

LAPORAN TUGAS PRAKTIKUM PEMROGRAMAN BERBASIS WEB

“Tugas 5 (Project Inventory Management with MVC + Sessions & Cookies)”



Oleh:

Nama : Adam Kaze

NPM : 4522210094

Kelas : A

Dosen:

Adi Wahyu Pribadi , S.Si., M.Kom

S1-Teknik Informatika

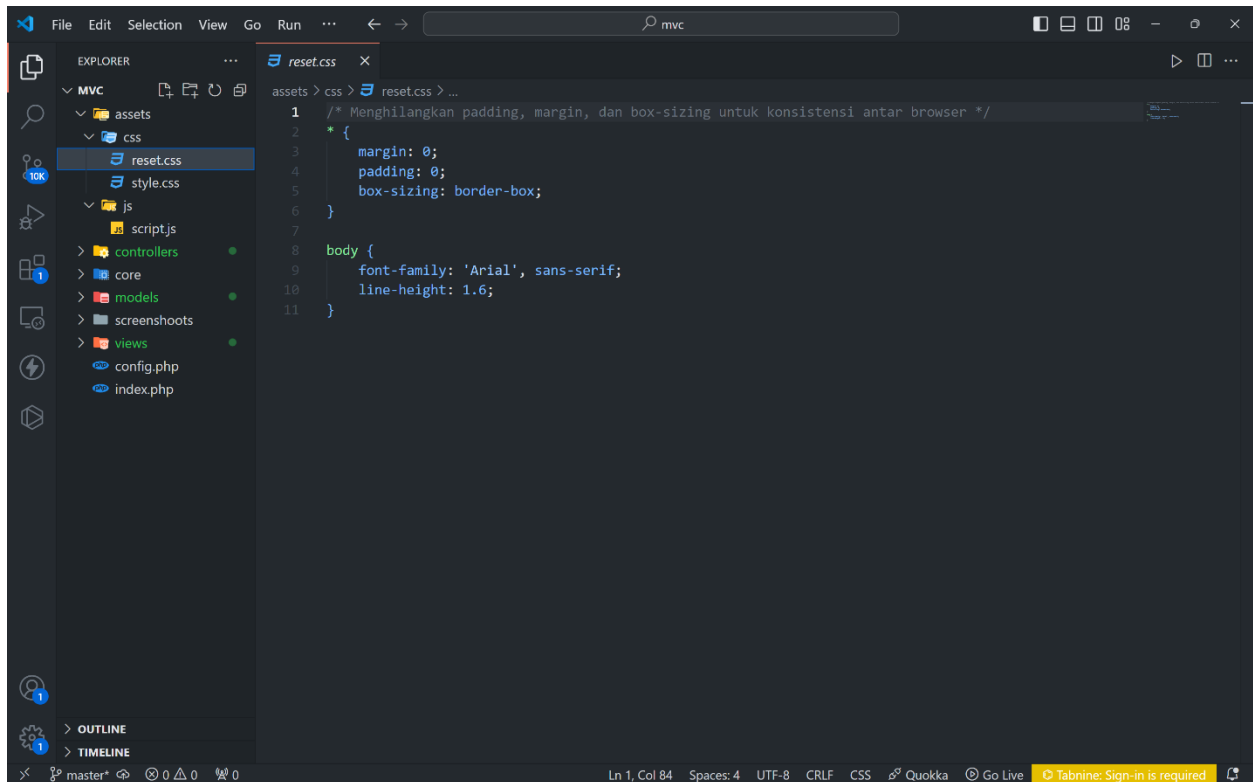
Fakultas Teknik Universitas Pancasila

2024

Tujuan Praktikum

Praktikum kali ini bertujuan untuk memahami cara membuat Aplikasi Manajemen Barang Sebagai Admin dengan konsep MVC, Sessions dan Cookies menggunakan Bahasa pemrograman PHP.

Screenshoot Source Code



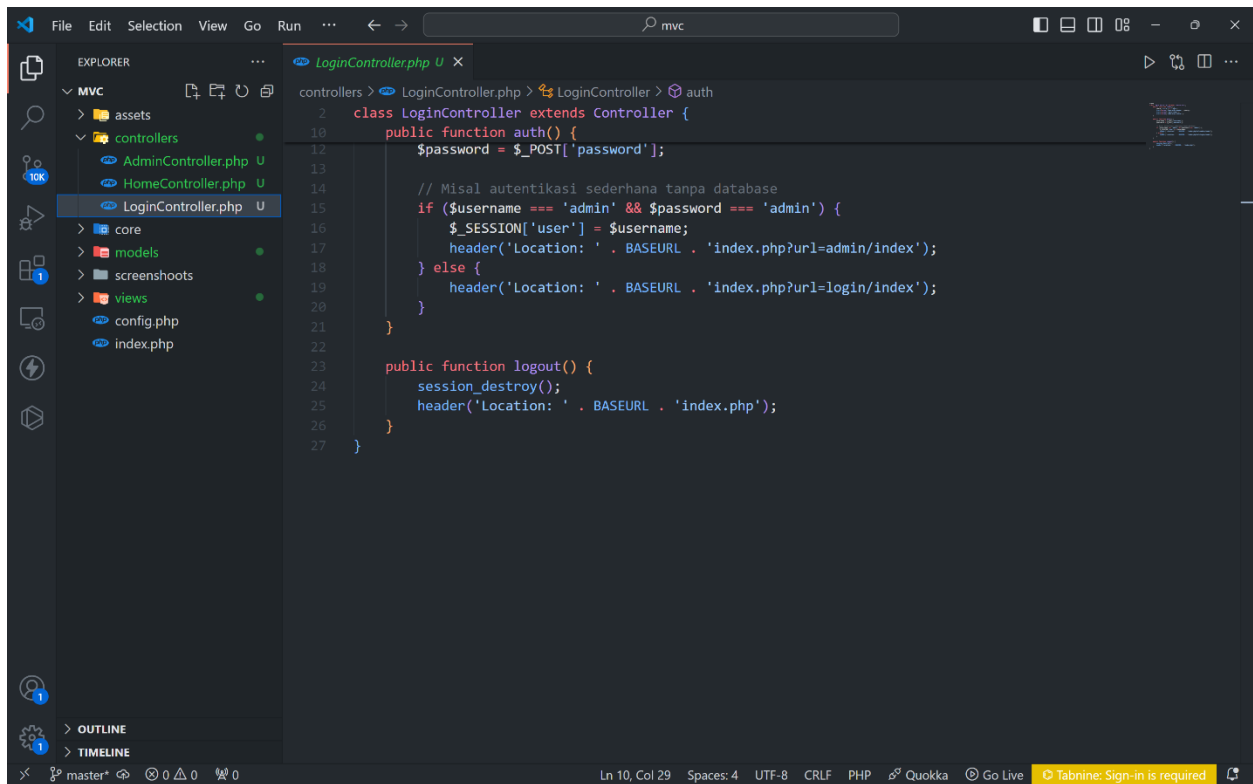
```
1  /* Menghilangkan padding, margin, dan box-sizing untuk konsistensi antar browser */
2  * {
3      margin: 0;
4      padding: 0;
5      box-sizing: border-box;
6  }
7
8  body {
9      font-family: 'Arial', sans-serif;
10     line-height: 1.6;
11 }
```

```
1  /* Global Styles */
2  body {
3    background-color: #f4f4f4;
4    color: #333;
5    font-family: 'Arial', sans-serif;
6  }
7
8  h1, h2 {
9    color: #444;
10   margin-bottom: 20px;
11 }
12
13 .container {
14   width: 80%;
15   margin: 0 auto;
16   padding: 20px;
17 }
18
19 /* Header Styles */
20 header {
21   background-color: #007bff;
22   color: #fff;
23   padding: 10px 0;
24   text-align: center;
25 }
26
27 header h1 {
28   margin: 0;
29 }
30
31 /* Form Styles */
32 form {
33   background-color: #fff;
```

```
1  // Modal confirmation for delete action
2  function confirmDelete(itemId) {
3    return confirm("Apakah Anda yakin ingin menghapus barang dengan ID " + itemId + "?");
4  }
5
6  // Smooth scrolling (optional enhancement)
7  document.querySelectorAll('a[href="#"]').forEach(anchor => {
8    anchor.addEventListener('click', function (e) {
9      e.preventDefault();
10
11      document.querySelector(this.getAttribute('href')).scrollIntoView({
12        behavior: 'smooth'
13      });
14    });
15  });
16
17 document.querySelector('.btn').addEventListener('click', function() {
18   this.innerHTML = 'Loading...';
19   this.style.pointerEvents = 'none'; // Disable button interaction
20 });
```

```
1 class AdminController extends Controller {
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33 public function edit($id) {
34     $data['title'] = 'Edit Barang';
35     $data['item'] = $this->model('ItemModel')->getItemById($id);
36     $this->view('templates/header', $data);
37     $this->view('admin/edit', $data);
38     $this->view('templates/footer');
39 }
40
41 public function update() {
42     if ($this->model('ItemModel')->updateItem($_POST) > 0) {
43         header('Location: ' . BASEURL . 'index.php?url=admin/index');
44     } else {
45         echo "Gagal mengubah data.";
46     }
47 }
48
49 public function delete($id) {
50     if ($this->model('ItemModel')->deleteItem($id) > 0) {
51         header('Location: ' . BASEURL . 'index.php?url=admin/index');
52     } else {
53         echo "Gagal menghapus data.";
54     }
55 }
56 }
```

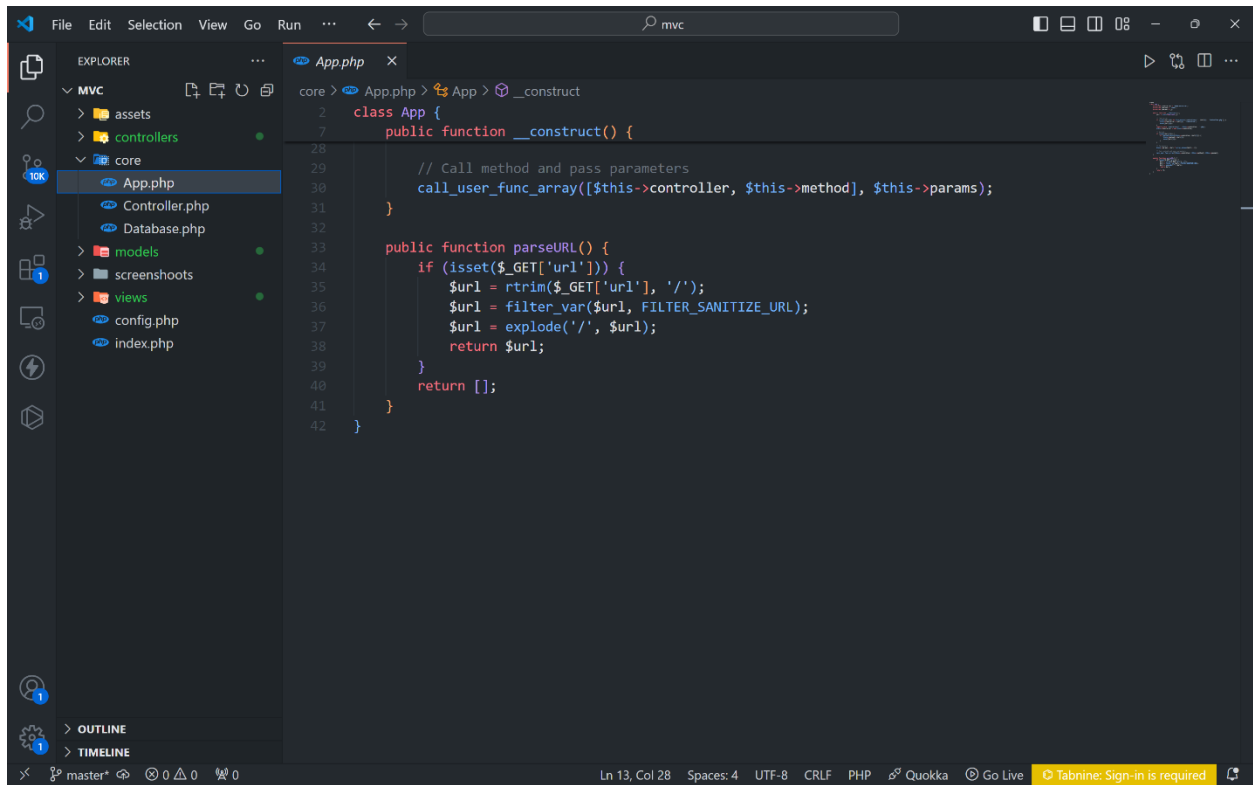
```
1 <?php
2 class HomeController extends Controller {
3     public function index() {
4         $data['title'] = 'Home';
5         $this->view('templates/header', $data);
6         $this->view('home/index', $data);
7         $this->view('templates/footer');
8     }
9 }
```



This screenshot shows the Visual Studio Code editor with the file `LoginController.php` open. The Explorer sidebar on the left shows a project structure with folders `assets`, `controllers`, `core`, `models`, `screenshots`, and `views`. The `controllers` folder contains `AdminController.php`, `HomeController.php`, and `LoginController.php`. The `core` folder contains `config.php` and `index.php`. The `models` folder is empty. The `views` folder is empty. The `controllers` folder is selected, and `LoginController.php` is the active file. The code in the editor is as follows:

```
2 class LoginController extends Controller {
18     public function auth() {
19         $password = $_POST['password'];
20
21         // Misal autentikasi sederhana tanpa database
22         if ($username === 'admin' && $password === 'admin') {
23             $_SESSION['user'] = $username;
24             header('Location: ' . BASEURL . 'index.php?url=admin/index');
25         } else {
26             header('Location: ' . BASEURL . 'index.php?url=login/index');
27         }
28     }
29
30     public function logout() {
31         session_destroy();
32         header('Location: ' . BASEURL . 'index.php');
33     }
34 }
```

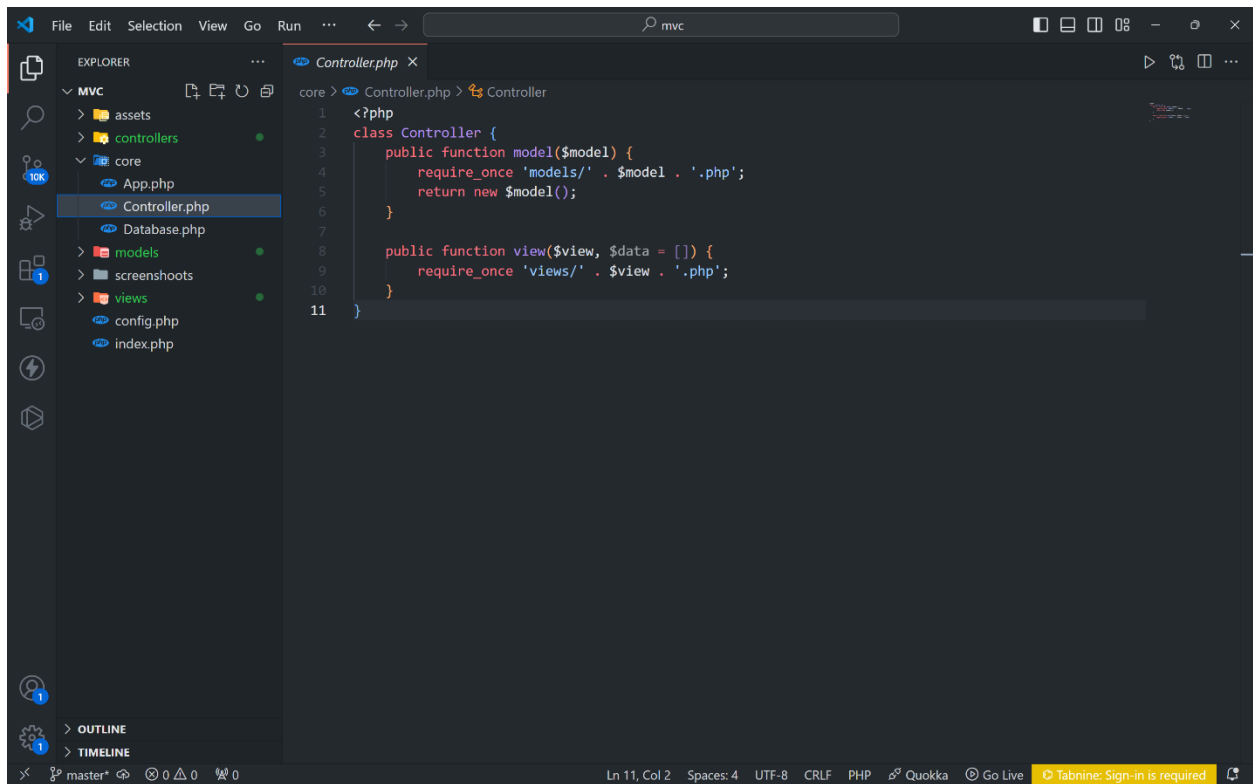
The status bar at the bottom indicates the file is at line 10, column 29, with 4 spaces, UTF-8 encoding, CRLF line endings, and PHP syntax. It also shows the Quokka extension and a Go Live button.



This screenshot shows the Visual Studio Code editor with the file `App.php` open. The Explorer sidebar on the left shows the same project structure as the previous screenshot. The `core` folder is selected, and `App.php` is the active file. The code in the editor is as follows:

```
2 class App {
7     public function __construct() {
8
9         // Call method and pass parameters
10        call_user_func_array([$this->controller, $this->method], $this->params);
11    }
12
13    public function parseURL() {
14        if (isset($_GET['url'])) {
15            $url = rtrim($_GET['url'], '/');
16            $url = filter_var($url, FILTER_SANITIZE_URL);
17            $url = explode('/', $url);
18            return $url;
19        }
20        return [];
21    }
22 }
```

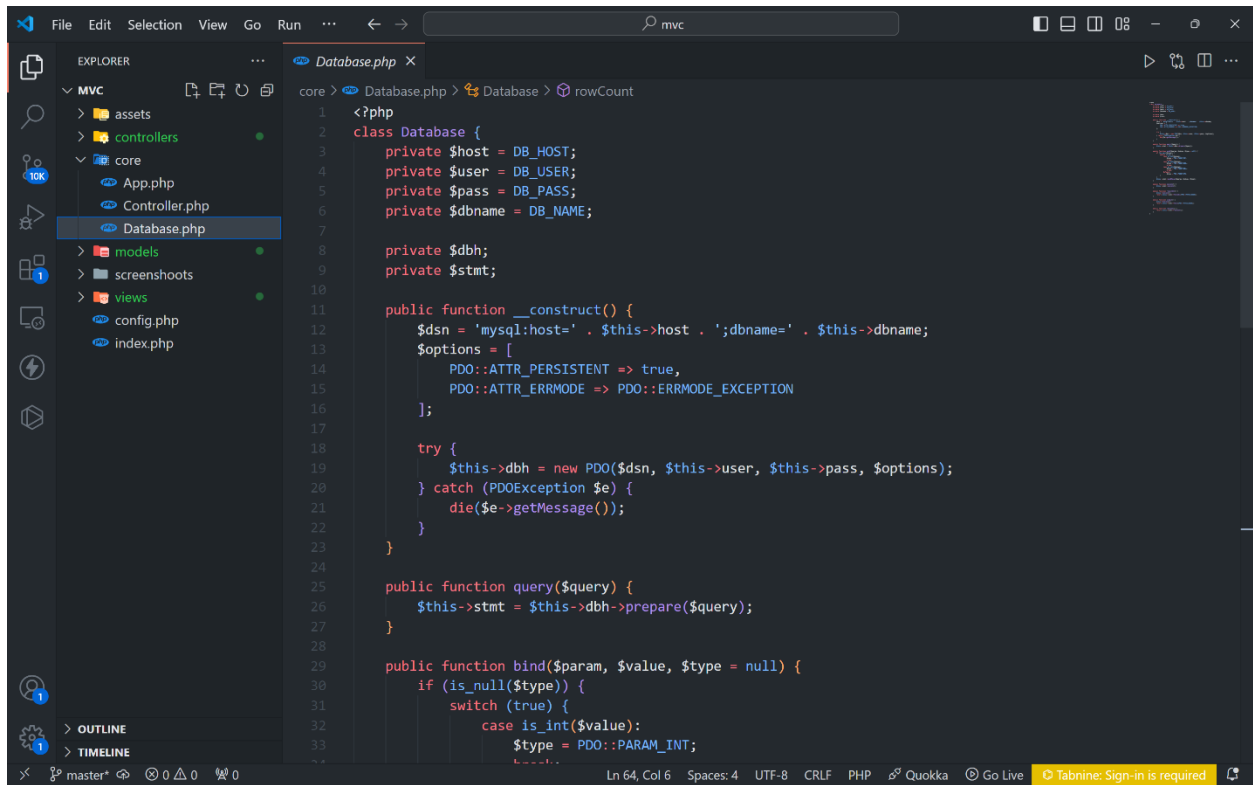
The status bar at the bottom indicates the file is at line 13, column 28, with 4 spaces, UTF-8 encoding, CRLF line endings, and PHP syntax. It also shows the Quokka extension and a Go Live button.



This screenshot shows the Visual Studio Code editor with the file `Controller.php` open. The Explorer sidebar on the left shows a project structure with folders `assets`, `controllers`, `core`, `models`, and `views`, and files `App.php`, `Controller.php`, `Database.php`, `config.php`, and `index.php`. The `Controller.php` file contains the following PHP code:

```
1 <?php
2 class Controller {
3     public function model($model) {
4         require_once 'models/' . $model . '.php';
5         return new $model();
6     }
7
8     public function view($view, $data = []) {
9         require_once 'views/' . $view . '.php';
10    }
11 }
```

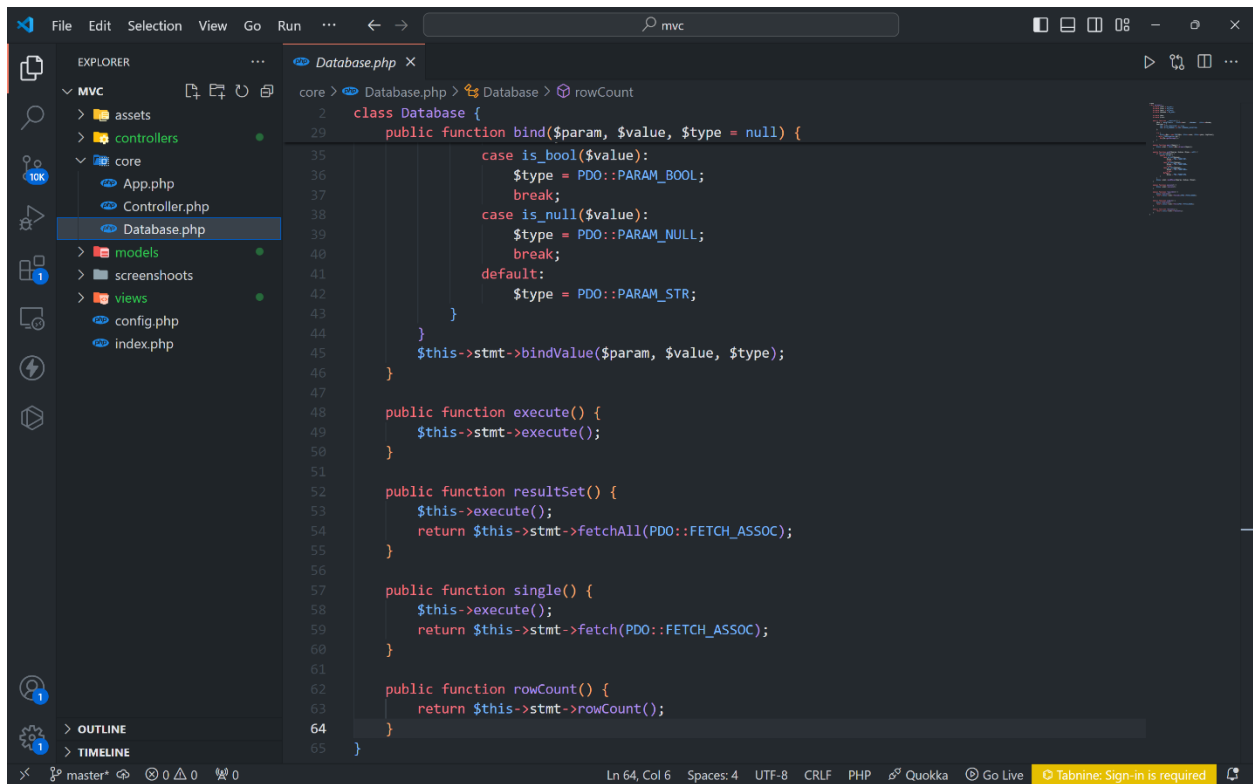
The status bar at the bottom indicates the cursor is at line 11, column 2, with 4 spaces, UTF-8 encoding, CRLF line endings, and the PHP language mode.



This screenshot shows the Visual Studio Code editor with the file `Database.php` open. The Explorer sidebar on the left shows the same project structure as the previous screenshot. The `Database.php` file contains the following PHP code:

```
1 <?php
2 class Database {
3     private $host = DB_HOST;
4     private $user = DB_USER;
5     private $pass = DB_PASS;
6     private $dbname = DB_NAME;
7
8     private $dbh;
9     private $stmt;
10
11    public function __construct() {
12        $dsn = 'mysql:host=' . $this->host . ';dbname=' . $this->dbname;
13        $options = [
14            PDO::ATTR_PERSISTENT => true,
15            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
16        ];
17
18        try {
19            $this->dbh = new PDO($dsn, $this->user, $this->pass, $options);
20        } catch (PDOException $e) {
21            die($e->getMessage());
22        }
23    }
24
25    public function query($query) {
26        $this->stmt = $this->dbh->prepare($query);
27    }
28
29    public function bind($param, $value, $type = null) {
30        if (is_null($type)) {
31            switch (true) {
32                case is_int($value):
33                    $type = PDO::PARAM_INT;
```

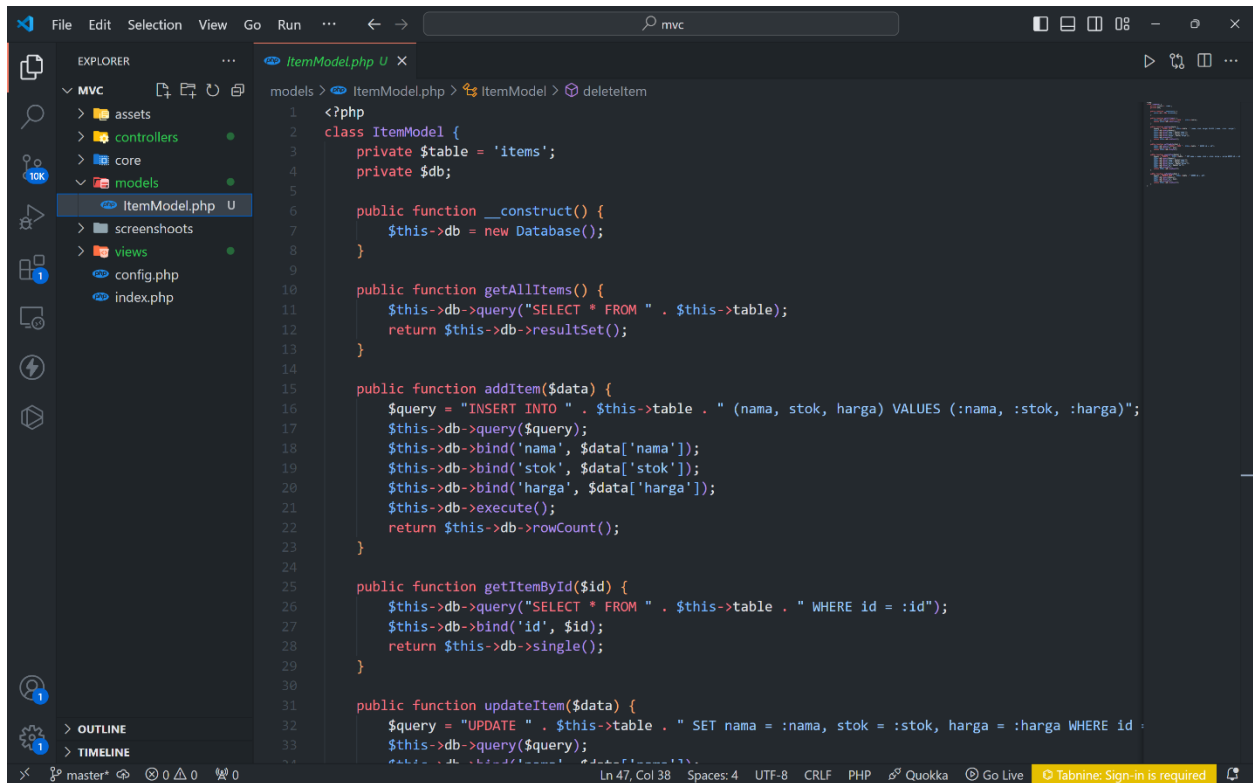
The status bar at the bottom indicates the cursor is at line 64, column 6, with 4 spaces, UTF-8 encoding, CRLF line endings, and the PHP language mode.



The screenshot shows the Visual Studio Code editor with the `Database.php` file open. The Explorer sidebar on the left shows the project structure with folders `assets`, `controllers`, `core`, `models`, `screenshots`, and `views`. The `Database.php` file is selected under `core`. The main editor displays the following PHP code:

```
core > Database.php > Database > rowCount
2  class Database {
29     public function bind($param, $value, $type = null) {
35         case is_bool($value):
36             $type = PDO::PARAM_BOOL;
37             break;
38         case is_null($value):
39             $type = PDO::PARAM_NULL;
40             break;
41         default:
42             $type = PDO::PARAM_STR;
43     }
44     $this->stmt->bindValue($param, $value, $type);
45 }
46
47     public function execute() {
48         $this->stmt->execute();
49     }
50
51     public function resultSet() {
52         $this->execute();
53         return $this->stmt->fetchAll(PDO::FETCH_ASSOC);
54     }
55
56     public function single() {
57         $this->execute();
58         return $this->stmt->fetch(PDO::FETCH_ASSOC);
59     }
60
61     public function rowCount() {
62         return $this->stmt->rowCount();
63     }
64 }
65 }
```

The status bar at the bottom indicates the file is at line 64, column 6, with 4 spaces, UTF-8 encoding, CRLF line endings, and PHP syntax. It also shows the Quokka extension and a Go Live button.



The screenshot shows the Visual Studio Code editor with the `ItemModel.php` file open. The Explorer sidebar on the left shows the project structure with folders `assets`, `controllers`, `core`, `models`, `screenshots`, and `views`. The `ItemModel.php` file is selected under `models`. The main editor displays the following PHP code:

```
models > ItemModel.php > ItemModel > deleteItem
1  <?php
2  class ItemModel {
3      private $table = 'items';
4      private $db;
5
6      public function __construct() {
7          $this->db = new Database();
8      }
9
10     public function getAllItems() {
11         $this->db->query("SELECT * FROM " . $this->table);
12         return $this->db->resultSet();
13     }
14
15     public function addItem($data) {
16         $query = "INSERT INTO " . $this->table . " (nama, stok, harga) VALUES (:nama, :stok, :harga)";
17         $this->db->query($query);
18         $this->db->bind('nama', $data['nama']);
19         $this->db->bind('stok', $data['stok']);
20         $this->db->bind('harga', $data['harga']);
21         $this->db->execute();
22         return $this->db->rowCount();
23     }
24
25     public function getItemById($id) {
26         $this->db->query("SELECT * FROM " . $this->table . " WHERE id = :id");
27         $this->db->bind('id', $id);
28         return $this->db->single();
29     }
30
31     public function updateItem($data) {
32         $query = "UPDATE " . $this->table . " SET nama = :nama, stok = :stok, harga = :harga WHERE id = :id";
33         $this->db->query($query);
34     }
35 }
```

The status bar at the bottom indicates the file is at line 47, column 38, with 38 spaces, UTF-8 encoding, CRLF line endings, and PHP syntax. It also shows the Quokka extension and a Go Live button.

```
2 class ItemModel {
15 public function addItem($data) {
19     $this->db->bind('stok', $data['stok']);
20     $this->db->bind('harga', $data['harga']);
21     $this->db->execute();
22     return $this->db->rowCount();
23 }
24
25 public function getItemById($id) {
26     $this->db->query("SELECT * FROM " . $this->table . " WHERE id = :id");
27     $this->db->bind('id', $id);
28     return $this->db->single();
29 }
30
31 public function updateItem($data) {
32     $query = "UPDATE " . $this->table . " SET nama = :nama, stok = :stok, harga = :harga WHERE id = :id";
33     $this->db->query($query);
34     $this->db->bind('nama', $data['nama']);
35     $this->db->bind('stok', $data['stok']);
36     $this->db->bind('harga', $data['harga']);
37     $this->db->bind('id', $data['id']);
38     $this->db->execute();
39     return $this->db->rowCount();
40 }
41
42 public function deleteItem($id) {
43     $query = "DELETE FROM " . $this->table . " WHERE id = :id";
44     $this->db->query($query);
45     $this->db->bind('id', $id);
46     $this->db->execute();
47     return $this->db->rowCount();
48 }
49 }
```

```
1 <h2>Tambah Barang</h2>
2 <form action="php echo BASEURL; ?/index.php?url=admin/store" method="POST">
3     <label for="nama">Nama:</label><br>
4     <input type="text" name="nama" id="nama" required><br><br>
5     <label for="stok">Stok:</label><br>
6     <input type="number" name="stok" id="stok" required><br><br>
7     <label for="harga">Harga:</label><br>
8     <input type="number" name="harga" id="harga" required><br><br>
9     <button type="submit">Simpan</button>
10 </form>
```


File Edit Selection View Go Run ...

EXPLORER

- MVC
 - assets
 - controllers
 - core
 - models
 - ItemModel.php U
 - screenshots
 - views
 - Admin
 - add.php U
 - edit.php U
 - index.php U
 - home
 - index.php U
 - Login
 - index.php U
 - templates
 - footer.php U
 - header.php U
 - config.php
 - index.php

OUTLINE

TIMELINE

views > Admin > edit.php > form

```
1 <h2>Edit Barang</h2>
2 <form action="?= BASEURL; ?>index.php?url=admin/update" method="POST">
3   <input type="hidden" name="id" value="?= $data['item']['id']; ?>"
4   <label for="nama">Nama:</label><br>
5   <input type="text" name="nama" id="nama" value="?= $data['item']['nama']; ?>" required><br><br>
6   <label for="stok">Stok:</label><br>
7   <input type="number" name="stok" id="stok" value="?= $data['item']['stok']; ?>" required><br><br>
8   <label for="harga">Harga:</label><br>
9   <input type="number" name="harga" id="harga" value="?= $data['item']['harga']; ?>" required><br><br>
10  <button type="submit">Update</button>
11 </form>
```

Ln 11, Col 8 Spaces: 4 UTF-8 CRLF PHP Quokka Go Live Tabnine: Sign-in is required

File Edit Selection View Go Run ...

EXPLORER

- MVC
 - assets
 - controllers
 - core
 - models
 - ItemModel.php U
 - screenshots
 - views
 - Admin
 - add.php U
 - edit.php U
 - index.php U
 - home
 - index.php U
 - Login
 - index.php U
 - templates
 - footer.php U
 - header.php U
 - config.php
 - index.php

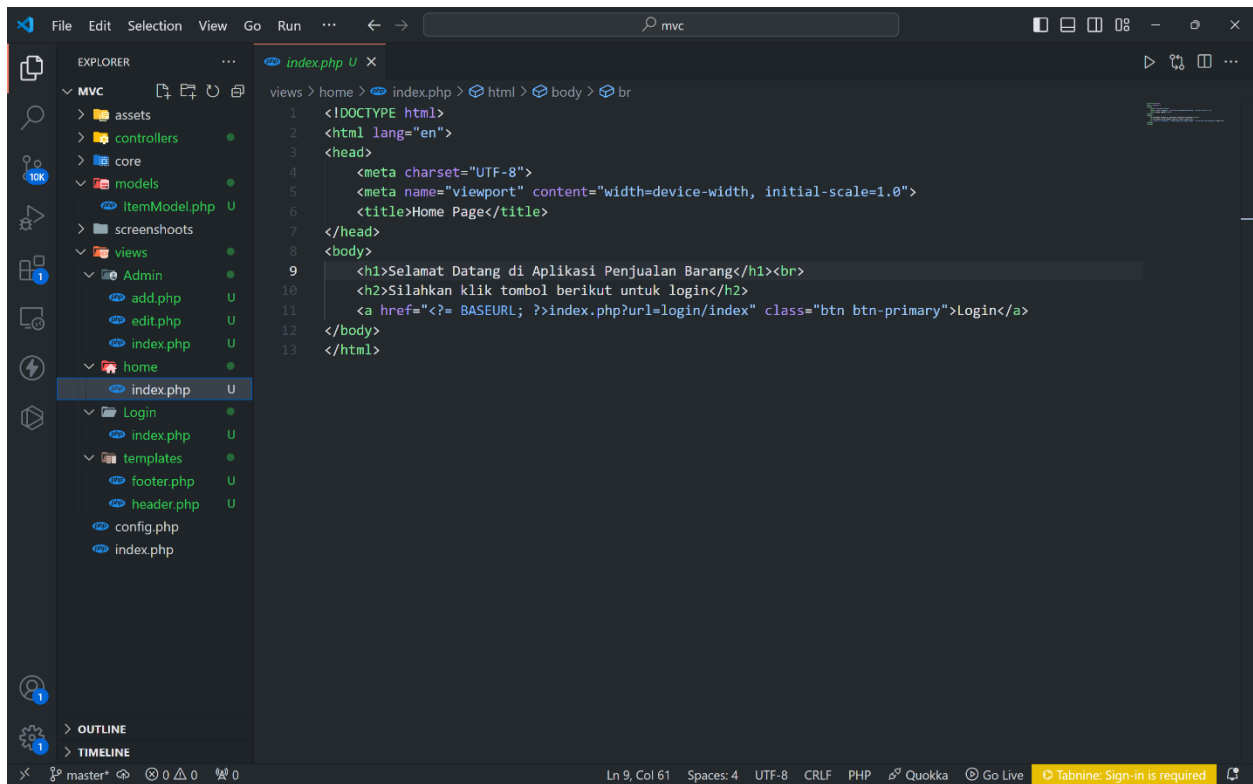
OUTLINE

TIMELINE

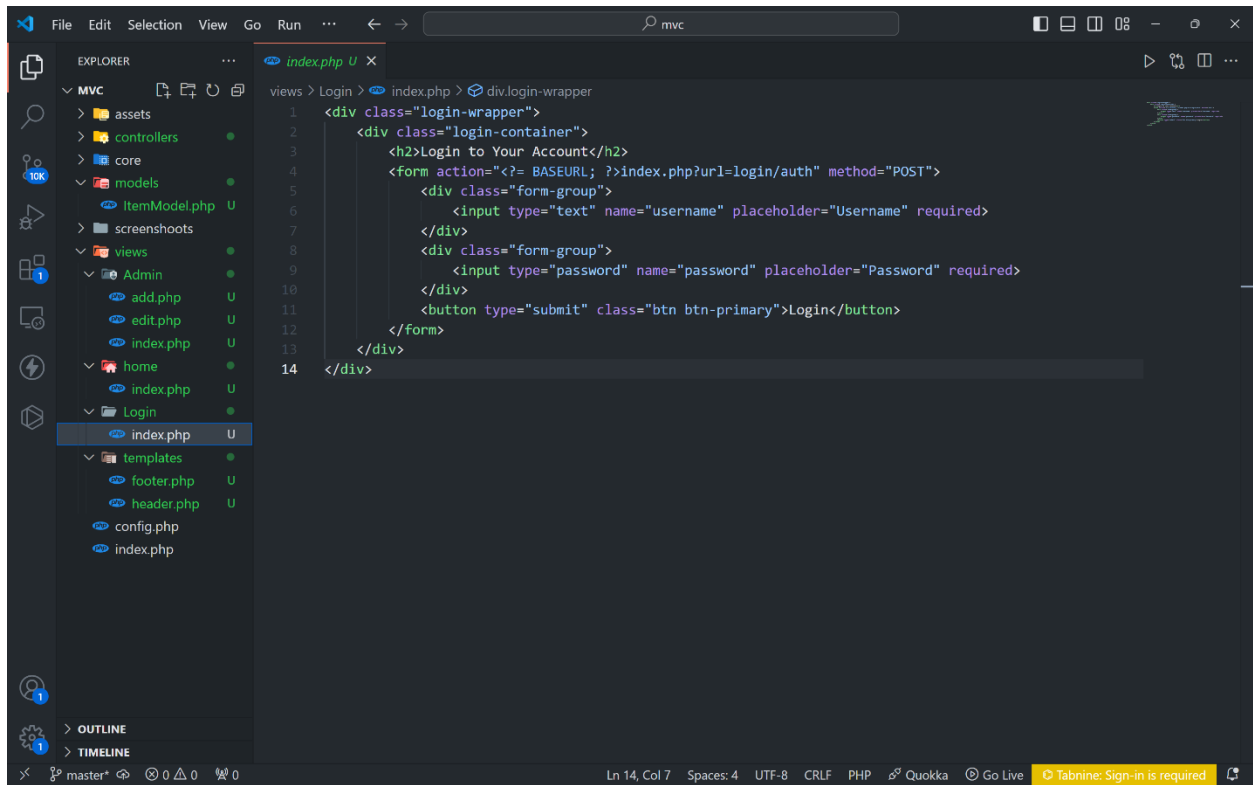
views > Admin > index.php > table > tr

```
2 <a href="?= BASEURL; ?>index.php?url=admin/add">Tambah Barang</a> | <a href="?= BASEURL; ?>index.php
3
4 <table border="1" cellpadding="10" cellspacing="0">
5   <tr>
6     <th>ID</th>
7     <th>Nama</th>
8     <th>Stok</th>
9     <th>Harga</th>
10    <th>Aksi</th>
11  </tr>
12  <?php foreach ($data['items'] as $item): ?>
13    <tr>
14      <td>?=$item['id']; ?></td>
15      <td>?=$item['nama']; ?></td>
16      <td>?=$item['stok']; ?></td>
17      <td>?=$item['harga']; ?></td>
18      <td>
19        <a href="?= BASEURL; ?>index.php?url=admin/edit/<?=$item['id']; ?>">Edit</a> |
20        <a href="?= BASEURL; ?>index.php?url=admin/delete/<?=$item['id']; ?>" onclick="return co
21      </td>
22    </tr>
23  <?php endforeach; ?>
</table>
```

Ln 12, Col 9 Spaces: 4 UTF-8 CRLF PHP Quokka Go Live Tabnine: Sign-in is required



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Home Page</title>
7 </head>
8 <body>
9   <h1>Selamat Datang di Aplikasi Penjualan Barang</h1><br>
10  <h2>Silahkan klik tombol berikut untuk login</h2>
11  <a href="<?= BASEURL; ?>index.php?url=login/index" class="btn btn-primary">Login</a>
12 </body>
13 </html>
```



```
1 <div class="login-wrapper">
2   <div class="login-container">
3     <h2>Login to Your Account</h2>
4     <form action="<?= BASEURL; ?>index.php?url=login/auth" method="POST">
5       <div class="form-group">
6         <input type="text" name="username" placeholder="Username" required>
7       </div>
8       <div class="form-group">
9         <input type="password" name="password" placeholder="Password" required>
10      </div>
11      <button type="submit" class="btn btn-primary">Login</button>
12    </form>
13  </div>
14 </div>
```

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders: assets, controllers, core, models, screenshots, views, and templates. The templates folder is expanded, showing files: footer.php, header.php, config.php, and index.php. The footer.php file is selected and its content is displayed in the main editor area. The content of footer.php is:

```
1 </body>
2 </html>
```

The status bar at the bottom indicates the file is at line 2, column 8 (16 selected), with 4 spaces, UTF-8 encoding, CRLF line endings, and PHP language. The status bar also shows a Tabnine sign-in prompt.

The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders: assets, controllers, core, models, screenshots, views, and templates. The templates folder is expanded, showing files: footer.php, header.php, config.php, and index.php. The header.php file is selected and its content is displayed in the main editor area. The content of header.php is:

```
2 <html>
3 <head>
4 <title><?= $data['title']; ?></title>
5 <link rel="stylesheet" href="<?= BASEURL; ?>assets/css/reset.css">
6 <link rel="stylesheet" href="<?= BASEURL; ?>assets/css/style.css">
7 <script src="<?= BASEURL; ?>assets/js/script.js" defer></script>
8 </head>
9 <body>
10 <header>
11 | <h1><?= $data['title']; ?></h1>
12 </header>
13 <div class="container">
```

The status bar at the bottom indicates the file is at line 12, column 14, with 4 spaces, UTF-8 encoding, CRLF line endings, and PHP language. The status bar also shows a Tabnine sign-in prompt.

```
1 <?php
2 define('DB_HOST', 'localhost');
3 define('DB_USER', 'root');
4 define('DB_PASS', '');
5 define('DB_NAME', 'projectmvc');
6
7 // BASEURL untuk memudahkan penulisan URL
8 define('BASEURL', 'http://localhost/mvc/');
```

Ln 8, Col 44 Spaces: 4 UTF-8 CRLF PHP Quokka Go Live Tabnine: Sign-in is required

```
1 <?php
2 session_start();
3 require_once 'config.php';
4 require_once 'core/App.php';
5 require_once 'core/Controller.php';
6 require_once 'core/Database.php';
7
8 $app = new App();
```

Ln 8, Col 18 Spaces: 4 UTF-8 CRLF PHP Quokka Go Live Tabnine: Sign-in is required

Screenshoot Hasil Tampilan

