Хід роботи

Login

Username: Uboz
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

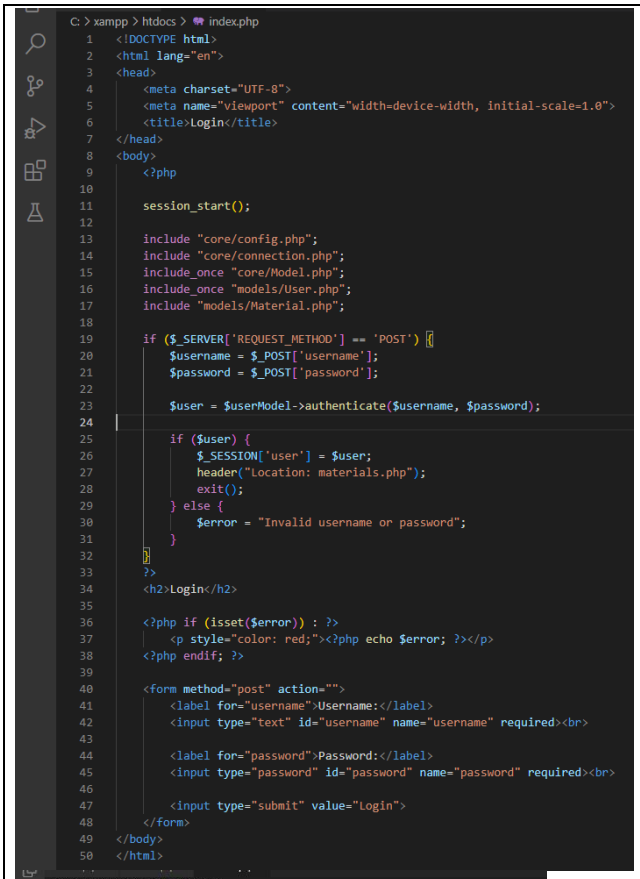
Back to Login

Create New Material

ID:
Product:
Amount:
Create

Back to Materials

Back to Login



```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
</head>
<body>
    <?php

    session_start();

    include "core/config.php";
    include "core/connection.php";
    include_once "core/Model.php";
    include_once "models/User.php";
    include "models/Material.php";

    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $username = $_POST['username'];
        $password = $_POST['password'];

        $user = $userModel->authenticate($username, $password);

        if ($user) {
            $_SESSION['user'] = $user;
            header("Location: materials.php");
            exit();
        } else {
            $error = "Invalid username or password";
        }
    }
    ?>
    <h2>Login</h2>

    <?php if (isset($error)) : ?>
        <p style="color: red;"><?php echo $error; ?></p>
    <?php endif; ?>

    <form method="post" action="">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br>

        <input type="submit" value="Login">
    </form>
</body>
</html>
```

```php
<?php
    class Model {
        static $pdo;
        static $table;

        public function __construct(PDO $pdo, $table) {
            $this->pdo = $pdo;
            $this->table = $table;
        }

        public function all() {
            $sql = "SELECT * FROM $this->table";
            $stmt = $this->pdo->query($sql);
            return $stmt->fetchAll(PDO::FETCH_ASSOC);
        }

        public function getById($id) {
            $sql = "SELECT * FROM $this->table WHERE id = :id";
            $stmt = $this->pdo->prepare($sql);
            $stmt->bindParam(':id', $id);
            $stmt->execute();
            return $stmt->fetch(PDO::FETCH_ASSOC);
        }

        public function create($data) {
            $columns = implode(', ', array_keys($data));
            $values = ':' . implode(', :', array_keys($data));

            $sql = "INSERT INTO $this->table ($columns) VALUES ($values)";
            $stmt = $this->pdo->prepare($sql);

            foreach ($data as $key => $value) {
                $stmt->bindValue(':' . $key, $value);
            }

            return $stmt->execute();
        }

        public function update($id, $data) {
            $set = [];
            foreach ($data as $key => $value) {
                $set[] = "$key = :$key";
            }
            $set = implode(', ', $set);

            $sql = "UPDATE $this->table SET $set WHERE id = :id";
            $stmt = $this->pdo->prepare($sql);

            $stmt->bindParam(':id', $id);
            foreach ($data as $key => $value) {
                $stmt->bindValue(':' . $key, $value);
            }

            return $stmt->execute();
        }

        public function delete($id) {
            $sql = "DELETE FROM $this->table WHERE id = :id";
            $stmt = $this->pdo->prepare($sql);
            $stmt->bindParam(':id', $id);
            return $stmt->execute();
        }

        public function authenticate($username, $password){
            $sql = "SELECT * FROM users WHERE username = :username AND password = :password";
            $stmt = $this->pdo->prepare($sql);
            $stmt->bindParam(':username', $username);
            $stmt->bindParam(':password', $password);
            $stmt->execute();
            return $stmt->fetch(PDO::FETCH_ASSOC);
        }
    }
?>
```

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 доступа 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 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.