Codeigniter version : 3.\*

\* System : user management system with auto navigation menu generation.

\* Authentication levels :

1. menu & sub menu authentication to user roles.

2. module wise(within the sub menu eg:-add/edit/delete privileges) authentication for users.

\* codeigniter hooks used for this authentication.

\* user can have multiple user roles.

\*controller,model,view are auto generating. no need to configure.

////\*\*\*\*\*\*\*\*installation guide\*\*\*\*\*\*\*\*////

1. extract the file

2. create a database with any name eg :- xxxxx in your sever

3. import the sql file("procurement\_vx.sql") to your database

4. create a application folder in your server location and you can give any name to it(in my case it is "procurement\_vx") and copy the files to that folder

5. change the database config data set with your sever config data set. and change the base url in the config file(in my case, i am using localhost so my file url = http://localhost/procurement\_vx/).

6. go to your browser and load the file (in my case, i am using localhost so my file url = http://localhost/procurement\_vx/).

7. give the full file permission to the dev user.

//\*\*\* login \*\*\*//

=============

url : localhost/procurement\_vx/index.php/users/index/

\* you can use this same link to direct login using url / form submit

\* for url login perameters : ($user\_id=\*\*\*\*, $password=\*\*\*\*, $login\_type=(direct/default/url) )

for testing only : login\_type = direct (you can change the login type in view file).

\* here the password not checked by the beckend.

\* default login controller : procurement\_vx/application/controllers/users.php

\* default login model : procurement\_vx/application/models/user.php

\* default login view page : procurement\_vx/application/views/users/index.php

\*if you are trying to develop the controller,model and views manually then this part is useful for you.

Rules for create a new menu

1. Create new controller in the application file

eg:- class Perameters extends CI\_Controller {}

2. then create a new menu using application interface and give that controller name to it.

Rules for create a new sub menu

1. create a function with any name in that controller in which you created,

\* make sure that function name must be meaning full and word must be separated by underscore("\_").

eg: - class Perameters extends CI\_Controller {

function manage\_cost\_centers (){}

}

2. then add these two lines to the above function and the set the data and the view then render.

a. $this->session->set\_userdata('controller', "Controller Name");

b. $this->session->set\_userdata('function',"Function Name");

eg:- class Perameters extends CI\_Controller {

function manage\_cost\_centers (){

$this->session->set\_userdata('controller', 'perameters');

$this->session->set\_userdata('function', 'manage\_cost\_centers');

$this->template->set();

$this->template->current\_view = 'perameters/manage\_cost\_centers'; $this->template->render();

}

}

3. Create a new view file

eg:- manage\_cost\_centers.php

4. create a new .js and .css files for the above view. or you can add your "js" and "CSS" codes directly to your view.

\* If you carete new ".js" or new ".css" file then,

5. add the above ". js" file path to " <!--page related js files-->" section in the "procurement\_vx/application/views/layouts/blocks/js.php" file.

6. add the above ".css" file path to " <!--page related css files--> section in the "procurement\_vx/application/views/layouts/blocks/css.php" file.

3. Then add the new sub menu using the system interface.

Rules for create a new sub menu related functions

1. create any function in controller with any name.

eg:- function add\_cost\_centres(){}

2. then add that function using the system interface by selecting the appropriate submenu and the module.

3. then add the rights to the roles using "manage user rights" sub menu in the "system" menu.

4. use the "manage module right" to change the functions level privilege (like "add") to particular user.

NOTE :-

\*in this system you cannot delete any parent item if you have any child item of that.

eg: - cannot delete the menu if you have sub menus for that.

\* User lists for testing purpose (no need password).

1. admin 2. 111 3. aaa