

2021/2022(2)

IF184605 Framework-Based Programming

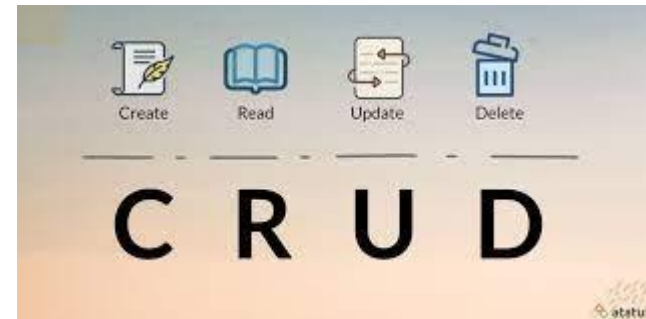
Lecture #7b

Bootstrap, jQuery & CRUD App

Misbakhul Munir **IRFAN SUBAKTI**

司馬伊凡

Мисбакхул Мунир **Ирфан Субакти**



Bootstrap: What is that?

- It's a free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.
- As of August 2021, Bootstrap is the tenth most starred project on GitHub, with over 152,000 stars, behind freeCodeCamp (over 328,000 stars), Vue.js framework, React library, TensorFlow and others.



jQuery: What is that?



- jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax.
- It is free, open-source software using the permissive MIT License.
- As of May 2019, jQuery is used by 73% of the 10 million most popular websites.
- Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having at least 3 to 4 times more usage than any other JavaScript library.

Todo App

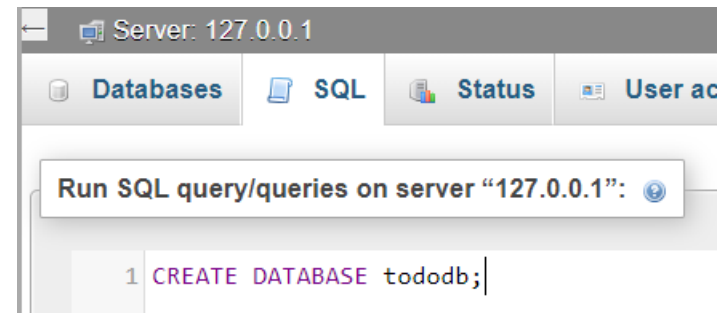
- <https://remotestack.io/create-codeigniter-crud-todo-app-with-mysql-bootstrap>

Create tododb & table todos

```
CREATE DATABASE tododb;
```

```
CREATE TABLE todos (  
  id int(11) NOT NULL AUTO_INCREMENT COMMENT 'Primary Key',  
  title varchar(100) NOT NULL COMMENT 'Title',  
  description varchar(255) NOT NULL COMMENT 'Description',  
  PRIMARY KEY (id)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
```

```
INSERT INTO todos(id, title, description) VALUES  
(1, 'Design Update', 'Send an email update to the team: 9am today'),  
(2, 'Mockup Agency', 'Call the design agency to finalize mockups'),  
(3, 'Recruiters Meeting', 'Touch base with recruiters about new role: Tuesday'),  
(4, 'Engineering Renosance', 'Meet with the engineering team.');
```



Creating table todos

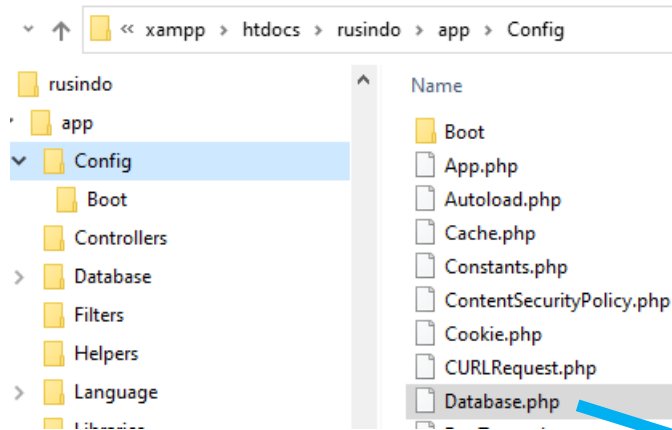
The screenshot shows the phpMyAdmin interface for a database named 'tododb'. The 'SQL' tab is active, displaying the following SQL code:

```
1 CREATE TABLE todos (  
2     id int(11) NOT NULL AUTO_INCREMENT COMMENT 'Primary Key',  
3     title varchar(100) NOT NULL COMMENT 'Title',  
4     description varchar(255) NOT NULL COMMENT 'Description',  
5     PRIMARY KEY (id)  
6 ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;  
7  
8 INSERT INTO todos(id, title, description) VALUES  
9 (1, 'Design Update', 'Send an email update to the team: 9am today'),  
10 (2, 'Mockup Agency', 'Call the design agency to finalize mockups'),  
11 (3, 'Recruiters Meeting', 'Touch base with recruiters about new role: Tuesday'),  
12 (4, 'Engineering Renosance', 'Meet with the engineering team.');
```

Below the SQL editor, the '+ Options' section is expanded, showing a table with 4 rows of data:

| | id | title | description |
|---|----|-----------------------|--|
| <input type="checkbox"/> Edit Copy Delete | 1 | Design Update | Send an email update to the team: 9am today |
| <input type="checkbox"/> Edit Copy Delete | 2 | Mockup Agency | Call the design agency to finalize mockups |
| <input type="checkbox"/> Edit Copy Delete | 3 | Recruiters Meeting | Touch base with recruiters about new role: Tuesday |
| <input type="checkbox"/> Edit Copy Delete | 4 | Engineering Renosance | Meet with the engineering team. |

Database.php: Update



```
Database.php x
33 public $default = [
34     'DSN' => '',
35     'hostname' => 'localhost',
36     'username' => 'root', // updated
37     'password' => '',
38     'database' => 'tododb', // updated
39     'DBDriver' => 'MySQLi',
40     'DBPrefix' => '',
41     'pConnect' => false,
42     // 'DBDebug' => (ENVIRONMENT !== 'production'),
43     'DBDebug' => (ENVIRONMENT !== 'development'), // updated
44     'charset' => 'utf8',
45     'DBCollat' => 'utf8_general_ci',
46     'swapPre' => '',
47     'encrypt' => false,
48     'compress' => false,
49     'strictOn' => false,
50     'failover' => [],
51     'port' => 3306,
52 ];
```


TodoController.php: Build



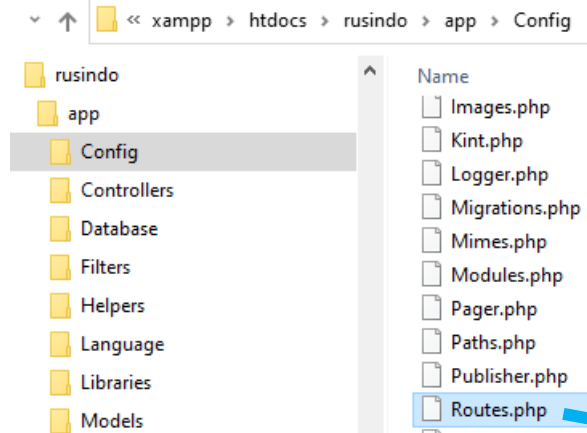
The screenshot shows a file explorer on the left with the following structure:

- rusindo
 - app
 - Config
 - Controllers
 - Database
 - Filters

The main editor displays the code for `TodoController.php`:

```
1 <?php
2 namespace App\Controllers;
3 use CodeIgniter\Controller;
4 use App\Models\TodoModel;
5 class TodoController extends Controller {
6     public function index() {
7         $todoModel = new TodoModel();
8         $data['todos'] = $todoModel->orderBy('id', 'DESC')->findAll();
9         return view('todo_view', $data);
10    }
11    public function create() {
12        return view('add_todo');
13    }
14    public function store() {
15        $todoModel = new todoModel();
16        $data = [
17            'title' => $this->request->getVar('title'),
18            'description' => $this->request->getVar('description')
19        ];
20        $todoModel->insert($data);
21        return $this->response->redirect(site_url('/todo-list'));
22    }
23    public function getTodo($id = null) {
24        $todoModel = new todoModel();
25        $data['todo_obj'] = $todoModel->where('id', $id)->first();
26        return view('edit_todo', $data);
27    }
28    public function update() {
29        $todoModel = new todoModel();
30        $id = $this->request->getVar('id');
31        $data = [
32            'title' => $this->request->getVar('title'),
33            'description' => $this->request->getVar('description')
34        ];
35        $todoModel->update($id, $data);
36        return $this->response->redirect(site_url('/todo-list'));
37    }
38    public function delete($id = null) {
39        $todoModel = new todoModel();
40        $data['todo'] = $todoModel->where('id', $id)->delete($id);
41        return $this->response->redirect(site_url('/todo-list'));
42    }
43 }
```

CRUD Routes: Update



```
Routes.php x
49 $routes->setDefaultController('TodoController');
50 $routes->get('todo-list', 'TodoController::index');
51 $routes->get('todo-form', 'TodoController::create');
52 $routes->post('add-todo', 'TodoController::store');
53 $routes->get('edit-todo/{:num}', 'TodoController::getTodo/{:num}');
54 $routes->post('update-todo', 'TodoController::update');
55 $routes->get('delete-todo/{:num}', 'TodoController::delete/{:num}');
56 // === UPDATED BY IRFAN ===
```

[Create] add_todo.php: Build



The image shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows the directory structure: `rusindo` > `app` > `Views`. The `add_todo.php` file is selected in the `Views` directory, indicated by a blue arrow. The code editor displays the following PHP code:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Codeigniter 4 CRUD Operations Example</title>
6   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
7   <style>
8     .error {
9       display: block;
10      color: red;
11      font-size: 13px;
12    }
13  </style>
14 </head>
15
16 <body>
17   <div class="container mt-3">
18     <form method="post" id="add_todo" name="add_todo"
19       action="<?> site_url('/add-todo') ?>">
20
21     <div class="mb-3">
22       <input type="text" name="title" class="form-control" placeholder="Title">
23     </div>
24
25     <div class="mb-3">
26       <input type="text" name="description" class="form-control" placeholder="Description">
27     </div>
28
```

[Create] add_todo.php: Build (continued)

```
29     <div class="d-grid">
30         <button type="submit" class="btn btn-dark">Create</button>
31     </div>
32 </form>
33 </div>
34
35 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
36 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.2/jquery.validate.js"></script>
37 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.2/additional-methods.min.js"></script>
38 <script>
39     if ($("#add_todo").length > 0) {
40         $("#add_todo").validate({
41             rules: {
42                 title: {
43                     required: true,
44                 },
45                 description: {
46                     required: true,
47                     maxlength: 120,
48                 },
49             }
50         })
51     }
52 </script>
53 </body>
54
55 </html>
```

[Retrieve] todo_view.php: Build



The screenshot displays a file explorer on the left and a code editor on the right. The file explorer shows the directory structure: `rusindo > app > Views`. The `Views` folder is selected, and its contents are listed: `errors`, `news`, `pages`, `templates`, `add_todo.php`, `todo_view.php` (highlighted with a blue box and a blue arrow), and `welcome_message.php`.

The code editor shows the content of `todo_view.php`:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Codeigniter 4 CRUD Operations Example</title>
6   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
7   <style>
8     .error {
9       display: block;
10      color: red;
11      font-size: 13px;
12    }
13  </style>
14 </head>
15
16 <body>
17   <div class="container">
18
19     <div class="d-flex justify-content-end">
20       <a href="<?php echo site_url('/todo-form') ?>" class="btn btn-danger mb-3">Create Todo</a>
21     </div>
22
```

[Retrieve] todo_view.php: Build (continued)

```
23     <?php if(isset($_SESSION['msg'])){
24         echo $_SESSION['msg'];
25     } ?>
26
27     <ul class="list-group">
28         <?php if($todos): ?>
29         <?php foreach($todos as $todo): ?>
30         <li class="list-group-item">
31             <h3><?php echo $todo['title']; ?></h3>
32             <p><?php echo $todo['description']; ?></p>
33             <p>
34                 <a href="<?php echo base_url('edit-todo/'.$todo['id']);?>" class="text-decoration-none">Edit</a>
35                 <a href="<?php echo base_url('delete-todo/'.$todo['id']);?>" class="text-decoration-none">Delete</a>
36             </p>
37         </li>
38         <?php endforeach; ?>
39         <?php endif; ?>
40     </ul>
41
42 </div>
43 </body>
44
45 </html>
```

[Update] edit_todo.php: Build



The image shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows the following structure:

- rusindo
 - app
 - Config
 - Controllers
 - Database
 - Filters
 - Helpers
 - Language
 - Libraries
 - Models
 - ThirdParty
 - Views
 - errors
 - news
 - pages
 - templates
 - add_todo.php
 - edit_todo.php
 - todo_view.php
 - welcome_message.php

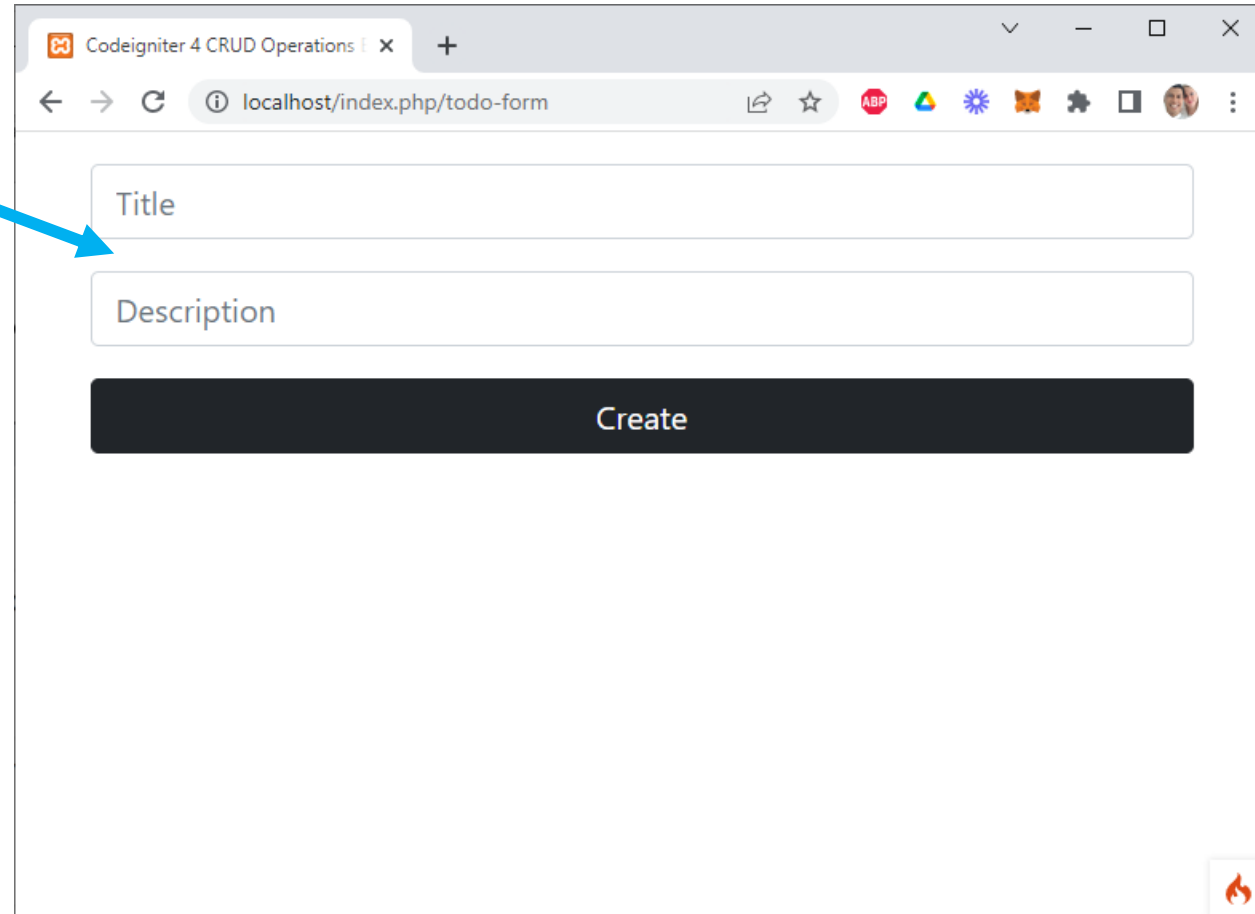
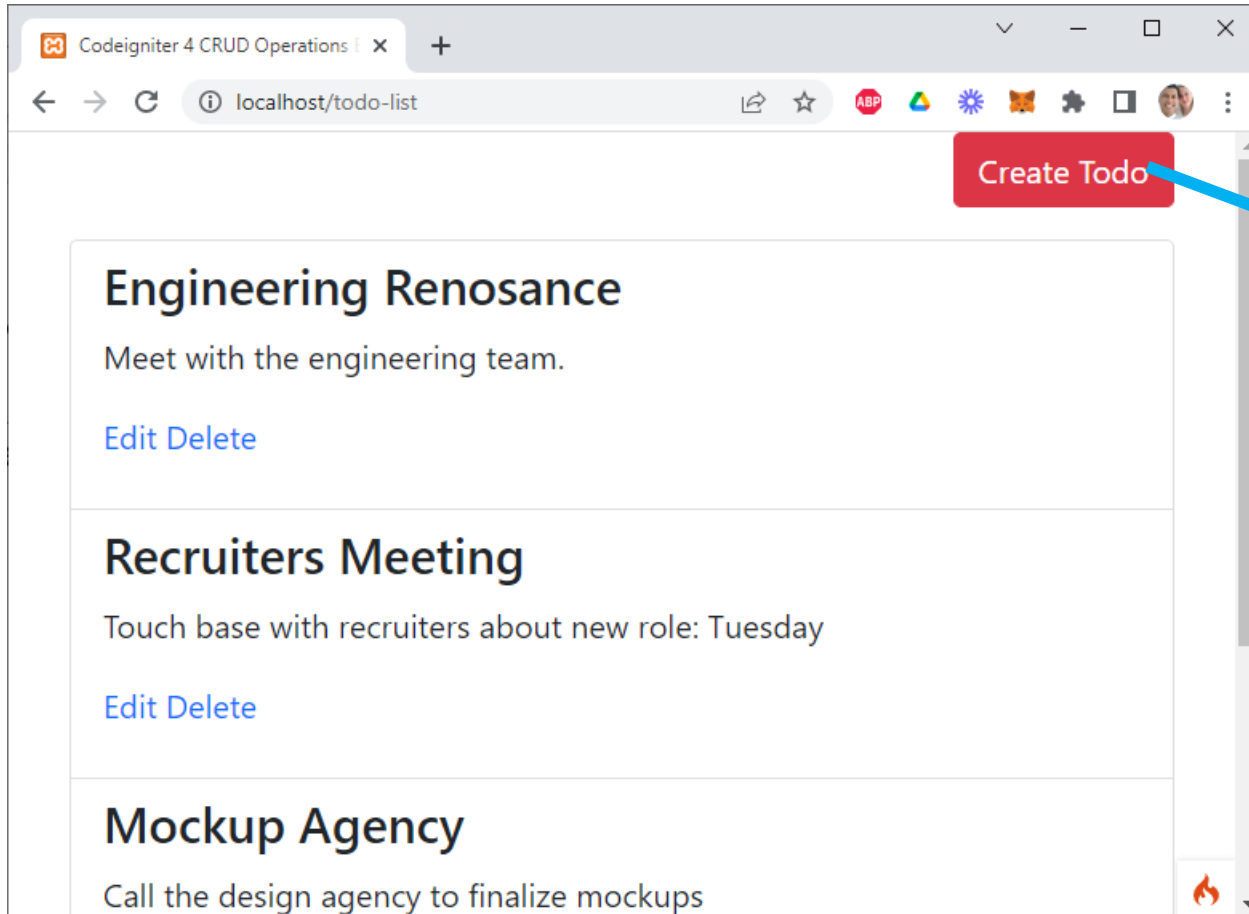
The code editor shows the content of `edit_todo.php`:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Codeigniter 4 TODO CRUD App</title>
6   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">
7   <style>
8     .error {
9       display: block;
10      color: red;
11      font-size: 13px;
12    }
13  </style>
14 </head>
15
16 <body>
17   <div class="container">
18     <form method="post" name="update_todo" id="update_todo" action="{?= site_url('/update-todo')} ?>">
19
20     <input type="hidden" name="id" id="id" value="{?php echo $todo_obj['id']; ?>">
21
22     <div class="mb-3">
23       <input type="text" name="title" class="form-control" value="{?php echo $todo_obj['title']; ?>">
24     </div>
25
26     <div class="mb-3">
27       <input type="text" name="description" class="form-control" value="{?php echo $todo_obj['description']; ?>">
28     </div>
```

[Update] edit_todo.php: Build (continued)

```
29
30     <div class="d-grid">
31         <button type="submit" class="btn btn-dark">Update</button>
32     </div>
33 </form>
34 </div>
35
36 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
37 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.2/jquery.validate.js"></script>
38 <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validate/1.19.2/additional-methods.min.js"></script>
39 <script>
40     if ($("#update_todo").length > 0) {
41         $("#update_todo").validate({
42             rules: {
43                 title: {
44                     required: true,
45                 },
46                 description: {
47                     required: true,
48                     maxlength: 120,
49                 },
50             }
51         })
52     }
53 </script>
54 </body>
55
56 </html>
```


Run & test App



Run & test App (continued)

