**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Кафедра інженерії програмного забезпечення в**

**енергетиці**

*КУРСОВА РОБОТА*

з дисципліни: ««Основи Веб-програмування»

на тему: «Додаток для медитацій»

**Київ – 2025**

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ**

**ЕНЕРГЕТИКИ**

**КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В**

**ЕНЕРГЕТИЦІ**

КУРСОВА РОБОТА

з дисципліни: ««Основи Веб-програмування»

на тему: «Додаток для медитацій»

Керівник: Виконала:

Гагарін. О.О. Хомякова Дар’я Вадимівна

студентка 2 курсу

«Допущено до захисту» групи ТВ-33

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(особистий підпис керівника)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ р. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Захищено з оцінкою (особистий підпис виконавця)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

(оцінка)

Посилання на **GitHub** репозиторій https://github.com/Daria123H/-W\_Ok\_--33\_-.git

**Київ – 2025**

**Національний технічний університет України**

**“Київський політехнічний інститут імені Ігоря Сікорського”**

Навчально-науковий інститут атомної та теплової енергетики \_

( повна назва )

Кафедра інженерії програмного забезпечення в енергетиці\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

( повна назва )

Напрям підготовки \_121\_\_\_\_ Інженерія програмного забезпечення\_\_\_\_\_\_\_\_\_\_

(шифр і назва)

**З А В Д А Н Н Я**

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Хомяковій Дар’ї Вадимівні

(прізвище, ім’я, по батькові)

1. Тема роботи – «Додаток для медитацій»

Керівник курсової роботи – Недашківський О.Л.

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи: «26» грудня 2025 р.

3. Вихідні дані до проекту (роботи): HTML/CSS/JS та PHP.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): розробити додаток медитацій .

5. Дата видачі завдання: 14 січня 2025

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів виконання дипломного  проекту (роботи) | Строк виконання етапів проекту  ( роботи ) | Примітка |
| 1. | Затвердження теми роботи | 15.02.2025 | **Виконано** |
| 2. | Вивчення та аналіз задачі | 20.02.2025 | **Виконано** |
| 3. | Тестування ідеї гри | 25.02.2025 | **Виконано** |
| 4. | Створення тест плану | 03.03.2025 | **Виконано** |
| 5. | Розробка програмного коду | 10.04.2025 | **Виконано** |
| 6. | Розробка інтерфейсу програми | 25.04.2025 | **Виконано** |
| 7. | Тестування програми | 15.05.2025 | **Виконано** |
| 8. | Оформлення пояснювальної записки | 20.05.2025 | **Виконано** |

**Студент \_\_\_\_\_\_\_\_\_** \_\_\_Хомякова Д. В

( підпис ) (прізвище та ініціали)

**Керівник курсової роботи \_\_\_\_\_\_\_\_\_** Недашківський О. Л.

( підпис ) (прізвище та ініціали)

**АНОТАЦІЯ**

У ході виконання курсової роботи було освоєно основи розробки веб-додатків для медитації з використанням сучасних веб-технологій. Використані знання застосовувались для створення інтуїтивно зрозумілої платформи для користувачів, які бажають покращити своє психічне та фізичне здоров’я через медитаційні практики. Під час виконання роботи увага приділялась кожному етапу розробки — від визначення вимог до проекту до реалізації та тестування функціоналу. Основною метою роботи було створення веб-додатку, який би допоміг користувачам ефективно практикувати медитацію. Додаток включає функції вибору медитацій за категоріями, тривалістю, а також збереження персональних даних та улюблених медитацій.

Обсяг пояснювальної записки 64 аркушів, кількість ілюстрацій - 18.

**ANNOTATION**

During the course of the course work, the basics of developing web applications for meditation using modern web technologies were mastered. The knowledge gained was used to create an intuitive platform for users who want to improve their mental and physical health through meditation practices. During the work, attention was paid to each stage of development - from defining project requirements to implementing and testing the functionality. The main goal of the work was to create a web application that would help users practice meditation effectively. The application includes features for selecting meditations by category, duration, and saving personal data and favorite meditations.

The explanatory note is 64 pages long, with 18 illustrations .

**ЗМІСТ**

**ВСТУП…………………………………………………………………………….6**

**РОЗДІЛУ 1. СТВОРЕННЯ ДІАГРАМ КОМПОНЕНТІВ, ВЗАЄМОДІЇ ТА КЛАСІВ ВЕБ………………………………………………………………….….7**

**РОЗДІЛУ 2. ПРОЕКТУВАННЯ СТРУКТУРИ МЕНЮ З УРАХУВАННЯМ ПОТРЕБ КОРИСТУВАЧА………………………………11**

**РОЗДІЛУ 3. РОЗМІЩЕННЯ ТА СТИЛІЗАЦІЯ ЕЛЕМЕНТІВ ІНТЕРФЕЙСУ ВИКОНАННЯ……………………………………………..…13**

**РОЗДІЛУ 4. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ДАНИХ ДЛЯ ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ……………………………………...….15**

**РОЗДІЛ 5. ХІД ВИКОНАННЯ РОБОТИ……………………………………18**

5.1. Використані технології та інструменти……………………………………18

5.2. Розробка механізмів отримання та оновлення даних…………………..…20

5.3 Розробка бази даних для збереження та отримання інформації…………22

5.4. Створення серверної архітектури додатку…………………………….…..24

5.5. Реалізація обробки запитів та взаємодії з базою даних на сервері………26

5.6. Тестування та налаштування веб додатку на сервері………………..……27

**РОЗДІЛ 6. КЕРІВНИЦТВО КОРИСТУВАЧА………………………………31**

6.1. Навігація в додатку………………………………………………………….31

6.2. Авторизація…………………………………………………………...……..31

6.3. Перегляд медитацій…………………………………………………………31

6.4. Профіль………………………………………………………….……...……32

**РОЗДІЛ 7. ВИГЛЯД СТОРІНОК WEB-САЙТУ……………………………33**

**ВИСНОВКИ…………………………………………………………………..…37**

**CПИCOК ВИКOРИCТAНИX ДЖEРEЛ…………………………………….38**

**ДОДАТКИ………………………………………………………………………..39**

**ВСТУП**

У сучасному світі медитації є одними з найвпливовіших технік для урівноваження та збереження спокою, бо постійне життя в стресі, швидкому темпі життя та перевантаженні інформацією, щоб і стали звичними явищами, але збереження здоров'я та психічного благополуччя все ж таки актуальне. Стрес і втома негативно впливають на фізичне та психічне здоров'я і знижують продуктивність.

Метою цього проєкту є створення веб-додатку для медитації, який дозволить користувачам ознайомитися з різними техніками медитації. Програмне забезпечення буде простим у використанні та адаптованим до вподобань користувачів. Крім того, у системі буде передбачена роль адміністратора.

Для реалізації проєкту будуть використані веб-технології: HTML, CSS, JavaScript для інтерфейсу, а також PHP і MySQL для роботи з базою даних. Додаток буде зручним для користувачів, дозволяючи легко вибирати медитації за категоріями та тривалістю.

**РОЗДІЛУ 1. СТВОРЕННЯ ДІАГРАМ КОМПОНЕНТІВ, ВЗАЄМОДІЇ ТА КЛАСІВ ВЕБ**

Проєкт "Додаток для медитації" створений для того, щоб допомогти людям почати медитувати та відчути себе краще. Цей веб-сайт допоможе знайти різні медитації, які допомагають розслабитися, зняти стрес, відновити дихання, інші техніки для покращення психічного та фізичного здоров'я. На сайті є інформація загальна про медитації, рекомендації та відео.

Кожен користувач може створити свій власний акаунт. Для цього потрібно вказати свою електронну пошту, прізвище та придумати пароль. Це потрібно для того, щоб кожен мав свій особистий доступ до програми і міг зберігати свої улюблені медитації, які сам же зможе додати.

На сайті буде кілька різних медитацій, які можна вибрати в залежності від того, що хочеш зробити - розслабитися, зняти стрес або покращити концентрацію. Кожна медитація буде мати інструкцію, що потрібно робити та також відео для наочного показу. Це допоможе людям, які тільки починають медитувати.

Передбачена можливість фільтрації медитацій за:

* категорією (релаксація, сон, концентрація тощо),
* тривалістю (коротка, середня, довга),
* рівнем складності (початковий, середній, просунутий).

Користувачі зможуть вибрати, скільки часу хочуть медитувати, та який тип медитації їм підходить. Наприклад, можна вибрати коротку медитацію на 5 хвилин або довшу на 30 хвилин. Це дасть змогу вибирати медитацію відповідно до особистих потреб і часу.

На сайті, окрім ролі звичайного користувача, передбачена роль адміністратора. Адміністратор має розширені можливості:

* додавати, редагувати або видаляти медитації,
* переглядати та керувати обліковими записами користувачів,
* призначати нових адміністраторів або змінювати їх статус на звичайного користувача.

Цей додаток створено для того, щоб кожен міг легко та зручно почати практику медитації, а завдяки простому інтерфейсу, фільтрам, інструкціям і відео – швидко знайти те, що підходить саме йому.

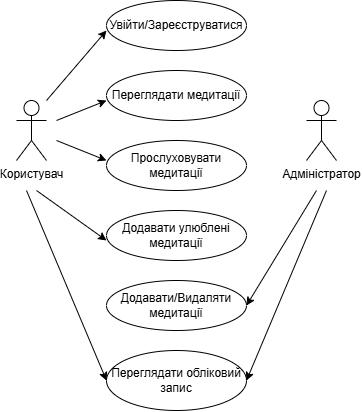


Рисунок 1.1. Діаграма варіантів використання.

Діаграма варіантів використання (Рисунок 1.1) ілюструє основні сценарії взаємодії користувачів із системою, включаючи авторизацію, перегляд і фільтрацію медитацій, додавання до улюблених, а також адміністративні функції, такі як додавання/видалення медитацій і керування обліковими записами.

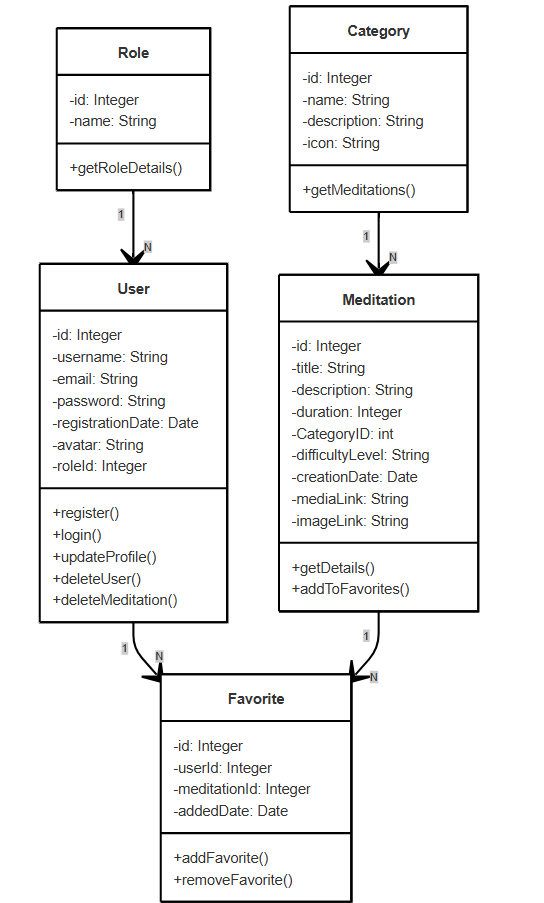


Рисунок 1.2. Діаграма класів.

Діаграма класів (Рисунок 1.2) відображає основні сутності системи (Користувач, Медитація, Категорія, Улюблене, Роль) із їхніми атрибутами та зв’язками між ними, такими як один-до-багатьох між користувачем і улюбленими медитаціями.

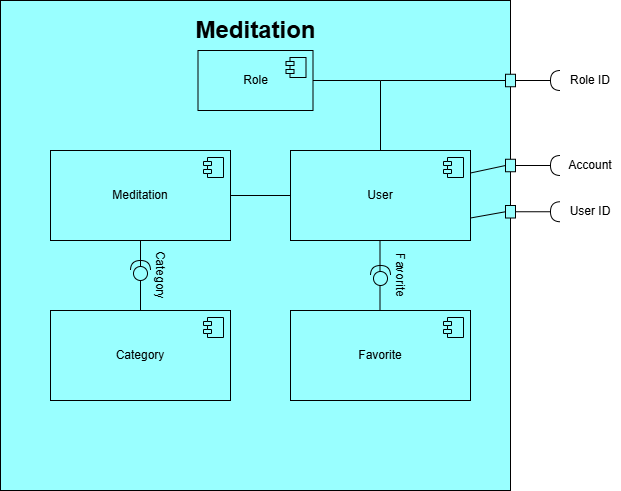


Рисунок 1.3. Діаграма компонентів

Діаграма компонентів (Рисунок 1.3) показує структуру системи, включаючи клієнтську частину (HTML, CSS, JavaScript), серверну частину (PHP) та базу даних (MySQL), а також їхню взаємодію через HTTP-запити та SQL-запити.

**РОЗДІЛУ 2. ПРОЕКТУВАННЯ СТРУКТУРИ МЕНЮ З УРАХУВАННЯМ ПОТРЕБ КОРИСТУВАЧА**

Меню додатку для медитації буде включати наступні пункти:

* Головна
* Медитації
* Про медитації
* Мій профіль

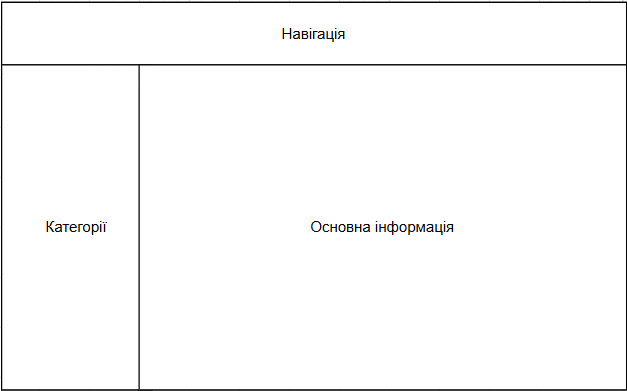


Рисунок 2.1 Попереднє схематичне зображення розміщення меню у веб-додатку.

В комп'ютерній версії веб-додатку меню буде знаходитись у верхній частині екрану у вигляді горизонтальної панелі навігації, а в верхній частині будуть додаткові фільтри для зручного доступу до різних категорій медитацій.

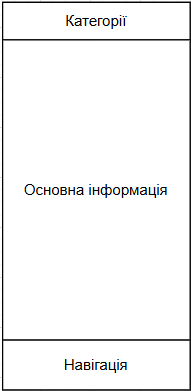


Рисунок 2.2 Розміщення елементів меню у мобільній версії веб-додатку.

В мобільній версії меню буде знаходитись у нижній частині екрану у вигляді панелі з іконками для швидкого доступу до основних розділів.

1. Головна
2. Елемент меню переадресовуватиме користувача на головну сторінку,

де будуть відображатися рекомендовані медитації, загальна інформація, новини.

2. Медитації

a. Елемент меню переадресовуватиме користувача на сторінку зі списком усіх доступних медитацій, згрупованих за категоріями.

b. У цьому розділі користувач зможе шукати та фільтрувати медитації за різними параметрами.

3. Про медитації

a. Елемент меню переадресовуватиме користувача на сторінку з детальною історією про медитацію, цікавими фактами та інше.

4. Мій профіль

a. Елемент меню переадресовуватиме користувача на сторінку з інформацією про його обліковий запис, збереженими медитаціями.

**РОЗДІЛУ 3. РОЗМІЩЕННЯ ТА СТИЛІЗАЦІЯ ЕЛЕМЕНТІВ ІНТЕРФЕЙСУ ВИКОНАННЯ**

1. Сторінка створення облікового запису користувача

a. На сторінці присутні чотири поля для введення даних:

* Ім'я користувача
* Електронна адреса
* Пароль
* Повторне введення паролю

b. Та кнопка:

* Створення облікового запису

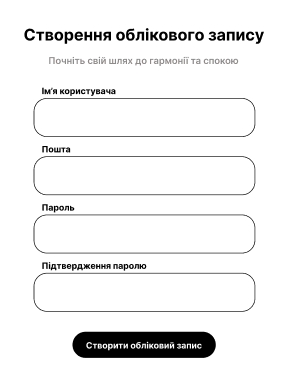


Рисунок 3.1 Схема розміщення елементів інтерфейсу сторінки створення облікового запису користувача.

1. Сторінка авторизації

a. На сторінці присутні два поля для введення даних:

* Електронна адреса
* Пароль

b. Та дві кнопки:

* Вхід в систему
* Переадресація на сторінку реєстрації (з текстом "Створити новий обліковий запис")



Рисунок 3.2 Схема розміщення елементів інтерфейсу сторінки авторизації.

**РОЗДІЛУ 4. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ МОДЕЛЕЙ ДАНИХ ДЛЯ ЗБЕРЕЖЕННЯ ІНФОРМАЦІЇ**

В системі передбачено наступні сутності даних:

* Користувач
* Медитація
* Категорія
* Улюблене

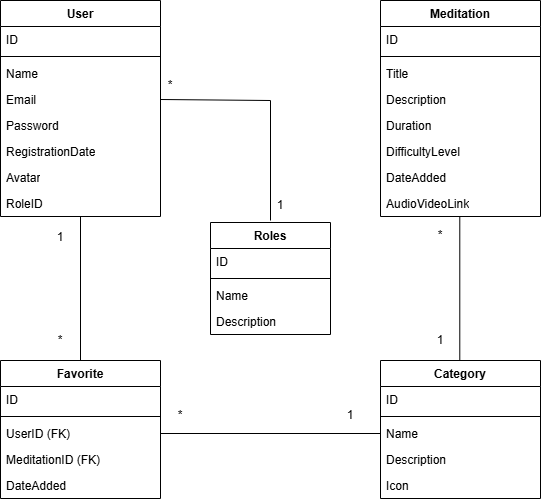


Рисунок 4.1 Діаграма моделей даних для збереження інформації.

1. Модель даних "Користувач" зберігає в собі необхідні атрибути для користувачів системи:

* ID користувача (первинний ключ)
* Ім'я користувача
* Електронна адреса
* Пароль
* Дата реєстрації
* Аватар (посилання на зображення)
* ID ролі (зовнішній ключ, що вказує на таблицю ролей)

1. Модель даних "Роль" визначає ролі користувачів у системі (наприклад, звичайний користувач, адміністратор тощо):

* ID ролі (первинний ключ)
* Назва ролі
* Опис ролі

1. Модель даних "Медитація" зберігає інформацію про доступні медитації:

* ID медитації (первинний ключ)
* Назва
* Опис
* Тривалість (у хвилинах)
* Рівень складності
* Дата додавання
* Посилання на аудіо/відео файл
* Посилання на зображення

1. Модель даних "Категорія" зберігає інформацію про категорії медитацій:

* ID категорії (первинний ключ)
* Назва категорії
* Опис категорії
* Іконка категорії

1. Модель даних "Улюблене" зберігає інформацію про улюблені медитації користувача:

* ID запису (первинний ключ)
* ID користувача (зовнішній ключ)
* ID медитації (зовнішній ключ)
* Дата додавання

Залежності між моделями:

* Один користувач може мати багато улюблених медитацій
* Одна медитація може бути прослухана багатьма користувачами
* Одна медитація може бути в списку улюблених у багатьох користувачів
* Одна категорія може містити багато медитацій

**РОЗДІЛ 5.ХІД ВИКОНАННЯ РОБОТИ**

**5.1. Використані технології та інструменти**

Щоб створити зручний і гарний додаток для медитацій, потрібно було обрати технології. Почнемо з фронтенду. Фронтенд - це частина додатка, з якою взаємодіє користувач, тобто інтерфейс веб-сторінок. Для його створення використано три основні технології: **HTML**, **CSS** і **JavaScript**. HTML - це мова, з якої починається будь-який сайт. Вона дозволяє створювати "скелет" сторінки, наприклад, заголовки, параграфи, кнопки, форми тощо. У додатку HTML відповідає за створення структури сторінок, таких як home.php, profile.html та сторінки окремих медитацій. Семантичні теги <header>, <nav>, <section> і <footer> допомогли організувати контент та зробити його зрозумілим для пошуку конкретних фрагментів[1].

**<header>**

**<nav>**

**<a href="#home">Головна</a>**

**<a href="#meditations">Медитації</a>**

**<a href="#about">Про медитації</a>**

**</nav>**

**</header>**

CSS3 — це мова стилів, яка відповідає за зовнішній вигляд сторінки: кольори, шрифти, розташування елементів, анімації тощо. CSS використано для створення дизайну[2]. Flexbox і Grid застосовувалися для розташування елементів, таких як картки медитацій у сітці. Також присутні анімації, наприклад ефект наведення на кнопки чи картки. Flexbox дозволяє легко вирівнювати елементи в рядок або стовпець, що ідеально підходить для адаптивного дизайну. Наприклад, властивість display: flex у секції .home-content забезпечує розташування тексту та зображення поруч.[3]

**.meditation-list {**

**display: grid;**

**grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));**

**gap: 30px;**

**}**

**.meditation-item:hover {**

**transform: scale(1.05);**

**box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);**

**}**

JavaScript - це мова програмування, яка додає інтерактивність до веб-сторінок, наприклад, обробку кліків, фільтрацію контенту чи взаємодію з сервером. JavaScript відповідає за такі елементи, як фільтрація медитацій, асинхронне завантаження даних через Fetch API, збереження даних у localStorage. Наприклад, функція addToFavorites додає медитацію до улюблених. Fetch API - це спосіб надсилання HTTP-запитів[4].

**document.getElementById('meditation-filters').addEventListener('submit', (e) => {**

**e.preventDefault();**

**const selectedCategory = document.getElementById('category').value;**

**const selectedDuration = document.getElementById('duration').value;**

**const meditations = document.querySelectorAll('.meditation-item');**

**meditations.forEach(article => {**

**const matchesCategory = selectedCategory === 'all' || selectedCategory === article.getAttribute('data-category');**

**const matchesDuration = selectedDuration === 'all' || selectedDuration === article.getAttribute('data-duration');**

**article.style.display = matchesCategory && matchesDuration ? 'block' : 'none';**

**});**

**});**

Бекенд обробляє запити користувачів, взаємодіє з базою даних і повертає дані на фронтенд. Використано **PHP** і **MySQL**. **PHP -** це мова програмування, яка працює на сервері. Її використовують для того, щоб сайт міг «думати» і взаємодіяти з базою даних. У моєму додатку PHP відповідає за такі дії, як реєстрація, вхід користувача, додавання медитацій у список улюблених або завантаження нових медитацій. Щоб захистити сайт від зламів (наприклад, від SQL-ін’єкцій), PHP використовує спеціальні запити, які не дозволяють зловмисникам вплинути на базу даних через форму[5,6].

**$stmt = $conn->prepare("SELECT \* FROM User WHERE Email = ?");**

**$stmt->bind\_param("s", $email);**

**$stmt->execute();**

**$result = $stmt->get\_result();**

**if ($result->num\_rows > 0) {**

**echo json\_encode(['success' => false, 'message' => 'Email вже зареєстрований']);**

**exit;**

**}**

**MySQL** — це система, яка зберігає всі дані сайту в таблицях. Наприклад, у мене є таблиці для користувачів, медитацій, категорій і списків улюбленого. Ці таблиці можуть бути пов’язані між собою через спеціальні ключі. Наприклад, щоб дістати всі медитації разом з їхніми категоріями, ми об’єднуємо дві таблиці за допомогою SQL-запиту.

**5.2. Розробка механізмів отримання та оновлення даних**

Реєстрація: скрипт отримує JSON-дані через php://input та декодує їх у масив PHP за допомогою json\_decode. Якщо дані відсутні або JSON некоректний, повертається помилка з відповідним повідомленням. Перевіряється наявність обов'язкових полів (username, email, password). Якщо хоча б одне поле відсутнє, повертається помилка. Виконується запит до бази даних для перевірки, чи існує користувач із вказаною електронною поштою. Якщо email вже зареєстрований, повертається повідомлення про помилку. Пароль хешується за допомогою функції password\_hash з алгоритмом за замовчуванням (PASSWORD\_DEFAULT) для безпечного зберігання. Якщо всі перевірки пройдено, дані користувача (ім'я, email, хеш пароля, дата реєстрації та аватар) додаються до таблиці User у базі даних за допомогою підготовленого запиту (prepared statement), що забезпечує захист від SQL-ін'єкцій. У разі успішної реєстрації повертається JSON-об'єкт із полем success: true та повідомленням про успішну реєстрацію. У разі помилки повертається success: false із описом проблеми.

Авторизація: дані отримуються через $\_POST (email та password). Виконується запит до бази даних для пошуку користувача за вказаним email. Якщо користувач існує, отримуються його дані. Пароль, отриманий від клієнта, перевіряється на відповідність збереженому хешу за допомогою функції password\_verify. Якщо авторизація успішна, повертається JSON-об'єкт із полем success та ім'ям користувача (username). У разі помилки повертається повідомлення про некоректний email або пароль.

Додавання медитації до улюблених:дані отримуються через $\_POST та приводяться до цілих чисел для безпеки.Перевіряється коректність даних (user\_id та meditation\_id мають бути більше 0). Якщо дані некоректні, повертається помилка. Виконується запит до таблиці Favorite, щоб перевірити, чи медитація вже додана до улюблених для цього користувача. Якщо запис існує, повертається повідомлення про це.Якщо медитація ще не додана, створюється новий запис у таблиці Favorite із UserID, MeditationID та поточною датою (NOW()).У разі успіху повертається текстовий рядок "added". У разі помилки повертається повідомлення з описом.

**5.3 Розробка бази даних для збереження та отримання інформації**

База даних - це місце, де зберігається вся важлива інформація для додатка. Наприклад, у базі зберігаються дані про користувачів, медитації, категорії й улюблені записи. Вона потрібна для того, щоб додаток міг швидко знаходити потрібну інформацію, додавати нову або змінювати стару. Для збереження всієї інформації я обрала MySQL — це програма, яка дозволяє створювати і керувати базами даних. У ній дані зберігаються в таблицях, схожих на звичайні таблиці в Excel. Але на відміну від них, MySQL дозволяє швидко шукати інформацію і зв’язувати таблиці між собою. У таких базах дані можна пов’язувати між собою - наприклад, кожна медитація має свою категорію, і зв’язок між ними відбувається через спеціальне поле, яке називається ключ. Перед тим як створювати базу, я вирішила, які саме дані нам потрібні. Це інформація про користувачів (ім’я, email, пароль, роль, дата реєстрації), медитації (назва, опис, тривалість, рівень складності, категорія, аудіо/відео), категорії (наприклад, "спокій", "сон"), ролі користувачів (звичайний користувач або адміністратор) і список улюблених медитацій. Для цього я створила п’ять таблиць: одна зберігає користувачів, інша — медитації, третя — категорії, четверта —куди зберігаються улюблені медитації, а п’ята зберігає назви ролей.

Щоб створити базу даних для мого додатку, я використала програму XAMPP. Вона дозволяє запускати сервер і працювати з базою даних локально на комп’ютері. Через інструмент phpMyAdmin я створила базу з назвою meditation. Усередині бази я створила п’ять таблиць - кожна для свого типу даних. Таблиця User зберігає дані про користувачів. У ній є поля: ID (унікальний номер), ім’я, email, пароль, дата реєстрації і фото (аватар). Email має бути унікальним — тобто такий, що не повторюється в інших користувачів. User.RoleID — встановлює зв’язок між користувачем і його роллю, що дозволяє керувати правами доступу.

Я використовувала SQL-команду, яка створює цю таблицю з усіма потрібними полями. Таблиця Category зберігає категорії медитацій, наприклад, "сон", "спокій", "фокус". Вона дуже проста - лише два поля: ID і назва категорії. Таблиця Meditation - це основна таблиця, в якій зберігаються всі медитації. Кожен запис має назву, опис, тривалість у хвилинах, рівень складності, дату додавання, посилання на аудіо чи відео, а також інформацію про категорію. Категорія підключена через спеціальне поле - CategoryID, яке зв’язує медитацію з її категорією. Це називається зовнішній ключ. Таблиця Favorite показує, які медитації додав улюблене конкретний користувач. У ній є поле ID, ID користувача, ID медитації і дата додавання в улюблене. Зв’язок із таблицями користувачів і медитацій також зроблено через зовнішні ключі. Після створення структури бази я додали в неї перші дані. Спочатку ми внесли категорії, такі як «Релаксація», «Сон», «Концентрація» та інші. Потім кілька прикладів медитацій із назвами, описом, тривалістю та посиланнями на відео. Щоб мій сайт міг працювати з базою, я зробила спеціальний файл bd.php, у якому прописала підключення до MySQL через функцію mysqli. Я вказала адресу сервера, логін, пароль і назву бази. Після цього почали писати запити до бази. Наприклад, у файлі load\_meditations.php є запит, який дістає всі медитації разом з назвами категорій.

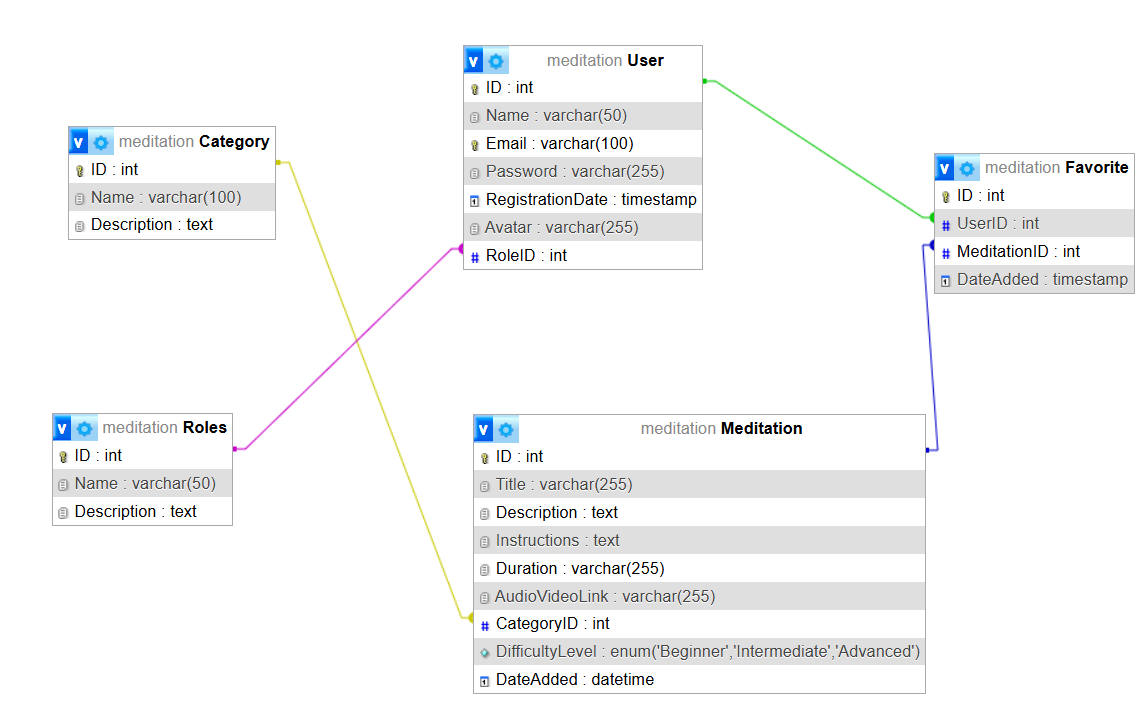


Рисунок 5.1. Модель бази даних

**5.4. Створення серверної архітектури додатку**

Для додатку для медитацій обрано **монолітну архітектуру**. Додаток має обмежену функціональність: реєстрація/авторизація користувачів, відображення медитацій, додавання медитацій до улюблених, перегляд профілю. Монолітна архітектура простіша для реалізації та підтримки на початковому етапі, коли проєкт невеликий.

Оскільки обрано монолітну архітектуру, взаємодія між компонентами відбувається **всередині одного додатку** через виклики функцій та методів. Для спілкування між клієнтом (браузером) і сервером використовується **REST API**:

* Фронтенд (home.php, profile.html) надсилає HTTP-запити до серверних ендпоінтів (register.php, login.php, favorites.php, get\_favorites.php).
* Відповіді повертаються у форматі JSON, що є стандартом для REST API.

У монолітній архітектурі серверна частина складається з кількох основних модулів (файлів PHP), які відповідають за різні аспекти функціональності.

Система складається з декількох PHP-модулів, кожен з яких виконує окрему функцію. Модуль bd.php відповідає за підключення до бази даних MySQL (зокрема, до бази meditation). У ньому налаштовуються параметри з'єднання — сервер, ім’я користувача, пароль і назва бази даних, а також реалізована обробка можливих помилок підключення. Цей модуль імпортується в усі інші PHP-файли для забезпечення доступу до БД.

Модуль register.php реалізує функцію реєстрації нових користувачів. Він приймає JSON-дані, що містять ім’я, email та пароль. Спочатку перевіряється унікальність email, після чого пароль хешується з використанням функції password\_hash, а дані нового користувача записуються до таблиці User. У відповідь повертається JSON, який інформує про результат реєстрації.

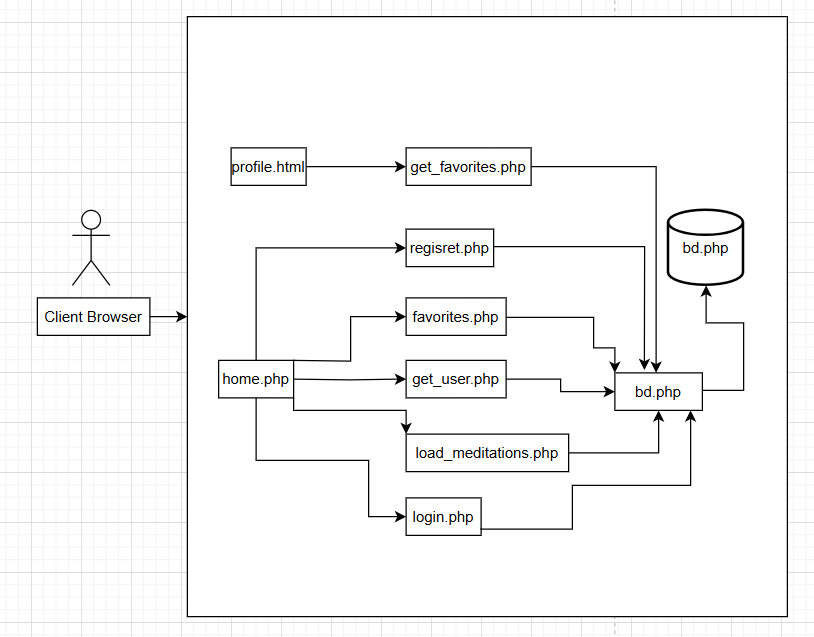
login.php забезпечує авторизацію користувачів. Він приймає дані email та пароль через POST-запит, перевіряє наявність користувача з таким email у базі, а також правильність пароля за допомогою password\_verify. У разі успішної авторизації повертається JSON-відповідь з результатом та ім’ям користувача.

favorites.php дозволяє користувачам додавати медитації до списку улюблених. Він приймає user\_id та meditation\_id, перевіряє, чи вже існує такий запис, і в разі відсутності — додає його до таблиці Favorite. У відповідь надсилається текстове повідомлення про успішність операції або помилку.

Щоб отримати список улюблених медитацій користувача, використовується модуль get\_favorites.php. Він приймає email користувача, знаходить його user\_id, після чого здійснює вибірку медитацій з таблиці Favorite, приєднуючи таблицю Meditation для отримання детальної інформації. Результат повертається у вигляді JSON-масиву.

Модуль get\_user.php призначений для отримання ідентифікатора користувача за його email. Він приймає email через POST, виконує відповідний запит до таблиці User і повертає JSON-відповідь з user\_id або повідомленням про помилку.

Останній модуль — load\_meditations.php — завантажує повний список медитацій. Він робить запит до таблиці Meditation та повертає JSON-масив із даними про кожну медитацію: назву, опис, категорію та тривалість.



**5.5. Реалізація обробки запитів та взаємодії з базою даних на сервері**

Серверна частина застосунку реалізована на PHP і включає кілька скриптів, кожен із яких виконує окрему функцію, таку як реєстрація, авторизація користувачів, робота з медитаціями та улюбленими. Запити надсилаються через HTTP, здебільшого методом POST, а відповіді повертаються у форматі JSON або простого тексту.

Реєстрація користувача здійснюється через скрипт register.php, що приймає JSON із полями username, email, password і avatar. Спочатку перевіряється унікальність email за допомогою SQL-запиту SELECT \* FROM User WHERE Email = ?. Якщо такий запис уже існує — повертається повідомлення про помилку. Інакше новий користувач додається до таблиці User через запит INSERT INTO User (Name, Email, Password, RegistrationDate, Avatar) VALUES (?, ?, ?, NOW(), ?), де пароль попередньо хешується. У разі успішної реєстрації повертається JSON-відповідь з повідомленням про успіх.

Скрипт login.php реалізує авторизацію. Він приймає email і пароль через POST-запит, виконує пошук у базі даних (SELECT \* FROM User WHERE Email = ?) і, у разі успіху, перевіряє пароль функцією password\_verify. Якщо дані коректні, повертається JSON з ім’ям користувача; інакше — повідомлення про помилку.

Щоб додати медитацію до улюблених, використовується favorites.php. Скрипт приймає user\_id та meditation\_id, перевіряє, чи вже існує такий запис (SELECT \* FROM Favorite WHERE UserID = ? AND MeditationID = ?). Якщо ні, додає його в таблицю Favorite запитом INSERT INTO Favorite (UserID, MeditationID, DateAdded) VALUES (?, ?, NOW()).

Скрипт get\_favorites.php отримує всі улюблені медитації певного користувача. Для цього спочатку виконується запит SELECT ID FROM User WHERE Email = ?, а потім — об’єднаний запит SELECT m.Title FROM Favorite f JOIN Meditation m ON f.MeditationID = m.ID WHERE f.UserID = ? ORDER BY f.DateAdded DESC, що повертає масив медитацій у форматі JSON.

Скрипт get\_user.php відповідає за отримання user\_id за email. Він виконує SQL-запит SELECT ID FROM User WHERE Email = ? і повертає JSON із результатом.

Скрипт load\_meditations.php реалізує завантаження всіх медитацій через запит SELECT \* FROM Meditation, результат якого повертається у форматі JSON без потреби у вхідних параметрах.

Для оптимізації роботи бази даних використовуються індекси. У таблиці User задано первинний ключ ID і унікальний індекс на Email, що пришвидшує пошук і забезпечує унікальність адрес. У таблиці Favorite наявні зовнішні ключі UserID і MeditationID, а також комбінований індекс UserID + MeditationID, який оптимізує перевірку наявності запису. У таблиці Meditation використовується індекс на поле CategoryID, що прискорює фільтрацію медитацій за категоріями.

**5.6. Тестування та налаштування веб додатку на сервері**

Фронтенд додатку складається з двох основних сторінок: home.php (головна сторінка з медитаціями, реєстрацією та авторизацією) і profile.html (профіль користувача з улюбленими медитаціями).

|  |  |  |
| --- | --- | --- |
| **Назва тесту** | **Опис** | **Очікуваний результат** |
| Відображення медитацій | Перевірити, чи медитації відображаються на головній сторінці (home.php). | Список медитацій завантажується з load\_meditations.php і відображається у вигляді карток. |
| Реєстрація нового користувача | Заповнити форму реєстрації та натиснути "Зареєструватися". | Після введення коректних даних (ім’я, email, пароль) з’являється повідомлення "Успішна реєстрація", і користувач може увійти. |
| Реєстрація з існуючим email | Спробувати зареєструвати користувача з уже існуючим email. | З’являється повідомлення "Email вже зареєстрований". |
| Авторизація з коректними даними | Увійти з правильним email і паролем. | З’являється повідомлення "Успішний вхід", посилання "Увійти" змінюється на "Профіль". |
| Авторизація з некоректними даними | Увійти з неправильним email або паролем. | З’являється повідомлення "Невірний email або пароль". |
| Додавання медитації до улюблених | Натиснути кнопку "Додати до улюблених" на картці медитації (авторизований користувач). | З’являється повідомлення "Медитація додана до улюблених!". |
| Перегляд улюблених медитацій | Перейти на сторінку профілю (profile.html) і перевірити список улюблених медитацій. | Список улюблених медитацій відображається коректно (збігається з доданими). |
| Вихід із профілю | На сторінці профілю натиснути кнопку "Вийти". | Користувач перенаправляється на home.php, посилання "Профіль" змінюється на "Увійти". |

Бекенд складається з PHP-скриптів, які обробляють запити від клієнта та взаємодіють із базою даних.

|  |  |  |
| --- | --- | --- |
| **Назва тесту** | **Опис** | **Очікуваний результат** |
| Реєстрація нового користувача | Надіслати POST-запит до register.php із коректними даними. | Повертається {"success": true, "message": "Успішна реєстрація"}, запис додається до таблиці User. |
| Реєстрація з існуючим email | Надіслати POST-запит до register.php із email, який уже є в базі. | Повертається {"success": false, "message": "Email вже зареєстрований"}. |
| Авторизація з коректними даними | Надіслати POST-запит до login.php із правильним email і паролем. | Повертається {"success": "Успішний вхід", "username": "Ім'я"}. |
| Авторизація з некоректними даними | Надіслати POST-запит до login.php із неправильним email або паролем. | Повертається {"error": "Невірний email або пароль"}. |
| Додавання медитації до улюблених | Надіслати POST-запит до favorites.php із user\_id і meditation\_id. | Повертається "added", запис додається до таблиці Favorite. |
| Повторне додавання до улюблених | Повторно надіслати той самий запит до favorites.php. | Повертається "Медитація вже додана до улюблених.". |
| Отримання улюблених медитацій | Надіслати POST-запит до get\_favorites.php із email користувача. | Повертається JSON-масив із назвами улюблених медитацій (або [], якщо їх немає). |
| Отримання ID користувача | Надіслати POST-запит до get\_user.php із email користувача. | Повертається {"success": true, "user\_id": 1}. |
| Завантаження всіх медитацій | Надіслати GET-запит до load\_meditations.php. | Повертається JSON-масив із усіма медитаціями з таблиці Meditation. |

**РОЗДІЛ 6. КЕРІВНИЦТВО КОРИСТУВАЧА**

**6.1. Навігація в додатку**

Меню додатку для медитації розроблено з урахуванням зручності користувача та доступне як у комп’ютерній, так і в мобільній версії. У комп’ютерній версії меню розташоване у верхній частині екрану у вигляді горизонтальної панелі навігації з додатковими фільтрами для швидкого доступу до категорій медитацій. У мобільній версії меню розміщене в нижній частині екрану у вигляді панелі з іконками для швидкого доступу до основних розділів.

**6.2. Авторизація**

Авторизація дозволяє увійти до облікового запису або створити новий, щоб отримати доступ до персоналізованих функцій, таких як додавання медитацій до улюблених. На головній сторінці у навігації натиснувши **"Увійти"**, у користувача з’явиться можливість, або увійти в уже існуючий профіль, або зареєструватись.

У формі "Створення облікового запису" треба ввести ім’я, **email**, п**ароль**, п**ідтвердження паролю**. Натиснути **"Зареєструватися"**. У разі успішної реєстрації користувача перенаправить до profile.html. У зворотньому ж випадку, виведе помилку (наприклад, "Email вже зареєстрований").

Щоб увійти в свій вже існуючий обліковий запис треба ввести **email** та п**ароль**. Натиснути **"Увійти"**. У разі успішного входу посилання "Увійти" змінюється на "Профіль". У зворотньому ж випадку, виведе помилку (наприклад, "Неправильний email/пароль"). Увійшовши як адміністратор з’являється можливість увійти в адмін-панель або ж вийти.

**6.3. Перегляд медитацій**

### Ця функція дозволяє переглядати, фільтрувати та відкривати медитації. Доступна без авторизації. Натиснувши на **"Медитації"** у меню, користувач побачить різні медитації та отримає можливість обрати фільтри:

* + - **Категорія** ("Релаксація", "Сон", "Усі").
    - **Тривалість** ("Коротка", "Середня", "Довга", "Усі").

Після чого натиснувши **"Знайти"**. Відображаються лише відповідні медитації.

Картки медитацій містять назву, опис, кнопки **"Почати"** і **"Додати до улюблених"**. Натиснувши **"Почати"** або назву, користувача перекине на сторінку медитації (наприклад, meditation-relaxation.php). Там він зможе ознайомитись з назва, описом, інструкцією, тривалістю та відео.

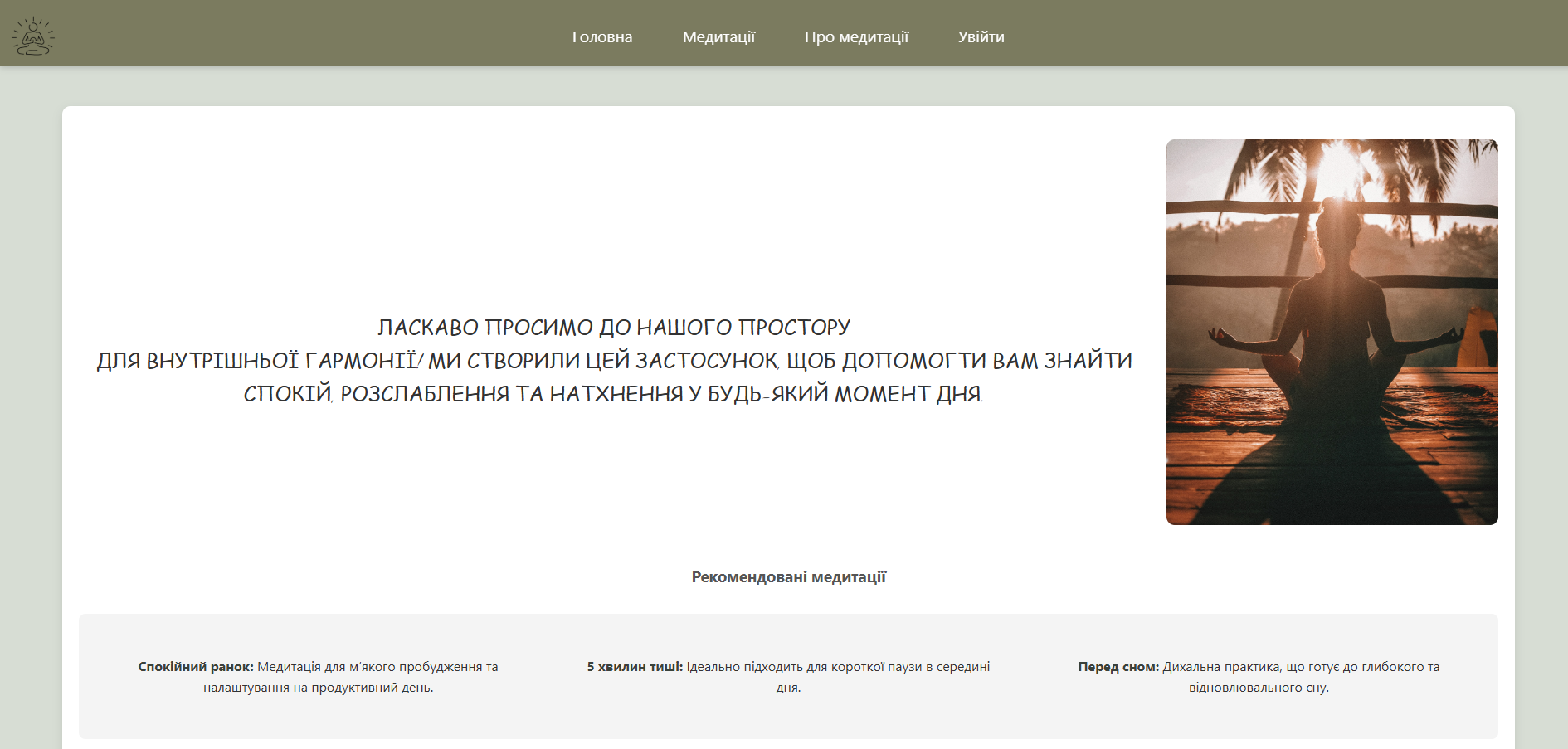
Натиснувши на **"Додати до улюблених"**, до профілю користувача буде додано цю медитацію, але це можливо лише після авторизації.

Зі сторони адміністратора, він може видалити або відредагувати медитації в адмін-панелі.

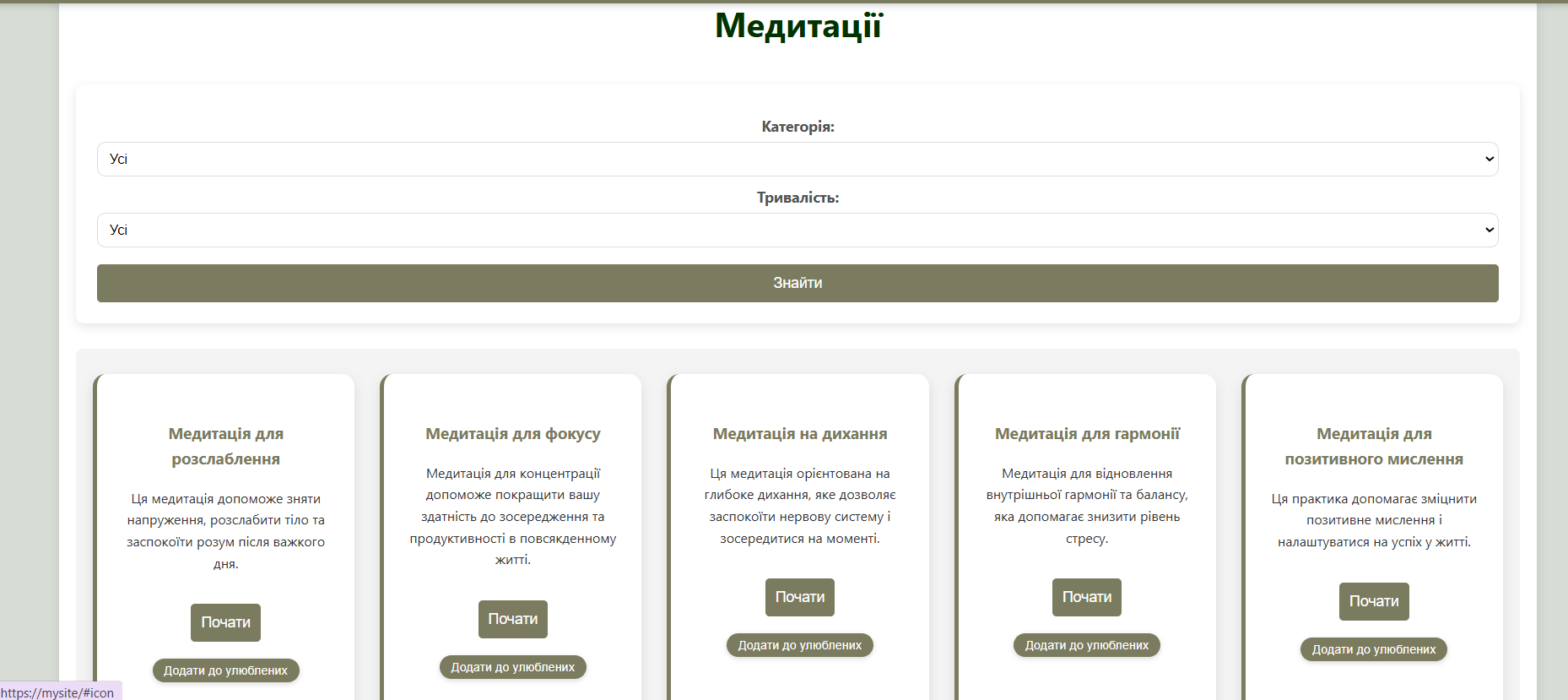
**6.4. Профіль**

Профіль дозволяє переглядати дані та улюблені медитації. Доступний після авторизації. Щоб зайти у профіль слід авторизуватись за посилання «Увійти», після чого воно зміниться на «Профіль». Там буде відображатися привітання (наприклад, "Ласкаво просимо, Олена!") і список улюблених медитацій. Адміністратор може ж видаляти користувачів, а також змінювати їх роль на адміністратора або навпаки користувача.

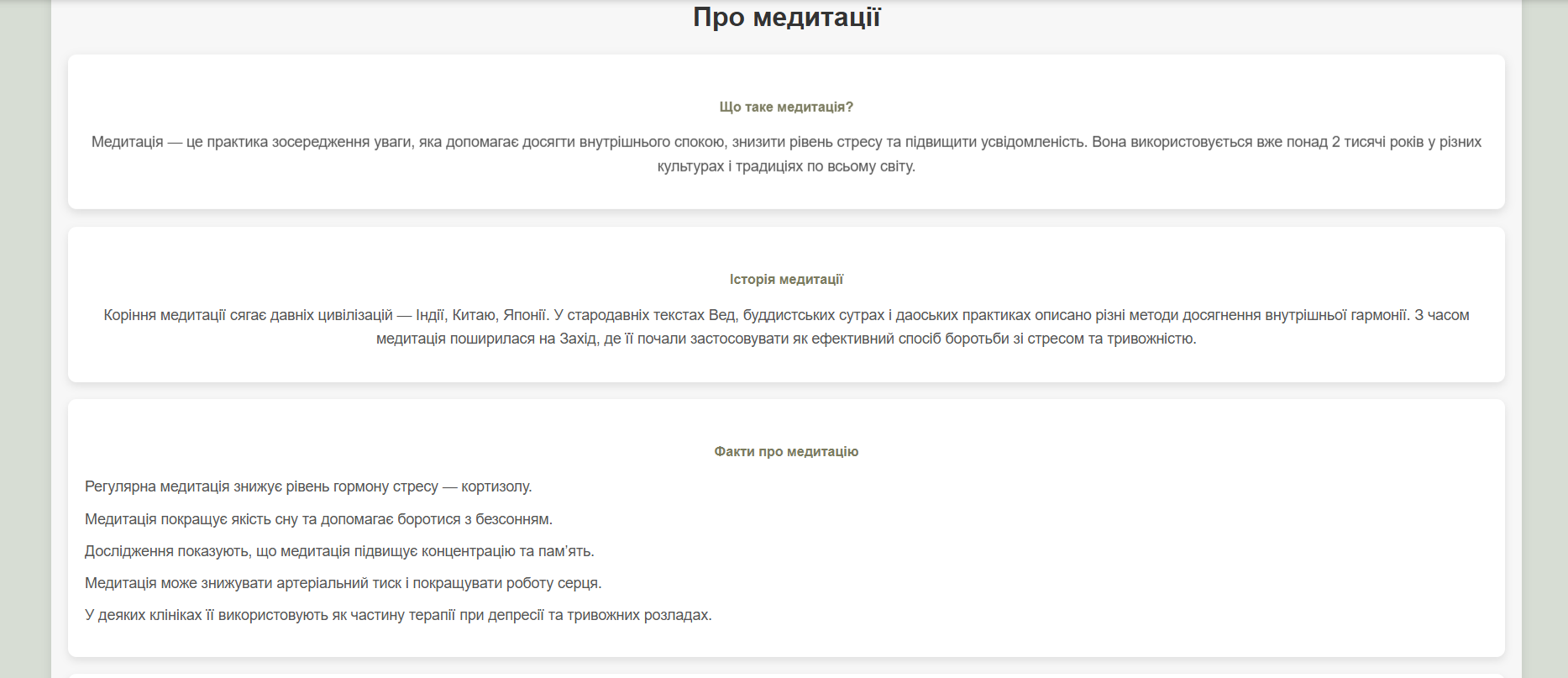
**РОЗДІЛ 7. ВИГЛЯД СТОРІНОК WEB-САЙТУ**

****

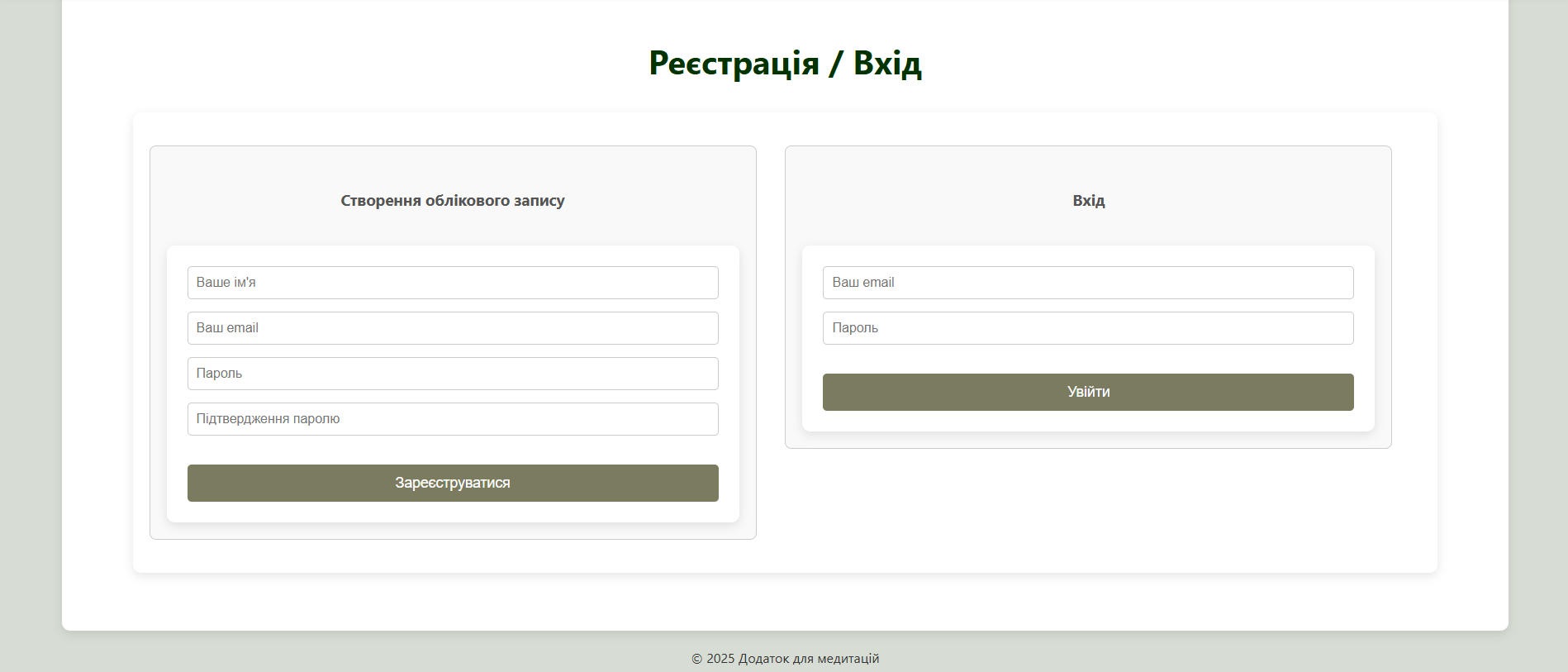
**Рис 7.1 – Меню на головній сторінці сайту**

****

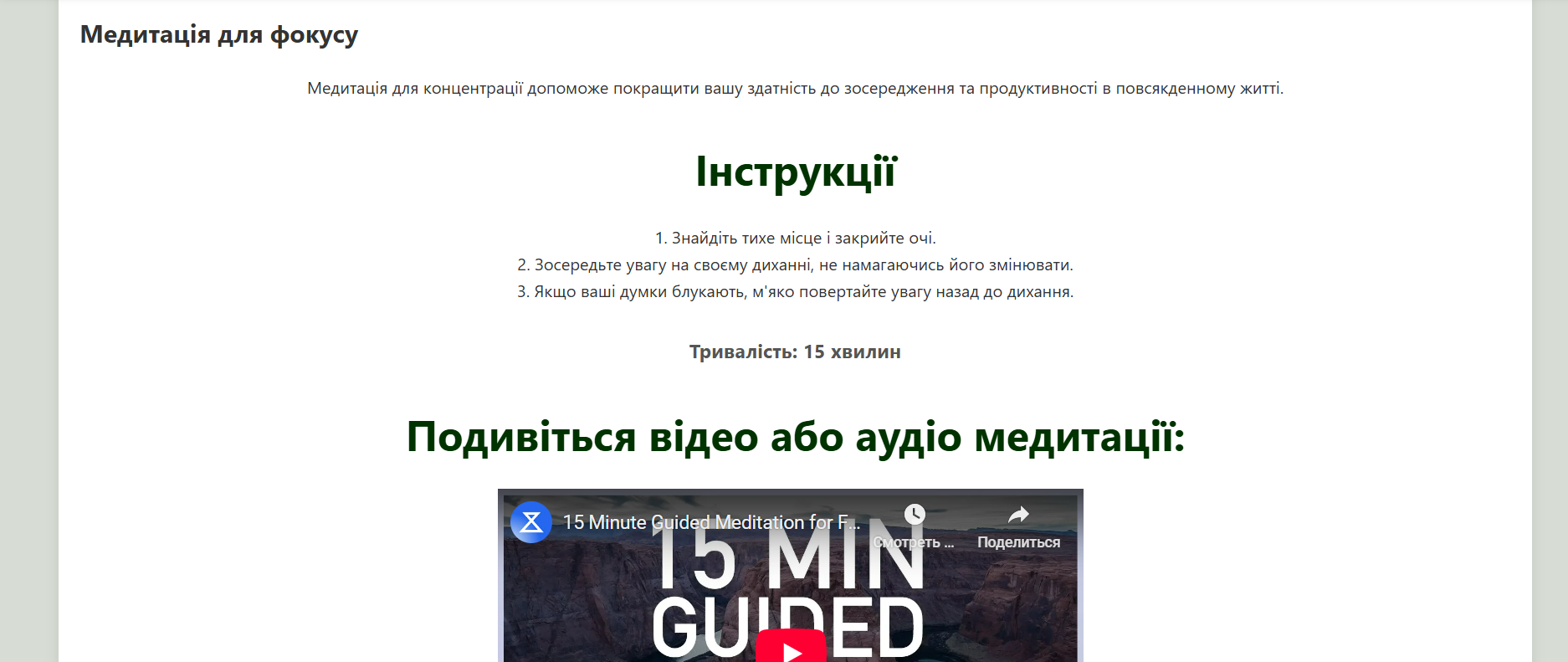
**Рис 7.2 - «Медитації» на головній сторінці сайту**

****

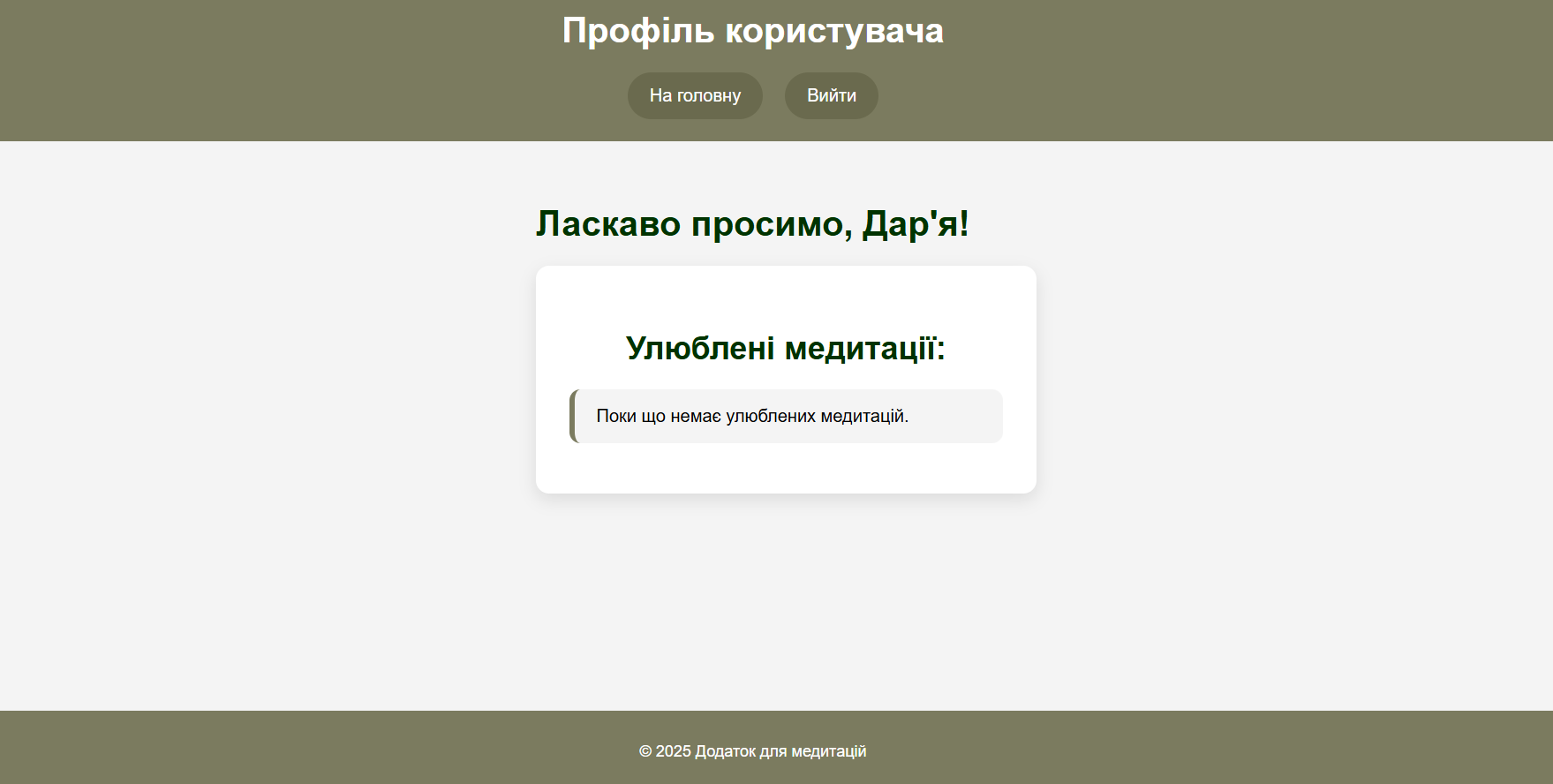
**Рис 7.3 - «Про медитації» на головній сторінці сайту**

****

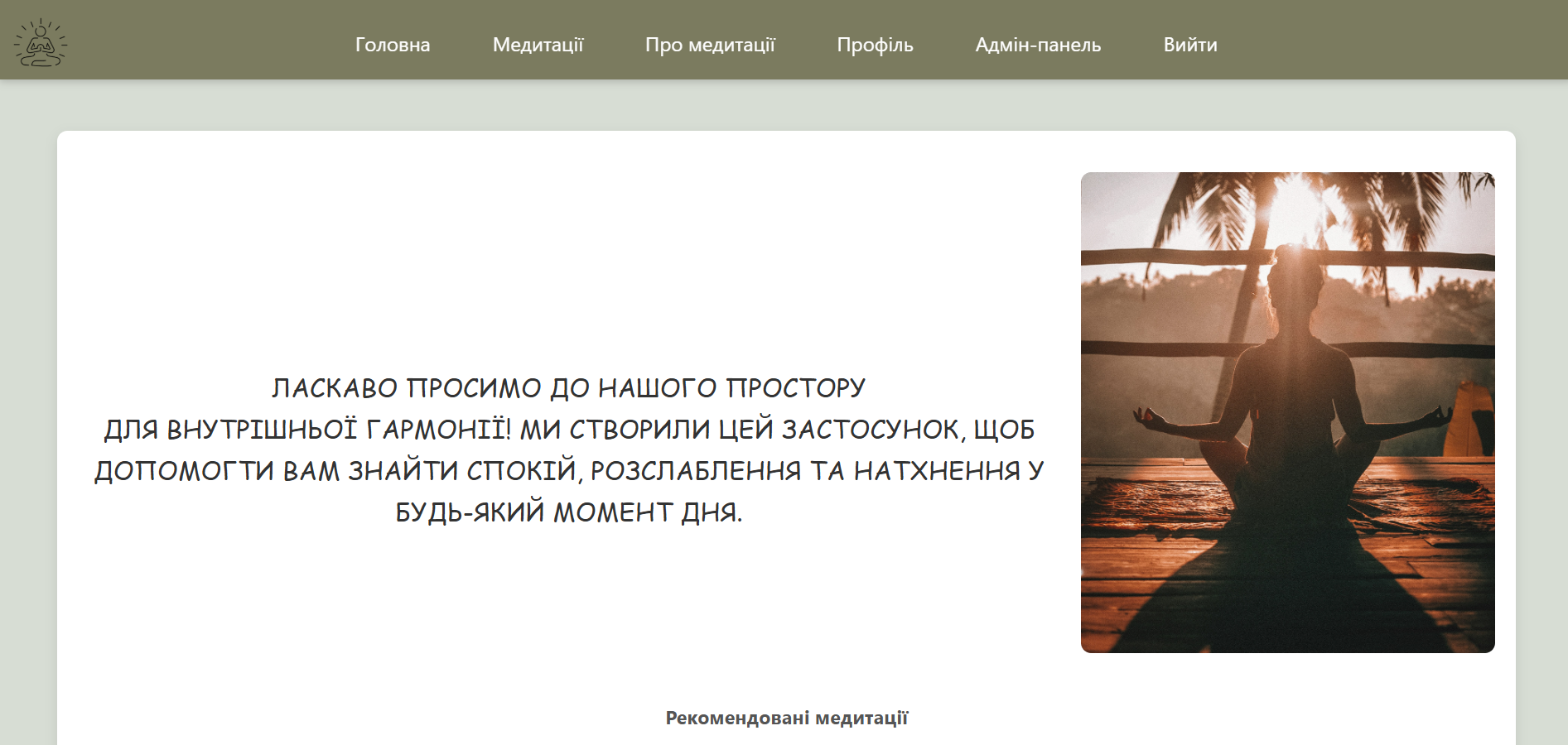
**Рис 7.4 - «Вхід/реєстрація» на головній сторінці сайту**

****

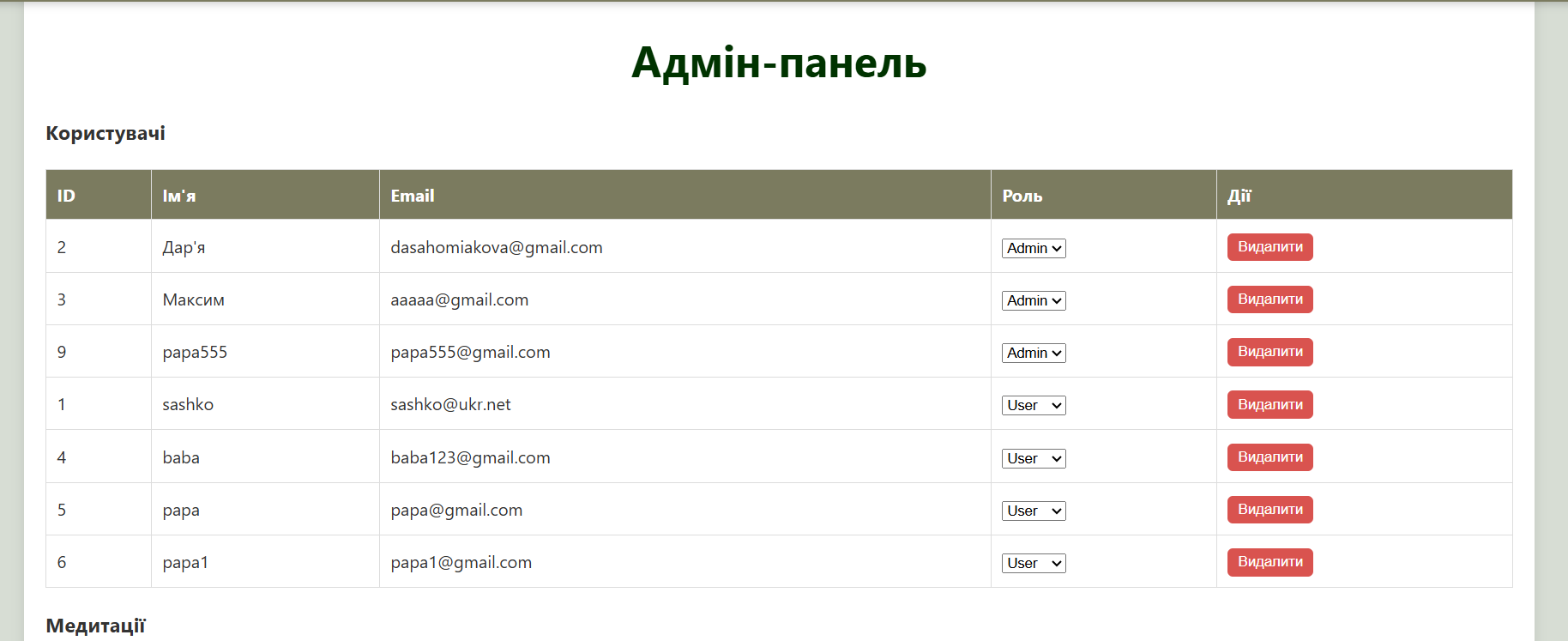
**Рис 7.5 – Окремо кожна медитація**

****

**Рис 7.6 – Профіль користувача**

****

**Рис 7.7 – Меню на головній сторінці сайту зі сторони адміністратора**

****

****

**Рис 7.8 – Адмін-панель**

**ВИСНОВКИ**

Цей проєкт завершився створенням веб-додатку для медитації, який допомагає користувачам вибирати медитаційні практики відповідно до їхніх потреб і вподобань. Додаток має кілька важливих функцій, таких як реєстрація користувачів, вибір медитацій за категоріями (наприклад, для зниження стресу, поліпшення сну, підвищення концентрації), збереження улюблених медитацій. Всі дані зберігаються в базі даних. Додаток був розроблений з акцентом на зручність і простоту використання, з інтуїтивно зрозумілим інтерфейсом. Окрім основних функцій для користувачів, система також передбачає роль адміністратора. Він може керувати обліковими записами користувачів, а також редагувати або видаляти медитації.

Оскільки інтерес до здоров’я і самопідтримки зростає, цей додаток має великий потенціал і може стати корисним інструментом для тих, хто хоче покращити своє психічне і фізичне здоров’я через медитацію.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. CSS, HTML, Tags. Доступно за посиланням: <https://css.in.ua/html/tags>.
2. W3Schools. Колірний вибірник. Доступно за посиланням: <https://w3schoolsua.github.io/colors/colors_picker.html#gsc.tab=0>.
3. WebSkill. HTML та CSS - Тема 3. Доступно за посиланням: <http://www.courses.webskill.com.ua/html&css/theme3/3%2023%2059%2012%203%204%20theme%203%2033%2012.html>.
4. JavaScript. Learn JavaScript. Доступно за посиланням: <https://learn.javascript.ru/>.
5. YouTube. JavaScript Tutorial. Доступно за посиланням: <https://www.youtube.com/watch?v=BE1ajZ5S7hE>.
6. **Freehost. Що таке phpMyAdmin?** Доступно за посиланням: <https://freehost.com.ua/faq/wiki/chto-takoe-phpmyadmin/>.

**ДОДАТКИ**

*Додаток А*

**Файл home.php**

<?php

session\_start();

?>

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Додаток для медитації</title>

<link href="https://fonts.googleapis.com/css2?family=Dancing+Script&display=swap" rel="stylesheet">

<style>

body {

margin: 0;

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

background-color: #d7ddd4;

color: #333;

line-height: 1.6;

display: flex;

flex-direction: column;

align-items: center;

padding-top: 100px;

}

header {

position: fixed;

top: 0;

left: 0;

width: 100%;

z-index: 1000;

background-color: #7b7b5f;

padding: 20px 0;

text-align: center;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.2);

}

nav {

display: flex;

justify-content: center;

gap: 60px;

margin-top: 10px;

font-size: 1.2em;

}

nav a {

color: white;

text-decoration: none;

transition: transform 0.3s;

}

nav a:hover {

transform: scale(1.05);

text-decoration: none;

}

section {

width: 90%;

max-width: 1800px;

padding: 40px 20px;

margin: 30px auto;

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);

transition: box-shadow 0.3s;

}

section:not(:last-child) {

margin-bottom: 10px;

}

h2 {

font-size: 2.5em;

text-align: center;

color: #003300;

margin-bottom: 20px;

}

h3 {

color: #555;

margin-top: 30px;

text-align: center;

}

ul {

list-style: none;

padding-left: 0;

}

.c\_text {

text-align: center;

font-family: 'Dancing+Script', cursive;

font-size: 25px;

text-transform: uppercase;

margin-top: 90px;

}

.home-content {

display: flex;

justify-content: space-between;

align-items: center;

gap: 20px;

margin-bottom: 40px;

}

.home-text {

flex: 1;

}

.home-image img {

width: 400px;

height: auto;

border-radius: 10px;

}

.meditations-container {

text-align: center;

}

.meditation-list {

list-style: none;

padding-left: 0;

margin-top: 40px;

display: block;

background-color: #f4f4f4;

margin: 10px 0;

padding: 20px;

border-radius: 8px;

color: #333;

transition: all 0.3s ease;

}

.meditation-item {

display: block;

background-color: #f4f4f4;

margin: 10px 0;

padding: 20px;

border-radius: 8px;

color: #333;

transition: all 0.3s ease;

}

.meditation-item:hover {

transform: scale(1.05);

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

text-decoration: none;

}

.meditation-item strong {

color: #3a413a;

font-weight: bold;

}

.meditation-list {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));

gap: 30px;

margin-top: 30px;

}

.meditation-list article {

background-color: #ffffff;

padding: 25px;

border-left: 5px solid #7b7b5f;

border-radius: 15px;

box-shadow: 0 6px 15px rgba(0, 0, 0, 0.1);

transition: transform 0.3s, box-shadow 0.3s;

}

.meditation-list article:hover {

transform: translateY(-10px);

box-shadow: 0 8px 20px rgba(0, 0, 0, 0.15);

}

button {

background-color: #7b7b5f;

color: white;

border: none;

padding: 12px 24px;

font-size: 1.1em;

border-radius: 50px;

cursor: pointer;

margin-top: 20px;

transition: background-color 0.3s, transform 0.3s;

}

button:hover {

background-color: #6a6a4e;

transform: scale(1.05);

}

form {

background-color: #ffffff;

padding: 25px;

border-radius: 10px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);

margin-top: 40px;

}

form label {

display: block;

margin-top: 10px;

font-weight: bold;

color: #555;

}

form select {

width: 100%;

padding: 10px;

border-radius: 10px;

border: 1px solid #ddd;

font-size: 1em;

}

#profile img {

max-width: 100%;

border-radius: 10px;

margin-top: 20px;

}

#about {

padding: 40px 20px;

background-color: #f7f7f7;

font-family: 'Arial', sans-serif;

}

.section-title {

font-size: 2em;

color: #333;

text-align: center;

margin-bottom: 10px;

}

.card {

background-color: #ffffff;

border-radius: 10px;

padding: 20px;

margin: 20px 0;

box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);

transition: transform 0.3s ease, box-shadow 0.3s ease;

}

.card:hover {

transform: translateY(-5px);

box-shadow: 0 6px 15px rgba(0, 0, 0, 0.2);

}

.card h3 {

font-size: 1em;

color: #7b7b5f;

margin-bottom: 15px;

}

.card p {

font-size: 1.1em;

line-height: 1.6;

color: #555;

}

.card ul {

list-style-type: none;

padding-left: 0;

margin-top: 15px;

}

.card ul li {

font-size: 1.1em;

color: #555;

margin-bottom: 10px;

}

.card ul li strong {

color: #7b7b5f;

}

.card ul li:last-child {

margin-bottom: 0;

}

header nav ul li a img {

width: 80px;

height: 80px;

margin-top: -25px;

}

header nav ul {

position: absolute;

left: 0;

margin: 0;

padding: 0;

}

.auth-container {

display: flex;

justify-content: space-between;

align-items: flex-start;

gap: 30px;

}

.auth-section {

width: 45%;

padding: 20px;

border: 1px solid #ccc;

border-radius: 8px;

background-color: #f9f9f9;

}

form {

display: flex;

flex-direction: column;

}

label {

font-size: 1.1em;

margin-bottom: 5px;

color: #555;

}

input {

font-size: 1em;

padding: 10px;

margin-bottom: 15px;

border: 1px solid #ccc;

border-radius: 5px;

outline: none;

}

input:focus {

border-color: #7b7b5f;

}

small {

font-size: 0.9em;

color: #555;

margin-top: -5px;

margin-bottom: 15px;

}

button {

font-size: 1.1em;

padding: 12px;

background-color: #7b7b5f;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

button:hover {

background-color: #6a6a4e;

}

p {

text-align: center;

margin-top: 10px;

font-size: 1em;

}

a {

color: #7b7b5f;

text-decoration: none;

}

a:hover {

text-decoration: underline;

}

.error-message {

font-size: 1.1em;

color: red;

text-align: center;

margin-top: 20px;

}

.favorite-btn {

background-color: #7b7b5f;

color: white;

border: none;

padding: 6px 14px;

font-size: 0.9em;

border-radius: 20px;

cursor: pointer;

transition: all 0.3s ease;

display: inline-flex;

align-items: center;

gap: 6px;

box-shadow: 0 2px 6px rgba(0, 0, 0, 0.15);

margin: 20px auto;

}

#loginMessage {

display: none;

color: red;

text-align: center;

margin-top: 10px;

}

</style>

</head>

<body>

<header>

<nav>

<a href="#home">Головна</a>

<a href="#meditations">Медитації</a>

<a href="#about">Про медитації</a>

<a href="#login" id="login-link" style="display: <?= isset($\_SESSION['user\_id']) ? 'none' : 'inline' ?>">Увійти</a>

<a href="profile.php" id="profile-link" style="display: <?= isset($\_SESSION['user\_id']) ? 'inline' : 'none' ?>">Профіль</a>

<a href="admin.php" id="admin-link" style="display: <?= (isset($\_SESSION['user\_id']) && $\_SESSION['user\_role'] == 1) ? 'inline' : 'none' ?>">Адмін-панель</a>

<a href="logout.php" id="logout-link" style="display: <?= isset($\_SESSION['user\_id']) ? 'inline' : 'none' ?>">Вийти</a>

<ul>

<li>

<a href="#icon">

<img src="image-removebg-preview.png" alt="Профіль">

</a>

</li>

</ul>

</nav>

</header>

<section id="home">

<div class="home-content">

<div class="home-text">

<p class="c\_text">Ласкаво просимо до нашого простору<br> для внутрішньої гармонії!

Ми створили цей застосунок, щоб допомогти вам знайти спокій,

розслаблення та натхнення у будь-який момент дня.</p>

</div>

<div class="home-image">

<img src="jared-rice-NTyBbu66\_SI-unsplash.jpg" alt="Медитація" />

</div>

</div>

<div class="meditations-container">

<h3>Рекомендовані медитації</h3>

<ul class="meditation-list">

<li><a href="meditation-relaxation.php?type=7" class="meditation-item">

<strong>Спокійний ранок:</strong> Медитація для мʼякого пробудження та налаштування на продуктивний день.

</a></li>

<li><a href="meditation-breathing.php?type=1" class="meditation-item">

<strong>5 хвилин тиші:</strong> Ідеально підходить для короткої паузи в середині дня.

</a></li>

<li><a href="meditation-sleep.php?type=8" class="meditation-item">

<strong>Перед сном:</strong> Дихальна практика, що готує до глибокого та відновлювального сну.

</a></li>

</ul>

</div>

</section>

<section id="meditations">

<h2>Медитації</h2>

<div class="meditations-container">

<div class="filters">

<form id="meditation-filters">

<label for="category">Категорія:</label>

<select id="category">

<option value="all">Усі</option>

<option value="1">Релаксація</option>

<option value="2">Сон</option>

<option value="3">Концентрація</option>

<option value="4">Енергія</option>

<option value="5">Зняття тривоги</option>

</select>

<label for="duration">Тривалість:</label>

<select id="duration">

<option value="all">Усі</option>

<option value="short">Коротка (до 10 хв)</option>

<option value="medium">Середня (10–20 хв)</option>

<option value="long">Довга (понад 20 хв)</option>

</select>

<button type="submit">Знайти</button>

</form>

</div>

<div class="meditation-list" id="meditationsContainer">

<article class="meditation-item" data-category="1" data-duration="long">

<h3><a href="meditation-relaxation.php?type=7">Медитація для розслаблення</a></h3>

<p>Ця медитація допоможе зняти напруження, розслабити тіло та заспокоїти розум після важкого дня.</p>

<a href="meditation-relaxation.php?type=7"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="7" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

<article class="meditation-item" data-category="3" data-duration="medium">

<h3><a href="meditation-focus.php?type=3">Медитація для фокусу</a></h3>

<p>Медитація для концентрації допоможе покращити вашу здатність до зосередження та продуктивності в повсякденному житті.</p>

<a href="meditation-focus.php?type=3"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="3" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

<article class="meditation-item" data-category="2" data-duration="short">

<h3><a href="meditation-breathing.php?type=1">Медитація на дихання</a></h3>

<p>Ця медитація орієнтована на глибоке дихання, яке дозволяє заспокоїти нервову систему і зосередитися на моменті.</p>

<a href="meditation-breathing.php?type=1"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="1" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

<article class="meditation-item" data-category="5" data-duration="medium">

<h3><a href="meditation-harmony.php?type=4">Медитація для гармонії</a></h3>

<p>Медитація для відновлення внутрішньої гармонії та балансу, яка допомагає знизити рівень стресу.</p>

<a href="meditation-harmony.php?type=4"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="4" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

<article class="meditation-item" data-category="4" data-duration="short">

<h3><a href="meditation-positive-thinking.php?type=6">Медитація для позитивного мислення</a></h3>

<p>Ця практика допомагає зміцнити позитивне мислення і налаштуватися на успіх у житті.</p>

<a href="meditation-positive-thinking.php?type=6"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="6" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

<article class="meditation-item" data-category="2" data-duration="short">

<h3><a href="meditation-sleep.php?type=8">Медитація для сну</a></h3>

<p>Медитація, яка допоможе вам розслабитися і заснути, знижуючи рівень тривожності перед сном.</p>

<a href="meditation-sleep.php?type=8"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="8" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

<article class="meditation-item" data-category="4" data-duration="medium">

<h3><a href="meditation-emotional-balance.php?type=2">Медитація для емоційної рівноваги</a></h3>

<p>Ця медитація допоможе відновити емоційну рівновагу та впоратися з важкими ситуаціями в житті.</p>

<a href="meditation-emotional-balance.php?type=2"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="2" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

<article class="meditation-item" data-category="5" data-duration="long">

<h3><a href="meditation-inner-peace.php?type=5">Медитація для внутрішнього спокою</a></h3>

<p>Практика для досягнення внутрішнього спокою та звільнення від негативних думок і переживань.</p>

<a href="meditation-inner-peace.php?type=5"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="5" onclick="addToFavorites(event)">Додати до улюблених</button>

</article>

</div>

</div>

</section>

<section id="about">

<h2 class="section-title">Про медитації</h2>

<div class="card">

<h3>Що таке медитація?</h3>

<p>Медитація — це практика зосередження уваги, яка допомагає досягти внутрішнього спокою, знизити рівень стресу та підвищити усвідомленість. Вона використовується вже понад 2 тисячі років у різних культурах і традиціях по всьому світу.</p>

</div>

<div class="card">

<h3>Історія медитації</h3>

<p>Коріння медитації сягає давніх цивілізацій — Індії, Китаю, Японії. У стародавніх текстах Вед, буддистських сутрах і даоських практиках описано різні методи досягнення внутрішньої гармонії. З часом медитація поширилася на Захід, де її почали застосовувати як ефективний спосіб боротьби зі стресом та тривожністю.</p>

</div>

<div class="card">

<h3>Факти про медитацію</h3>

<ul>

<li>Регулярна медитація знижує рівень гормону стресу — кортизолу.</li>

<li>Медитація покращує якість сну та допомагає боротися з безсонням.</li>

<li>Дослідження показують, що медитація підвищує концентрацію та памʼять.</li>

<li>Медитація може знижувати артеріальний тиск і покращувати роботу серця.</li>

<li>У деяких клініках її використовують як частину терапії при депресії та тривожних розладах.</li>

</ul>

</div>

<div class="card">

<h3>Користь медитації</h3>

<p>Медитація допомагає зміцнити психічне і фізичне здоровʼя, знизити рівень стресу, підвищити якість життя та досягти внутрішньої гармонії. Завдяки регулярним практикам можна покращити емоційну стійкість і загальне самопочуття.</p>

</div>

</section>

<section id="auth">

<h2>Реєстрація / Вхід</h2>

<section class="auth-container">

<div id="register" class="auth-section">

<h3>Створення облікового запису</h3>

<form id="registerForm" method="POST" action="register.php">

<input type="text" id="username" name="username" required placeholder="Ваше ім'я">

<input type="email" id="email" name="email" required placeholder="Ваш email">

<input type="password" id="password" name="password" required placeholder="Пароль">

<input type="password" id="confirm-password" name="confirm-password" required placeholder="Підтвердження паролю">

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

</form>

</div>

<div id="login" class="auth-section">

<h3>Вхід</h3>

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

<input type="email" id="login-email" name="email" required placeholder="Ваш email">

<input type="password" id="login-password" name="password" required placeholder="Пароль">

<button type="submit">Увійти</button>

</form>

</div>

<div id="errorMessage" class="error-message"></div>

<div id="loginMessage" class="error-message">Будь ласка, увійдіть, щоб додати до улюблених.</div>

</section>

</section>

<section id="profile-section" style="display:none;">

<h2>Мій профіль</h2>

<p id="welcome-msg"></p>

<h3>Улюблені медитації:</h3>

<ul id="favorites-list"></ul>

</section>

<footer>

<p>© 2025 Додаток для медитацій</p>

</footer>

<script>

document.addEventListener("DOMContentLoaded", function () {

// Перевірка авторизації при завантаженні сторінки

checkAuthStatus();

// Функція для перевірки статусу авторизації

function checkAuthStatus() {

fetch('session\_check.php')

.then(res => res.json())

.then(data => {

const isLoggedIn = data.logged\_in;

const isAdmin = data.role\_id === 1;

document.getElementById('login-link').style.display = isLoggedIn ? 'none' : 'inline';

document.getElementById('profile-link').style.display = isLoggedIn ? 'inline' : 'none';

document.getElementById('admin-link').style.display = isLoggedIn && isAdmin ? 'inline' : 'none';

document.getElementById('logout-link').style.display = isLoggedIn ? 'inline' : 'none';

})

.catch(error => console.error('Помилка при перевірці авторизації:', error));

}

// Завантаження медитацій

fetch('load\_meditations.php')

.then(response => {

if (!response.ok) throw new Error(`HTTP error! status: ${response.status}`);

return response.json();

})

.then(data => {

const meditationsContainer = document.getElementById('meditationsContainer');

meditationsContainer.innerHTML = '';

const meditationFiles = {

1: 'meditation-breathing.php',

2: 'meditation-emotional-balance.php',

3: 'meditation-focus.php',

4: 'meditation-harmony.php',

5: 'meditation-inner-peace.php',

6: 'meditation-positive-thinking.php',

7: 'meditation-relaxation.php',

8: 'meditation-sleep.php'

};

data.forEach(meditation => {

const file = meditationFiles[meditation.id] || '#';

const article = document.createElement('article');

article.className = 'meditation-item';

article.setAttribute('data-category', meditation.category\_id);

article.setAttribute('data-duration', getDurationCategory(meditation.duration));

article.innerHTML = `

<h3><a href="${file}?type=${meditation.id}">${meditation.title}</a></h3>

<p>${meditation.description}</p>

<a href="${file}?type=${meditation.id}"><button>Почати</button></a>

<button class="favorite-btn" data-meditation-id="${meditation.id}" onclick="addToFavorites(event)">Додати до улюблених</button>

`;

meditationsContainer.appendChild(article);

});

})

.catch(error => {

console.error('Помилка завантаження медитацій:', error);

});

function getDurationCategory(duration) {

if (duration <= 10) return 'short';

if (duration <= 20) return 'medium';

return 'long';

}

document.getElementById('meditation-filters').addEventListener('submit', (e) => {

e.preventDefault();

const selectedCategory = document.getElementById('category').value;

const selectedDuration = document.getElementById('duration').value;

const meditations = document.querySelectorAll('.meditation-item');

meditations.forEach(article => {

const articleCategory = article.getAttribute('data-category');

const articleDuration = article.getAttribute('data-duration');

const matchesCategory = selectedCategory === 'all' || selectedCategory === articleCategory;

const matchesDuration = selectedDuration === 'all' || selectedDuration === articleDuration;

article.style.display = matchesCategory && matchesDuration ? 'block' : 'none';

});

});

document.getElementById('registerForm').addEventListener('submit', function(e) {

e.preventDefault();

const formData = new FormData(this);

fetch('register.php', {

method: 'POST',

body: formData

})

.then(response => {

if (response.redirected) {

window.location.href = response.url; // Перенаправлення після успішної реєстрації

} else {

return response.json();

}

})

.then(data => {

if (data) {

document.getElementById('errorMessage').textContent = data.message;

}

})

.catch(error => {

console.error('Помилка при реєстрації:', error);

document.getElementById('errorMessage').textContent = 'Помилка сервера: ' + error.message;

});

});

document.getElementById('loginForm').addEventListener('submit', function(e) {

e.preventDefault();

const formData = new FormData(this);

fetch('login.php', {

method: 'POST',

body: formData

})

.then(response => {

if (response.redirected) {

window.location.href = response.url;

} else {

return response.json();

}

})

.then(data => {

if (data) {

if (data.error) {

document.getElementById('errorMessage').textContent = data.error;

}

}

})

.catch(error => {

console.error('Помилка при вході:', error);

document.getElementById('errorMessage').textContent = 'Помилка сервера: ' + error.message;

});

});

window.addToFavorites = function(event) {

const button = event.target;

const meditationId = button.getAttribute('data-meditation-id');

fetch('session\_check.php')

.then(res => res.json())

.then(data => {

if (!data.logged\_in) {

document.getElementById('loginMessage').style.display = 'block';

setTimeout(() => {

document.getElementById('loginMessage').style.display = 'none';

}, 3000);

return;

}

fetch('favorites.php', {

method: 'POST',

headers: {

'Content-Type': 'application/x-www-form-urlencoded'

},

body: `user\_id=${data.user\_id}&meditation\_id=${meditationId}`

})

.then(response => response.text())

.then(result => {

if (result === 'added') {

alert('Медитація додана до улюблених!');

} else {

alert(result);

}

})

.catch(error => {

console.error('Помилка при додаванні до улюблених:', error);

document.getElementById('errorMessage').textContent = 'Помилка при додаванні до улюблених.';

});

})

.catch(error => console.error('Помилка при перевірці сесії:', error));

};

});

</script>

</body>

</html>

**admin.php**

<?php

session\_start();

include 'bd.php';

// Перевірка ролі користувача

if (!isset($\_SESSION['user\_id']) || $\_SESSION['user\_role'] != 1) {

header("Location: home.php");

exit;

}

// Отримання списку користувачів

$usersStmt = $conn->prepare("SELECT u.ID, u.Name, u.Email, r.ID AS RoleID, r.Name AS RoleName FROM User u JOIN Roles r ON u.RoleID = r.ID");

$usersStmt->execute();

$usersResult = $usersStmt->get\_result();

$users = [];

while ($row = $usersResult->fetch\_assoc()) {

$users[] = $row;

}

// Отримання списку ролей для випадаючого списку

$rolesStmt = $conn->prepare("SELECT ID, Name FROM Roles");

$rolesStmt->execute();

$rolesResult = $rolesStmt->get\_result();

$roles = [];

while ($row = $rolesResult->fetch\_assoc()) {

$roles[] = $row;

}

// Отримання списку медитацій

$meditationsStmt = $conn->prepare("SELECT m.ID, m.Title, m.Description, m.Duration, m.DifficultyLevel, m.AudioVideoLink, c.ID AS CategoryID, c.Name AS CategoryName FROM Meditation m LEFT JOIN Category c ON m.CategoryID = c.ID");

$meditationsStmt->execute();

$meditationsResult = $meditationsStmt->get\_result();

$meditations = [];

while ($row = $meditationsResult->fetch\_assoc()) {

$meditations[] = $row;

}

// Завантаження категорій для випадаючого списку

$categories = [];

$result = $conn->query("SELECT ID, Name FROM Category");

while ($row = $result->fetch\_assoc()) {

$categories[] = $row;

}

// Обробка редагування медитації

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

$meditation\_id = $\_POST['meditation\_id'] ?? '';

$title = $\_POST['title'] ?? '';

$description = $\_POST['description'] ?? '';

$duration = $\_POST['duration'] ?? '';

$difficulty = $\_POST['difficulty'] ?? '';

$category\_id = $\_POST['category\_id'] ?? '';

$audio\_video\_link = $\_POST['audio\_video\_link'] ?? '';

if ($meditation\_id && $title && $description && $duration && $difficulty && $category\_id && $audio\_video\_link) {

$stmt = $conn->prepare("UPDATE Meditation SET Title = ?, Description = ?, Duration = ?, DifficultyLevel = ?, CategoryID = ?, AudioVideoLink = ? WHERE ID = ?");

$stmt->bind\_param("ssisssi", $title, $description, $duration, $difficulty, $category\_id, $audio\_video\_link, $meditation\_id);

if ($stmt->execute()) {

header("Location: admin.php?message=Медитація успішно оновлена");

exit;

} else {

$error = "Помилка при оновленні медитації: " . $conn->error;

}

$stmt->close();

} else {

$error = "Усі поля обов'язкові для заповнення";

}

}

// Обробка видалення медитації

if (isset($\_GET['delete\_meditation'])) {

$meditation\_id = (int)$\_GET['delete\_meditation'];

$stmt = $conn->prepare("DELETE FROM Meditation WHERE ID = ?");

$stmt->bind\_param("i", $meditation\_id);

if ($stmt->execute()) {

header("Location: admin.php?message=Медитація успішно видалена");

exit;

} else {

$error = "Помилка при видаленні медитації: " . $conn->error;

}

$stmt->close();

}

// Обробка зміни ролі користувача

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

$user\_id = $\_POST['user\_id'] ?? '';

$role\_id = $\_POST['role\_id'] ?? '';

if ($user\_id && $role\_id) {

$stmt = $conn->prepare("UPDATE User SET RoleID = ? WHERE ID = ?");

$stmt->bind\_param("ii", $role\_id, $user\_id);

if ($stmt->execute()) {

header("Location: admin.php?message=Роль користувача успішно оновлена");

exit;

} else {

$error = "Помилка при оновленні ролі: " . $conn->error;

}

$stmt->close();

} else {

$error = "Помилка: Не вибрано роль";

}

}

// Обробка видалення користувача

if (isset($\_GET['delete\_user'])) {

$user\_id = (int)$\_GET['delete\_user'];

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

$error = "Ви не можете видалити власний акаунт.";

} else {

$stmt = $conn->prepare("DELETE FROM Favorite WHERE UserID = ?");

$stmt->bind\_param("i", $user\_id);

$stmt->execute();

$stmt->close();

$stmt = $conn->prepare("DELETE FROM User WHERE ID = ?");

$stmt->bind\_param("i", $user\_id);

if ($stmt->execute()) {

header("Location: admin.php?message=Користувач успішно видалений");

exit;

} else {

$error = "Помилка при видаленні користувача: " . $conn->error;

}

$stmt->close();

}

}

$conn->close();

?>

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Адмін-панель</title>

<link href="https://fonts.googleapis.com/css2?family=Dancing+Script&display=swap" rel="stylesheet">

<style>

body {

margin: 0;

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

background-color: #d7ddd4;

color: #333;

line-height: 1.6;

display: flex;

flex-direction: column;

align-items: center;

padding-top: 100px;

}

header {

position: fixed;

top: 0;

left: 0;

width: 100%;

z-index: 1000;

background-color: #7b7b5f;

padding: 20px 0;

text-align: center;

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.2);

}

nav {

display: flex;

justify-content: center;

gap: 60px;

margin-top: 10px;

font-size: 1.2em;

}

nav a {

color: white;

text-decoration: none;

transition: transform 0.3s;

}

nav a:hover {

transform: scale(1.05);

text-decoration: none;

}

section {

width: 90%;

max-width: 1800px;

padding: 40px 20px;

margin: 30px auto;

background-color: white;

border-radius: 10px;

box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);

transition: box-shadow 0.3s;

}

h2 {

font-size: 2.5em;

text-align: center;

color: #003300;

margin-bottom: 20px;

}

table {

width: 100%;

border-collapse: collapse;

margin-top: 20px;

}

table, th, td {

border: 1px solid #ddd;

}

th, td {

padding: 10px;

text-align: left;

}

th {

background-color: #7b7b5f;

color: white;

}

.edit-btn, .delete-btn {

padding: 5px 10px;

border: none;

border-radius: 5px;

cursor: pointer;

margin-right: 5px;

}

.edit-btn {

background-color: #5bc0de;

color: white;

}

.edit-btn:hover {

background-color: #31b0d5;

}

.delete-btn {

background-color: #d9534f;

color: white;

}

.delete-btn:hover {

background-color: #c9302c;

}

.error-message, .success-message {

font-size: 1.1em;

text-align: center;

margin-top: 20px;

}

.error-message {

color: red;

}

.success-message {

color: green;

}

.edit-form {

display: none;

margin-top: 20px;

padding: 15px;

border: 1px solid #ddd;

border-radius: 5px;

background-color: #f9f9f9;

}

</style>

</head>

<body>

<header>

<nav>

<a href="home.php#home">Головна</a>

<a href="home.php#meditations">Медитації</a>

<a href="home.php#about">Про медитації</a>

<a href="home.php#login" id="login-link" style="display: none;">Увійти</a>

<a href="profile.html" id="profile-link" style="display: inline;">Профіль</a>

<a href="admin.php" id="admin-link" style="display: inline;">Адмін-панель</a>

<a href="logout.php" id="logout-link" style="display: inline;">Вийти</a>

</nav>

</header>

<section id="admin-panel">

<h2>Адмін-панель</h2>

<!-- Повідомлення про успіх або помилку -->

<?php if (isset($\_GET['message'])): ?>

<div class="success-message"><?= htmlspecialchars($\_GET['message']) ?></div>

<?php endif; ?>

<?php if (isset($error)): ?>

<div class="error-message"><?= htmlspecialchars($error) ?></div>

<?php endif; ?>

<!-- Список користувачів -->

<h3>Користувачі</h3>

<table>

<thead>

<tr>

<th>ID</th>

<th>Ім'я</th>

<th>Email</th>

<th>Роль</th>

<th>Дії</th>

</tr>

</thead>

<tbody>

<?php foreach ($users as $user): ?>

<tr>

<td><?= htmlspecialchars($user['ID']) ?></td>

<td><?= htmlspecialchars($user['Name']) ?></td>

<td><?= htmlspecialchars($user['Email']) ?></td>

<td>

<form method="POST" style="display: inline;">

<input type="hidden" name="user\_id" value="<?= $user['ID'] ?>">

<select name="role\_id" onchange="this.form.submit()">

<?php foreach ($roles as $role): ?>

<option value="<?= $role['ID'] ?>" <?= $role['ID'] == $user['RoleID'] ? 'selected' : '' ?>>

<?= htmlspecialchars($role['Name']) ?>

</option>

<?php endforeach; ?>

</select>

<input type="hidden" name="update\_role" value="1">

</form>

</td>

<td>

<button class="delete-btn" onclick="if(confirm('Ви впевнені, що хочете видалити цього користувача?')) location.href='admin.php?delete\_user=<?= $user['ID'] ?>'">Видалити</button>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<!-- Список медитацій -->

<h3>Медитації</h3>

<table>

<thead>

<tr>

<th>ID</th>

<th>Назва</th>

<th>Категорія</th>

<th>Дії</th>

</tr>

</thead>

<tbody>

<?php foreach ($meditations as $meditation): ?>

<tr>

<td><?= htmlspecialchars($meditation['ID']) ?></td>

<td><?= htmlspecialchars($meditation['Title']) ?></td>

<td><?= htmlspecialchars($meditation['CategoryName'] ?? 'Невідомо') ?></td>

<td>

<button class="edit-btn" onclick="showEditForm(<?= $meditation['ID'] ?>)">Редагувати</button>

<button class="delete-btn" onclick="if(confirm('Ви впевнені, що хочете видалити цю медитацію?')) location.href='admin.php?delete\_meditation=<?= $meditation['ID'] ?>'">Видалити</button>

</td>

</tr>

<!-- Форма для редагування медитації -->

<tr class="edit-form-row" id="edit-form-<?= $meditation['ID'] ?>" style="display: none;">

<td colspan="4">

<form method="POST">

<input type="hidden" name="meditation\_id" value="<?= $meditation['ID'] ?>">

<label for="title-<?= $meditation['ID'] ?>">Назва:</label>

<input type="text" id="title-<?= $meditation['ID'] ?>" name="title" value="<?= htmlspecialchars($meditation['Title']) ?>" required>

<label for="description-<?= $meditation['ID'] ?>">Опис:</label>

<textarea id="description-<?= $meditation['ID'] ?>" name="description" required><?= htmlspecialchars($meditation['Description']) ?></textarea>

<label for="duration-<?= $meditation['ID'] ?>">Тривалість (хвилини):</label>

<input type="number" id="duration-<?= $meditation['ID'] ?>" name="duration" value="<?= htmlspecialchars($meditation['Duration']) ?>" required>

<label for="difficulty-<?= $meditation['ID'] ?>">Рівень складності:</label>

<select id="difficulty-<?= $meditation['ID'] ?>" name="difficulty" required>

<option value="Beginner" <?= $meditation['DifficultyLevel'] == 'Beginner' ? 'selected' : '' ?>>Початковий</option>

<option value="Intermediate" <?= $meditation['DifficultyLevel'] == 'Intermediate' ? 'selected' : '' ?>>Середній</option>

<option value="Advanced" <?= $meditation['DifficultyLevel'] == 'Advanced' ? 'selected' : '' ?>>Просунутий</option>

</select>

<label for="category\_id-<?= $meditation['ID'] ?>">Категорія:</label>

<select id="category\_id-<?= $meditation['ID'] ?>" name="category\_id" required>

<?php foreach ($categories as $category): ?>

<option value="<?= $category['ID'] ?>" <?= $category['ID'] == $meditation['CategoryID'] ? 'selected' : '' ?>>

<?= htmlspecialchars($category['Name']) ?>

</option>

<?php endforeach; ?>

</select>

<label for="audio\_video\_link-<?= $meditation['ID'] ?>">Посилання на відео/аудіо:</label>

<input type="url" id="audio\_video\_link-<?= $meditation['ID'] ?>" name="audio\_video\_link" value="<?= htmlspecialchars($meditation['AudioVideoLink']) ?>" required>

<button type="submit" name="edit\_meditation">Зберегти зміни</button>

</form>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

</section>

<footer>

<p>© 2025 Додаток для медитацій</p>

</footer>

<script>

document.addEventListener("DOMContentLoaded", function () {

fetch('session\_check.php')

.then(res => res.json())

.then(data => {

if (!data.logged\_in || data.role\_id != 1) {

window.location.href = 'home.php';

}

});

window.showEditForm = function(meditationId) {

// Приховуємо всі форми редагування

document.querySelectorAll('.edit-form-row').forEach(form => {

form.style.display = 'none';

});

// Показуємо потрібну форму

const form = document.getElementById(`edit-form-${meditationId}`);

if (form) {

form.style.display = 'table-row';

}

};

});

</script>

</body>

</html>