CodeIgniter 4 CRUD with Bootstrap and MySQL Example

Last Updated on December 18, 2023 by **Digamber** in CodeIgniter Tutorial

If you are willing to learn how to create CRUD operations in Codeigniter 4 application. We will also shed light on how to integrate Bootstrap 4 and display data using Datatables jQuery plug-in.Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for

DataTables is a table enhancing plug-in for the jQuery Javascript library, adding sorting, paging and filtering abilities to plain HTML tables with minimal effort.

The following topics will be covered in this tutorial:

- Installing Codeigniter with the composer package.
- Creating views to handle CRUD operations.
- Implementing Bootstrap 4 in Codeigniter.
- Implementing DataTables in Codeigniter.

- Retrieving data from MySQL.
- Update data from MySQL.
- Delete user object from the database.
- Enable Codeigniter Errors.

Codeigniter Bootstrap CRUD Application Example

- **Step 1:** Download or Install Codeigniter 4
- **Step 2:** Enable Codeigniter Errors
- **Step 3:** Set Up Database
- **Step 4:** Create New Model
- **Step 5:** Create CRUD Controller
- **Step 6:** Create Routes
- **Step 7:** Insert Data into Database
- **Step 8:** Show Data List & Delete
- **Step 9:** Edit and Update Data

Download or Install Codeigniter 4

Invoke the first step by downloading the fresh Codeigniter 4 application using the composer package.

```
composer create-project codeigniter4/appstarter
```

Once the project has downloaded, rename the folder name as per your choice. In our case, we will name it **codeigniter-crud-example**. Afterward, go

```
cd codeigniter-crud-example
```

If you are seeing following error on the terminal:

“Your requirements could not be resolved to an installable set of packages”

Run the following command.

```
composer install --ignore-platform-reqs
```

You can also manually download the Codeigniter 4 application to impetus the CRUD operations development process.

Enable Codeigniter Errors

This step is essential. Let me tell you why? When i was first learning Codeigniter, then i was getting stuck at every level, because i was not aware of how to debug the errors. After spending some time on google, i got to know that you can enhance the impetus of development process by enabling the error reporting and can display on-screen by following the below procedure.

Open **app/Config/Boot/development.php** file and set the `display_errors` to **1** instead of **0**. Repeat the same process in **app/Config/Boot/production.php** file.

```
ini_set('display_errors', '1');
```

In general, this step interprets how to give impetus to the database related work.

Add your database name, username and password in **app/config/database.php** file.

```
public $default = [
        'DSN'      => '',
        'hostname' => 'localhost',
        'username' => 'root',
        'password' => 'root',
        'database' => 'demo',
        'DBDriver' => 'MySQLi',
        'DBPrefix' => '',
        'pConnect' => false,
        'DBDebug'  => (ENVIRONMENT !== 'development'),
        'cacheOn'  => false,
        'cacheDir' => '',
        'charset'  => 'utf8',
        'DBCollat' => 'utf8_general_ci',
        'swapPre'  => '',
        'encrypt'  => false,
        'failover' => [],
        'port'     => 3306,
    ];
```

CodeIgniter\Database\Exceptions\DatabaseException #8
Unable to connect database : Codeigniter

If anyhow you get the **Codeigniter – cannot connect to MySQL database** error, then change the hostname value based on your local server e.g MAMPP or XAMPP.

```
# MAMPP
public $default = [
  ...
     'hostname' => '/Applications/MAMP/tmp/mysql/mysql.sock',
  ...
]
# XAMPP
public $default = [
  ...
     'hostname' => '/Applications/XAMPP/xamppfiles/var/mysql/mysql.s
  ...
]
```

To create the users table we need to run the following SQL query; make sure to head over to PHPMyAdmin tab, look for SQL tab then execute the following query.

```
    `name` VARCHAR(150) NOT NULL,
    `email` VARCHAR(50) NOT NULL
)
```

Create New Model

The model is used for defining the schema that is an archetype of table values. So, we have to manifest a new **UserModel.php** file in the **app/Models/** folder. Insert the following code inside the same file to establish the User Model.

```
<?php
namespace App\Models;
use CodeIgniter\Model;
class UserModel extends Model
{
    protected $table = 'users';
    protected $primaryKey = 'id';

    protected $allowedFields = ['name', 'email'];
}
```

The model class provides impetus access to database connection. It concurrently supports database the queries to propel forward the data in the database.

Codeigniter offers multiple configuration options to handle Model values, and you can check those values here.

In this part of this tutorial, we are going to manifest a new controller and name it **UserCrud.php**. This controller will hold the CRUD methods to handle CRUD operations such as Create, Read, Update, Delete.

We have to place the following code inside the **app/Controllers/ UserCrud.php** file and invoke the CRUD operations.

```
<?php
namespace App\Controllers;
use App\Models\UserModel;
use CodeIgniter\Controller;
class UserCrud extends Controller
{
    // show users list
    public function index(){
        $userModel = new UserModel();
        $data['users'] = $userModel->orderBy('id', 'DESC')->findAll(
        return view('user_view', $data);
    }
    // add user form
    public function create(){
        return view('add_user');
    }

    // insert data
    public function store() {
        $userModel = new UserModel();
        $data = [
            'name' => $this->request->getVar('name'),
            'email'  => $this->request->getVar('email'),
        ];
        $userModel->insert($data);
        return $this->response->redirect(site_url('/users-list'));
        $userModel = new UserModel();
        $data['user_obj'] = $userModel->where('id', $id)->first();
        return view('edit_user', $data);
    }
    // update user data
    public function update(){
        $userModel = new UserModel();
        $id = $this->request->getVar('id');
        $data = [
            'name' => $this->request->getVar('name'),
            'email'  => $this->request->getVar('email'),
        ];
        $userModel->update($id, $data);
        return $this->response->redirect(site_url('/users-list'));
    }

    // delete user
    public function delete($id = null){
        $userModel = new UserModel();
        $data['user'] = $userModel->where('id', $id)->delete($id);
        return $this->response->redirect(site_url('/users-list'));
    }
}
```

Create Routes in Codeigniter

Now, we have to create routes to handle CRUD operations. To know more about creating RESTful routes you can check out the official documents here.

To enable the routes in Codeigniter CRUD application place the following code inside the **app/Config/Routes.php** file.

```
$routes->get('users-list', 'UserCrud::index');
$routes->get('user-form', 'UserCrud::create');
$routes->post('submit-form', 'UserCrud::store');
$routes->get('edit-view/(:num)', 'UserCrud::singleUser/$1');
$routes->post('update', 'UserCrud::update');
$routes->get('delete/(:num)', 'UserCrud::delete/$1');
```

Codeigniter Insert Data in Database Example

Now, we will learn how to propel data into the MySQL database via Codeigniter. Create **add_user.php** file inside the **app/Views/** folder and then incorporate the following code inside the same file.

```
<!DOCTYPE html>
<html>
<head>
  <title>Codeigniter 4 Add User With Validation Demo</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bo
  <style>
    .container {
      max-width: 500px;
    }
    .error {
      display: block;
      padding-top: 5px;
      font-size: 14px;
      color: red;
    }
  </style>
</head>
```

```
  action="<?= site_url('/submit-form') ?>">
    <div class="form-group">
      <label>Name</label>
      <input type="text" name="name" class="form-control">
    </div>
    <div class="form-group">
      <label>Email</label>
      <input type="text" name="email" class="form-control">
    </div>
    <div class="form-group">
      <button type="submit" class="btn btn-primary btn-block">Upda
    </div>
  </form>
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></s
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validat
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validat
<script>
  if ($("#add_create").length > 0) {
    $("#add_create").validate({
      rules: {
        name: {
          required: true,
        },
        email: {
          required: true,
          maxlength: 60,
          email: true,
        },
      },
      messages: {
        name: {
          required: "Name is required.",
        },
        email: {
            maxlength: "The email should be or equal to 60 chars.",
          },
        },
      })
    }
  </script>
</body>
</html>
```

We have covered the following topics in the above code:

- Inserting data into the Database.
- Integrating and using Bootstrap 4 in Codeigniter application.
- Implementing client side form validation in Codeigniter with **jquery.validate.js**.

Show Data List & Delete

In this step, we will fetch data from MySQL database and display data in Codeigniter application using DataTables. Simultaneously we will also look at how to delete a single user from the database.

Generically, DataTables is a jQuery (Javascript library) based table advancement plug-in, It brings coherence in the data analysis process, ideally offers adding, sorting, paging and filtering abilities to plain HTML tables with minimal effort.

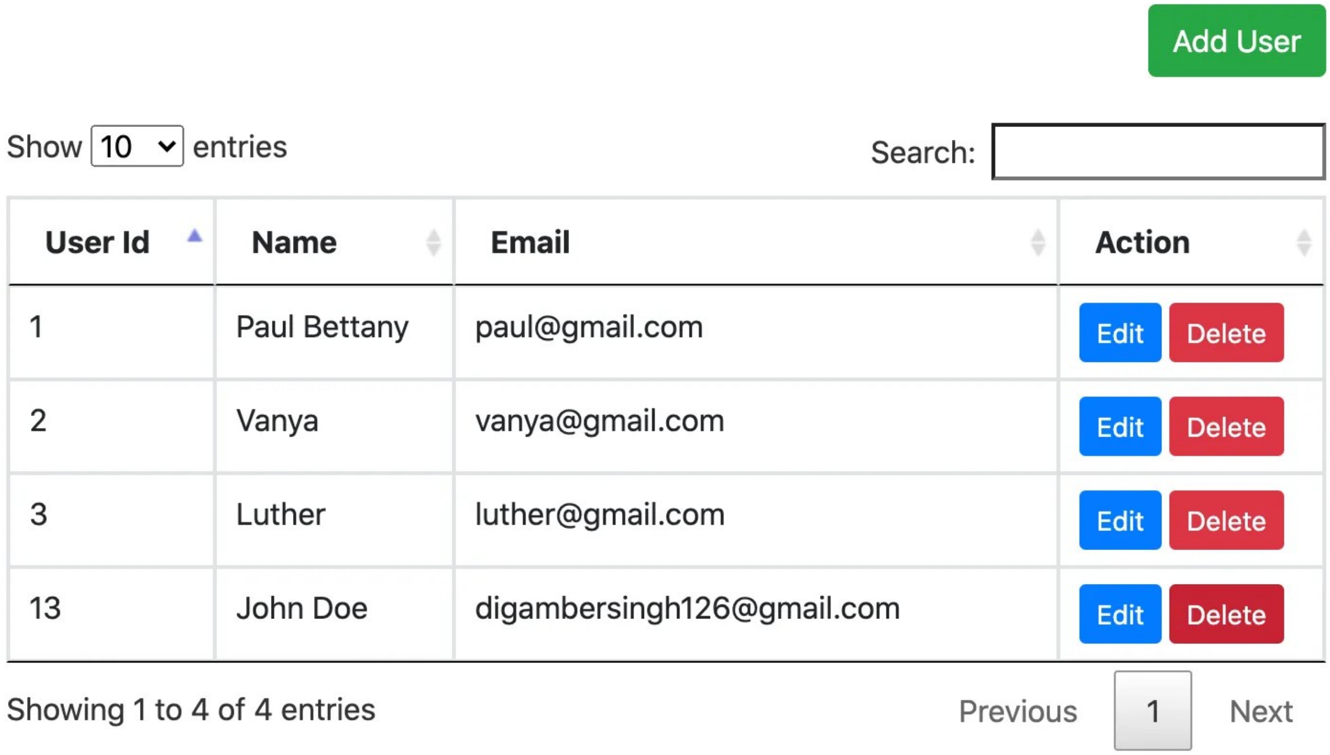
Create **app/Views/user_view.php** file and add the following code inside of it.

```
<!doctype html>
```

```
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1
  <title>Codeigniter 4 CRUD App Example - positronx.io</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bo
</head>
<body>
<div class="container mt-4">
    <div class="d-flex justify-content-end">
        <a href="<?php echo site_url('/user-form') ?>" class="btn bt
    </div>
    <?php
     if(isset($_SESSION['msg'])){
        echo $_SESSION['msg'];
      }
     ?>
  <div class="mt-3">
     <table class="table table-bordered" id="users-list">
       <thead>
          <tr>
             <th>User Id</th>
             <th>Name</th>
             <th>Email</th>
             <th>Action</th>
          </tr>
       </thead>
       <tbody>
          <?php if($users): ?>
          <?php foreach($users as $user): ?>
          <tr>
             <td><?php echo $user['id']; ?></td>
             <td><?php echo $user['name']; ?></td>
             <td><?php echo $user['email']; ?></td>
             <td>
              <a href="<?php echo base_url('edit-view/'.$user['id'])
              <a href="<?php echo base_url('delete/'.$user['id']);?>
```

```
         <?php endforeach; ?>
         <?php endif; ?>
       </tbody>
     </table>
  </div>
</div>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></scr
<link rel="stylesheet" type="text/css" href="https://cdn.datatables.
<script type="text/javascript" src="https://cdn.datatables.net/1.10.
<script>
    $(document).ready( function () {
      $('#users-list').DataTable();
  } );
</script>
</body>
</html>
```



User Id	Name	Email	Action
1	Paul Bettany	paul@gmail.com	Edit Delete
2	Vanya	vanya@gmail.com	Edit Delete
3	Luther	luther@gmail.com	Edit Delete
13	John Doe	digambersingh126@gmail.com	Edit Delete

Showing 1 to 4 of 4 entries

Edit and Update Data

We have to create **app/Views/edit_view.php** file and insert the following code inside of it to update the user data directly in the database.

```
<!DOCTYPE html>
<html>
<head>
  <title>Codeigniter 4 CRUD - Edit User Demo</title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bo
  <style>
    .container {
      max-width: 500px;
    }
    .error {
      display: block;
      padding-top: 5px;
      font-size: 14px;
      color: red;
    }
  </style>
</head>
<body>
  <div class="container mt-5">
    <form method="post" id="update_user" name="update_user"
    action="<?= site_url('/update') ?>">
      <input type="hidden" name="id" id="id" value="<?php echo $user
      <div class="form-group">
        <label>Name</label>
        <input type="text" name="name" class="form-control" value="<
      </div>
      <div class="form-group">
        <label>Email</label>
      <div class="form-group">
        <button type="submit" class="btn btn-danger btn-block">Save
      </div>
    </form>
  </div>
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></s
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validat
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validat
  <script>
    if ($("#update_user").length > 0) {
      $("#update_user").validate({
        rules: {
          name: {
            required: true,
          },
          email: {
            required: true,
            maxlength: 60,
            email: true,
          },
        },
        messages: {
          name: {
            required: "Name is required.",
          },
          email: {
            required: "Email is required.",
            email: "It does not seem to be a valid email.",
            maxlength: "The email should be or equal to 60 chars.",
          },
        },
      })
    }
  </script>
</body>
```

Start the Application

Execute the following command to run this application:

```
php spark serve
```

Here is the URL that you have to enter in the browser's address bar to initiate the CRUD application:

```
http://localhost:8080/index.php/users-list
```

The Bottom Line

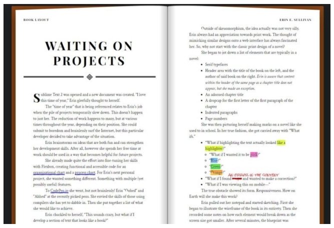
Eventually, we have completed the Codeigniter 4 CRUD Operations tutorial. In this tutorial, we have assorted everything at the right place from the starting and considered all the imperatives.

If you are a beginner in Codeigniter development, then this tutorial thrives your development career.

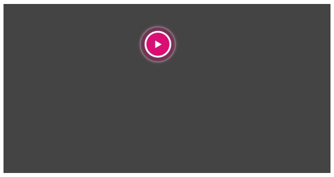
The impetus to create this tutorial comes from the initial days of my Codeigniter development. I hope you must have liked my intensive efforts and will surely share this tutorial with others.

You can download the full code of this tutorial from GitHub, Have a good day.

❤️ Checkout Popular CSS Collection

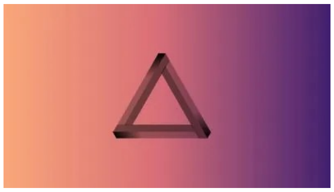


25+ CSS Book Effects



20+ CSS Play/ Pause Buttons

40+ CSS Flip Cards



40+ CSS Triangles

Digamber

Digamber, the founder of PositronX, is a certified web developer with over 10 years of experience. He possesses a curiosity for learning new things and enjoys playing cricket on his days off.

Subscribe Our Newsletter

Email Address *

Subscribe

Social

- Twitter
- GitHub
- LinkedIn
- Dev
- YouTube

Explore

- About
- Affiliate Disclosure
- Privacy Policy
- Disclaimer
- Contact

Categories

- CSS Animations
- React
- Angular
- Vue
- Ionic
- Firebase
- PHP
- Laravel
- CodeIgniter
- Node
- Angular Material
- JavaScript
- React Native