

# Лабораторна робота №18

## Створення контролерів для обробки користувацьких запитів

### Хід роботи

```
1 <?php
2 class MaterialController extends Controller {
3     public function index() {
4         $materialsModel = new Material($this->pdo, 'materials');
5         $materials = $materialsModel->all();
6         $this->renderMaterials($materials);
7     }
8
9     public function show($id) {
10        $materialsModel = new Material($this->pdo, 'materials');
11        $material = $materialsModel->find($id);
12        $this->renderMaterial($material);
13    }
14
15    protected function renderMaterials($materials) {
16        echo "<h1>Material List</h1>";
17        echo "<table border='1'>";
18        echo "<tr><th>ID</th><th>Product</th><th>Amount</th></tr>";
19
20        foreach ($materials as $material) {
21            echo "<tr>";
22            echo "<td>" . $material['id'] . "</td>";
23            echo "<td>" . $material['product'] . "</td>";
24            echo "<td>" . $material['amount'] . "</td>";
25            echo "</tr>";
26        }
27
28        echo "</table>";
29    }
30
31    protected function renderMaterial($material) {
32        echo "<h1>Material Details</h1>";
33        echo "<table border='1'>";
34        echo "<tr><th>ID</th><th>Product</th><th>Amount</th></tr>";
35
36        if ($material != null) {
37            echo "<tr>";
38            echo "<td>" . $material['id'] . "</td>";
39            echo "<td>" . $material['product'] . "</td>";
40            echo "<td>" . $material['amount'] . "</td>";
41            echo "</tr>";
42        } else {
43            echo "<tr><td colspan='3'>Material not found</td></tr>";
44        }
45
46        echo "</table>";
47    }
48 }
49 ?>
50
```

```
1 <?php
2 class UserController extends Controller {
3     public function index() {
4         $userModel = new User($this->pdo, 'users');
5         $users = $userModel->all();
6         $this->renderUsers($users);
7     }
8
9     public function show($id) {
10        $userModel = new User($this->pdo, 'users');
11        $user = $userModel->find($id);
12        $this->renderUser($user);
13    }
14
15    protected function renderUsers($users) {
16        echo "<h1>User List</h1>";
17        echo "<table border='1'>";
18        echo "<tr><th>ID</th><th>Username</th><th>Email</th></tr>";
19
20        foreach ($users as $user) {
21            echo "<tr>";
22            if ($user != null) {
23                echo "<td>" . $user['id'] . "</td>";
24                echo "<td>" . $user['username'] . "</td>";
25                echo "<td>" . $user['email'] . "</td>";
26            } else {
27                echo "<td colspan='3'>User not found</td>";
28            }
29            echo "</tr>";
30        }
31
32        echo "</table>";
33    }
34
35    protected function renderUser($user) {
36        if ($user != null) {
37            echo "<td>" . $user['id'] . "</td>";
38            echo "<td>" . $user['username'] . "</td>";
39            echo "<td>" . $user['email'] . "</td>";
40        } else {
41            echo "User not found";
42        }
43    }
44 }
45 ?>
```

					Лабораторна робота № 18					
Зм	Лист	№ докум	Підпис	Дата						
Розробив	Убоженко				Створення контроллерів для обробки користувацьких запитів			Літ.	Лист.	Листів.
Перевірів	Левицький									
								Група 451		
Оцінка										
Затв										

## Контрольні питання

1. Що таке контролер в контексті веб-розробки?

У веб-розробці контролер – це компонент, відповідальний за обробку HTTP-запитів, взаємодіючи з моделями та видами. Контролер призначений для керування логікою обробки запитів і подій веб-додатку.

2. Яка роль контролера в архітектурі MVC (Model-View-Controller)?

У парадигмі MVC, контролер відповідає за прийняття вхідних запитів від користувача, обробку цих запитів, та ініціювання відповідних змін у моделях та відображеннях (видах). Він виконує роль по-середника між моделями, які представляють дані, і видами, які відображають інформацію для користувача.

3. Які завдання можуть виконувати контролери в процесі обробки HTTP-запитів?

Контролери можуть виконувати різні завдання, такі як:

- Прийом та аналіз вхідних HTTP-запитів.
- Взаємодія з моделями для отримання чи збереження даних.
- Ініціювання відображення відповідей для користувача.
- Обробка подій та виконання бізнес-логіки.

4. Яка різниця між функціональними контролерами та класами-контролерами в контексті веб-розробки?

Функціональні контролери представляють собою прості функції, що виконують логіку обробки запитів. Класи-контролери, навпаки, це об'єкти класів, які можуть містити більше складну логіку та забезпечувати можливості об'єктно-орієнтованого програмування.

5. Які методи можуть бути присутні в класі-контролері, і що вони роблять?

Методи в класі-контролері можуть включати, наприклад:

- Методи для обробки різних видів HTTP-запитів (GET, POST).
- Методи для виконання бізнес-логіки та інтеракції з моделями.
- Методи для ініціювання та відображення відповідей користувачу.

6. Чому використання контролерів важливо для правильної організації логіки веб-додатку?

Контролери важливі для відокремлення логіки обробки запитів від логіки відображення та роботи з даними. Вони сприяють створенню масштабованих та легко змінюваних веб-додатків, де різні компоненти виконують свої конкретні функції.

7. Як контролери взаємодіють з моделями та видами у парадигмі MVC?

Контролери взаємодіють з моделями, щоб отримати чи зберегти дані, і з видами для ініціювання відображення інформації користувачу. Вони служать посередниками між цими двома компонентами, забезпечуючи правильний обмін даними та керуючи потоком веб-додатку.

					<b>Лабораторна робота № 18</b>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		