**Лабораторна робота №14**

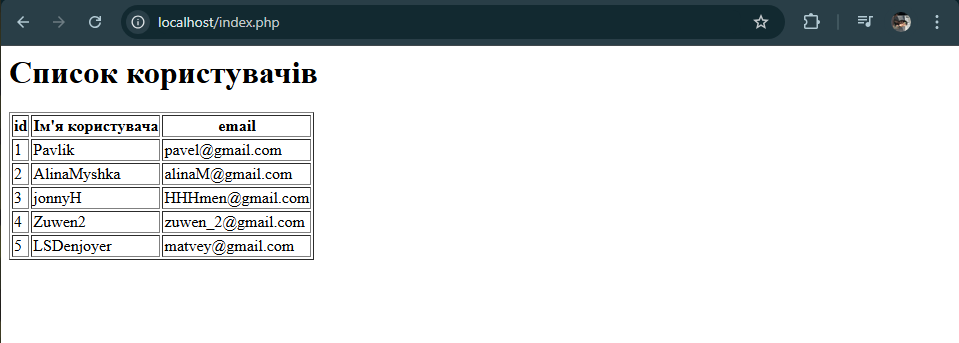
**Виконала Будяк Анастасія 451 група**

**Тема:** Використання класів для створення моделей для архітектури MVC

**Мета:** опанувати основні принципи роботи з моделями, навчитися створювати основу для моделі.

**Хід роботи**

1. Впевнитись, що пакет XAMPP встановлено та web-сервер Apache запущений
2. Перейти до каталогу C:\xampp\htdocs\ та очистити його
3. Розгляньте приклад 1, де клас User виконано як просту модель даних.



1. На основі попередньої лабораторної роботи та індивідуального завдання створіть новий проект.
2. Для кожної сутності зробіть окрему модель.

**index.php:**

<?php

session\_start();

include "Book.php";

try {

    $pdoLibrary = new PDO('mysql:host=localhost;dbname=library;charset=utf8', 'root', '');

    $pdoLibrary->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

    $books = Book::all($pdoLibrary);

    echo "<h1>Каталог книг</h1>";

    echo "<table border='1'>";

    echo "<tr><th>ID</th><th>Назва</th><th>Рік</th><th>Жанр</th></tr>";

    foreach ($books as $book) {

        echo "<tr>";

        echo "<td>{$book->id}</td>";

        echo "<td>{$book->name}</td>";

        echo "<td>{$book->year}</td>";

        echo "<td>{$book->genre}</td>";

        echo "</tr>";

    }

    echo "</table>";

    include "login\_form.php";

} catch (PDOException $e) {

    echo "Помилка: " . $e->getMessage();

}

?>

**User.php:**

<?php

class User

{

    private $pdo;

    public $id;

    public $username;

    public $email;

    private $password;

    public function \_\_construct(PDO $pdo)

    {

        $this->pdo = $pdo;

    }

public static function authenticate(PDO $pdo, $email, $password)

{

    $sql = "SELECT \* FROM users WHERE email = :email";

    $stmt = $pdo->prepare($sql);

    $stmt->bindParam(':email', $email);

    $stmt->execute();

    $userRow = $stmt->fetch(PDO::FETCH\_ASSOC);

    if ($userRow && password\_verify($password, $userRow['password'])) {

        $user = new User($pdo);

        $user->id = $userRow['id'];

        $user->username = $userRow['username'];

        $user->email = $userRow['email'];

        $user->password = $userRow['password'];

        return $user;

    }

    return null; // повертаємо null, якщо авторизація неуспішна

}

};

?>

**Book.php:**

<?php

class Book

{

    private $pdo;

    public $id;

    public $name;

    public $year;

    public $genre;

    public function \_\_construct(PDO $pdo)

    {

        $this->pdo = $pdo;

    }

    public static function all(PDO $pdo)

    {

        $stmt = $pdo->query("SELECT \* FROM books");

        return $stmt->fetchAll(PDO::FETCH\_OBJ);

    }

    public static function find(PDO $pdo, $id)

    {

        $stmt = $pdo->prepare("SELECT \* FROM books WHERE id = :id");

        $stmt->bindParam(':id', $id);

        $stmt->execute();

        return $stmt->fetch(PDO::FETCH\_OBJ);

    }

    public static function delete(PDO $pdo, $id)

    {

        $stmt = $pdo->prepare("DELETE FROM books WHERE id = :id");

        $stmt->bindParam(':id', $id);

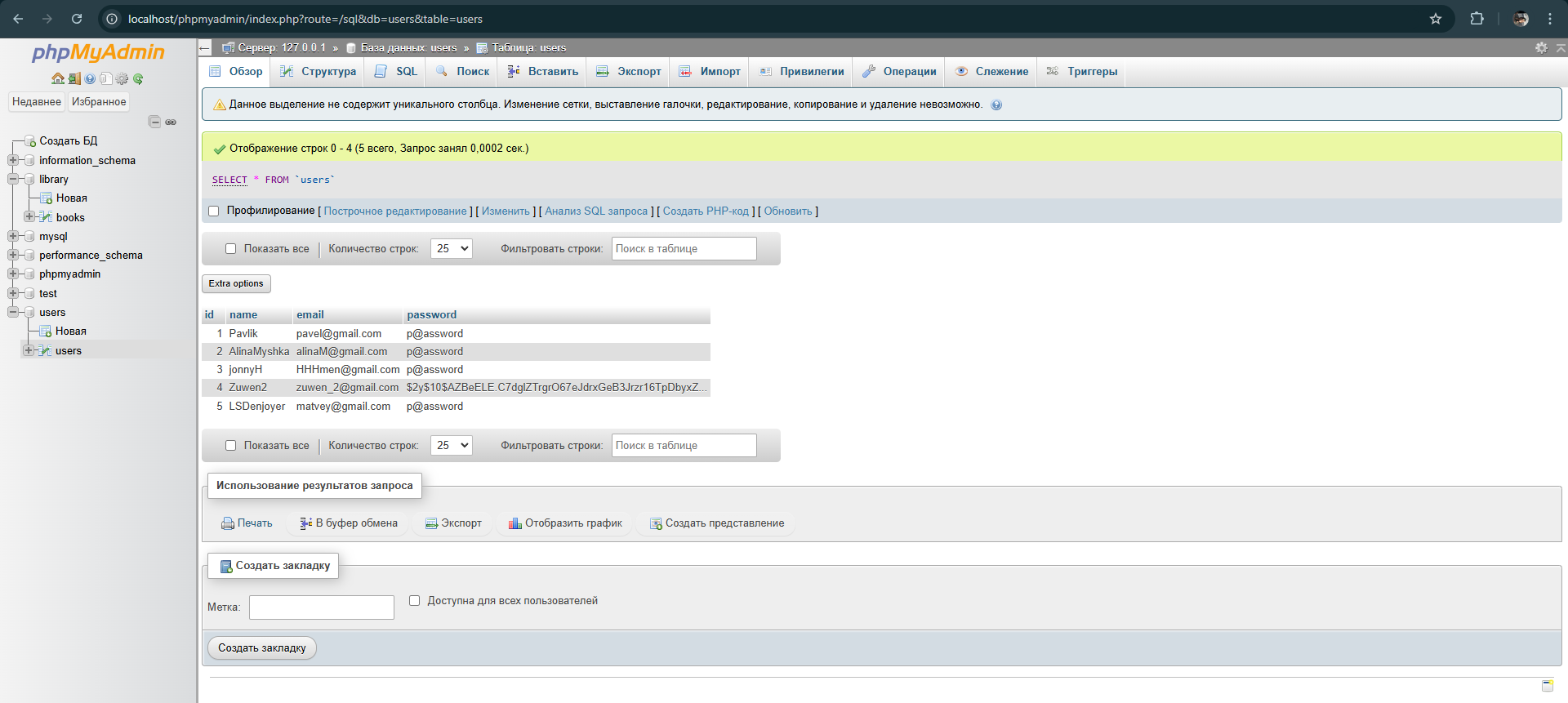
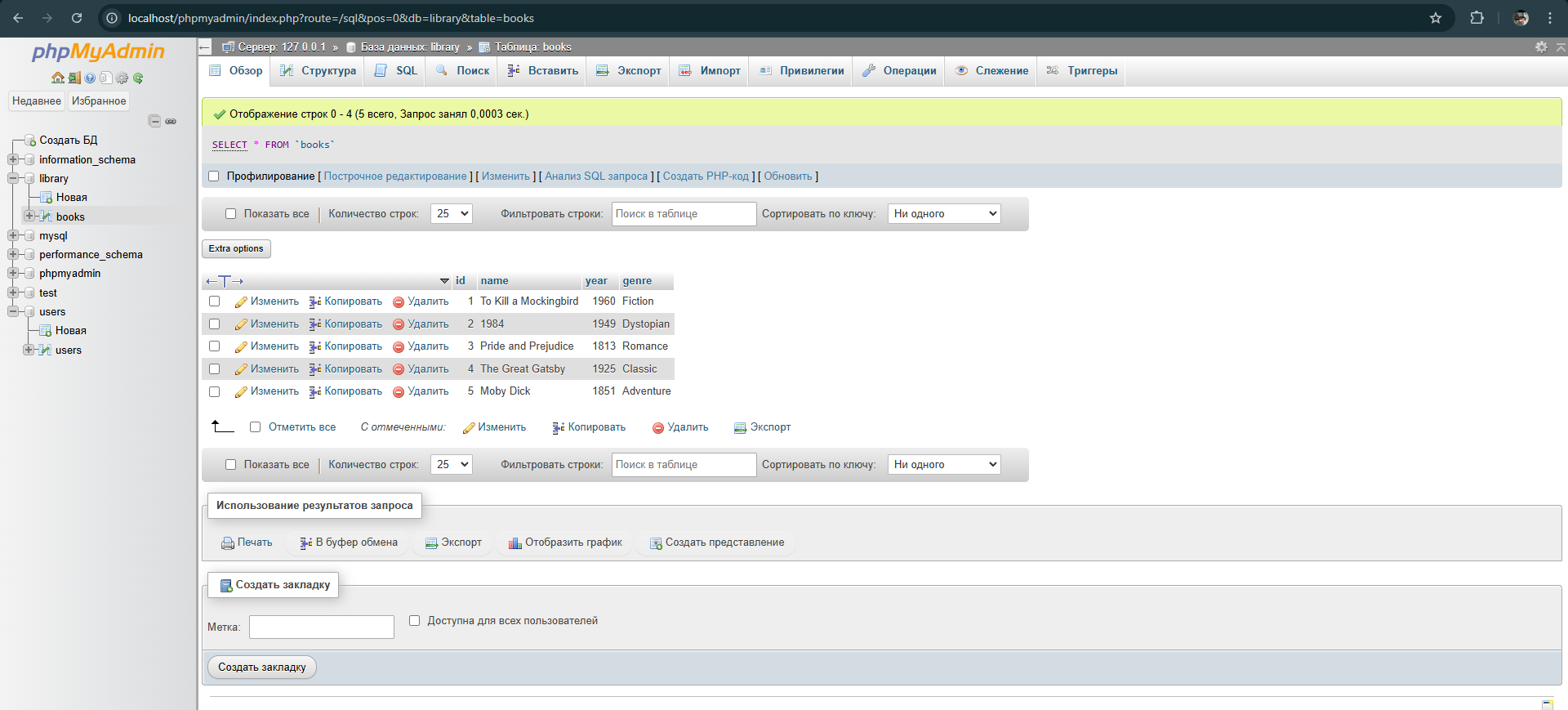
        return $stmt->execute();

    }

}

?>

1. Файл конфігурації з налаштуваннями підключення до БД винесіть окремо.

1. Для авторизованих користувачів створіть сторінку редагування та можливість видалення записів.

**login.php:**

<?php

session\_start();

include "User.php";

try {

    $pdoUsers = new PDO('mysql:host=localhost;dbname=users;charset=utf8', 'root', '');

    $pdoUsers->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

    if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

        $email = $\_POST['email'];

        $password = $\_POST['password'];

        $user = User::authenticate($pdoUsers, $email, $password);

        if ($user) {

            $\_SESSION['user\_id'] = $user->id;

            header("Location: edit\_books.php");

            exit();

        } else {

            echo "Невірний email або пароль";

        }

    }

} catch (PDOException $e) {

    echo "Помилка: " . $e->getMessage();

}

?>

**login\_form.php:**

<form action="login.php" method="POST">

    <h3>Авторизація</h3>

    <label>Email: <input type="email" name="email" required></label><br>

    <label>Пароль: <input type="password" name="password" required></label><br>

    <button type="submit">Увійти</button>

</form>

**edit\_books.php:**

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

    header("Location: index.php");

    exit();

}

include "Book.php";

try {

    $pdoLibrary = new PDO('mysql:host=localhost;dbname=library;charset=utf8', 'root', '');

    $pdoLibrary->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

    if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['delete\_id'])) {

        $bookId = $\_POST['delete\_id'];

        Book::delete($pdoLibrary, $bookId);

    }

    $books = Book::all($pdoLibrary);

    echo "<h1>Редагування каталогу</h1>";

    echo "<table border='1'>";

    echo "<tr><th>ID</th><th>Назва</th><th>Рік</th><th>Жанр</th><th>Дії</th></tr>";

    foreach ($books as $book) {

        echo "<tr>";

        echo "<td>{$book->id}</td>";

        echo "<td>{$book->name}</td>";

        echo "<td>{$book->year}</td>";

        echo "<td>{$book->genre}</td>";

        echo "<td><form method='POST'><button name='delete\_id' value='{$book->id}'>Видалити</button></form></td>";

        echo "</tr>";

    }

    echo "</table>";

} catch (PDOException $e) {

    echo "Помилка: " . $e->getMessage();

}

?>

Також я використала додатково **hash\_password.php** для хешування паролю:

<?php

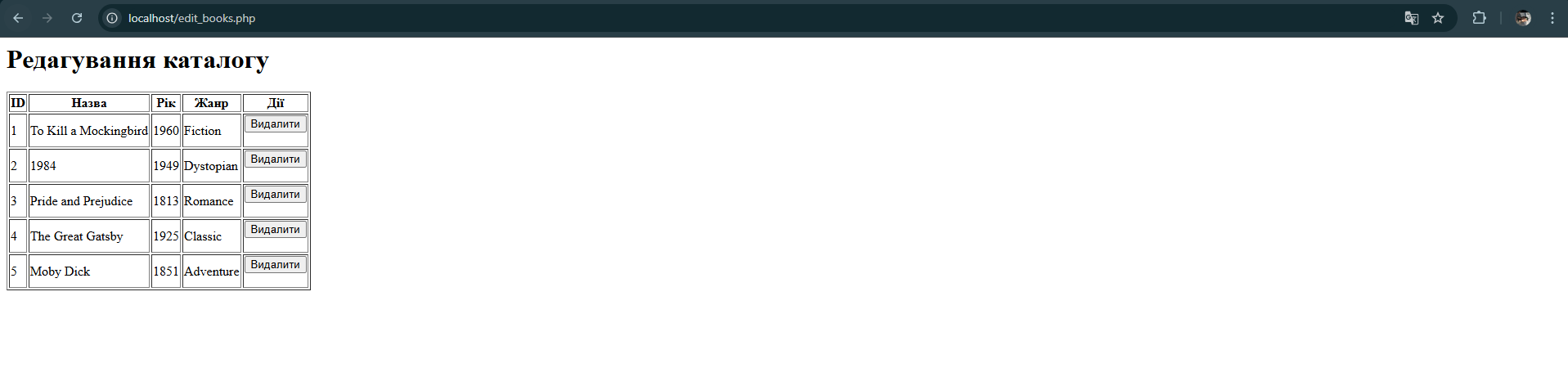
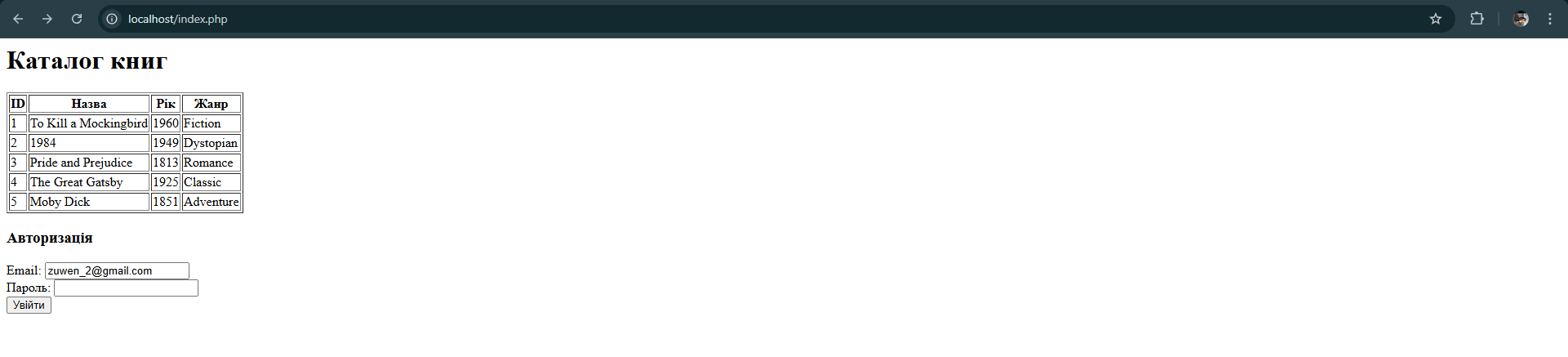
$hashedPassword = password\_hash('p@assword', PASSWORD\_DEFAULT);

echo "Хеш пароля: " . $hashedPassword;

?>

1. Впевніться, що всі вихідні HTML-сторінки є валідними, використавши валідатор HTML-коду [validator.w3.org](https://validator.w3.org/). За необхідності, виправити помилки та зауваження
2. Для проекту створіть папку "lab 14" в своєму репозиторії "web-progr" на Github. Завантажте кінцеві файли лабораторної роботи.
3. Виконайте звіт

**РЕЗУЛЬТАТ:**



**Контрольні питання:**

1. Що таке клас та екземпляр класу в ООП?

Клас - це структура для створення об'єктів, що містять властивості і методи; екземпляр класу - це створений об'єкт на основі цього класу.

1. Коли відбувається ініціалізація властивостей класу?

Відбувається у конструкторі (метод `\_\_construct`) або при їх оголошенні.

1. Для чого використовується вказівник $this?

Для доступу до властивостей та методів поточного екземпляра класу.

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

За допомогою `parent::\_\_construct()`.

1. За допомогою якого методу доступа можна отримати доступ з поточного та дочірнього класу?

Модифікатор доступу `protected` дозволяє отримати доступ з поточного і дочірнього класів.

1. Якою функцією можна імітувати роботу деструктора?

За допомогою функції `\_\_destruct`.

1. Для чого призначений оператор new?

Для створення нового екземпляра класу.