**Experiment No. 8**

**Title: Form validation and dealing with database**

**Batch:B1 RollNo.:1514033 ExperimentNo.:8**

### Aim: Using appropriate helpers create a CodeIgniter application which accept name, email ID, mobile number, address of user through form and store it in database. When user clicks on show the contents of database must be displayed. Perform appropriate validation before storing data in database.

### --------------------------------------------------------------------------------------------

**Resources needed:**CodeIgniter, Notepad

### Theory:

### CodeIgniter is an application development framework, which can be used to develop websites, using PHP. It is an Open Source framework. It has a very rich set of functionality, which will increase the speed of website development work.

### Installing CodeIgniter and creating application

### 1. Download CodeIgniter from its official website.

### https://www.codeigniter.com

### 2. Unzip CodeIgniter package: Downloaded CodeIgniter will be in zip format. Unzip it

### 3. Create folder namedCodeIgniter in C>xampp>htdocs

### 4. Copy content of unzip folder insideCodeIgniter folder.

### 5. Do database configurations in configuration file located at application/config/database.php

### 

### Specify the name of database created by you. For ex. Database name is “college” then

### ‘database’=>’college’

### 6. To connect automatically to database add the word database to the library array, as indicated in the file:application/config/autoload.php

### 

### Also form\_validation library can be added here to load automatically. In that case code will be like this

### $autoload[‘libraries’]=array(‘database’, ‘form\_validation’);

### 7. Load appropriate helper in file: application/config/autoload.php

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

### 8.Create controller (.php) and save in C:\xampp\htdocs\CodeIgniter\application\controllers

### 9. Create view (.php) and save in C:\xampp\htdocs\CodeIgniter\application\views

### 10. Create model (.php) and save in C:\xampp\htdocs\CodeIgniter\application\models

### Results: (Screen shots of application development steps, program code and web browser displaying the specified message.)

### Controller:

### 

### Controller:form.php

### <?php

### class form extends CI\_Controller {

### public function index() {

### // load view\_form.php present in views folder

### $this->load->view("view\_form");

### }

### }

### ?>

### Controller:register.php

### <?php

### class register extends CI\_Controller {

### function \_\_construct() {

### parent::\_\_construct();

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

### }

### public function index() {

### 

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

### $this->form\_validation->set\_error\_delimiters('<div class="error">', '</div>');

### $this->form\_validation->set\_rules('emp\_name', 'Name', 'required|min\_length[5]|max\_length[15]');

### $this->form\_validation->set\_rules('email\_id', 'Email', 'required|valid\_email');

### $this->form\_validation->set\_rules('password', 'Password', 'required|regex\_match[/^(?=.\*[A-Za-z])(?=.\*\d)(?=.\*[$@$!%\*#?&])[A-Za-z\d$@$!%\*#?&]{8,}$/]');

### if ($this->form\_validation->run() == FALSE) {

### $this->load->view('view\_form');

### } else {

### $data = array(

### 'employee\_name' => $this->input->post('emp\_name'),

### 'employee\_email' => $this->input->post('email\_id'),

### 'employee\_password' => $this->input->post('password'),

### 'employee\_gender' => $this->input->post('select')

### );

### if($this->input->post('update'))

### $this->Insert\_model->form\_update($data);

### else if($this->input->post('delete'))

### $this->Insert\_model->form\_delete($data);

### else

### $this->Insert\_model->form\_insert($data);

### $data['message'] = 'Data Inserted Successfully';

### $data['h']=$this->Insert\_model->select();

### $this->load->view("view\_form1", $data);

### }

### }

### }

### ?>

### Model

### 

### Model:Insert\_model.php

### <?php

### class Insert\_model extends CI\_Model{

### function \_\_construct() {

### parent::\_\_construct();

### }

### function form\_insert($data){

### //echo 'insert';

### // Inserting in Table(students) of Database(college)

### $this->db->insert('student', $data);

### 

### }

### 

### public function select()

### {

### //data is retrive from this query

### $query = $this->db->get('student');

### return $query;

### }

### 

### public function form\_update($data)

### {

### $sname = $data['employee\_name'];

### $this->db->where('employee\_name',$sname);

### $this->db->update('student',$data);

### }

### public function form\_delete($data)

### {

### $srollno = $data['employee\_name'];

### $this->db->where('employee\_name',$srollno);

### $this->db->delete('student');

### }

### }

### ?>

### Views:

### 

### View:view\_form.php

### <html>

### <head>

### <title>Codeigniter Form Helper</title>

### <meta name="robots" content="noindex, nofollow">

### </head>

### <body>

### <center>

### <?php

### echo form\_open('register');

### echo form\_label('Name :','emp\_name');

### echo form\_error('emp\_name');

### echo br();

### $data=array('id' => 'emp\_name', 'name' => 'emp\_name');

### echo form\_input($data);

### 

### echo br();

### 

### echo form\_label('Email-ID','email\_id');

### echo form\_error('email\_id');

### echo br();

### $data\_email = array('type' => 'email','name' => 'email\_id','id' => 'email\_id','class' => 'input\_box','placeholder' => 'Please Enter Email');

### echo form\_input($data\_email);

### 

### echo br();

### echo form\_label('Password');

### echo form\_error('password');

### echo br();

### $data\_password = array('name' => 'password','id' => 'password','class' => 'input\_box','placeholder' => 'Please Enter Password');

### echo form\_password($data\_password);

### echo br();

### 

### echo form\_label('Gender');

### echo br();

### $data\_gender = array('Male' => 'Male','Female' => 'Female');

### echo form\_dropdown('select', $data\_gender, 'Male', 'class="dropdown\_box"');

### echo br();

### ?><input type="submit" name="update" value="Update">

### <input type="submit" name="delete" value="Delete"><?php

### echo form\_submit('submit', 'Insert', "class='submit'");

### echo form\_close();

### 

### 

### ?>

### </center>

### </body>

### </html>

### Views:view\_form1.php

### <html>

### <head>

### <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

### <title>Untitled Document</title>

### </head>

### <body>

### <table border="10">

### <tr>

### <td>Employee Name</td>

### <td>Employee Email</td>

### <td>Employee Password</td>

### <td>Employee Gender</td>

### </tr>

### <?php

### foreach ($h->result() as $row)

### {

### echo "<tr>

### <td>$row->employee\_name</td>

### <td>$row->employee\_email</td>

### <td>$row->employee\_password</td>

### <td>$row->employee\_gender</td>

### </tr> ";

### }

### ?>

### </table>

### </body>

### </html>

### Output:

### Validation

### 

### Insert:

### 

### 

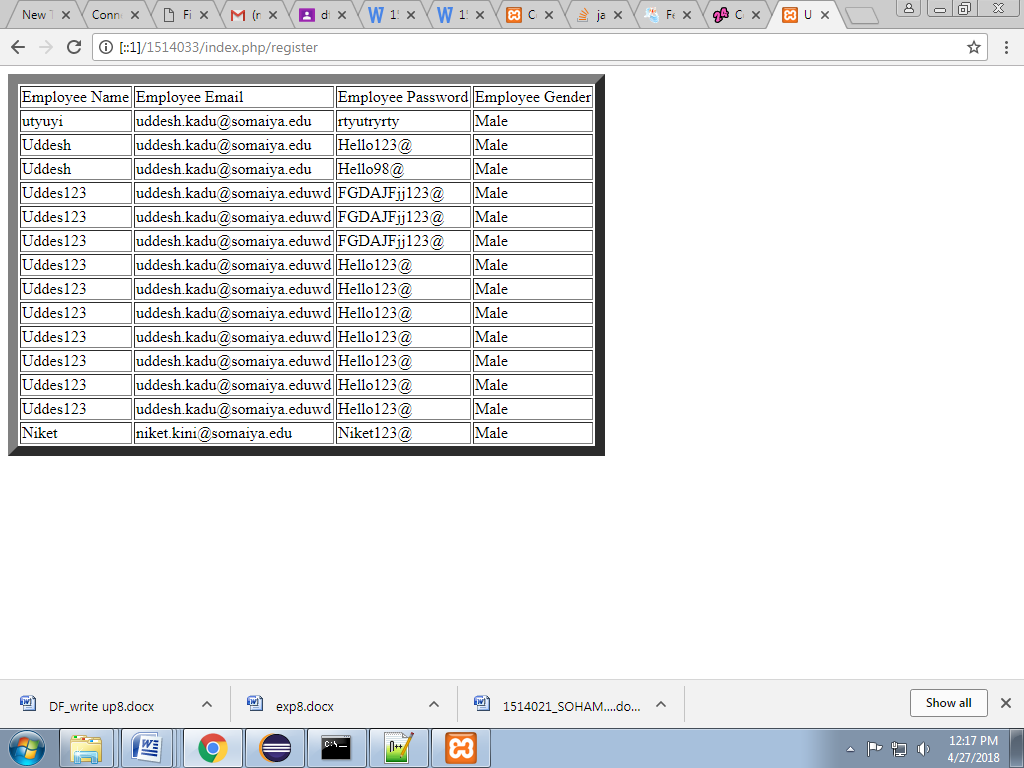
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Questions:**

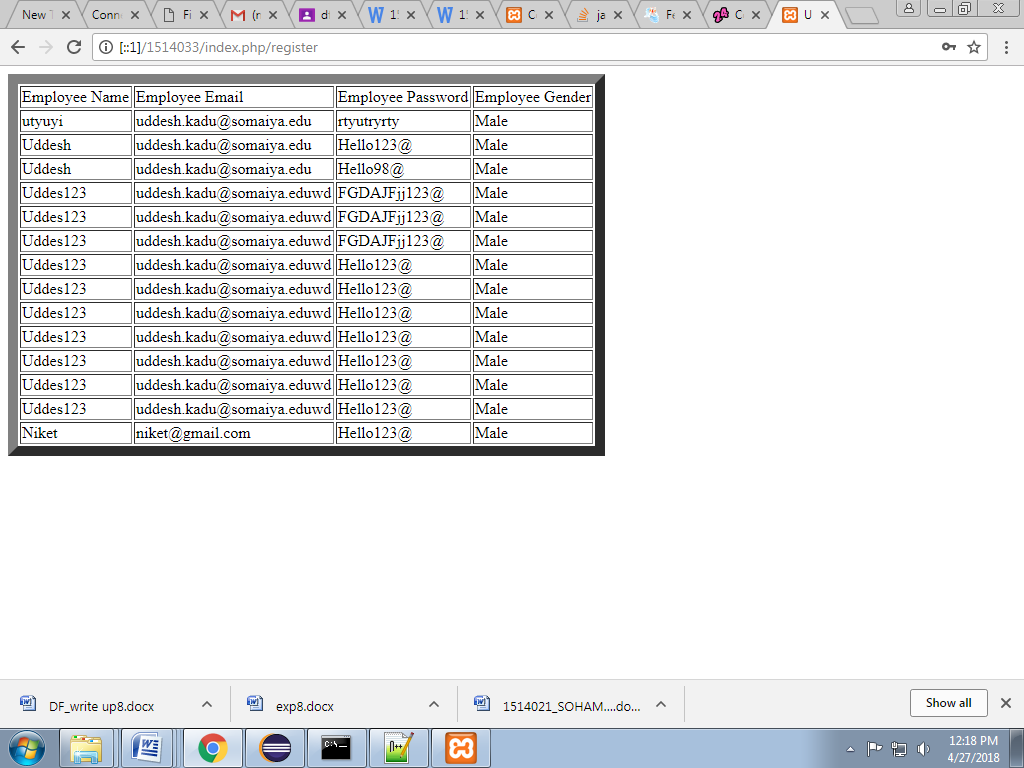
1. What is helper. Explain various methods of form helper.
2. In CodeIgniter there are helpers which are there to help you with different tasks. Every helper
3. file is a collection of functions aiming towards a particular role. Some of the helpers are &#39;file
4. helpers&#39; which help you to deal with the file, &#39;text helpers&#39; to perform various text formatting
5. routines, &#39;form helpers&#39; to create form element, &#39;cookie helpers&#39; set and read cookies, &#39;URL
6. helpers&#39; which assist in creating links, etc.
7. Helpers are not written in Object Oriented format, instead they are simple, procedural
8. functions, independent of each other.
9. To use helper files, you need to load it. Once loaded it is globally available to your controller
10. and views. They are located at two places in CodeIgniter. CodeIgniter will look first for a
11. helper in application/helpers folder and if not found there then it will go
12. to system/helpers folder.
13. Form Helper methods are:
14. Form\_open()
15. Form\_input()
16. Form\_label()
17. Form\_checkbox()
18. Form\_radio
19. Form\_hidden()
20. Form\_password()
21. And many more.
22. All of these methods take object or array as a parameter where we can specify its attributes.
23. Create CI application form which will accept the student name and roll number from user, store it in database. The form should have delete and update button which will accept the roll number and will perform delete or update operation respectively.

Update:

Before:

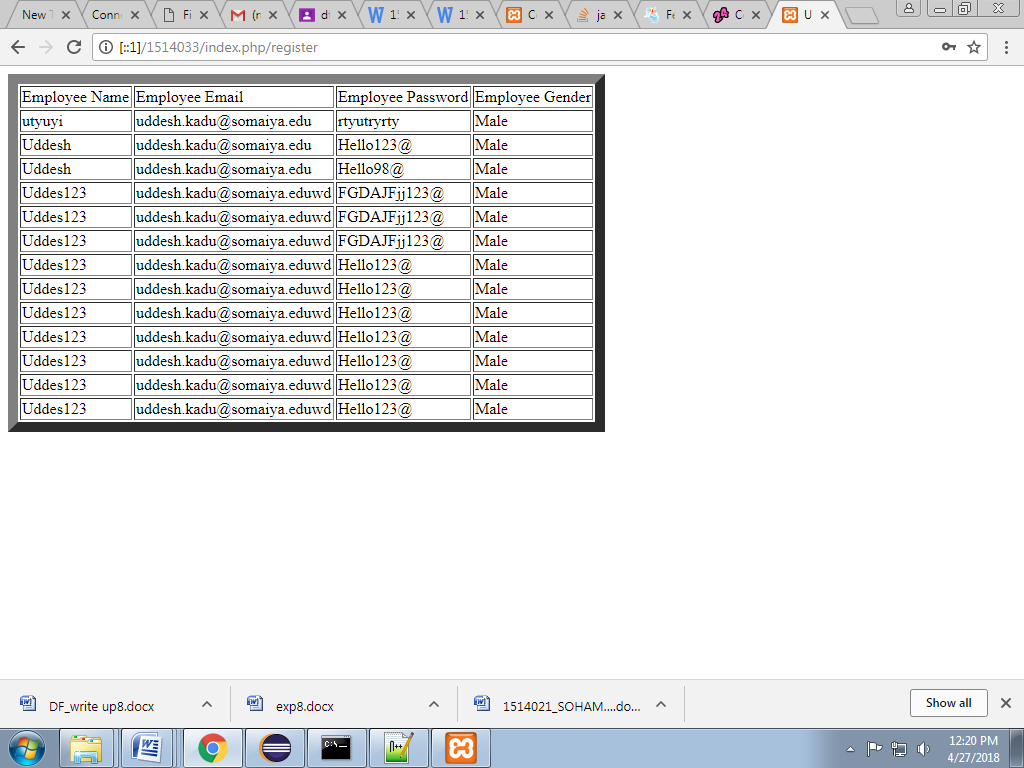


After:



Delete:

After:



### Outcomes:Develop application using framework

**Conclusion: (Conclusion to be based on the objectives and outcomesachieved)**

**Employee data was validated and inserted in database using codeignitter**

**Grade: AA / AB / BB / BC / CC / CD/DD**

**Signature of faculty in-charge withdate**

**References:**

1.By David Upton ;“Codeigniter for Rapid PHP Application Development”; Packt

Publishing, July 2007