main title that shows in the banner
            $data['main_title'] = mb_convert_case(lang('title_main'),
MB_CASE_TITLE);
            $data['contents']='includes/'.$page;
            $this ->load->view('main_view', $data);
      }
}
```

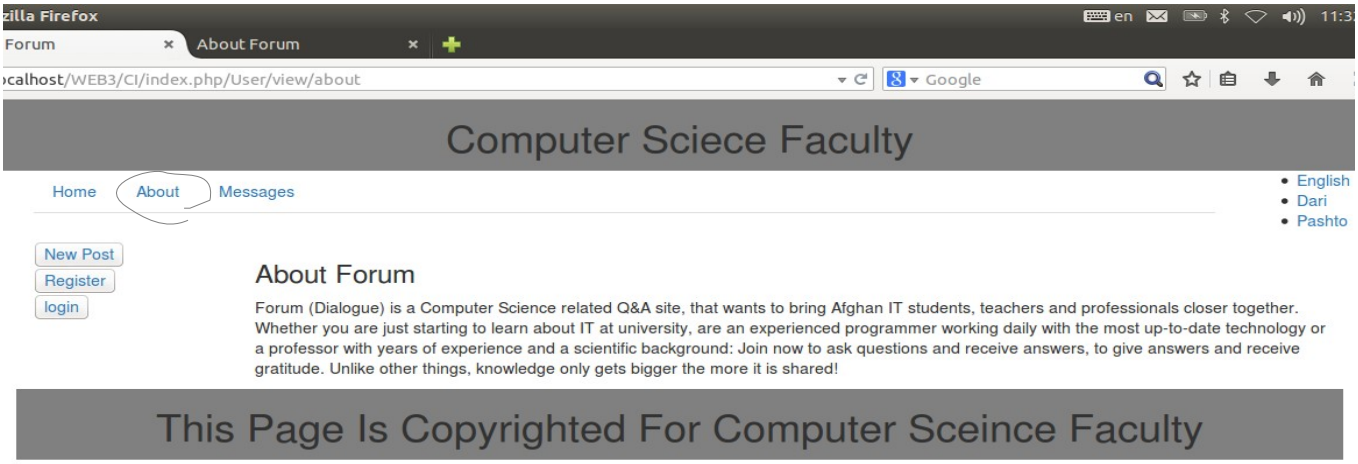
8.7 for clearity we now create one other view called about. Go to C/wamp/www/CI/application/views/includes and create new file called about and place below codes.

```
<div id="body">
<h3><?php echo lang('title_about') ?></h3>
// we have par_about variable in all our languages.
<p><?php echo lang('par_about') ?></p>
</div>
```

8.8 now if we load our project it will look like below.



8.8.1 if we click on about tab it will look like below.



8.8.1 if we click on dari language it will look like below.

8.8.1 if we click on pashto language it will look like below.