

## Хід роботи

Username: Edamenko

Password:

Login

## Materials

ID	Product	Amount	Action
1	wood	100	Delete
2	nails	97	Delete
3	brick	234	Delete
4	stone	78	Delete

New Material record

[Back to Login](#)

## Create New Material

ID: 

Product:	
----------	--

Amount:

Create

[Back to Materials](#)

[Back to Login](#)

```

C:\xampp\htdocs> Notepad.php
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 </head>
7 <html:login/>
8 <body>
9
10 </body>
11
12 session_start();
13
14 include "core/config.php";
15 include "core/connection.php";
16 include_once "core/Model.php";
17 include_once "Model/User.php";
18 include "Model/Material.php";
19
20 if (($_SERVER['REQUEST_METHOD'] == 'POST') &&
21     $username = $_POST['username'];
22     $password = $_POST['password'];
23
24     $user = UserModel::authenticate($username, $password);
25
26     if ($user) {
27         $_SESSION['user'] = $user;
28         header('location: materials.php');
29         exit();
30     } else {
31         $error = "Invalid username or password";
32     }
33 }
34 </?>
35 </?>
36 </?>
37 </?>
38 </?>
39
40 <form method="post" action="">
41     <label for="username">Username:</label>
42     <input type="text" id="username" name="username" required/>
43
44     <label for="password">Password:</label>
45     <input type="password" id="password" name="password" required/>
46
47     <input type="submit" value="login">
48
49 </form>
50 </body>
51 </html>

```

					<i>Лабораторна робота № 17</i>										
Зм	Лист	№ докум	Підпис	Дата	Створення класу визначеної моделі					Літ.		Лист.		Листів.	
Розробив	Єдаменко														
Перевірів	Левицький														
Оцінка															
Затв										Група 451					

```

C:\xampp\htdocs> core > Model.php
1 <?php
2 class Model {
3     static $pdo;
4     static $table;
5
6     public function __construct(PDO $pdo, $table) {
7         $this->pdo = $pdo;
8         $this->table = $table;
9     }
10
11    public function all() {
12        $sql = "SELECT * FROM $this->table";
13        $stmt = $this->pdo->query($sql);
14        return $stmt->fetchAll(PDO::FETCH_ASSOC);
15    }
16
17    public function getId($id) {
18        $sql = "SELECT * FROM $this->table WHERE id = :id";
19        $stmt = $this->pdo->prepare($sql);
20        $stmt->bindParam(':id', $id);
21        $stmt->execute();
22        return $stmt->fetch(PDO::FETCH_ASSOC);
23    }
24
25    public function create($data) {
26        $columns = implode(', ', array_keys($data));
27        $values = ':' . implode(':', array_keys($data));
28
29        $sql = "INSERT INTO $this->table ($columns) VALUES ($values)";
30        $stmt = $this->pdo->prepare($sql);
31
32        foreach ($data as $key => $value) {
33            $stmt->bindValue(':'. $key, $value);
34        }
35
36        return $stmt->execute();
37    }
38
39    public function update($id, $data) {
40        $set = [];
41        foreach ($data as $key => $value) {
42            $set[] = "$key = :$key";
43        }
44        $set = implode(', ', $set);
45
46        $sql = "UPDATE $this->table SET $set WHERE id = :id";
47        $stmt = $this->pdo->prepare($sql);
48
49        $stmt->bindParam(':id', $id);
50        foreach ($data as $key => $value) {
51            $stmt->bindValue(':'. $key, $value);
52        }
53
54        return $stmt->execute();
55    }
56
57    public function delete($id) {
58        $sql = "DELETE FROM $this->table WHERE id = :id";
59        $stmt = $this->pdo->prepare($sql);
60        $stmt->bindParam(':id', $id);
61        return $stmt->execute();
62    }
63
64    public function authenticate($username, $password){
65        $sql = "SELECT * FROM users WHERE username = :username AND password = :password";
66        $stmt = $this->pdo->prepare($sql);
67        $stmt->bindParam(':username', $username);
68        $stmt->bindParam(':password', $password);
69        $stmt->execute();
70        return $stmt->fetch(PDO::FETCH_ASSOC);
71    }
72 }
73 >>

```

## Контрольні питання

1. Що таке клас та екземпляр класу в ООП?

Клас в об'єктно-орієнтованому програмуванні (ООП) є шаблоном або типом даних, який описує структуру та поведінку об'єктів. Екземпляр класу, або об'єкт, є конкретною реалізацією цього класу, представляючи конкретний об'єкт або екземпляр даних згідно із визначеними класом властивостями та методами.

2. Коли відбувається ініціалізація властивостей класу?

Ініціалізація властивостей класу може відбуватися в конструкторі класу. Конструктор — це спеціальний метод класу, який викликається при створенні нового екземпляру класу і служить для встановлення початкових значень властивостей.

3. Для чого використовується вказівник \$this?

\$this вказівник використовується для звертання до властивостей та методів поточного об'єкта (екземпляра класу) всередині самого класу. Він вказує на поточний об'єкт, що дозволяє уникнути конфліктів імен та працювати з властивостями чи методами об'єкта всередині його власного класу.

4. Як можна викликати батьківський конструктор в конструкторі дочірнього класу?

Для виклику батьківського конструктора в конструкторі дочірнього класу використовується ключове слово parent::, за яким слідує виклик конструктора батьківського класу.

5. За допомогою якого методу доступу можна отримати доступ з поточного та дочірнього класу?

За допомогою методу доступу protected можна отримати доступ до властивостей або методів з поточного та дочірнього класу. Властивості та методи, оголошені з

					<p style="text-align: center;"><i>Лабораторна робота № 17</i></p>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		

модифікатором `protected`, будуть доступні для використання в класі та його дочірніх класах.

6. Якою функцією можна імітувати роботу деструктора?

Функцію `__destruct()` можна використовувати для імітації роботи деструктора. Цей метод буде викликано автоматично при знищенні об'єкта або завершенні його життєвого циклу.

7. Для чого призначений оператор `new`?

Оператор `new` використовується для створення нового екземпляру класу, тобто для виділення пам'яті та ініціалізації об'єкта на основі визначеного класу. Він повертає посилання на новий об'єкт та викликає його конструктор для налаштування початкових значень.

					<i>Лабораторна робота № 17</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		