**ЛАБОРАТОРНА РОБОТА №6**

***Створення індивідуального блогу користувача.***

**Мета**: Оволодіння базовими навичками основних функцій PHP. Використання СКБД MySQL. Робота з сервером, налаштування БД. Вибірка елементів з БД. Виконання операцій над об’єктами.

**Завдання для виконання**

**Необхідно використати:**

**1. Мова програмування:**

• Мова програмування: PHP.

• Потрібно буде організувати структуру проекту самостійно. Спочатку почати з простого PHP-проекту, а потім поступово розширювати його функціональність.

**2. База даних:**

• СУБД: MySQL, зверніть увагу на те, що вам потрібно буде самостійно створювати БД, використовуючи phpmyadmin, adminer або інше.

• Вам потрібно буде створювати та виконувати SQL-запити самостійно, без автоматичного мапування об'єктів на таблиці бази даних.

**3. Фронтенд:**

• Необхідно буде розробити/взяти готовий шаблон/використати з попередніх проектів інтуїтивно зрозумілий адаптивний дизайн.

• HTML, CSS, JavaScript: Використовуйте ці три основні засоби для розробки клієнтської частини вашого додатку.

• Bootstrap, jQuery: можна використати готові компоненти, сітку для реалізації проекту.

**4. Аутентифікація та авторизація:**

• Механізми аутентифікації: Розробіть механізми аутентифікації на базі сесій та кукі для ідентифікації користувачів.

• Механізми авторизації: Ручно визначте права доступу користувачів та контролюйте доступ до різних частин додатку.

**Завдання для виконання**

**Частина адміністратора:**

**1. Авторизація адміністратора:**

• Створіть сторінку входу для адміністратора, де він може ввести свої облікові дані (логін та пароль). Бажано для цього створити окрему папку на сервері та додатково обмежити до неї доступ.

• Реалізуйте перевірку введених даних на відповідність збереженим обліковим записам в базі даних. За замовчуванням створити потрібно мати користувача admin, який у подальшому може змінити пароль.

• Після успішної авторизації перенаправте адміністратора на адмін-панель, інакше виведіть повідомлення про помилку.

**2. Додавання нових постів:**

• Створіть форму додавання нового поста, яка міститиме поля для введення категорії поста, завантаження зображення, введення заголовку та опису.

• Забезпечте перевірку та валідацію введених даних перед їх збереженням у базі даних.

• Додайте функціональність для збереження нового поста у базі даних після натискання на кнопку "Опублікувати".

**3. Редагування та видалення постів:**

• На адмін-панелі виведіть список всіх наявних постів разом з можливістю редагування та видалення кожного поста.

• Реалізуйте функціональність для редагування полів категорії, зображення, заголовку та опису поста.

• Забезпечте підтвердження видалення поста перед видаленням його з бази даних.

**Частина користувача:**

**1. Реєстрація та авторизація користувачів:**

• Створіть сторінки реєстрації та входу для користувачів з полями для введення обов'язкової інформації (наприклад, ім'я, email, пароль). В залежності від предметної області необхідно додати поля.

• Перевірте унікальність введеного email під час реєстрації.

• Зберігайте дані про користувачів у базі даних та забезпечте безпечне зберігання паролів (наприклад, за допомогою хешування).

**2. Перегляд постів:**

• На головній сторінці виведіть список доступних постів з можливістю переходу до детального перегляду кожного поста.

• Додайте функціональність фільтрації постів за категоріями для зручності користувачів.

• Передбачити вибірку постів за датою публікації(вказавши період з - по). Також надати можливість пошуку по заголовку.

**3. Коментування та вподобання:**

• Реалізуйте можливість додавання коментарів до кожного поста з відображенням автора, дати та тексту коментаря.

• Додайте можливість користувачам виражати вподобання до постів та відображайте кількість вподобань для кожного поста.

**4. Авторизований доступ:**

• Забезпечте доступ до функціоналу коментування та вподобань тільки для авторизованих користувачів.

• Перенаправте неавторизованих користувачів на сторінку входу або реєстрації, якщо вони спробують взаємодіяти з функціоналом, який потребує авторизації.

**5. Додаткові можливості:**

• Додайте можливість перегляду детальної інформації про окремий пост, включаючи категорію, зображення, заголовок, опис та коментарі.

• Реалізуйте функціональність пошуку постів за ключовими словами або категоріями для зручності користувачів.

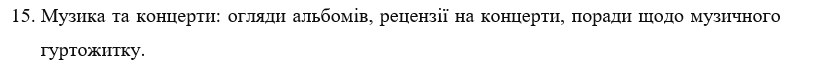
**6. Робота з сесіями:**

• При авторизації користувача створіть нову сесію та збережіть в ній необхідні дані, такі як ідентифікатор користувача або інше ідентифікаційне значення.

• Використовуйте сесії для збереження даних, які повинні залишатися доступними протягом всього сеансу взаємодії з додатком, наприклад, інформацію про авторизацію, корзину покупок тощо.

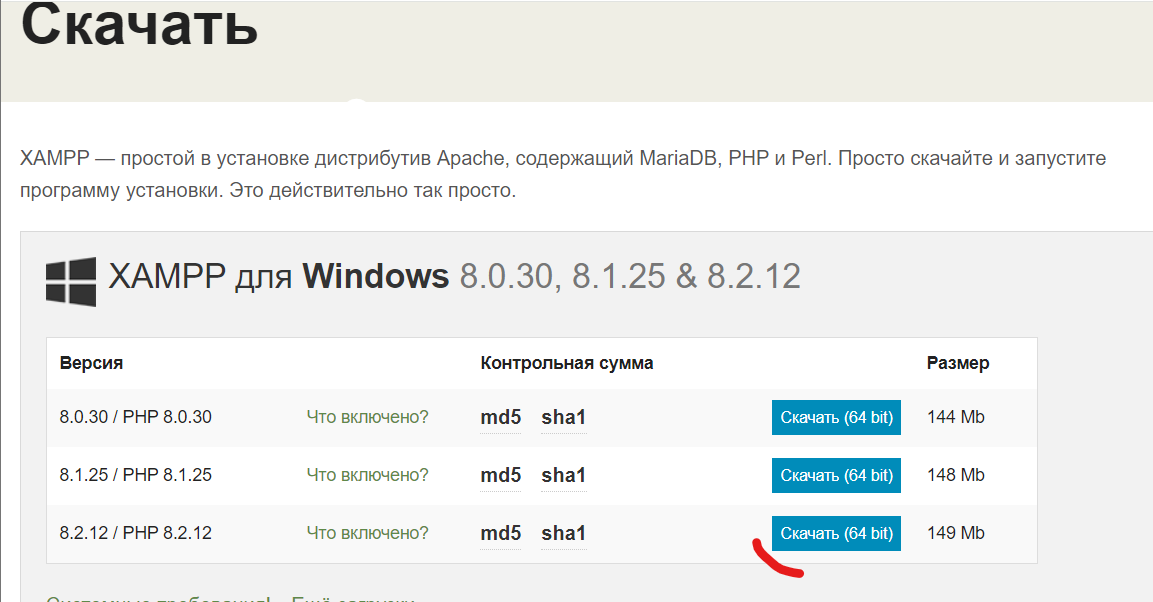
• Забезпечте можливість завершення сесії користувача після виходу або закінчення сеансу.

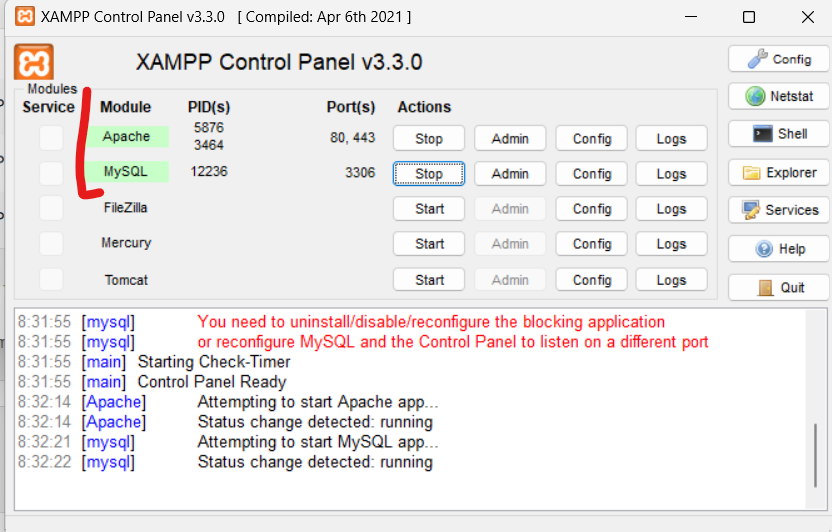
Варіант завдання:

****

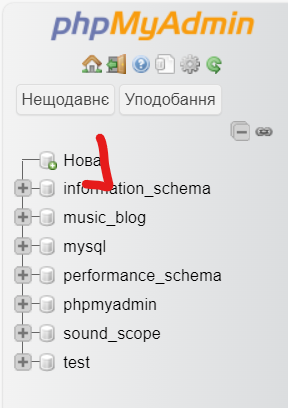
**Хід виконання роботи**

Згідно завдань

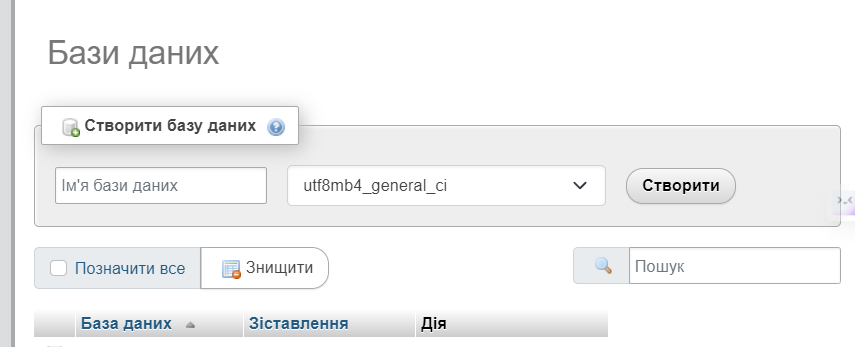
1. **Далі для роботи із php я встановив XAMP:**
2. **Результат встановлення:**



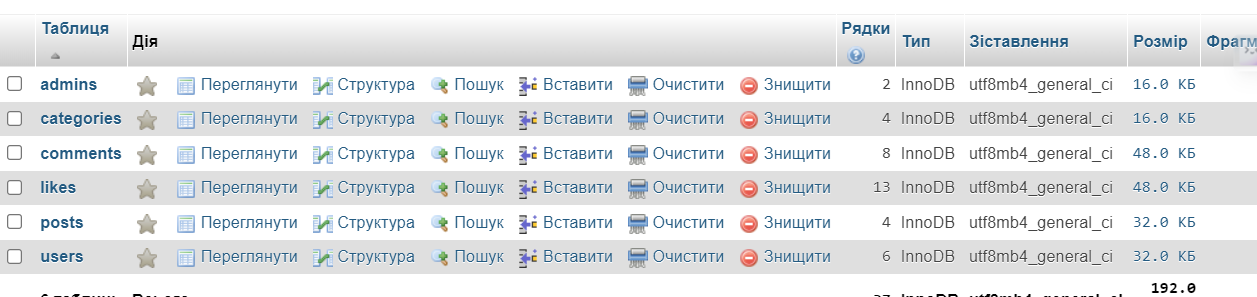
Для того, щоб коректно працювати із php нам потрібно запусти модулі Apache та MySQL

1. **Тепер використовуючи команду:** [**http://localhost/phpmyadmin/**](http://localhost/phpmyadmin/)**; Я перейшов до створення своєї БД:**

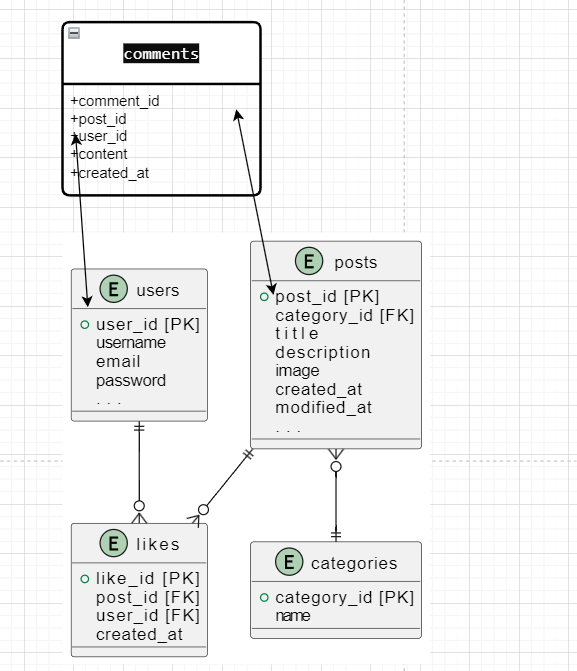
Далі переходимо та вводимо ім’я нової БД і натискаємо створити:



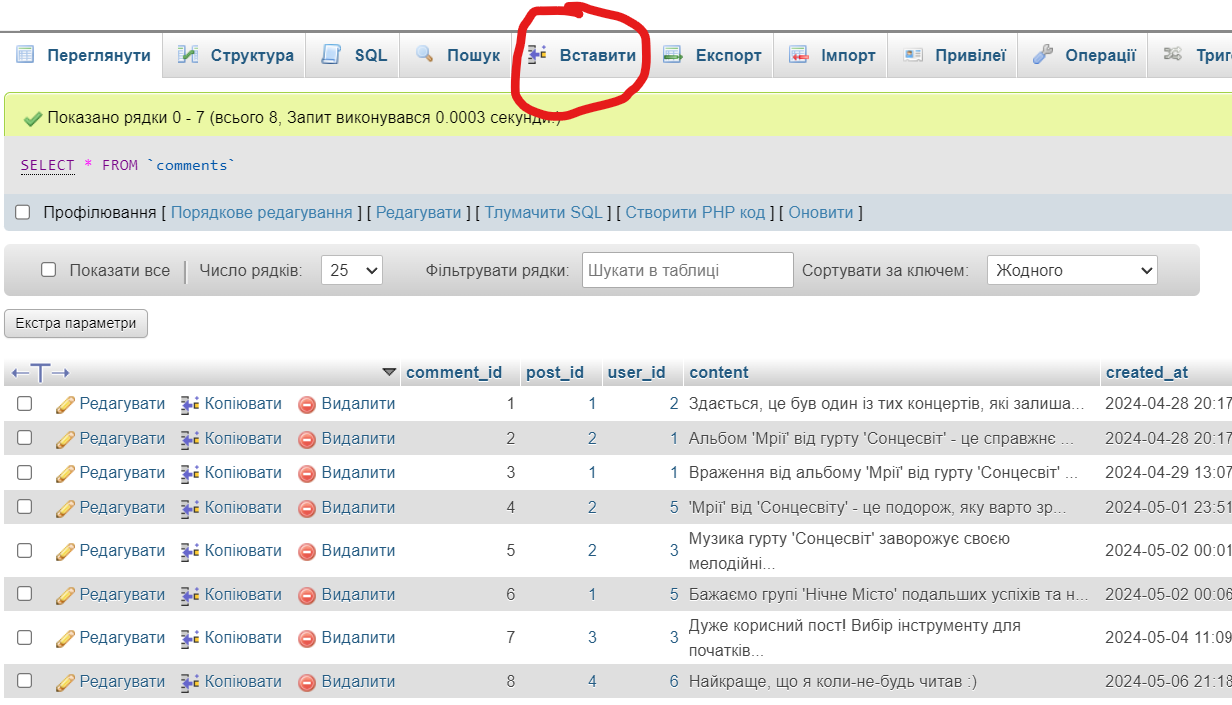
Далі я спроектував свої таблички:



За ось такою схемою:

****

1. **Далі я за допомогою функції php вставив перші початкові дані, які буду ть відображатися із самого початку, для того, щоб адмін панель робити у самому кінці**

****

**І так із усіма**

1. **Наступним кроком було знаходження шаблона для подальшої розробки блога, тепер коли ми зробили наш шаблон, то можемо переходити до завдань:**

**Завдання для виконання**

**Необхідно використати:**

**1. Мова програмування:**

• Мова програмування: PHP.

• Потрібно буде організувати структуру проекту самостійно. Спочатку почати з простого PHP-проекту, а потім поступово розширювати його функціональність.

Із самого початку я робив основний файл  на різні частини, щоб збирати мої сторінки потім по частинах, для цього я створив папку includes, у які у мене знаходяться усі частини моєї сторніки:



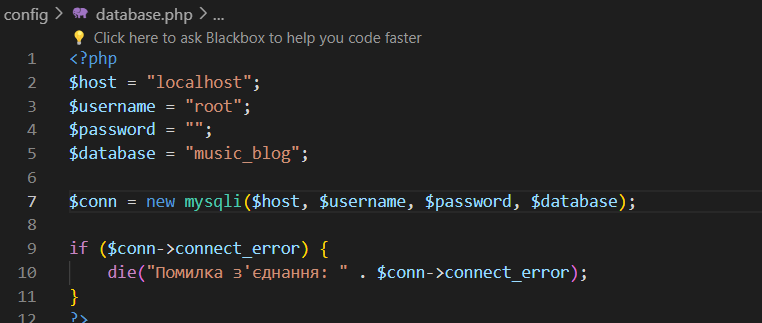
І відповідно тіло цього файлу: 

**2. База даних:**

• СУБД: MySQL, зверніть увагу на те, що вам потрібно буде самостійно створювати БД, використовуючи phpmyadmin, adminer або інше.

• Вам потрібно буде створювати та виконувати SQL-запити самостійно, без автоматичного мапування об'єктів на таблиці бази даних.

Для роботи із базою даних, у мене є спеціальна папка із файлом: 



* **$host**: Адреса сервера бази даних. У цьому випадку використовується **localhost**, що означає поточний сервер.
* **$username**: Ім'я користувача бази даних. У даному випадку — "root", за замовчуванням користувач MySQL.
* **$password**: Пароль користувача бази даних. Тут він порожній, що зазвичай вказується під час налаштування MySQL.
* **$database**: Назва бази даних, до якої потрібно підключитися. В цьому коді — "music\_blog".

**3. Фронтенд:**

• Необхідно буде розробити/взяти готовий шаблон/використати з попередніх проектів інтуїтивно зрозумілий адаптивний дизайн.

• HTML, CSS, JavaScript: Використовуйте ці три основні засоби для розробки клієнтської частини вашого додатку.

• Bootstrap, jQuery: можна використати готові компоненти, сітку для реалізації проекту.

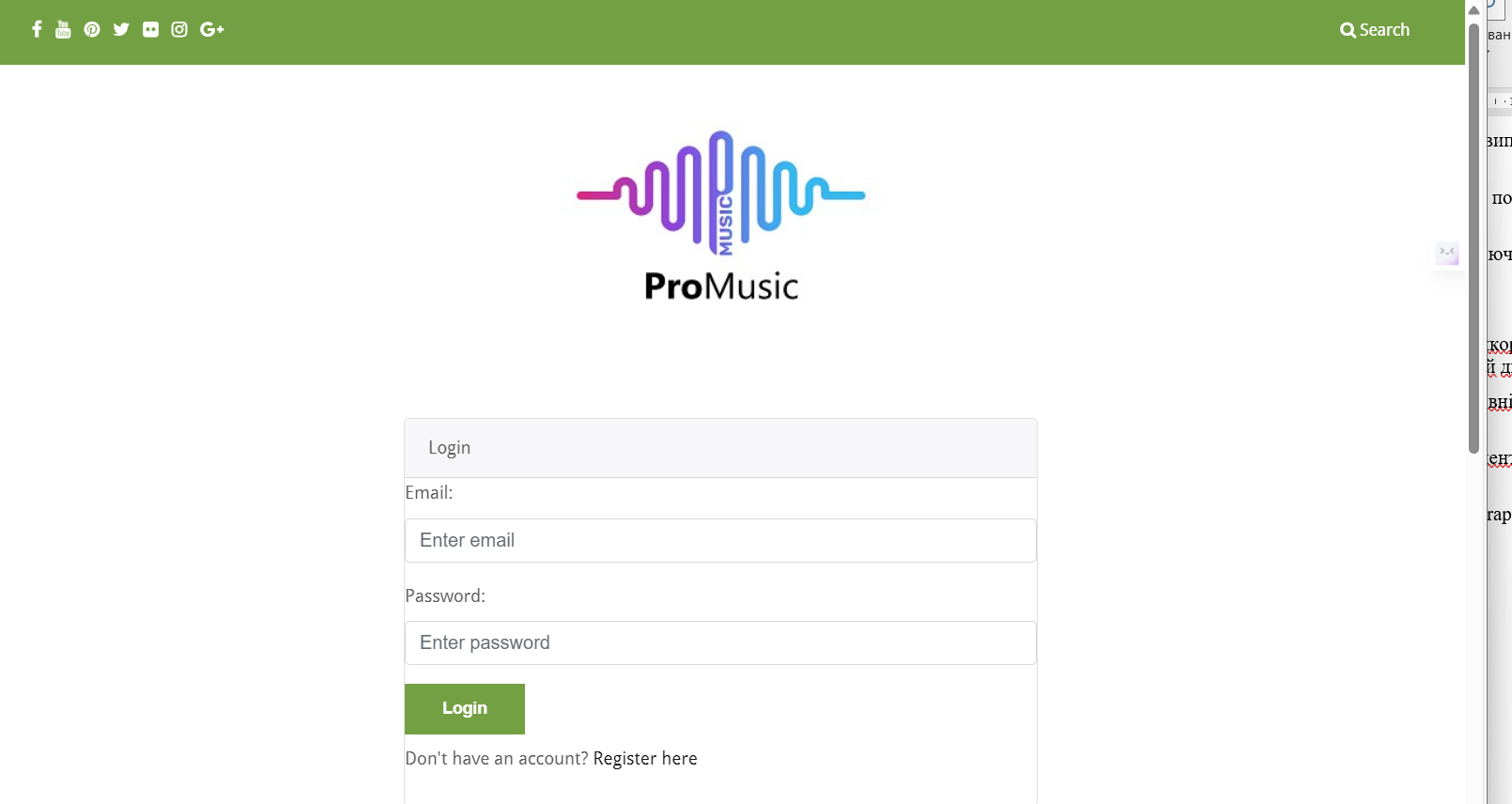
Я використав встановлений шалон побудований на Bootstrap та Jquery

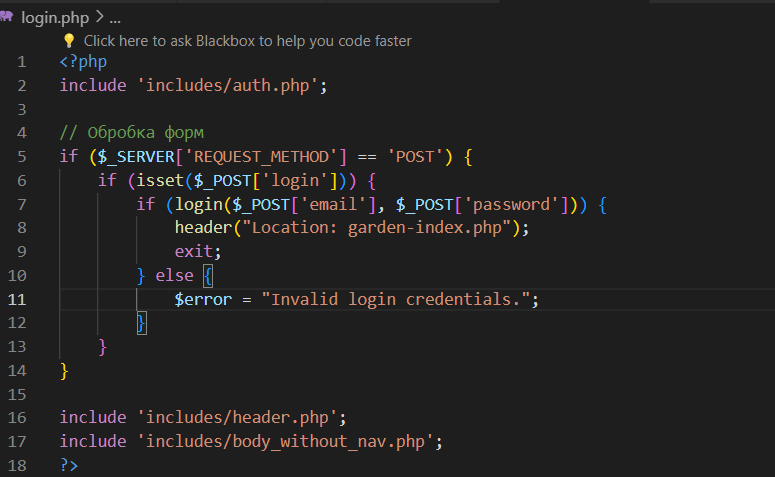
**4. Аутентифікація та авторизація:**

• Механізми аутентифікації: Розробіть механізми аутентифікації на базі сесій та кукі для ідентифікації користувачів.

• Механізми авторизації: Ручно визначте права доступу користувачів та контролюйте доступ до різних частин додатку.

Форма для авторизації:





<?php

session\_start();

require 'config/database.php';

// Функція реєстрації

function register($username, $email, $password) {

    global $conn;

    $password = password\_hash($password, PASSWORD\_DEFAULT); // Хешування пароля

    $query = "INSERT INTO users (username, email, password) VALUES (?, ?, ?)";

    $stmt = $conn->prepare($query);

    $stmt->bind\_param("sss", $username, $email, $password);

    $stmt->execute();

    if ($stmt->affected\_rows == 1) {

        $user\_id = $stmt->insert\_id;

        $\_SESSION['user\_id'] = $user\_id;

        return true;

    }

    return false;

}

// Функція входу

function login($email, $password) {

    global $conn;

    $query = "SELECT user\_id, password FROM users WHERE email = ?";

    $stmt = $conn->prepare($query);    $stmt->bind\_param("s", $email);    $stmt->execute();    $result = $stmt->get\_result();    if ($result->num\_rows == 1) {        $user = $result->fetch\_assoc();

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

            $\_SESSION['user\_id'] = $user['user\_id'];

            return true;       } }

    return false;

}

// Перевірка, чи користувач авторизований

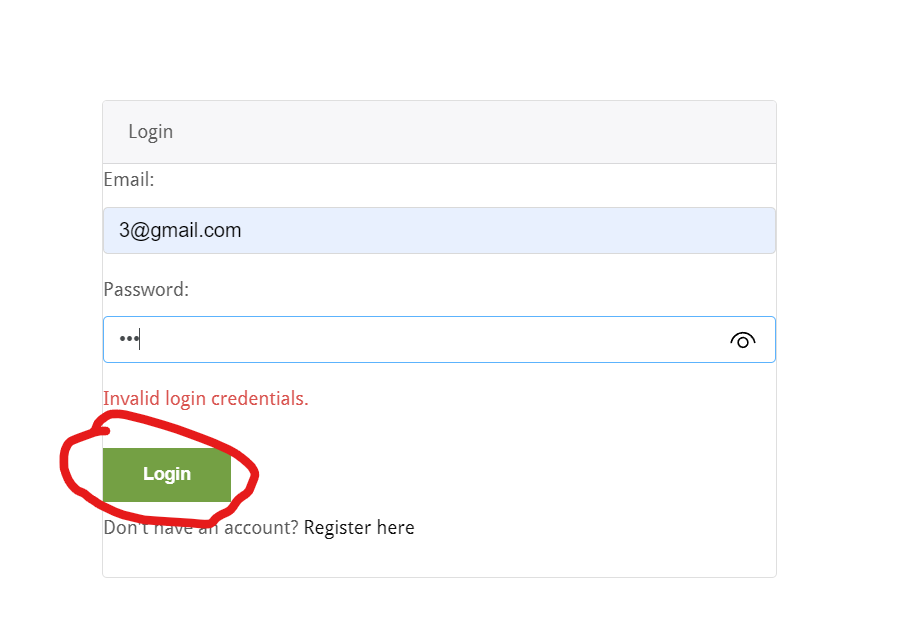
function isLoggedIn() {

    return isset($\_SESSION['user\_id']);

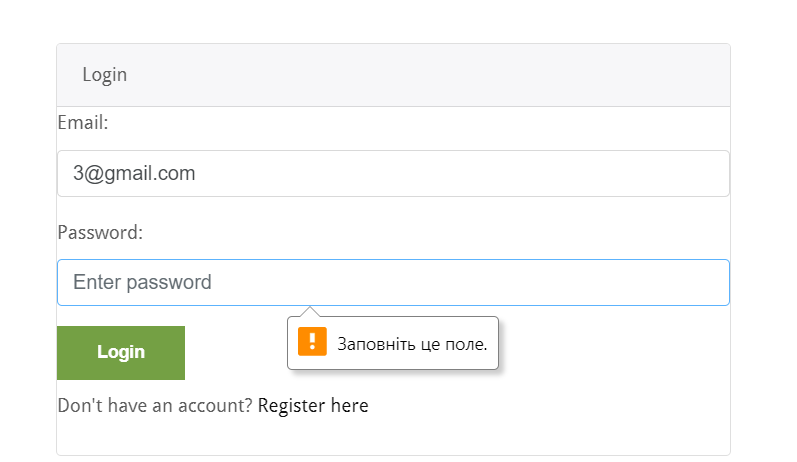
}

?>

Перевірка роботи аутентифікації:



Валідація:



**Завдання для виконання**

**Частина адміністратора:**

**1. Авторизація адміністратора:**

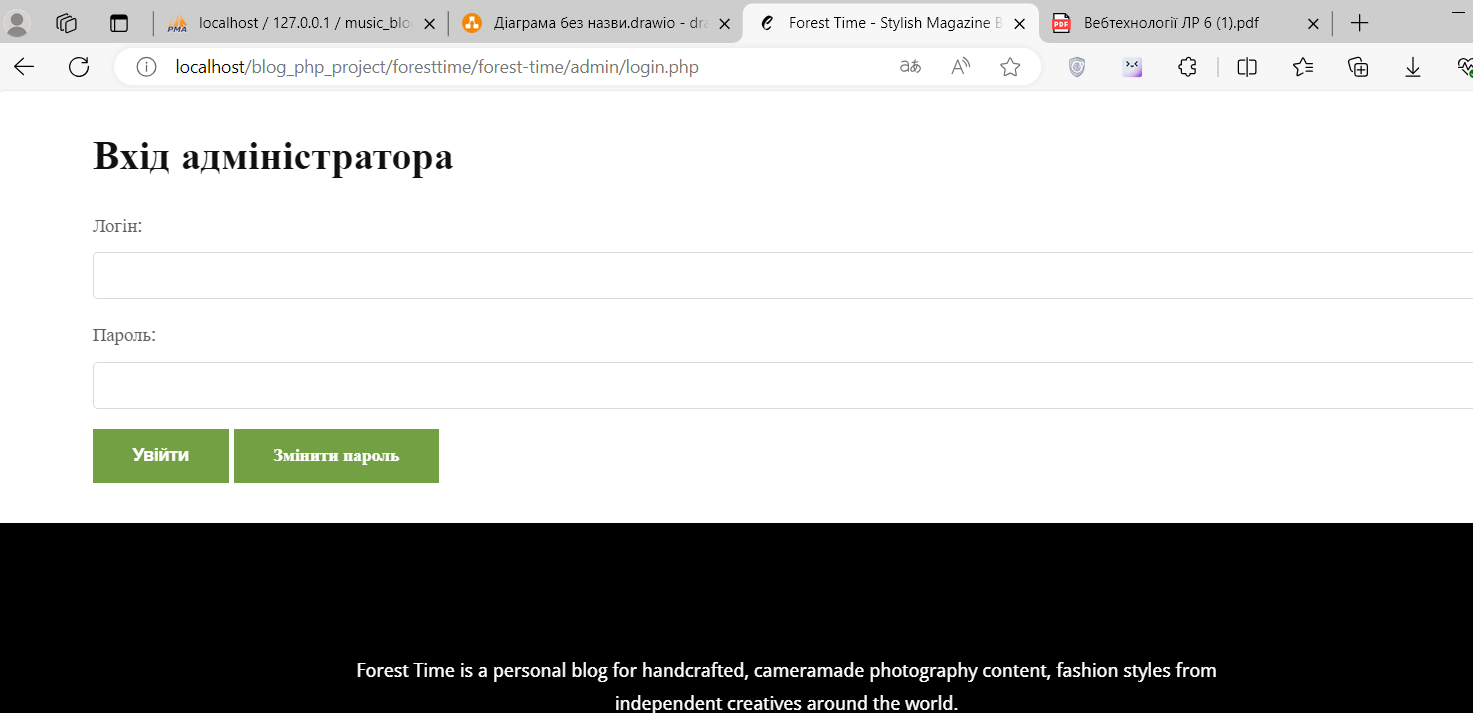
• Створіть сторінку входу для адміністратора, де він може ввести свої облікові дані (логін та пароль). Бажано для цього створити окрему папку на сервері та додатково обмежити до неї доступ.

• Реалізуйте перевірку введених даних на відповідність збереженим обліковим записам в базі даних. За замовчуванням створити потрібно мати користувача admin, який у подальшому може змінити пароль.

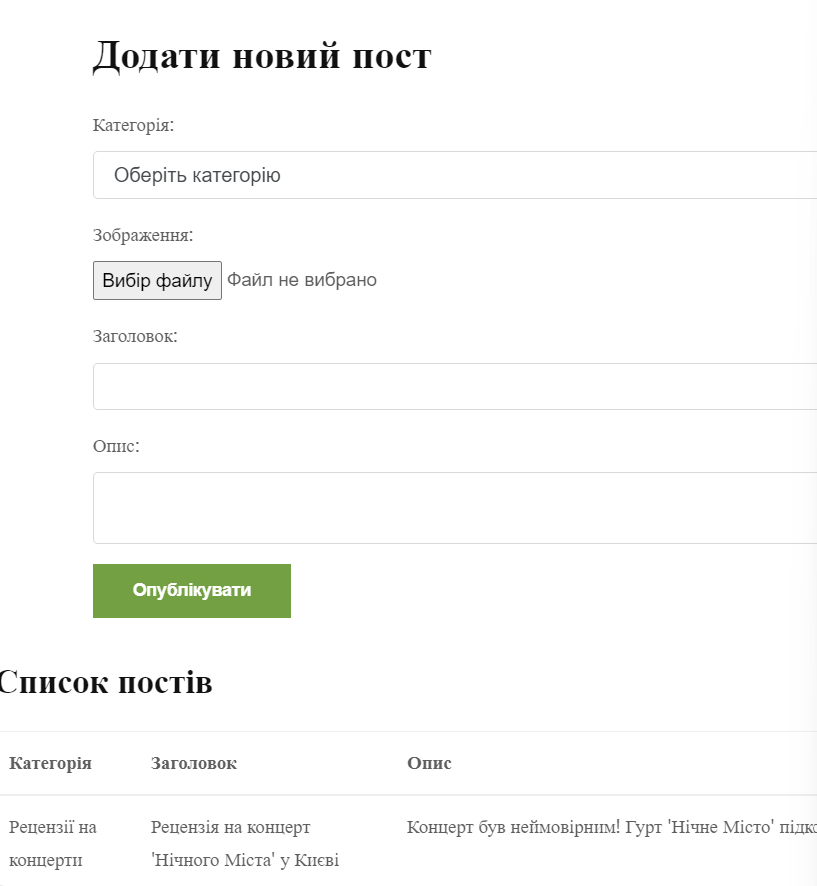
• Після успішної авторизації перенаправте адміністратора на адмін-панель, інакше виведіть повідомлення про помилку.

Якщо я спробую перейти на адмін-панель не авторизованим, то мене одразу ж перекинить на форму авторизації  


То як бачимо мене перекидує ось сюди:



При успішній авторизації:



Бачимо адмін-панель

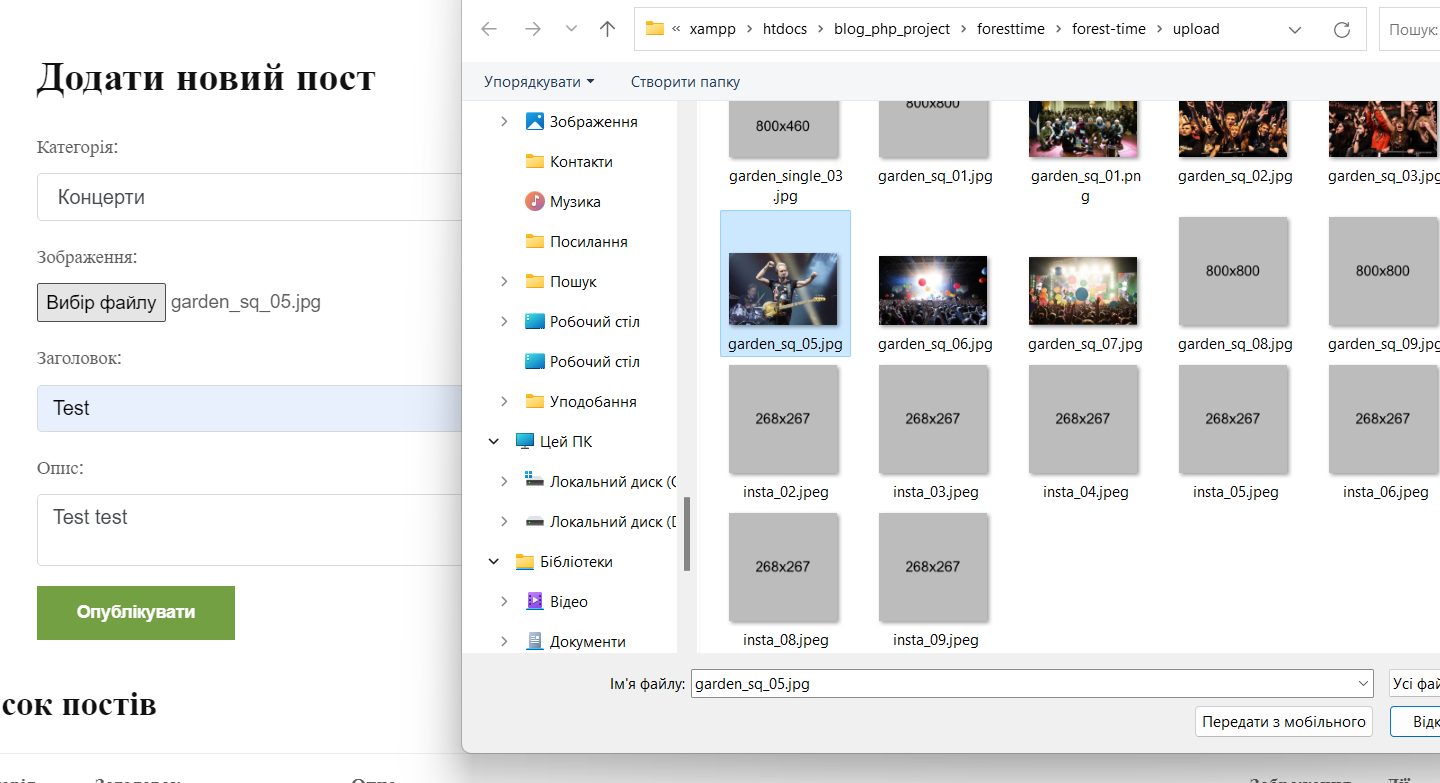
Також доступно зміна пароля:  

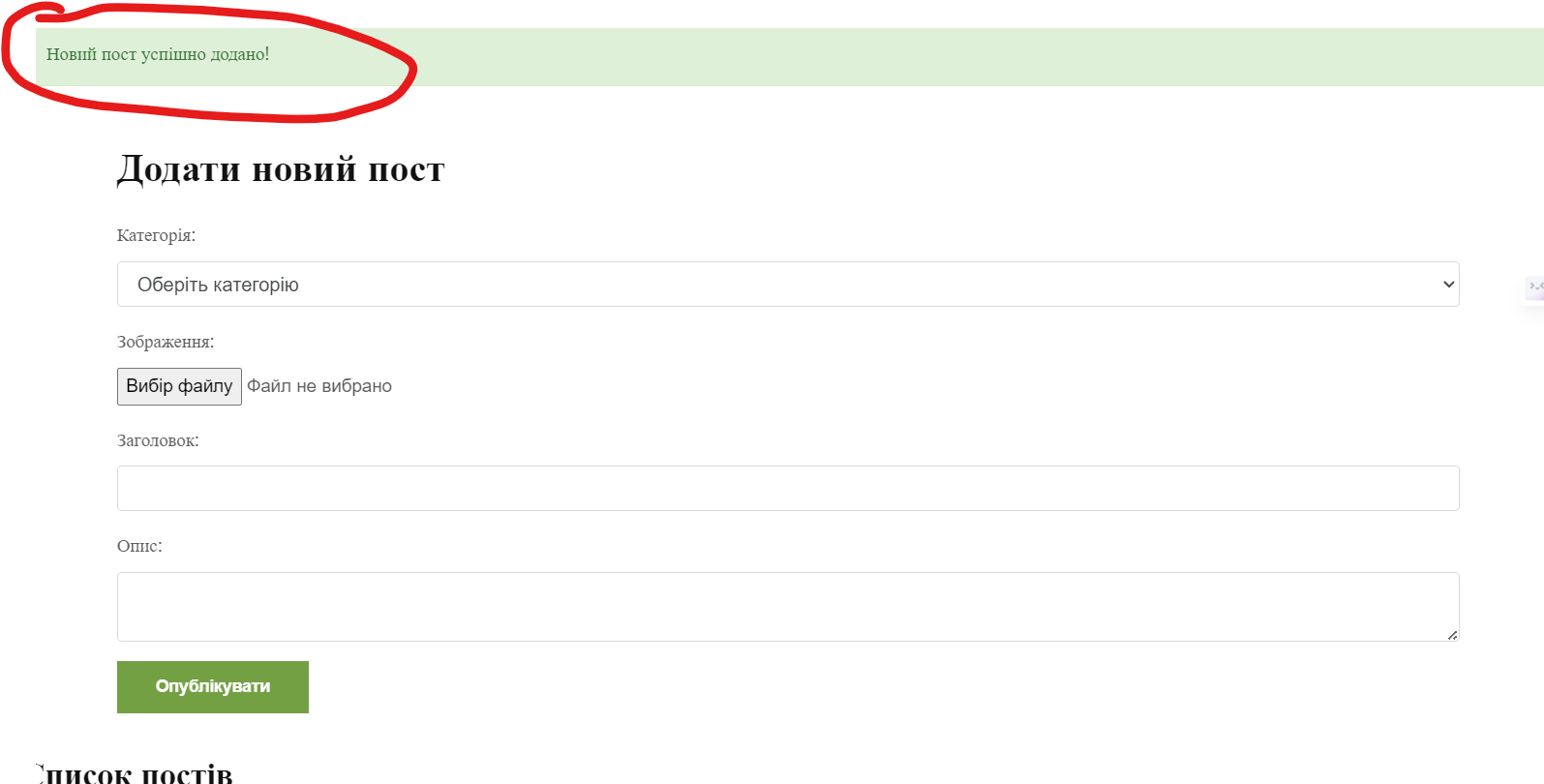

**2. Додавання нових постів:**

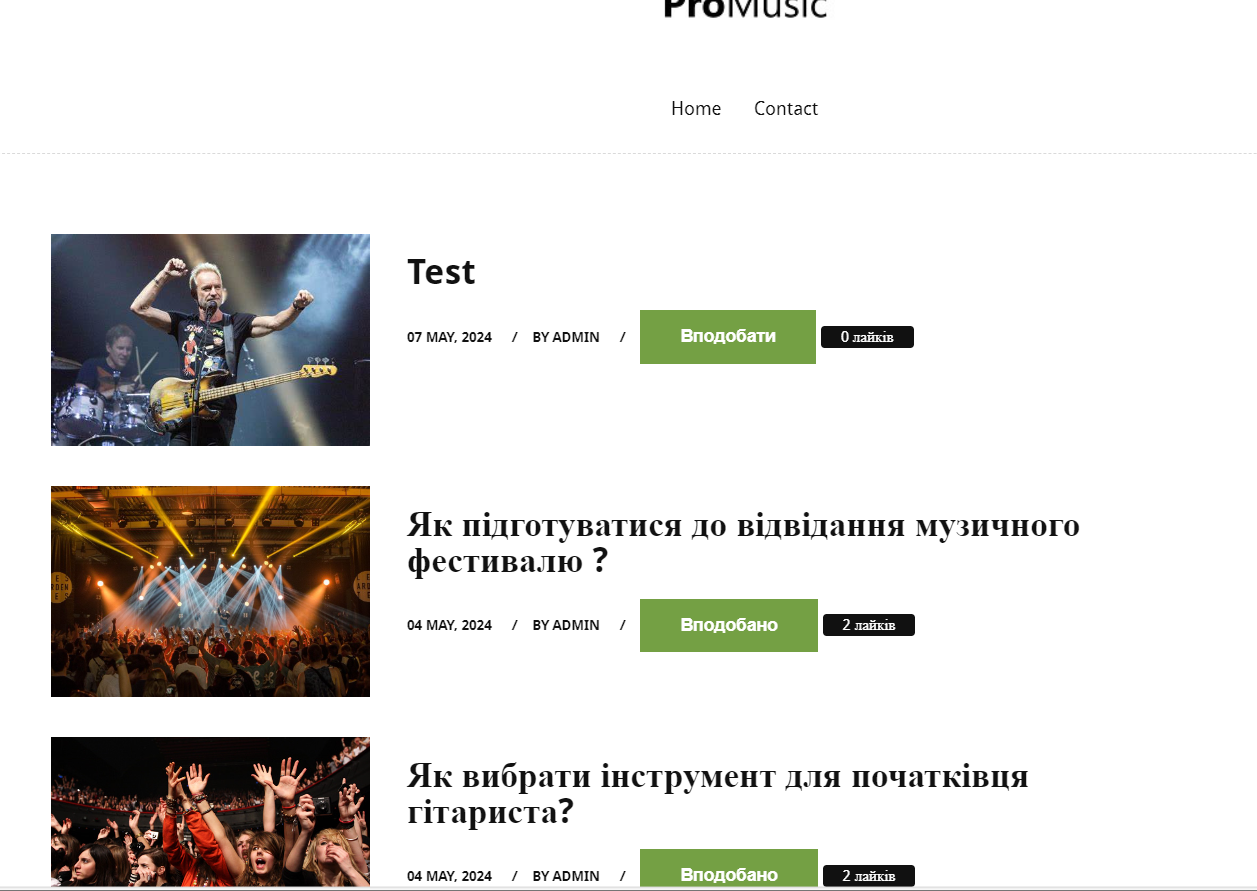
• Створіть форму додавання нового поста, яка міститиме поля для введення категорії поста, завантаження зображення, введення заголовку та опису.

• Забезпечте перевірку та валідацію введених даних перед їх збереженням у базі даних.

• Додайте функціональність для збереження нового поста у базі даних після натискання на кнопку "Опублікувати".





Результат:  


**3. Редагування та видалення постів:**

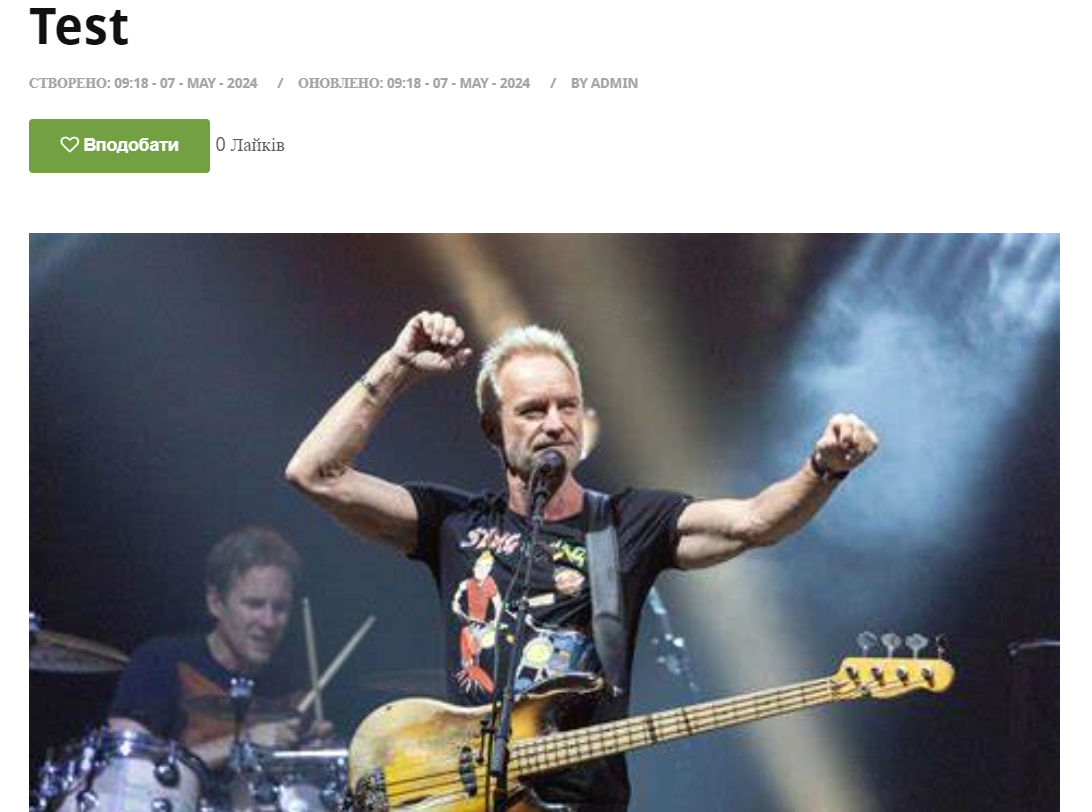
• На адмін-панелі виведіть список всіх наявних постів разом з можливістю редагування та видалення кожного поста.

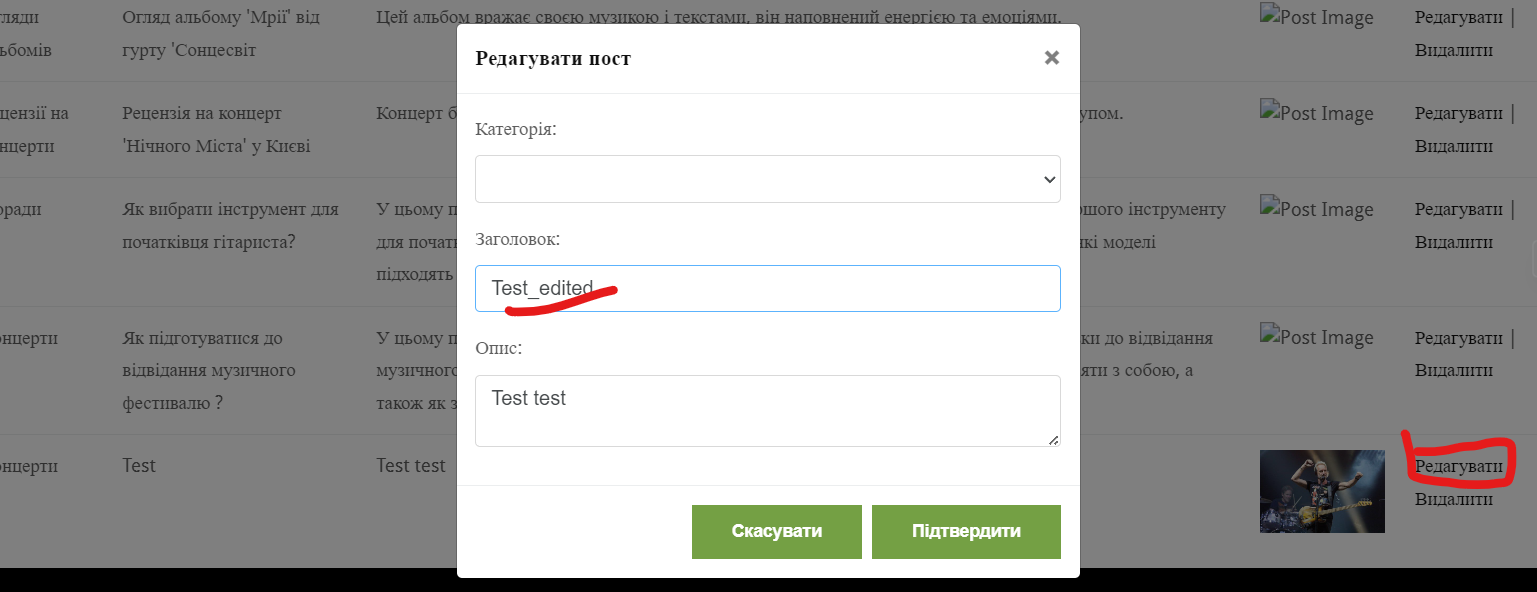
• Реалізуйте функціональність для редагування полів категорії, зображення, заголовку та опису поста.

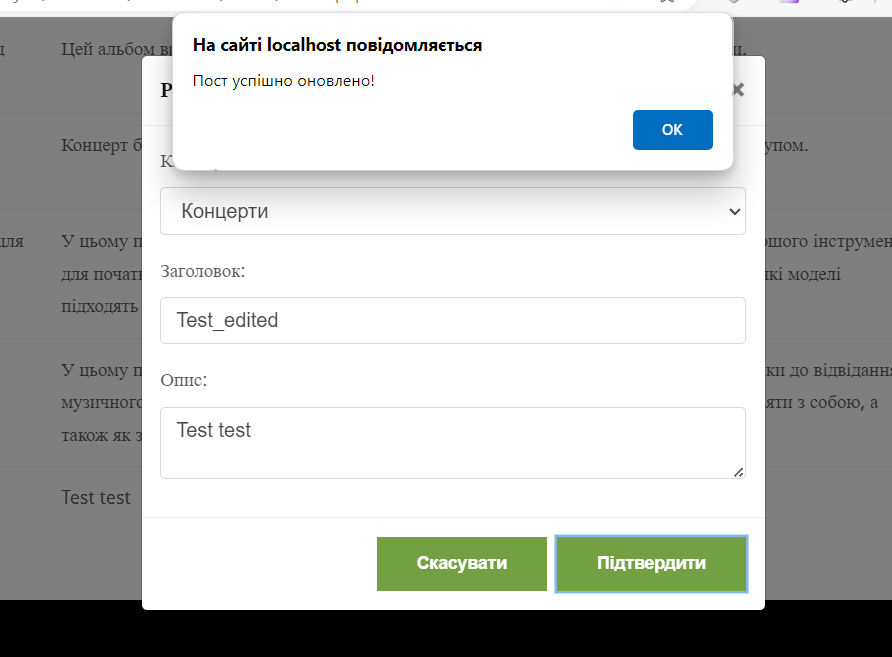
• Забезпечте підтвердження видалення поста перед видаленням його з бази даних.

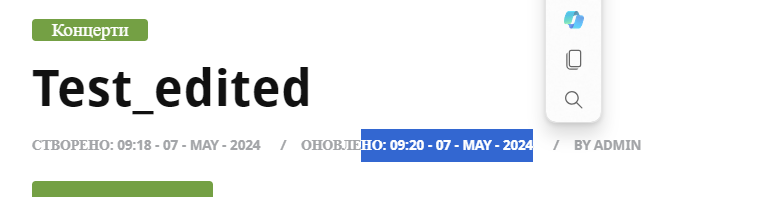
Тепер спробуємо його видалити та редагувати:

Ось детальний перегляд:

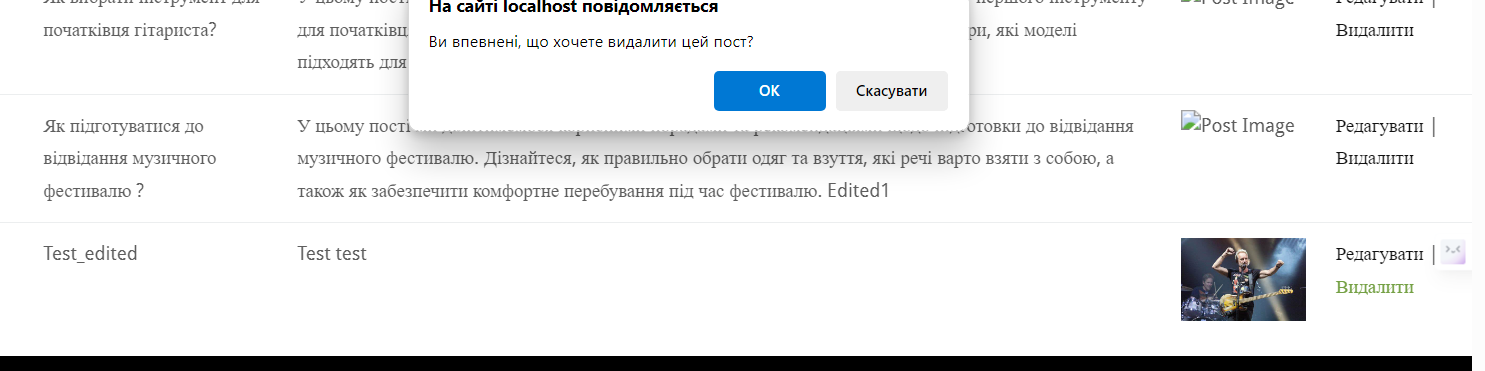




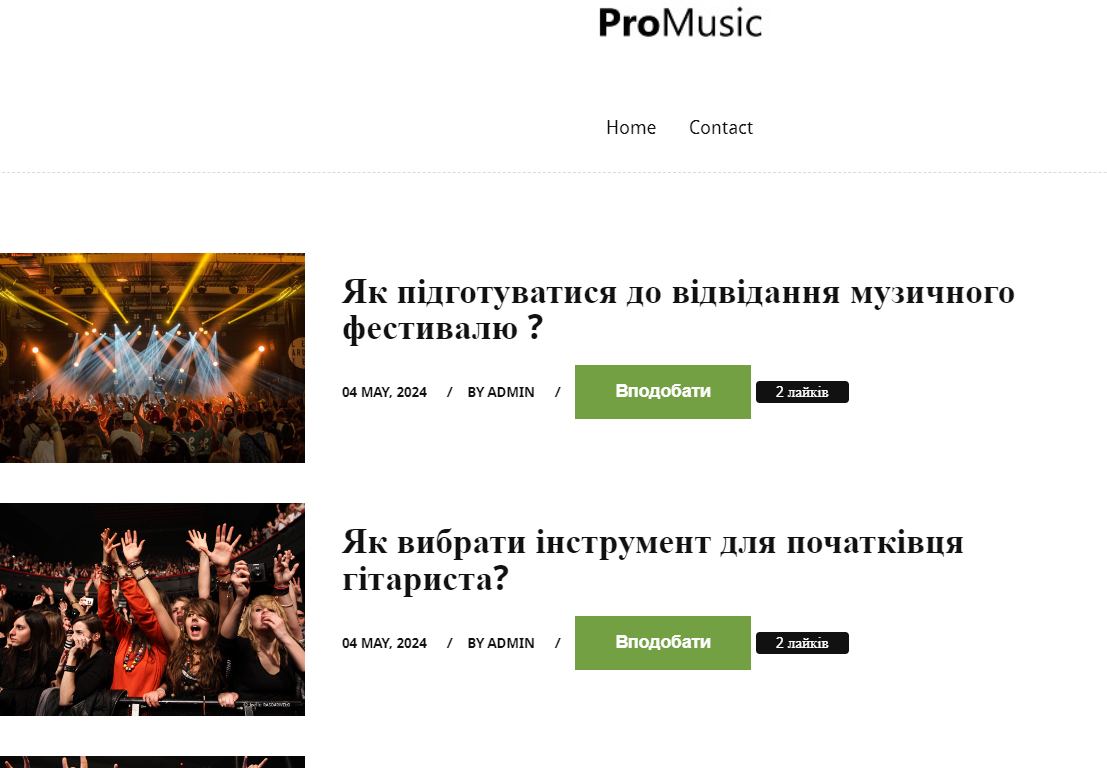


Результат:  


Тепер видалимо його:



Тепер, як бачимо його немає:



**Частина користувача:**

**1. Перегляд постів:**

Забезпечив перегляд постів:  


Код виведення:

<section class="section wb">

    <div class="container">

        <div class="row">

            <div class="col-lg-9 col-md-12 col-sm-12 col-xs-12">

                <div class="page-wrapper">

                    <div class="blog-list clearfix">

                        <?php if ($result && $result->num\_rows > 0): ?>

                            <?php while ($row = $result->fetch\_assoc()): ?>

                                <?php

                                $post\_id = $row['post\_id'];

                                $liked = false;

                                // Перевірка, чи користувач вже вподобав пост

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

                                    $user\_id = $\_SESSION['user\_id'];

                                    $like\_query = "SELECT \* FROM likes WHERE post\_id = $post\_id AND user\_id = $user\_id";

                                    $like\_result = $conn->query($like\_query);

                                    if ($like\_result && $like\_result->num\_rows > 0) {

                                        $liked = true;

                                    }

                                }

                                ?>

                                <div class="blog-box row">

                                    <div class="col-md-4">

                                        <div class="post-media">

                                            <a href="garden-single.php?id=<?php echo $post\_id; ?>" title="">

                                                <img src="upload/<?php echo $row['image']; ?>" alt="" class="img-fluid">

                                            </a>

                                        </div>

                                    </div>

                                    <div class="blog-meta big-meta col-md-8">

                                        <h4><a href="garden-single.php?id=<?php echo $post\_id; ?>"><?php echo htmlspecialchars($row['title']); ?></a></h4>

                                        <small><a href="garden-single.php?id=<?php echo $row['post\_id']; ?>" title=""><?php echo date('d M, Y', strtotime($row['created\_at'])); ?></a></small>

                                        <small>by admin</small>

                                        <button class="btn btn-primary" onclick="likePost(<?php echo $post\_id; ?>, this)">

                                            <?php echo $liked ? 'Вподобано' : 'Вподобати'; ?>

                                        </button>

                                        <span class="like-count"><?php echo $row['like\_count']; ?> лайків</span>

                                    </div>

                                </div>

                                <hr class="invis">

                                <?php endwhile; ?>

                        <?php else: ?>

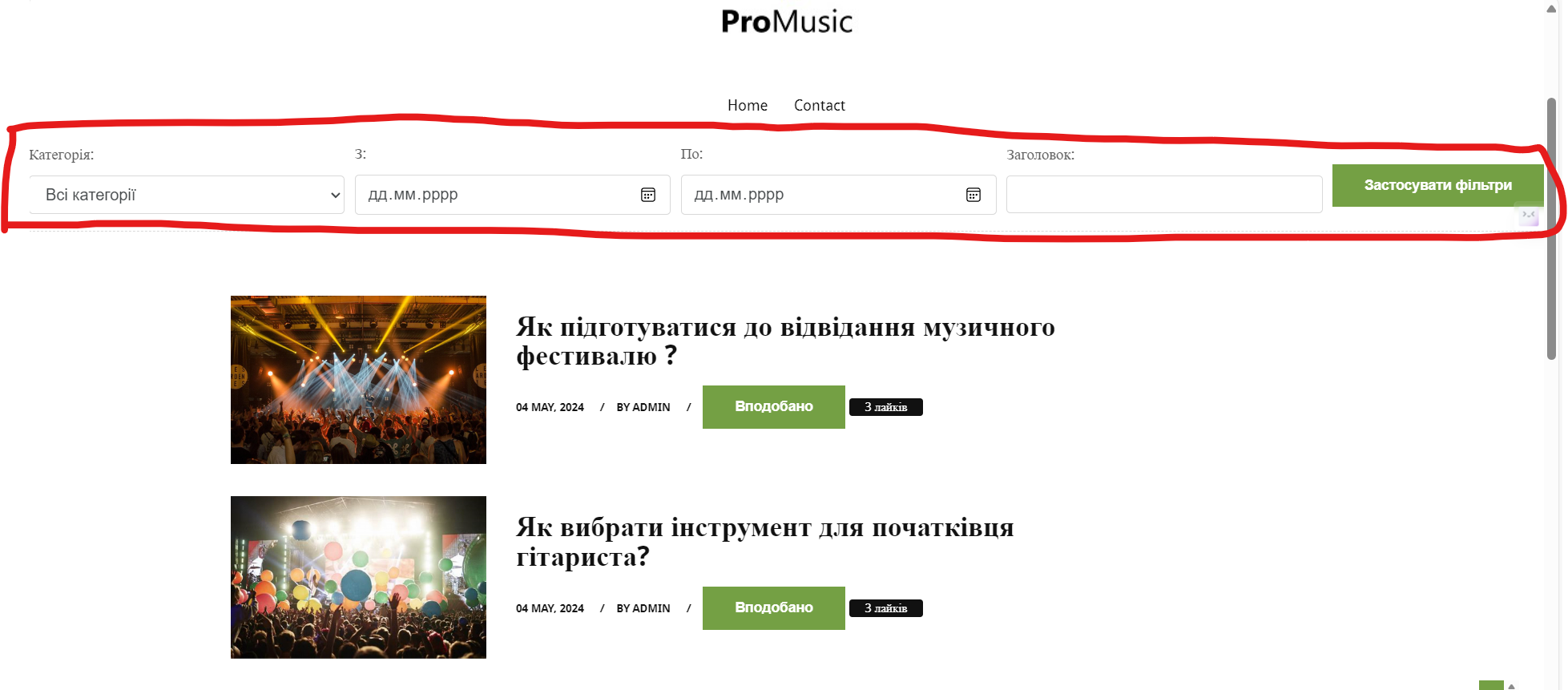
                            <p>No posts found.</p>

                        <?php endif; ?>

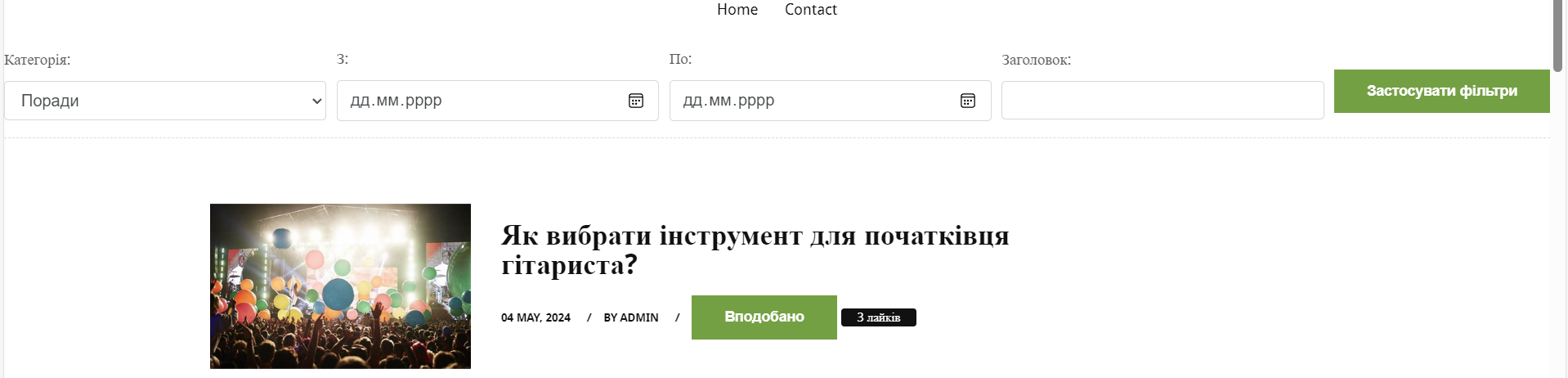
                        <?php $conn->close(); ?>

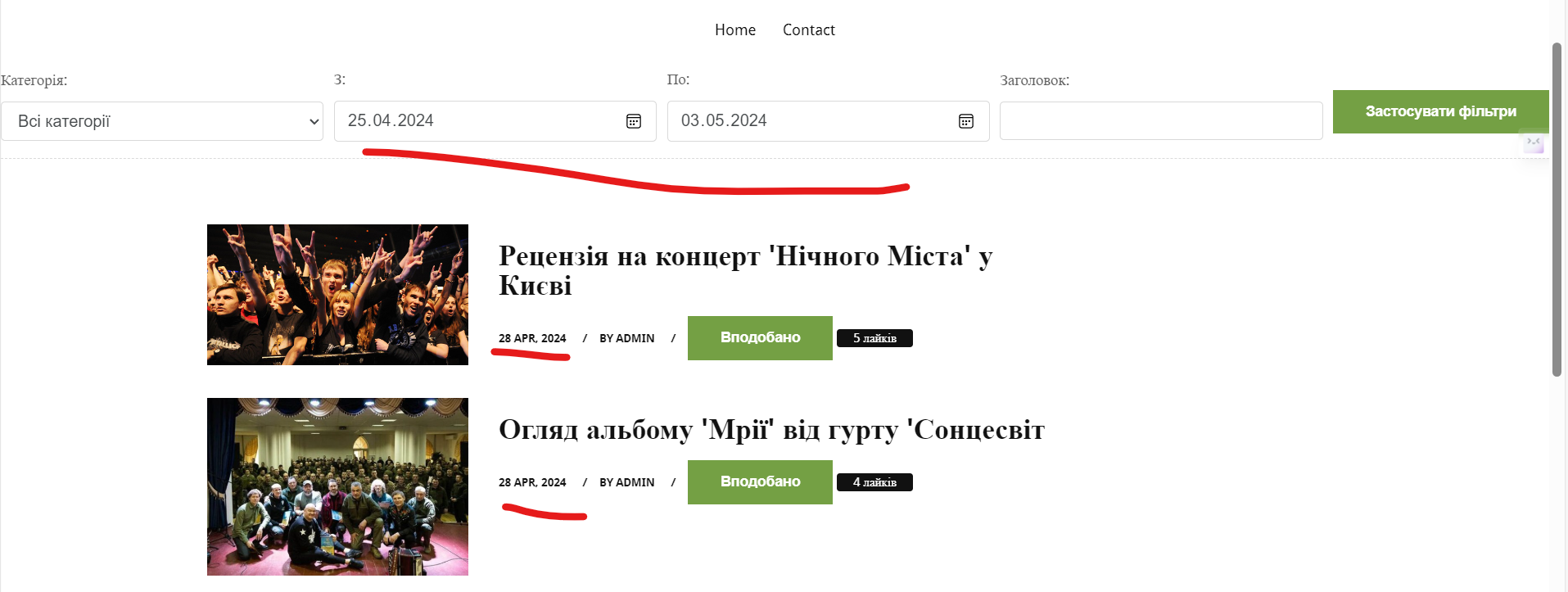
                    </div>

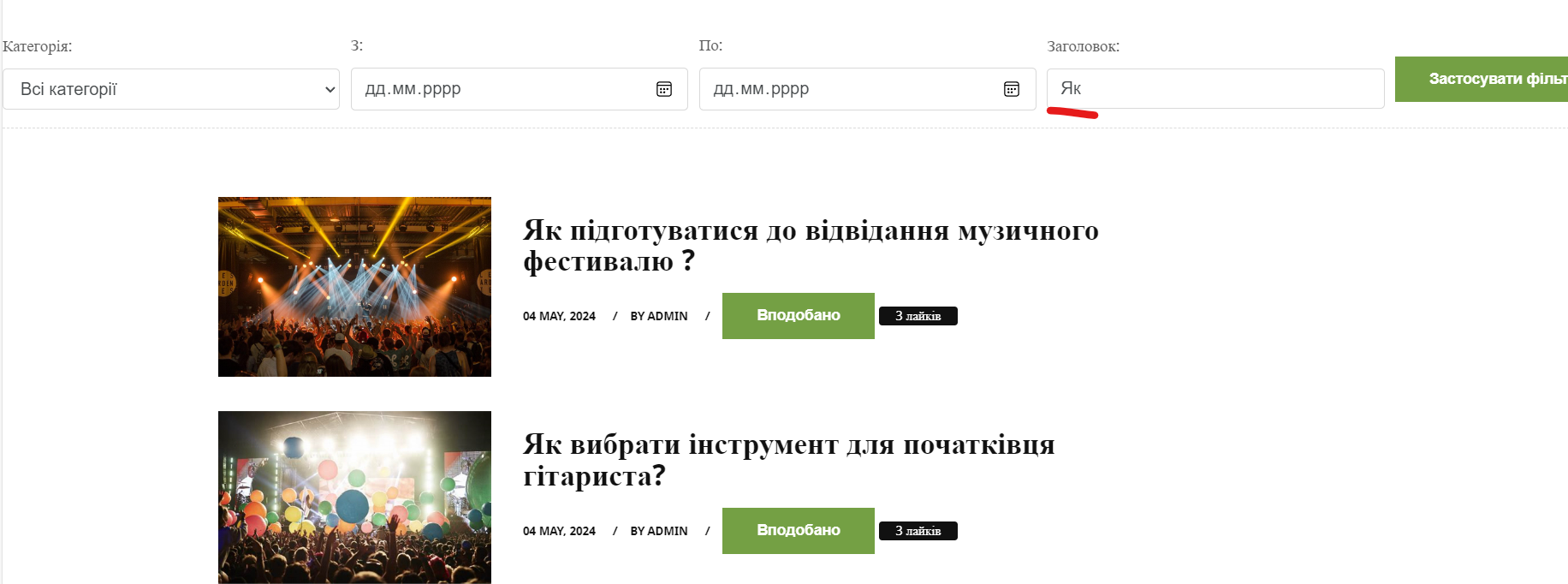
Створення та застосування фільтрів:



Фільтрація за категоріями:

  
Фільтрація за датою:



За заголовком:  


**3. Коментування та вподобання:**

• Реалізуйте можливість додавання коментарів до кожного поста з відображенням автора, дати та тексту коментаря.

• Додайте можливість користувачам виражати вподобання до постів та відображайте кількість вподобань для кожного поста.

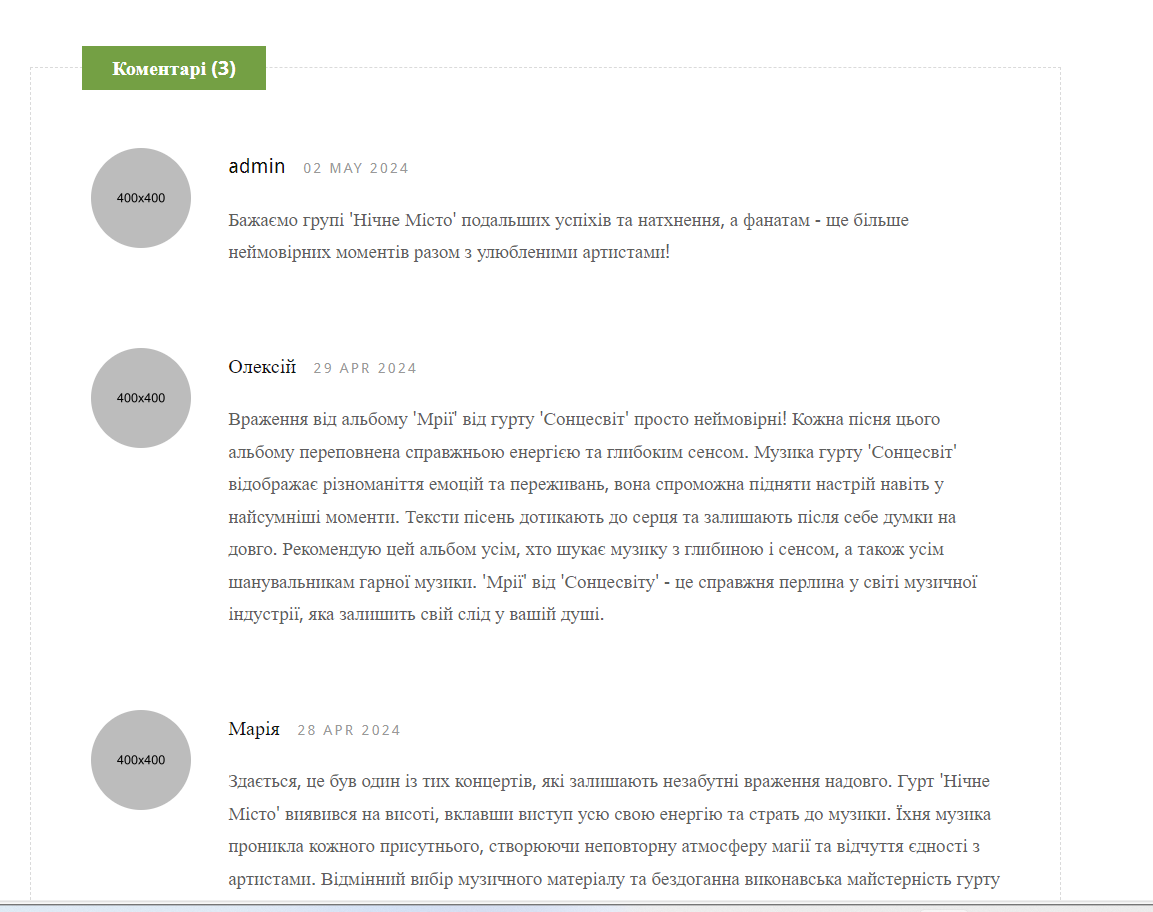
Перед натисканням:



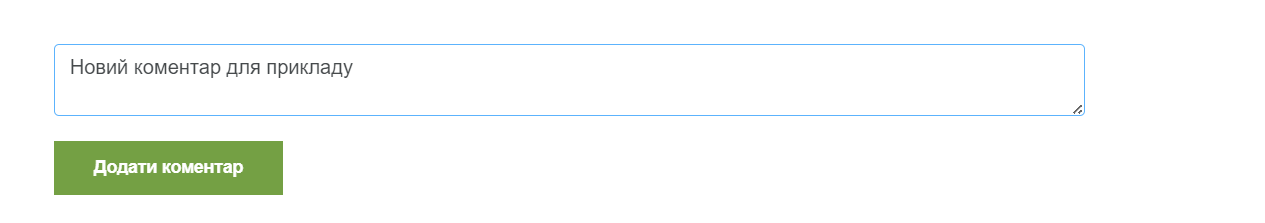
Після:



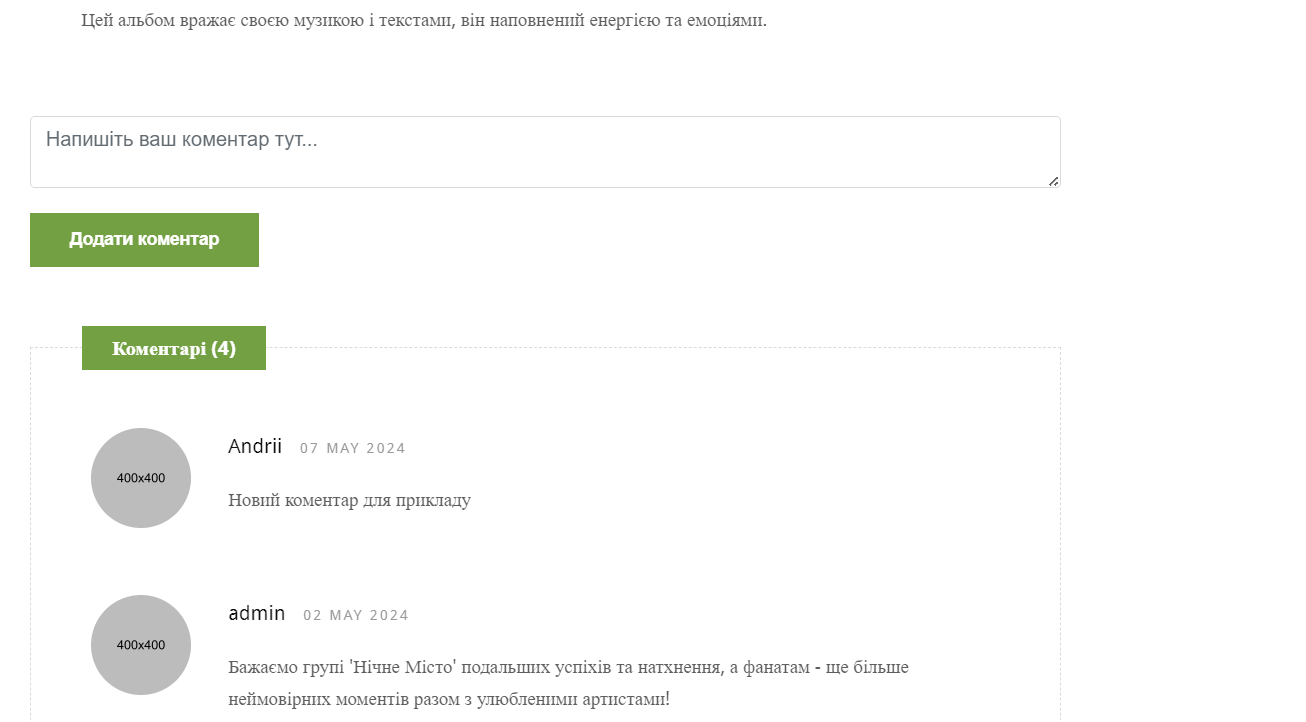
Тепер коментування під кожним постом:



Додаймо коментар:



Результат:



**4. Авторизований доступ:**

• Забезпечте доступ до функціоналу коментування та вподобань тільки для авторизованих користувачів.

• Перенаправте неавторизованих користувачів на сторінку входу або реєстрації, якщо вони спробують взаємодіяти з функціоналом, який потребує авторизації.

Для garden-index.php:

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

    header("Location: login.php");

    exit;

}

Із адмінки:

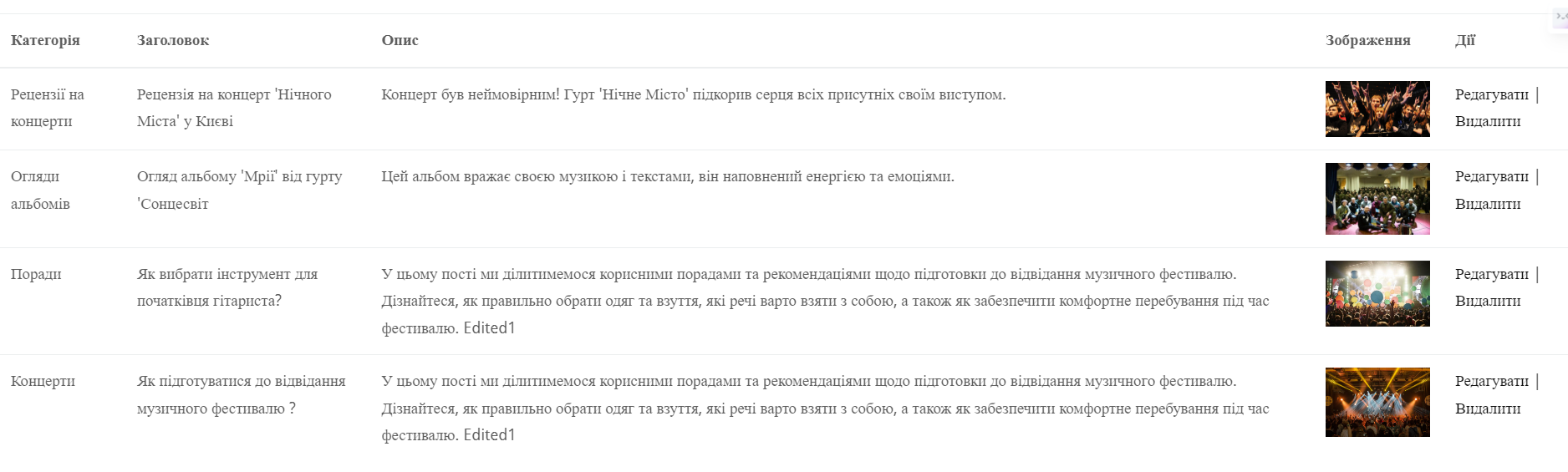
if (!isset($\_SESSION['admin\_id'])) {

    header("Location: login.php");

    exit;

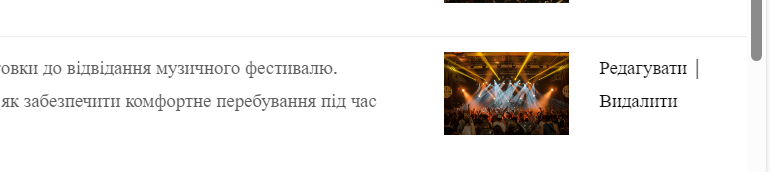
}

**Додавання редагування за фото та правильне виведення постів із фото:**

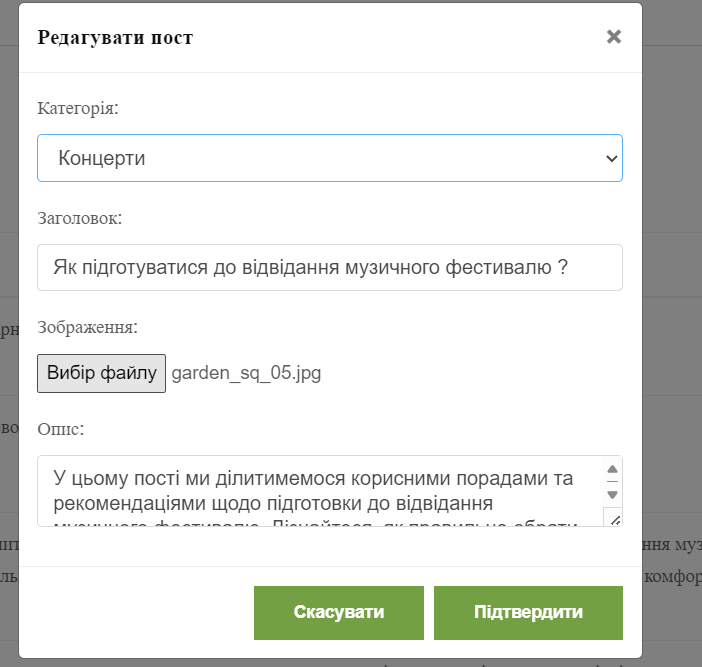
****

**Редагування фото:**

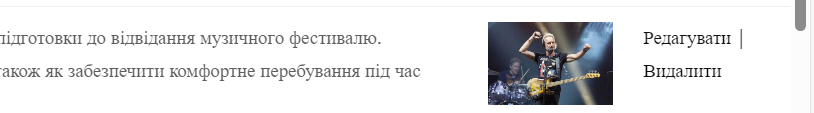
**До:**

****

**Зміна:**

****

**Після:**

****

**Висновок:** я оволодів базовими навичками основних функцій PHP. Освоїв використання СКБД MySQL та робота із сервером та налаштуванням БД. Вибірка елементів з БД. Виконання операцій над об’єктами.