

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «**РОЗРОБКА ВЕБСАЙТУ ДЛЯ СПІЛЬНОТИ
МУЗИКАНТІВ ТА БІТМЕЙКЕРІВ**»

Виконав: студент 4 курсу, групи 6.1210-2пi
спеціальності 121 інженерія програмного забезпечення
(шифр і назва спеціальності)

освітньої програми програмна інженерія
(назва освітньої програми)

В.В. Храпко

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,
к.ф.-м.н. Мильцев О.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри фундаментальної та прикладної
математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри програмної
інженерії, к.ф.-м.н., доцент

Лісняк А.О.

(підпис)

“ _____ ” _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Храпку Валерію Віталійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка вебсайту для спільноти музикантів та бітмейкерів

керівник роботи Мильцев Олександр Михайлович, к.ф.-м.н.

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Розробка вебсайту для спільноти музикантів та бітмейкерів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.12.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	09.01.2024	
2.	Збір вихідних даних.	30.01.2024	
3.	Обробка методичних та теоретичних джерел.	23.02.2024	
4.	Розробка першого та другого розділу.	12.04.2024	
5.	Розробка третього розділу.	20.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	21.06.2024	

Студент _____
(підпис)

В.В. Храпко
(ініціали та прізвище)

Керівник роботи _____
(підпис)

О.М. Мильцев
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

А.В. Столярова
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка вебсайту для спільноти музикантів та бітмейкерів»: 51 с., 50 рис., 13 джерел.

ВЕБКОМУНІКАЦІЯ, ВЕБСАЙТ, РОЗРОБКА, СПІЛЬНОТА МУЗИКАНТІВ, СТВОРЕННЯ ТА ПЕРЕГЛЯД КОНТЕНТУ.

Об'єкт дослідження – вебсайт для спільноти музикантів та бітмейкерів.

Мета роботи – створення вебсайту для спільноти музикантів та бітмейкерів, що надає можливості для безкоштовного отримання авторських звуків для створення музики новачкам.

Метод дослідження – аналіз існуючих вебтехнологій та вебсайтів з схожим функціоналом, розробка технічного завдання, реалізація проєкту та тестування.

Розробка вебсайту для музикантів та бітмейкерів є актуальною завданням у зв'язку зі зростаючою потребою в онлайн-платформах, що підтримують творчу діяльність та співпрацю. Сайт повинен забезпечувати користувачам зручний доступ до ресурсів, можливість завантаження та обміну музичними файлами, участь в обговореннях та створення спільних проєктів.

Цей вебсайт призначений для об'єднання музикантів, бітмейкерів та інших творчих особистостей, які зацікавлені у створенні та поширенні музичних творів. Об'єкт дослідження охоплює всі аспекти розробки, від проєктування структури та дизайну до реалізації функціональних можливостей і забезпечення безпеки даних користувачів.

SUMMARY

Bachelor's qualifying paper "Development of a Website for the Community of Musicians and Beatmakers": 51 pages, 50 figures, 13 references.

WEBCOMMUNICATION, WEBSITE, DEVELOPMENT, MUSICIAN COMMUNITY, CONTENT CREATION AND VIEWING.

The object of the study is a website for the community of musicians and beatmakers.

The aim of the study is to create a website for the community of musicians and beatmakers, which provides opportunities for free access to author's sounds for creating music for beginners.

The method of research is analysis of existing web technologies and websites with similar functionality, development of a technical task, project implementation and testing.

Website development for musicians and beatmakers is an urgent task due to the growing need for online platforms that support creativity and collaboration. The site should provide users with convenient access to resources, the ability to download and share music files, participate in discussions and create joint projects.

This website is intended to bring together musicians, beatmakers and other creative individuals who are interested in creating and sharing musical works. The object of research covers all aspects of development, from the design of the structure and design to the implementation of functionality and ensuring the security of user data.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	8
1 Методи розробки вебсайтів.....	9
1.1 Основні підходи	9
1.1.1 Статичні вебсайти	9
1.1.2 Системи управління контентом (CMS)	10
1.1.3 Односторінкові додатки (SPA).....	10
1.1.4 Розподілені вебдодатки (PWA)	11
1.1.5 ДевОпс та CI/CD підходи.....	11
1.2 Вибір технологій	12
1.2.1 Фронтенд-технології.....	12
1.2.2 JavaScript фреймворки та бібліотеки	13
1.2.3 Інші технології	13
1.2.4 Бекенд-технології.....	13
1.2.5 Фреймворки та платформи	13
1.2.6 Бази даних.....	14
1.2.7 Інші технології та інструменти.....	14
1.3 Планування структури.....	15
2 Дизайн та функціональність	16
2.1 Інтерфейс користувача	16
2.2 Бекенд.....	16
2.2.1 Мови програмування	16
2.2.2 Фреймворки	17
2.2.3 Бази даних.....	17
2.2.4 Архітектура.....	17

2.2.5 Безпека	17
2.2.6 Тестування та моніторинг	18
2.2.7 Документація	18
2.2.8 Скалабельність	18
2.3 Інтеграція з базами даних.....	19
2.3.1 Приклади технологій і методів інтеграції	19
2.3.2 Переваги інтеграції з базами даних	20
2.4 Етапи створення вебсайту	20
2.4.1 Розробка прототипу	21
2.4.2 Створення макетів сторінок.....	21
2.5 Схема бази даних	28
2.6 Проектування інтерфейсу.....	30
3 Реалізація та тестування	32
3.1 Фронтенд	32
3.2 Бекенд.....	36
3.3 Тестування вебсайту	40
Висновки	48
Перелік посилань.....	50

ВСТУП

В останні десятиліття розвиток інтернет-технологій значно змінив спосіб створення музики, та саунд дизайну, для музикантів та бітмейкерів важливо мати платформу, яка дозволяє легко знайти потрібні їм компоненти та файли для створення свого унікального звуку, ділитися своїми роботами та отримувати зворотній зв'язок. Вебсайт для спільноти музикантів та бітмейкерів може стати таким місцем, забезпечуючи інтерактивний простір для обміну досвідом та спільної творчості.

Метою даної роботи є розробка вебсайту, який дозволить музикантам та бітмейкерам створювати профілі, ділитися музичними композиціями, знаходити для себе самий підходящий матеріал, спілкуватися та об'єднуватися у групи для спільної роботи. Для досягнення цієї мети необхідно дослідити сучасні вебтехнології, вибрати оптимальні інструменти для розробки та протестувати кінцевий продукт на користувачах.

1 МЕТОДИ РОЗРОБКИ ВЕБСАЙТІВ

1.1 Основні підходи

Розробка вебсайтів включає різні підходи, кожен з яких має свої переваги та недоліки. Розглянемо основні підходи до розробки вебсайтів.

1.1.1 Статичні вебсайти

Статичні вебсайти складаються з фіксованого контенту, який не змінюється в залежності від користувача чи часу. Вони створюються за допомогою HTML, CSS та JavaScript [1].

Переваги:

- простота: легше розробляти та розгортати;
- швидкість: відсутність серверних запитів робить їх швидкими;
- безпека: менше точок входу для атак, оскільки немає серверної частини.

Недоліки:

- обмежена функціональність: не підходять для динамічних сайтів, що вимагають інтерактивності;
- складність оновлення: кожне оновлення контенту вимагає втручання розробника;
- динамічні вебсайти.

Динамічні вебсайти використовують серверну частину для генерування контенту на ходу. Вони можуть реагувати на запити користувачів і змінювати контент в залежності від їхніх дій [6].

Переваги:

- гнучкість: можливість персоналізації контенту для кожного користувача;

- можливості інтеграції: легше інтегруються з базами даних та іншими сервісами.

Недоліки:

- складність: більш складні у розробці та підтримці.
- швидкість: можуть бути повільнішими через необхідність обробки серверних запитів;
- безпека: потребують захисту від різних типів атак, таких як sql-ін'єкції.

1.1.2 Системи управління контентом (CMS)

CMS, такі як WordPress, Joomla, Drupal, дозволяють створювати та керувати контентом без необхідності писати код [2].

Переваги:

- зручність: інтуїтивні інтерфейси для створення та оновлення контенту;
- модульність: багато плагінів та тем для розширення функціональності.

Недоліки:

- виконання: можуть бути повільнішими через завантаженість зайвими функціями;
- безпека: потребують регулярних оновлень для захисту від відомих вразливостей.

1.1.3 Односторінкові додатки (SPA)

SPA використовують JavaScript-фреймворки, такі як Angular, React або Vue.js, для створення вебдодатків, які завантажуються один раз і динамічно оновлюють контент без перезавантаження сторінки [3].

Переваги:

- швидкість: користувацький досвід подібний до десктопних додатків;
- інтерактивність: висока інтерактивність і швидка реакція на дії користувача.

Недоліки:

- складність SEO: можливі проблеми з індексацією контенту пошуковими системами;
- початковий час завантаження: може бути довшим через великий обсяг початкового завантаження.

1.1.4 Розподілені вебдодатки (PWA)

PWA використовують сучасні вебтехнології для надання досвіду, подібного до нативних мобільних додатків, з підтримкою офлайн-режиму та push-сповіщень [7].

Переваги:

- офлайн-підтримка: можливість роботи без підключення до інтернету;
- швидкість: швидкий доступ завдяки кешуванню контенту.

Недоліки:

- складність розробки: потребують значних зусиль для налаштування та підтримки;
- сумісність: може не підтримуватися старими браузерами.

1.1.5 ДевОпс та CI/CD підходи

Ці підходи включають використання автоматизації для безперервної інтеграції (ContinuousIntegration, CI) та безперервного розгортання (ContinuousDeployment, CD) вебдодатків.

Переваги:

- автоматизація: швидке розгортання та оновлення додатків;
- якість: підвищена якість коду через автоматизоване тестування.

Недоліки:

- складність налаштування: потребує часу та ресурсів для налаштування автоматизованих процесів;
- залежність від інструментів: може бути залежність від конкретних інструментів та платформ.

При розробці вебсайту для спільноти музикантів та бітмейкерів було обрано динамічний тип сайту.

1.2 Вибір технологій

Для розробки вебсайтів використовуються різні технології, які можна умовно поділити на три основні категорії: фронтенд-технології, бекенд-технології та бази даних [8]. Розглянемо одні з найпопулярніших технологій у кожній категорії.

1.2.1 Фронтенд-технології

Мови розмітки і стилізації:

- HTML (HyperText Markup Language): основна мова розмітки для створення вебсторінок;
- CSS (Cascading Style Sheets): мова стилізації для оформлення HTML-документів.

Мови програмування:

- JavaScript: основна мова програмування для створення динамічного контенту на вебсторінках.

1.2.2 JavaScript фреймворки та бібліотеки

React: бібліотека для створення користувацьких інтерфейсів, розроблена Facebook [10].

Angular: фреймворк для розробки односторінкових додатків, створений Google.

1.2.3 Інші технології

TypeScript: надмножина JavaScript, яка додає статичну типізацію.

Sass/SCSS: препроцесори для CSS, які додають можливість використання змінних, вкладених правил та інших зручностей [4].

1.2.4 Бекенд-технології

Мови програмування:

- Node.js: серверна платформа на базі JavaScript;
- Python: використовується з фреймворками, такими як Django і Flask;
- Ruby: використовується з фреймворком RubyonRails;
- PHP: широко використовується мова для веброзробки, популярна завдяки WordPress;
- Java: використовується з фреймворком Spring;
- C#: використовується з фреймворком .NET.

1.2.5 Фреймворки та платформи

Express.js: мінімалістичний вебфреймворк для Node.js. Laravel: популярний PHP-фреймворк. ASP.NET: вебплатформа для C# від Microsoft.

1.2.6 Бази даних

Реляційні бази даних [11]:

- MySQL: популярна СУБД з відкритим вихідним кодом;
- SQLite: легка вбудована СУБД.

Нереляційні бази даних (NoSQL) [11]:

- MongoDB: документо-орієнтована СУБД;
- CouchDB: документо-орієнтована СУБД з підтримкою JSON;
- Cassandra: розподілена NoSQL база даних для великих даних.

1.2.7 Інші технології та інструменти

Системи контролю версій:

- Git: система контролю версій для відстеження змін у коді;
- GitHub/GitLab/Bitbucket: платформи для хостингу репозиторіїв Git.

Хмарні платформи:

- AWS (AmazonWebServices): широкий спектр хмарних послуг;
- GoogleCloudPlatform: платформа хмарних обчислень від Google;
- MicrosoftAzure: хмарна платформа від Microsoft.

API та інші сервіси:

- GraphQL: мова запитів для API;
- REST: архітектурний стиль для створення вебсервісів;
- Firebase: платформа для створення мобільних і вебдодатків від Google.

Кожна з цих технологій має свої особливості та області застосування, і вибір конкретної технології залежить від вимог проєкту, ресурсів команди та інших факторів.

Для розробки вебсайту для спільноти музикантів та бітмейкерів було обрано наступні технології:

- HTML/CSS для створення структури та стилізації сторінок;
- JavaScript для реалізації інтерактивних елементів;
- PHP для серверної частини та обробки запитів;
- MySQL як базу даних для зберігання інформації про користувачів та їх композиції.

1.3 Планування структури

Планування структури вебсайту є критичним кроком у процесі розробки, оскільки правильно спроектована структура забезпечує зручну навігацію для користувачів і сприяє покращенню SEO. Для вебсайту було виділено наступні основні розділи:

- головна сторінка з новинами та оголошеннями;
- профілі користувачів з можливістю завантаження музичних творів, та редагування профілю;
- сторінка з стрічкою завантажених користувачами композицій;
- сторінка з стрічкою готових компонентів для створення музики.

2 ДИЗАЙН ТА ФУНКЦІОНАЛЬНІСТЬ

2.1 Інтерфейс користувача

Інтерфейс користувача (UI) – це платформа, через яку користувачі взаємодіють з вебсайтом. Створення ефективного і привабливого інтерфейсу користувача є важливим етапом в розробці вебпродукту [5].

Інтерфейс користувача повинен бути інтуїтивно зрозумілим та зручним. Було обрано мінімалістичний дизайн з акцентом на зручність навігації та доступ до основних функцій. Основні елементи інтерфейсу включають:

- дизайн: дизайн з збалансованою кольоровою палітрою, та легкі для сприйняття шрифти;
- контент: контент з легко читабельною контентною структурою;
- користувацький досвід (UX): проста взаємодія з елементами, валідація форм та адаптивність вебсайту;
- доступність: доступність інтерфейсу для користувачів з обмеженими можливостями та адаптивність до взаємодії з технічними пристроями.

2.2 Бекенд

Бекенд – це та частина вебдодатка, яка відповідає за обробку запитів користувачів, взаємодію з базою даних, бізнес-логіку та відповідь на запити, що надходять з фронтенду.

2.2.1 Мови програмування

Популярні мови програмування для бекенду включають JavaScript (з Node.js), Python, Ruby, PHP, Java, C# тощо.

2.2.2 Фреймворки

Використання фреймворків потрібно для прискорення розробки, забезпечення структури та готових інструментів для роботи.

Наприклад, для JavaScript є фреймворки Express.js, Nest.js, для Python – Django, Flask, для Ruby – Ruby on Rails, для PHP – Laravel, Symfony тощо.

2.2.3 Бази даних

Вибір бази даних проєкту включаючи в себе реляційні (наприклад, MySQL, PostgreSQL) та нереляційні (наприклад, MongoDB, Redis) системи управління базами даних.

Врахування потреби щодо швидкодії, масштабованості, консистентності та інших факторів при виборі бази даних.

2.2.4 Архітектура

Розробка чистої та ефективної архітектури бекенду, для забезпечення легкості розширення та підтримки.

Поширені архітектурні шаблони включають RESTful API, Model-View-Controller (MVC), Microservices, Serverless тощо.

2.2.5 Безпека

Забезпечення належного рівню безпеки, використовуючи захист від SQL-ін'єкцій, Cross-SiteScripting (XSS), Cross-SiteRequestForgery (CSRF) та інших загроз безпеці.

Використання методів аутентифікації та авторизації, таких як JWT (JSONWebTokens), OAuth, HTTPS тощо.

2.2.6 Тестування та моніторинг

Проведення тестування, для переконання, що бекенд працює правильно і без помилок.

Використання моніторингу для відстеження продуктивності, надійності та масштабованості бекенду.

2.2.7 Документація

Забезпечення докладної документації API та коду для спрощення розробки, розуміння та підтримки бекенду.

2.2.8 Скалабельність

Розробка бекенду з урахуванням можливості масштабування, щоб забезпечити ефективну роботу при зростанні обсягів даних та трафіку.

Серверна частина для вебсайту реалізована на базі PHP з використанням MySQL як бази даних.

Основні функції включають:

- обробку запитів від користувачів;
- аутентифікацію та авторизацію користувачів;
- зберігання та обробку даних у базі даних.

2.3 Інтеграція з базами даних

Інтеграція з базами даних [12] – це процес об'єднання даних із різних джерел в єдину систему, що дозволяє отримувати, маніпулювати та зберігати інформацію у зручний та ефективний спосіб. Цей процес є критичним для організацій, які працюють з великими обсягами інформації з різних джерел.

Ось кілька ключових аспектів інтеграції з базами даних:

- збирання даних: збір даних з різних джерел, таких як вебсайти, CRM-системи, ERP-системи, файли CSV, Excel тощо;
- трансформація даних: зміна формату даних, очищення даних, нормалізація, агрегація, щоб дані стали узгодженими і придатними для аналізу та використання в спільних базах даних;
- завантаження даних: завантаження оброблених даних у централізовану базу даних або сховище даних (DataWarehouse);
- синхронізація даних: забезпечення того, щоб всі зміни в даних відображалися в реальному часі у всіх інтегрованих системах;
- забезпечення якості даних: перевірка цілісності даних, видалення дублікатів, виправлення помилок, щоб дані були точними і надійними;
- інструменти інтеграції: використання різних інструментів та технологій для автоматизації процесу інтеграції, таких як ETL-процеси (Extract, Transform, Load), API, інтерфейси обміну даними.

2.3.1 Приклади технологій і методів інтеграції

ETL-процеси: інструменти типу Talend, Informatica, Apache Nifi.

API: використання RESTfulAPI або SOAP для обміну даними між системами.

Middleware: проміжні програмні продукти, які забезпечують з'єднання між різними програмними додатками.

Віртуалізація даних: створення віртуальних представлень даних, які інтегрують дані з різних джерел без їх фізичного переміщення.

2.3.2 Переваги інтеграції з базами даних

Централізація даних: об'єднання даних з різних джерел в одному місці полегшує доступ та управління ними.

Покращення якості даних: завдяки інтеграції та очищенню даних підвищується їх точність і надійність.

Ефективність бізнес-процесів: зниження дублювання зусиль і помилок, що сприяє більш ефективному веденню бізнесу.

Аналітика і звітність: полегшення процесу аналізу та створення звітів на основі інтегрованих даних.

Для зберігання даних про користувачів та їх музичні твори була обрана база даних MySQL. Основні переваги цієї бази даних включають:

- гнучка структура даних;
- висока продуктивність;
- легкість масштабування.

2.4 Етапи створення вебсайту

Процес розробки вебсайту було поділено на декілька етапів:

- постановка цілей і завдань сайту;
- створення, опрацювання технічного завдання на розробку сайту;
- розробка прототипу;
- створення макетів сторінок сайту;
- вибір технологій для реалізації;
- верстка сторінок;

- програмна частина;
- наповнення контентом;
- тестування сайту;
- задача проєкту.

2.4.1 Розробка прототипу

Метою розробки прототипу вебсайту є створення інтерактивної моделі, яка дозволить перевірити функціональність та зручність використання сайту [13].

Для створення такої моделі було використано інструмент Adobe Photoshop, прототип сайту зображено на рисунку 2.1.

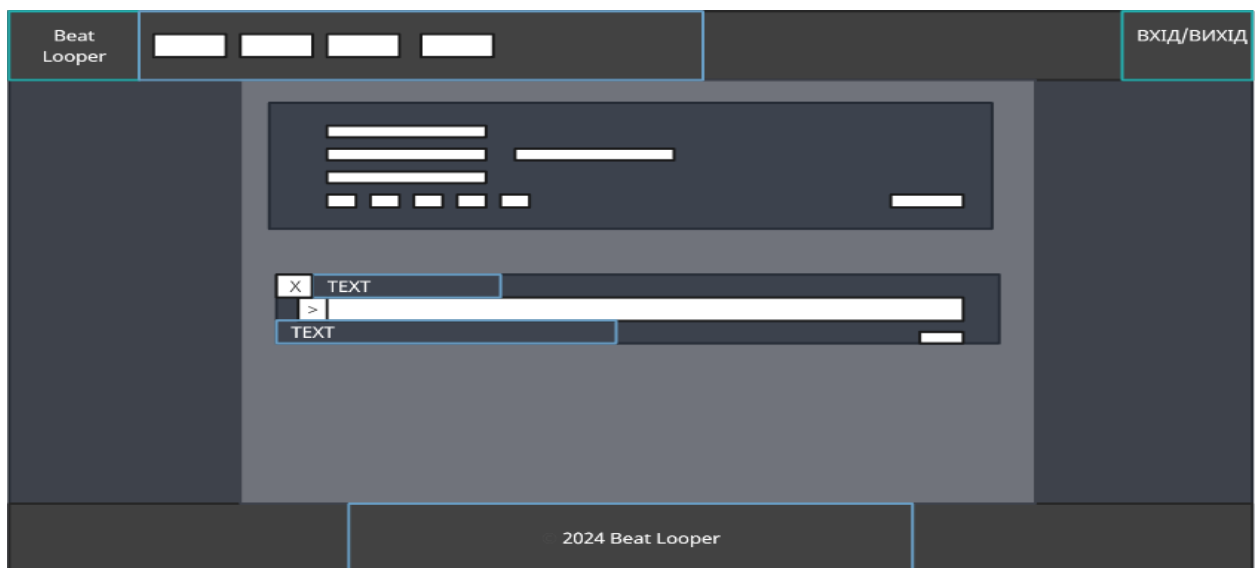


Рисунок 2.1 – Прототип вебсайту

2.4.2 Створення макетів сторінок

При створенні макетів сторінок було дотримано єдиного стилю. Назви пунктів меню були визначені відповідно сторінкам, на які ці пункти посилаються. Вигляд кнопок відповідає діям, які відбудуться після їх

натискання. Усі сторінки сайту виконані за принципами матеріального дизайну.

При проектуванні бази даних застосунку було враховано призначення та технічні вимоги до програмного продукту.

За допомогою програмного застосунку Adobe Photoshop був розроблений дизайн сторінок, всього зроблено 9 сторінок:

- головна сторінка;
- сторінка з завантаженими композиціями;
- сторінка з драм кітами;
- сторінка з програмними компонентами для секвенсорів;
- сторінка редагування профілю користувача;
- сторінка завантаження композицій;
- сторінка користувача;
- сторінка реєстрації;
- сторінка входу в аккаунт користувача.

На головній сторінці вебсайту по центру інформаційного блоку розміщено блоки з новинами, останніми завантаженими звуками, файлами. Також на головній сторінці розміщений блок з списком нових користувачів. Макет головної сторінки сайту представлений на рисунку 2.2.

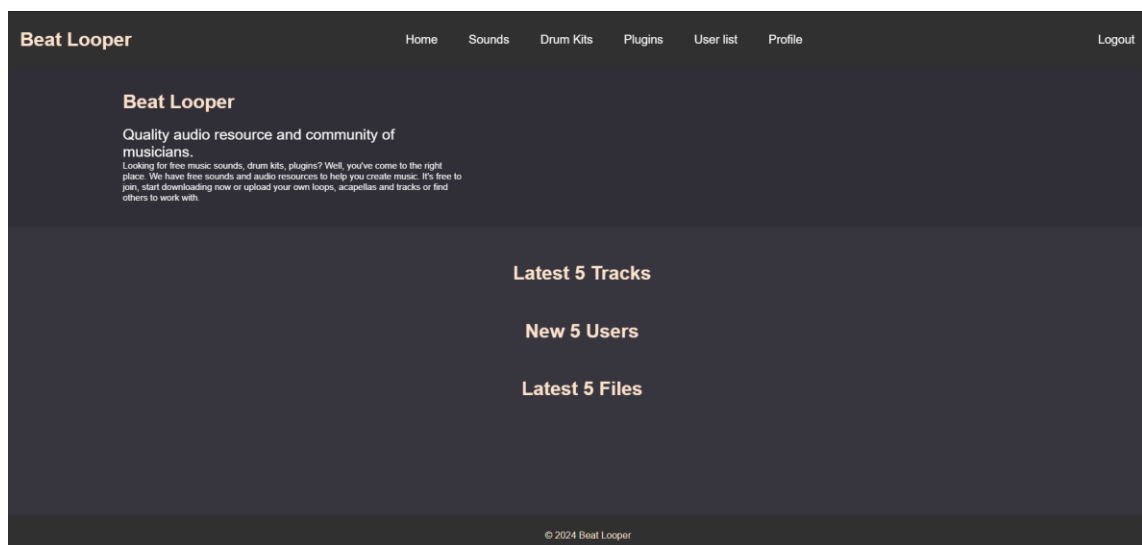


Рисунок 2.2 – Макет головної сторінки вебсайту

На сторінці з завантаженими композиціями розміщений блок фільтрації звуків по категоріям, та список завантажених користувачами композицій, кожна з якої має свій аудіо плеєр з інформацією про композицію, її автора, кнопкою для завантаження цієї композиції. Також в плеєрі буде розміщена хвиля з візуалізацією для кращого сприйняття звуку користувачем. Сторінку з завантаженими композиціями зображено на рисунку 2.3.

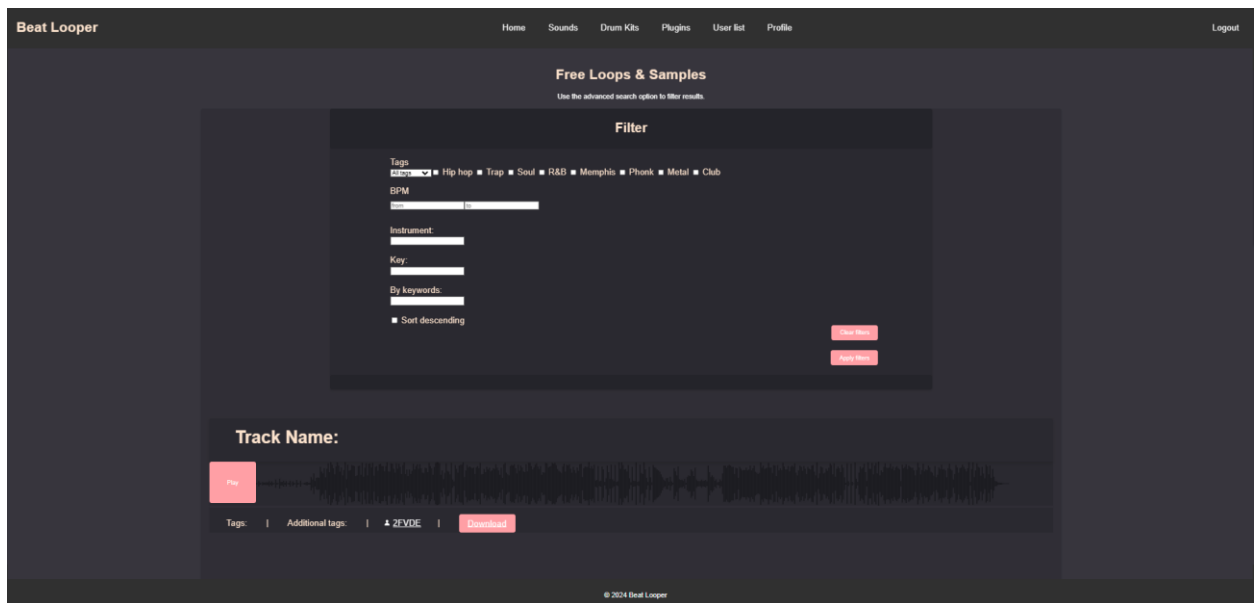


Рисунок 2.3 – Макет сторінки з завантаженими композиціями

Драм кіт (з англ. Drum Kit) – це спеціально підготовлений архів з звуками та ефектами, які можуть бути використані для створення композиції або саунд дизайну.

На сторінці з драм кітами розміщені блоки з інформацією про драм кіт, приклад роботи який створений на основі цього драм кіта, та кнопка для скачування файлу. Завантаження таких файлів на вебсайт доступно тільки адміністратору вебсайту. Сторінку з драм кітами зображено на рисунку 2.4.

На сторінці з програмними компонентами для секвенсорів розміщений список з плагінами які можуть бути інтегровані та додані в секвенсор для розширення спектру ефектів та інструментів композицій. Сторінку з програмними компонентами для секвенсорів зображено на рисунку 2.5.

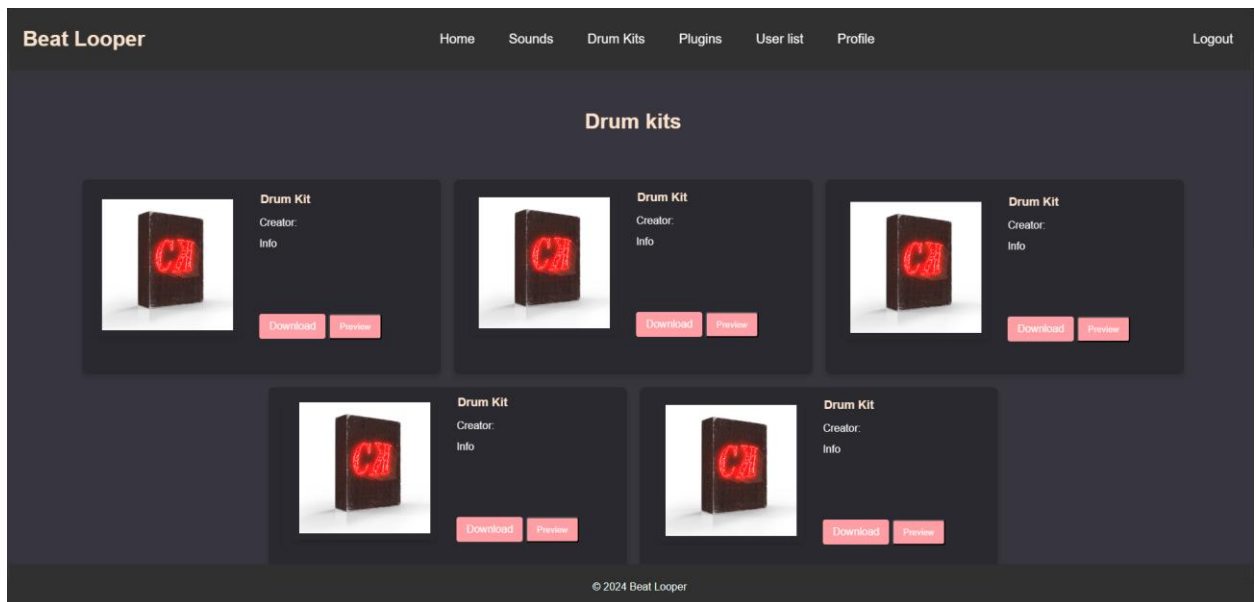


Рисунок 2.4 – Макет сторінки з драм кітами

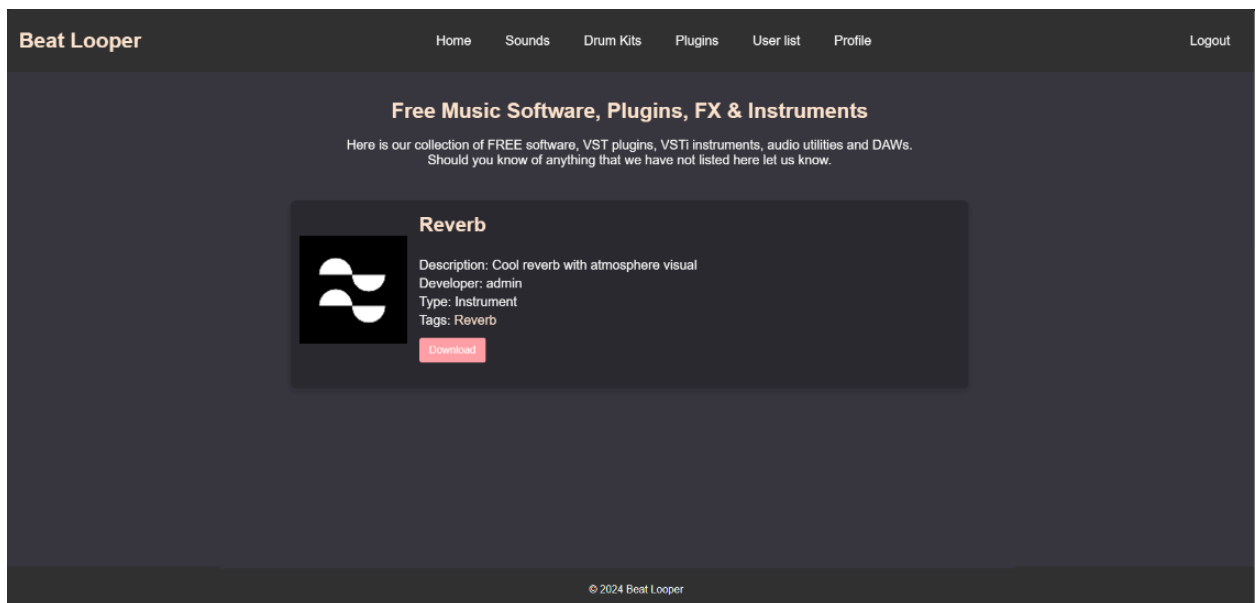


Рисунок 2.5 – Макет сторінки з програмними компонентами

На сторінці з редагуванням профілю користувача розміщено блок з редагуванням інформації такої як:

- ім'я;
- електронна адреса;
- аватар;
- пароль;
- соціальні мережі.

Сторінку з редагуванням профілю користувача зображено на рисунку 2.6.

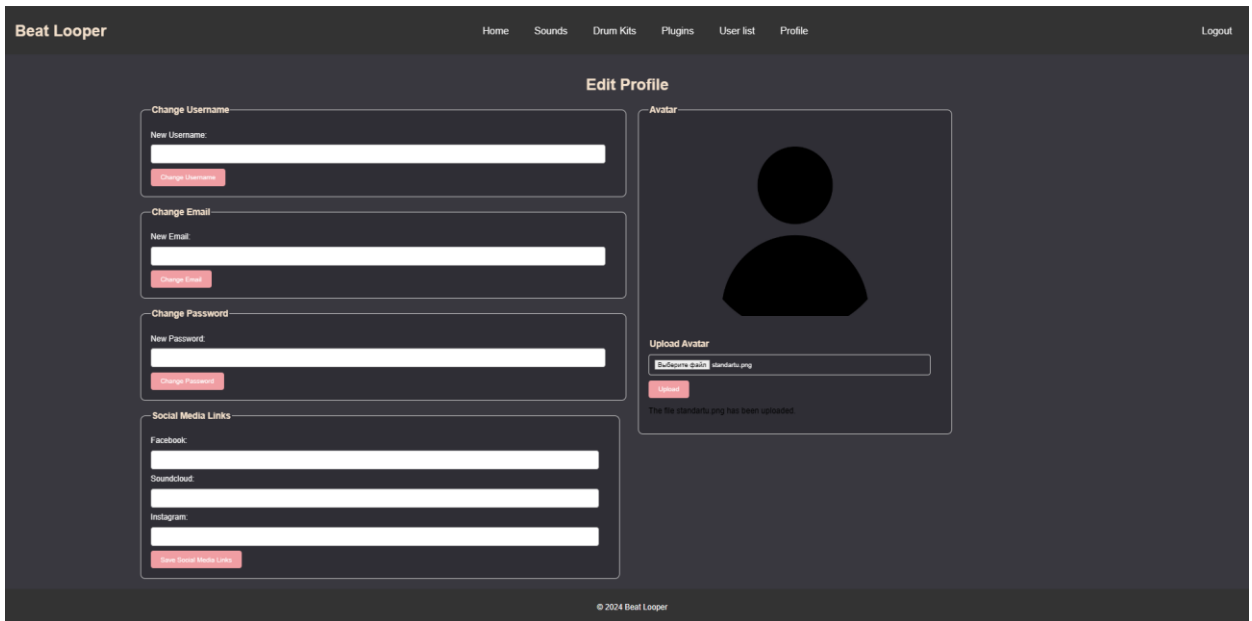


Рисунок 2.6 – Макет сторінки редагування профілю користувача

На сторінці завантаження композицій розміщено блок з полями для заповнення інформації, поле для завантаження аудіо файлів, та кнопка завантаження.

Сторінку завантаження композицій зображено на рисунку 2.7.

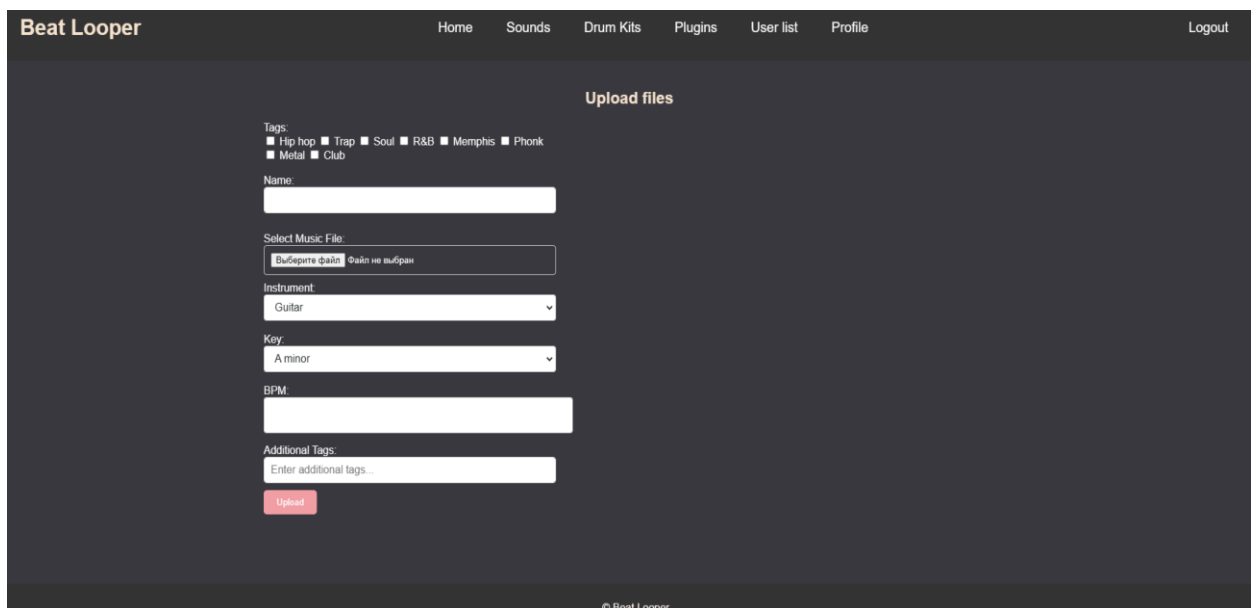


Рисунок 2.7 – Макет сторінки завантаження композицій

На сторінці користувача розміщено блоки в які буде розміщена інформація про користувача за допомогою PHP коду.

Сторінку сторінку користувача зображено на рисунку 2.8.

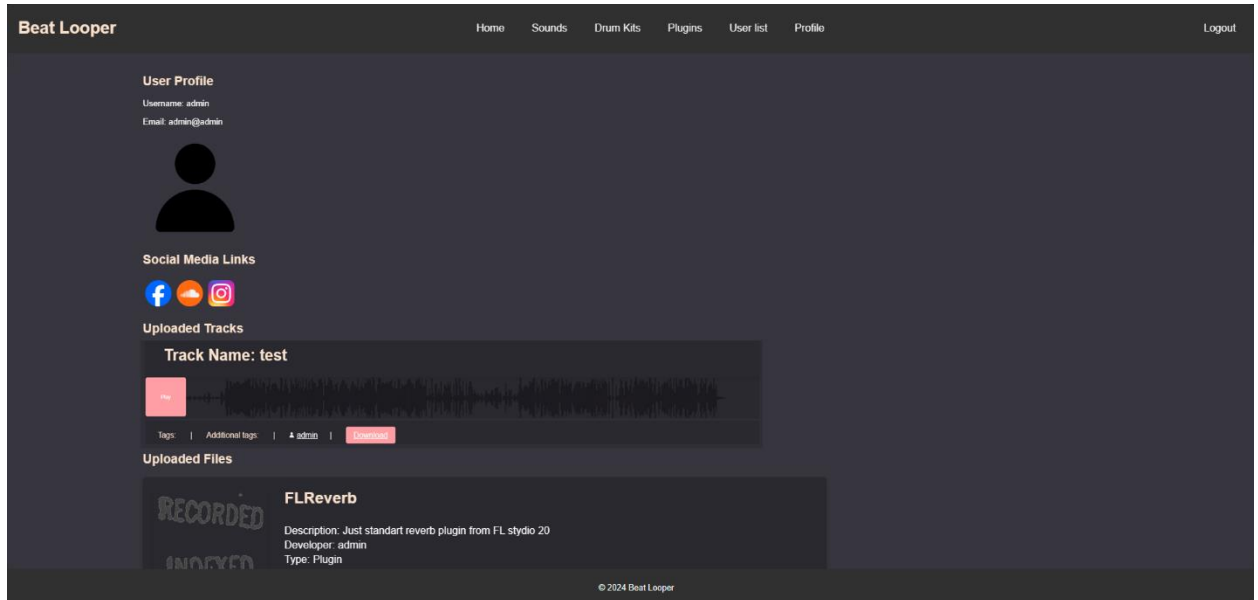


Рисунок 2.8 – Макет сторінки користувача

На сторінці авторизації та реєстрації користувача розміщено блоки для заповнення логіну або електронної пошти та пароллю.

Сторінку користувача зображено: для реєстрації – на рисунку 2.9, для авторизації – на рисунку 2.10. Діаграма авторизації користувача зображена на рисунку 2.11.

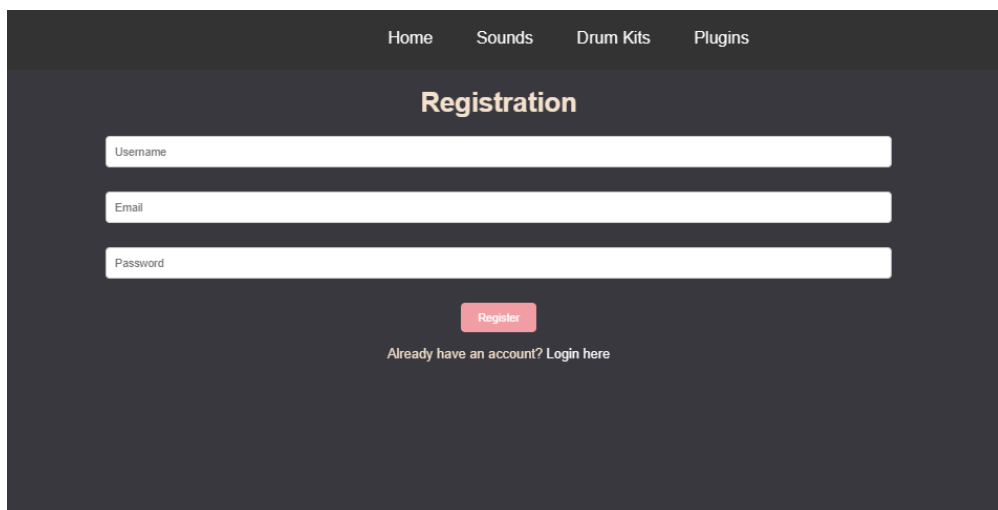


Рисунок 2.9 – Макет сторінки реєстрації

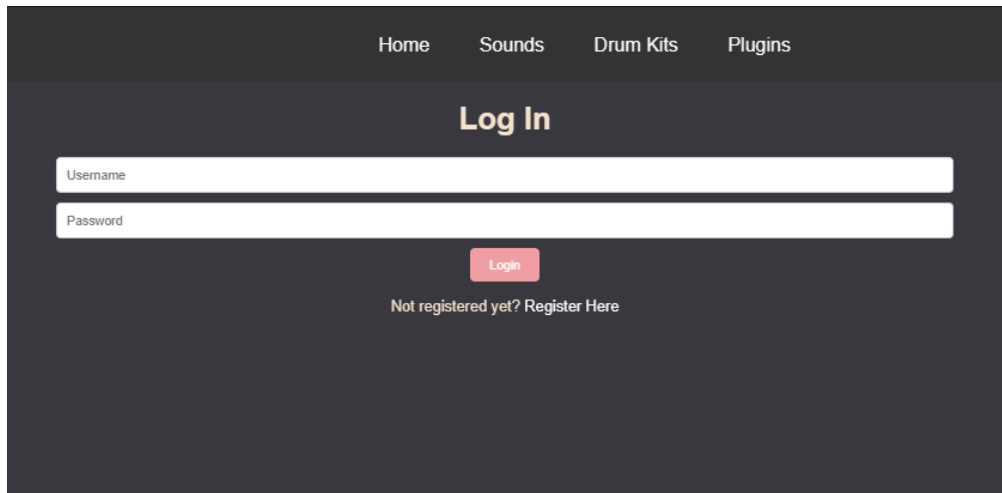


Рисунок 2.10 – Макет сторінки авторизації

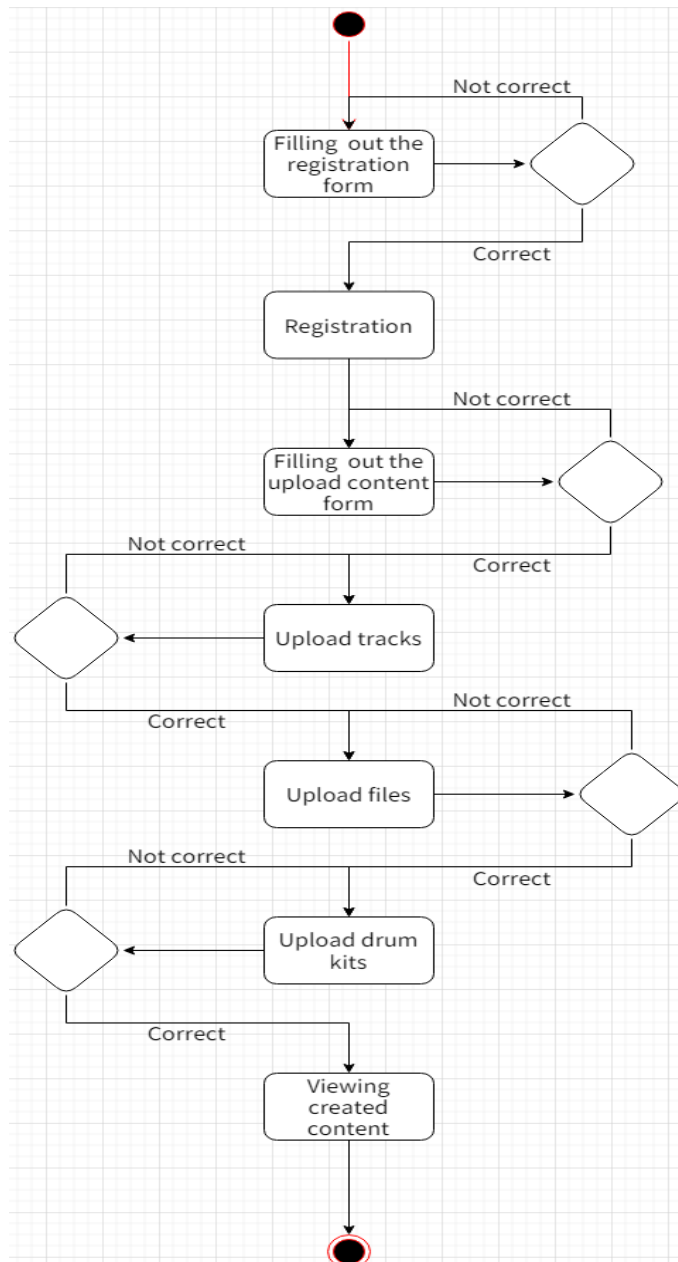


Рисунок 2.11 – Діаграма авторизації користувача

2.5 Схеми бази даних

Проектування бази даних було виконано відповідно до технічного завдання, а також було створено діаграму класів, яку зображено на рисунку 2.12. Модель бази даних складається з трьох сутностей:

- tracks – дані про завантажені аудіо файли;
- users – дані про користувачів;
- files – дані про завантажені користувачами файли.

Кожній сутності моделі відповідає таблиця бази даних.

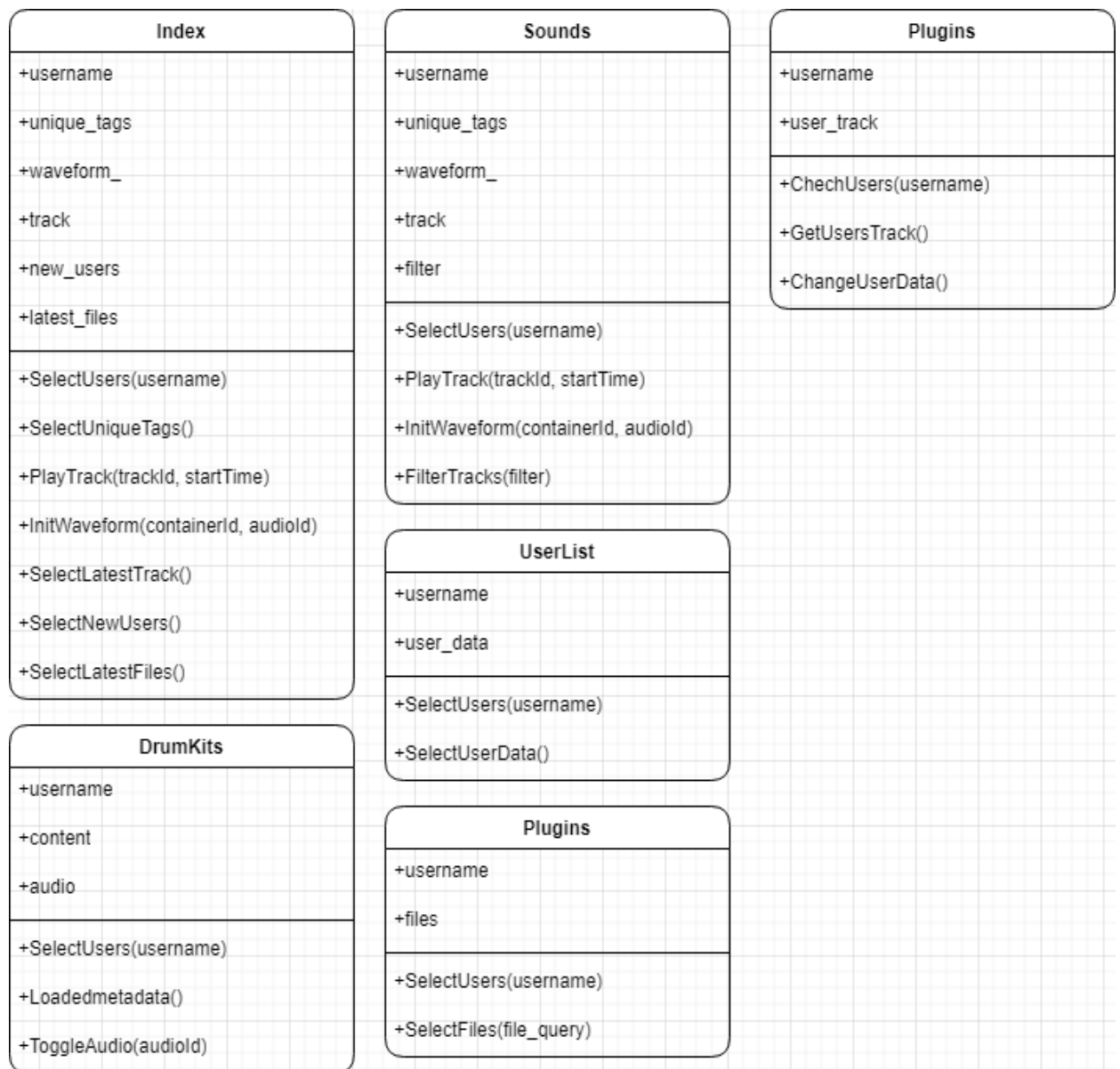


Рисунок 2.12 – Діаграма класів

На рисунку 2.13 зображено модель таблиці tracks.

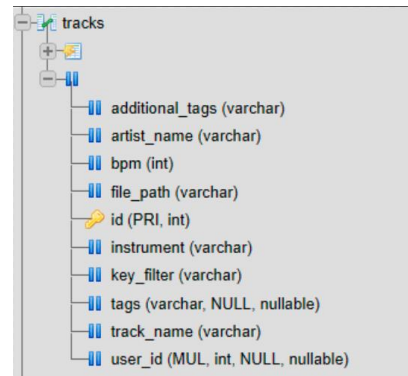


Рисунок 2.13 – Модель таблиці tracks

На рисунку 2.14 зображено модель таблиці users.

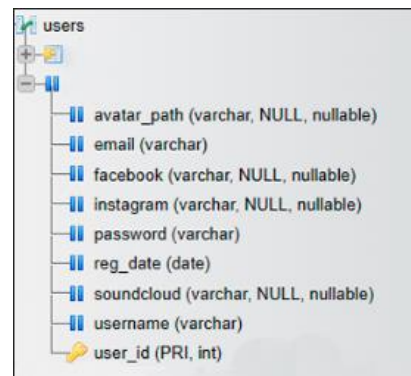


Рисунок 2.14 – Модель таблиці users

На рисунку 2.15 зображено модель таблиці files.

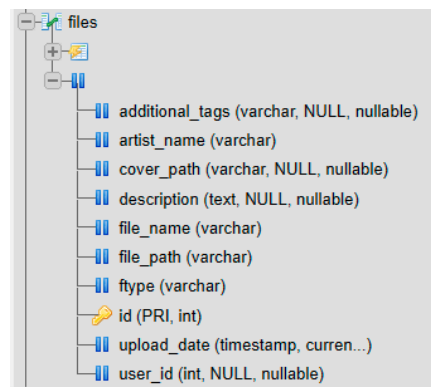


Рисунок 2.15 – Модель таблиці files

2.6 Проектування інтерфейсу

Зручний і привабливий інтерфейс користувача – важливий показник якості сайту. Елементи інтерфейсу мають бути логічно структуровані та взаємопов'язані. Проектування інтерфейсу необхідно ґрунтувати на законах візуального дизайну, іконографіки та друкарні. Застосовувати сучасні закономірності інформаційної архітектури та орієнтуватися на людську психологію.

Для створення палітри кольорів яка буде використовуватись для сайту було використано кольоровий круг. Кольоровий круг зображено на рисунку 2.16.



Рисунок 2.16 – Кольоровий круг

Для підбору кольорової палітри було обрано метод «Аналогова Тріада».

Метод аналогової тріади – це поєднання від 2 до 5 (в ідеалі від 2 до 3) кольорів, що знаходяться поряд один з одним на колірному колі. Прикладом можуть бути поєднання приглушених кольорів: жовто-жовтогарячий, жовтий, жовто-зелений, зелений, синьо-зелений. Приклад зображено на рисунку 2.17. Результат підбору кольорової палітри зображено на рисунку 2.18.



Рисунок 2.17 – Приклад підбору палітри методом «Аналогової тради»

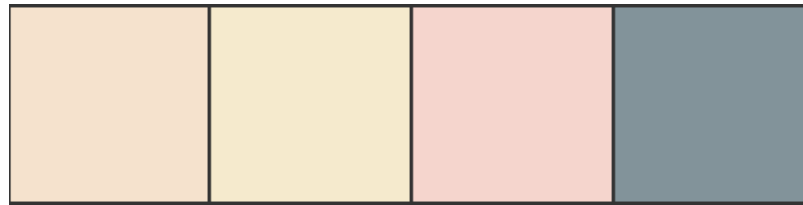


Рисунок 2.18 – Готова кольорова палітра для стилізації вебсайту

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Фрондент

Розробку було розпочато з розмітки сторінок та їх стилізації. Для початку було створено головну сторінку вебсайту, а саме головне меню, тому що навігаційна система на сайті грає найважливішу роль, яка надає змогу зручно переміщатися поміж контентом на сайті.

За допомогою html-тегів була зроблена розмітка, додані посилання і текст у меню. CSS код зображено на рисунку 3.1. Головне меню з застосуванням стилів та коду розмітки зображено на рисунку 3.2. Приклад коду розмітки вебсайту показано на рисунку 3.3.

```
nav {
  display: contents;
  width: 100%;
  justify-content: space-between;
}

.nav-links {
  list-style: none;
  display: flex;
  margin: 0;
  padding: 0;
}

.nav-links li {
  margin-right: 20px;
}

.nav-links a, .auth-links a {
  text-decoration: none;
  color: white;
  padding: 16px;
  transition: background-color 0.3s;
}

.nav-links a:hover, .auth-links a:hover {
  background-color: #F5E2CD;
}
```

Рисунок 3.1 – HTML код навігаційного меню

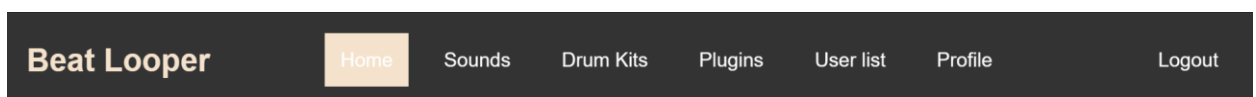


Рисунок 3.2 – Головне навігаційне меню вебсайту


```

<header>
  <h1 style="">Beat Looper</h1>
  <nav>
    <ul class="nav-links">
      <li><a href="index.php">Home</a></li>
      <li><a href="sounds.php">Sounds</a></li>
      <li><a href="drumkits.php">Drum Kits</a></li>
      <li><a href="plugins.php">Plugins</a></li>
      <li><a href="userlist.php">User list</a></li>
      <li><a href="userprofile.php?user_id=<?php echo isset($_SESSION['user_id']) ? $_SESSION['user_id'] : ''; ?>">Profile</a></li>
    </ul>
    <div class="auth-links">
      <?php
      if(isset($_SESSION["username"])) {
        echo "<a href='logout.php'>Logout</a>";
      } else {
        echo "<a href='login.php'>Login</a>";
      }
      ?>
    </div>
  </nav>
</header>

```

Рисунок 3.3 – HTML код навігаційного меню

На сторінці з списком завантажених композицій користувачами було створено аудіо плеєр.

Для розробки плеєра на етапі фронтенду було використано:

- HTML;
- CSS;
- JS.

Код аудіо плеєру зображено на рисунку 3.4. Реалізація плеєру зображена на рисунку 3.5.

```

// Перевірка треків
if (mysqli_num_rows($result) > 0) {
  // Вивід кожного з існуючих треків
  while($row = mysqli_fetch_assoc($result)) {
    echo "<div class='track_block'>";
    echo "<div align='left' class='playerhad'>";
    echo "<h1 style='margin-left: 3%; padding-top: 20px;'>Track Name: " . $row["track_name"] . "</h1>";
    echo "</div>";

    echo "<div class='track'>";
    // Відображення треку
    echo "<div style='right:0 px;' align='left'> <button class='btn' id='playPauseButton_" . $row["id"] . "' onclick='playTrack(" . $row["id"] . "'
    )>Play</button> </div>";
    echo "<div class='track-wrapper'>";
    echo "<div id='waveform_" . $row["id"] . "' class='waveform'></div>";
    echo "<audio id='" . $row["id"] . "'>";
    echo "<source src='" . $row["file_path"] . "' type='audio/mpeg'>";
    echo "Your browser does not support the audio element.";
    echo "</audio>";

    echo "</div>";
    echo "<script>initWaveform('waveform_" . $row["id"] . "', '" . $row["id"] . "')</script>";
    echo "</div>";

    echo "<div align='left' style='background-color: rgba(0, 0, 0, 0.1); padding-bottom: 10px;'>";
    echo "<div style='margin-left: 2%; padding-top: 25px;'><a>Tags: " . $row["tags"] . "</a> <a style='margin-left:2%'></a> <a style='margin-left:2%'>
    </a> <a style='margin-left:2%'></a> <a style='margin-left:2%'></a> <img style='margin-left: 30px; margin-right:0px;
    src='avatars/whiteava.png' width='15'
    height='15'></a> <a style='color: white;' href='profile.php?user_id=" . $row["user_id"] . "'> " . $row["artist_name"] . "</a> ";
    echo "<a style='margin-left:2%'>| <a style='margin-left:2%; align='right' class='abutton' download href='" . $row["file_path"] . "'>Download </a>";
    echo "</div>";
    echo "</div>";

    echo "</div>";|
    echo "<br>";
  }
} else {
  echo "No tracks found.";
}
?>

```

Рисунок 3.4 – Код плеєру

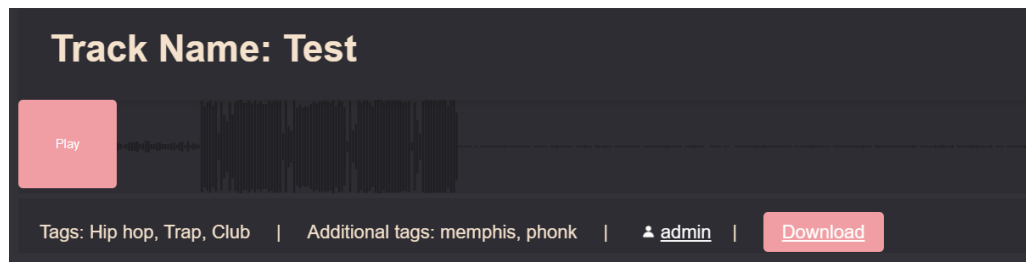


Рисунок 3.5 – Плеєр вебсайту

Також, для створення динамічної форми хвилі еквалізації композиції було використано бібліотеки `Wavesurfer.js`.

`Wavesurfer.js` – це бібліотека JavaScript для відтворення та візуалізації аудіо на вебсайтах [9]. Вона дозволяє легко інтегрувати аудіоплеєр з хвилястим зображенням аудіохвиль у ваші вебдодатки. Ось деякі основні функції та можливості `Wavesurfer.js`:

Основні функції:

- відтворення аудіо: `Wavesurfer.js` дозволяє відтворювати аудіофайли на вашому вебсайті;
- візуалізація аудіохвиль: вона генерує хвилясте зображення аудіохвиль, що відображає звукову хвилю;
- підтримка різних форматів: `Wavesurfer.js` підтримує багато різних аудіоформатів, включаючи MP3, WAV, OGG тощо.

Розширені можливості:

- плагіни: `Wavesurfer.js` має велику кількість плагінів, які розширюють його функціональність, наприклад, плагіни для вибору області відтворення, роботи з метаданими, збереження зображень тощо;
- керування відтворенням: ви можете керувати відтворенням аудіо, зупиняти, паузувати, перемотувати, змінювати гучність тощо;
- кастомізація: `Wavesurfer.js` дозволяє налаштовувати вигляд хвилястого зображення, включаючи колір, товщину ліній, швидкість оновлення тощо.

Форма хвилі в плеєрі вебсайту зображена на рисунку 3.6.

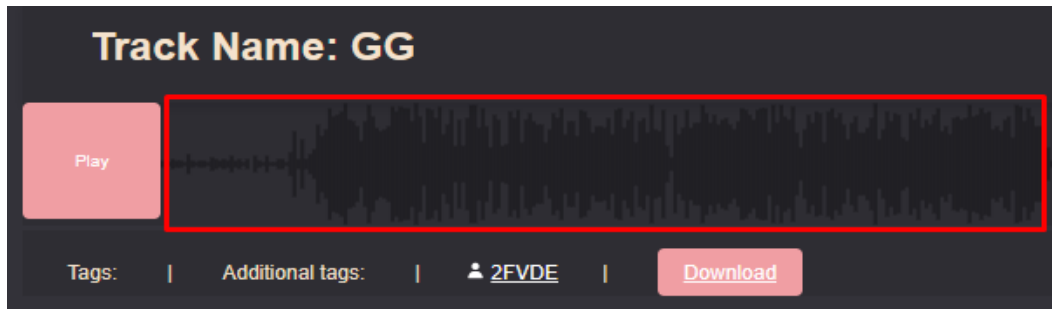


Рисунок 3.6 – Форма хвилі в плеєрі

Для відображення драмкітів та плагінів було створено блоки з інформаційними картками, в яких міститься інформація про файли, приклад сторінки з драмкітами та плагінами на яких використовуються інформаційні картки зображено на рисунках 3.7 та 3.8 відповідно. Код для реалізації карток зображено на рисунках 3.9 та 3.10.

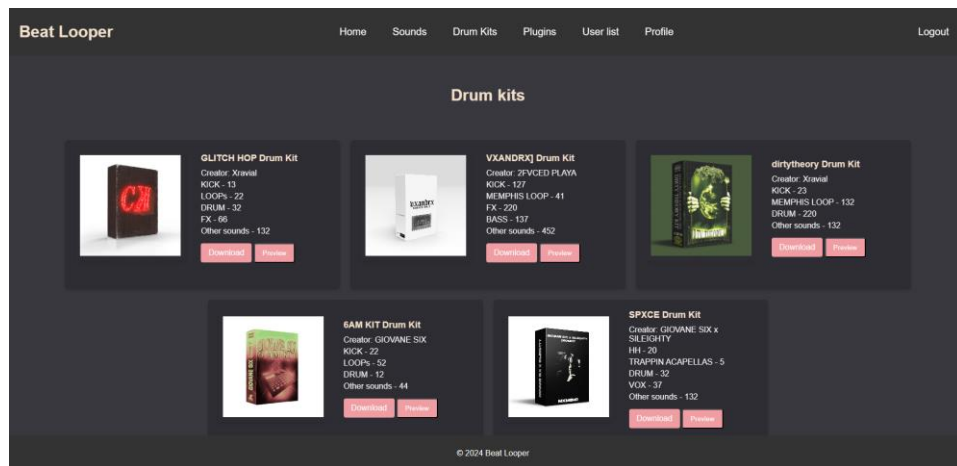


Рисунок 3.7 – Сторінка драмкітів

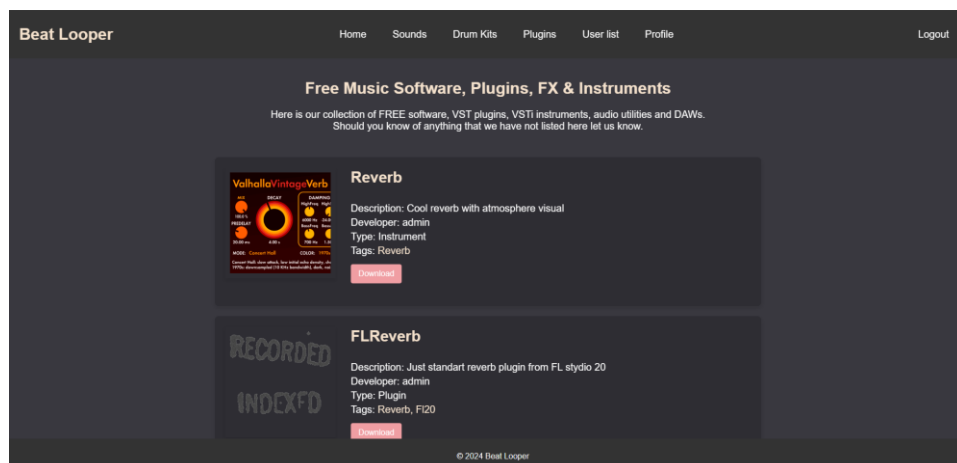


Рисунок 3.8 – Сторінка плагінів

```

<!--card-->
<div class="card" data-drumkit-id="1">
  <div class="cover-container paused">
    
  </div>
  <div class="info">
    <h2>GLITCH HOP Drum Kit</h2>
    <p>Creator: Xrivial</p>
    <p>KICK - 13</p>
    <p>LOOPS - 22</p>
    <p>DRUM - 32</p>
    <p>FX - 66</p>
    <p>Other sounds - 132</p>
    <a href="download2.zip" class="download-btn">Download</a>
    <audio id="audio2" src="audio/ghop.mp3"></audio>
    <button class="download-btn" onclick="toggleAudio('audio2')">Preview</button>
  </div>
</div><!--card end-->

```

Рисунок 3.9 – HTML код інформаційної картки

```

.container {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
  padding: 20px;
}

.card {
  display: flex;
  background-color: #2e2d33;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  border-radius: 8px;
  overflow: hidden;
  margin: 10px;
  width: 30%;
  padding-bottom: 2%;
  align-items: center;
  position: relative;
}

```

Рисунок 3.10 – CSS код інформаційної картки

3.2 Бекенд

Наступним кроком було створення фронтенд частини вебсайту. Для створення та запуску серверу було використано інструмент XAMPP Control Panel.

XAMPP Control Panel – це інструмент для управління серверним середовищем XAMPP, яке надає користувачам можливість легко

встановлювати і запускати вебсервери на локальному комп'ютері. ХАМРР є аббревіатурою від “Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) і Perl (P)”. Це безкоштовний і простий у використанні пакет, який містить всі необхідні інструменти для локальної розробки вебдодатків.

Основні компоненти ХАМРР:

- Apache – популярний вебсервер, який дозволяє розміщувати вебсайти;
- MariaDB – система управління базами даних (попередньо використовувався MySQL);
- PHP – мова програмування для серверної частини вебдодатків;
- Perl – мова програмування, часто використовується для обробки текстів і веброзробки;
- phpMyAdmin – інструмент для управління базами даних через вебінтерфейс.

Для релізації авторизації користувача на вебсайті було створено таблицю users в базі даних MySQL, структура таблиці користувачів приведена на рисунку 3.11.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 username	varchar(100)	utf8mb4_general_ci		Нет	Нем			Изменить Удалить Ещё
<input type="checkbox"/>	2 email	varchar(100)	utf8mb4_general_ci		Нет	Нем			Изменить Удалить Ещё
<input type="checkbox"/>	3 password	varchar(100)	utf8mb4_general_ci		Нет	Нем			Изменить Удалить Ещё
<input type="checkbox"/>	4 reg_date	date			Нет	Нем			Изменить Удалить Ещё
<input type="checkbox"/>	5 avatar_path	varchar(255)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	6 user_id	int(11)			Нет	Нем		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	7 facebook	varchar(255)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	8 soundcloud	varchar(255)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	9 instagram	varchar(255)	utf8mb4_general_ci		Да	NULL			Изменить Удалить Ещё
<input type="checkbox"/>	10 is_admin	tinyint(1)			Да	0			Изменить Удалить Ещё

Рисунок 3.11 – Таблица users

Після створення таблиці в базі даних було написано PHP код який зможе записувати вхідні данні з поля вводу в відповідні стовбці. PHP код зображено на рисунку 3.12.

```
require('db.php');

if (isset($_REQUEST['username']) && isset($_REQUEST['email'])) {
    $username = stripslashes($_REQUEST['username']);
    $username = mysqli_real_escape_string($con, $username);
    $email = stripslashes($_REQUEST['email']);
    $email = mysqli_real_escape_string($con, $email);
    $password = stripslashes($_REQUEST['password']);
    $password = mysqli_real_escape_string($con, $password);
    $reg_date = date("Y-m-d H:i:s");
```

Рисунок 3.12 – PHP код для реєстрації

Було реалізовано завантаження звуків на серверну частину вебсайту за допомогою PHP коду. Після завантаження файлу, заповнення полів та відправки форми користувачем, в базі даних генерується папка з іменем користувача якщо вона не була створена і в цю папку завантажуються відправлені користувачем файли.

Приклад завантажених даних на серверній часті зображено на рисунку 3.13. PHP код для завантаження файлів на серверну частину вебсайту зображено на рисунку 3.14. Діаграма завантаження користувачем звуків зображена на рисунку 3.15.

Имя	Дата изменения	Тип
6666ce490c0f1_memph...	10.06.2024 12:58	Файл "MP3"
6666ce490c0f1_memph...	10.06.2024 12:58	Chrome HTML Do

Рисунок 3.13 – Завантажені файли користувачем admin

```
// Шлях до папки для збереження
$target_dir = "music_files";

// Перевірка відправки файлу
if (isset($_FILES["music_file"]) && $_FILES["music_file"]["error"] == 0) {
    // Генерація імені файлу
    $file_name = uniqid() . "." . basename($_FILES["music_file"]["name"]);
    $target_file = $target_dir . $file_name;

    // Переміщення файлу в папку користувача
    if (move_uploaded_file($_FILES["music_file"]["tmp_name"], $target_file)) {
        // Отримуємо значення з форми
        $track_name = mysqli_real_escape_string($con, $_POST['track_name']);
        $artist_name = mysqli_real_escape_string($con, $_POST['artist_name']);
        $instrument = mysqli_real_escape_string($con, $_POST['instrument']);
        $key = mysqli_real_escape_string($con, $_POST['key']);
        $bpm = mysqli_real_escape_string($con, $_POST['bpm']);
        $additional_tags = mysqli_real_escape_string($con, $_POST['additional_tags']);
        $tags = isset($_POST["tags"]) ? implode(', ', $_POST["tags"]) : '';

        $username = $_SESSION["username"];

        // Створити папку користувача якщо вона не існує
        $user_folder = "users/" . $username;
        if (!file_exists($user_folder)) {
            mkdir($user_folder, 0777, true);
        }

        // Шлях до папки користувача
        $user_track_file = $user_folder . "/" . $file_name;

        // Переміщення файлу в папку користувача
        rename($target_file, $user_track_file);
```

Рисунок 3.14 – PHP код для завантаження файлів на серверну частину

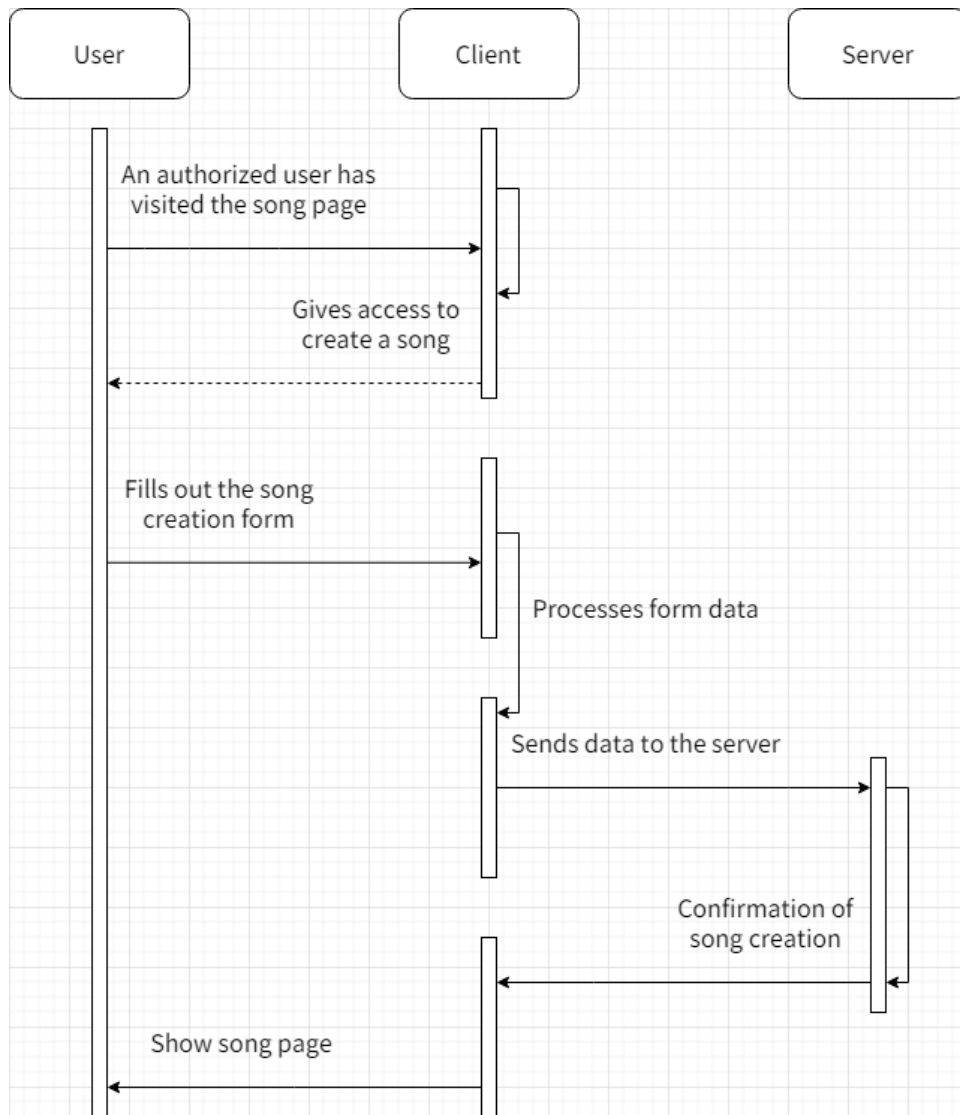


Рисунок 3.15 – Діаграма завантаження звуків користувачем

Після завантаження файлів в базу даних записується шлях до файлу та цьому файлу надається унікальний ідентифікатор який зв'язаний з ідентифікатором користувача який завантажив цей файл, для відображення інформації та звуку в плеєрі, приклад показано на рисунку 3.16.

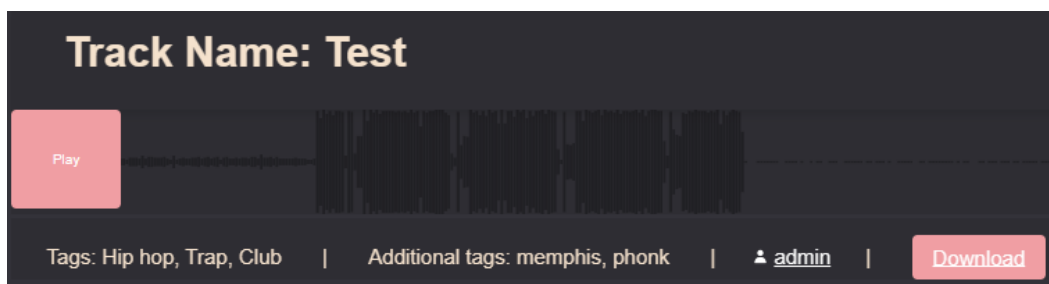


Рисунок 3.16 – Інтегрована інформація з бази даних та сервера в плеєрі

3.3 Тестування вебсайту

Коли користувач потрапляє на сторінку вебсайту він бачить головну сторінку на який коротко описаний функціонал вебсайту та останні дії які були виконані, а саме:

- останні завантаженні користувачами композиції;
- останні завантаженні драмкіти;
- останні зареєстровані користувачі;
- останні завантаженні користувачами файли.

Вид головної сторінки для неавторизованих користувачів зображено на рисунку 3.17, а вид сторінки для авторизованих користувачів зображено на рисунку 3.18, також було створено діаграму компонентів для сторінки з звуками завантажених користувачами, діаграму зображено на рисунку 3.19.

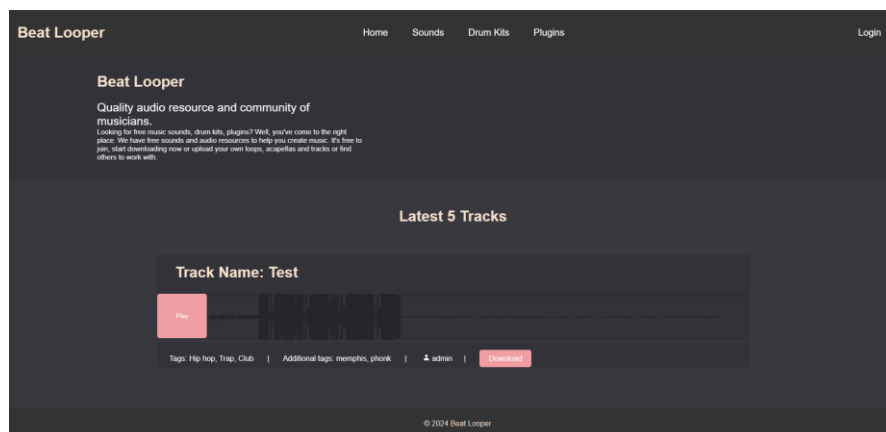


Рисунок 3.17

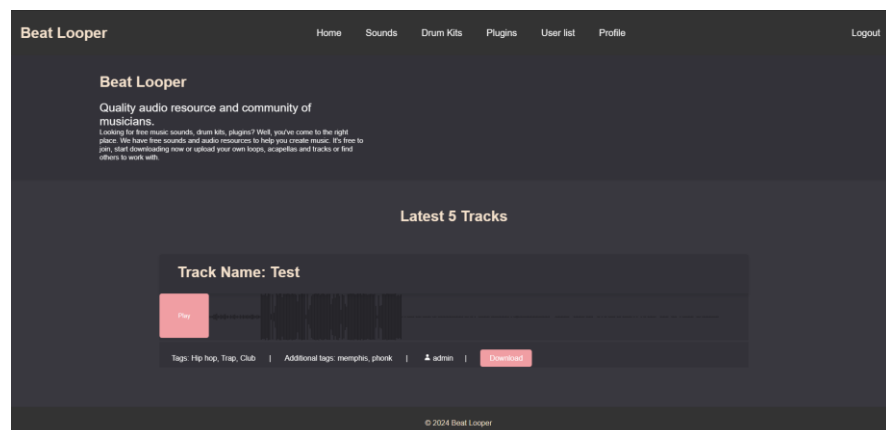


Рисунок 3.18

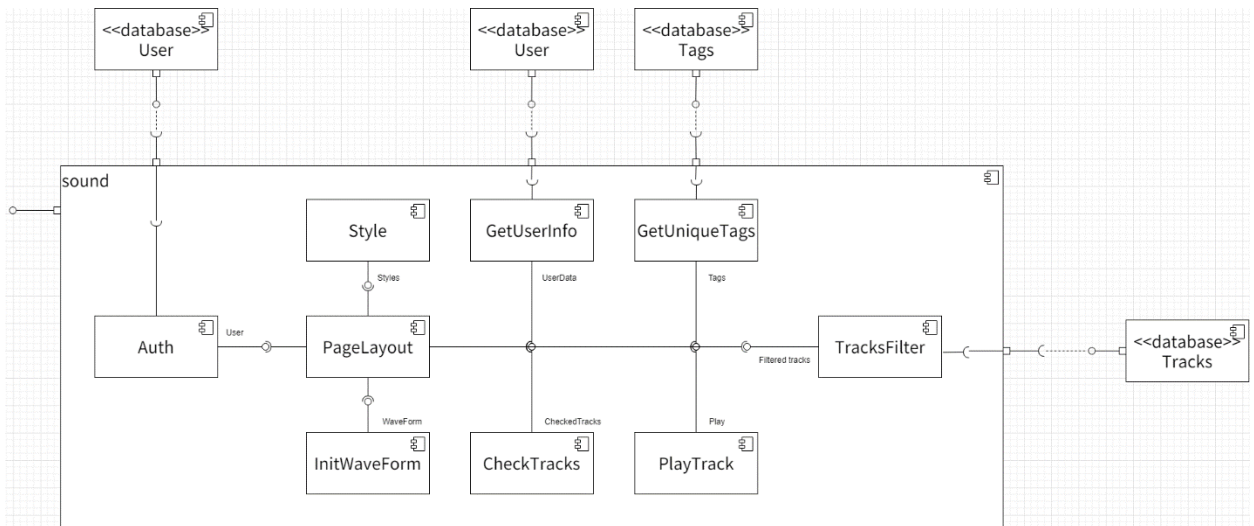


Рисунок 3.19 – Діаграма компонентів для сторінки Sounds

Для завантаження своїх файлів або композицій користувачу треба авторизуватись або зареєструватись, при натисканні на кнопку login користувача переносить на сторінку авторизації, сторінка авторизації зображена на рисунку 3.20, після чого користувач може ввести свої данні для входу або натиснувши на кнопку register зареєструватись на вебсайті, сторінка реєстрації зображена на рисунку 3.21.

Після авторизації користувач може перейти на свій профіль, та відредагувати свої данні, такі як: ім'я; електронна адреса; аватар; пароль; соціальні мережі.

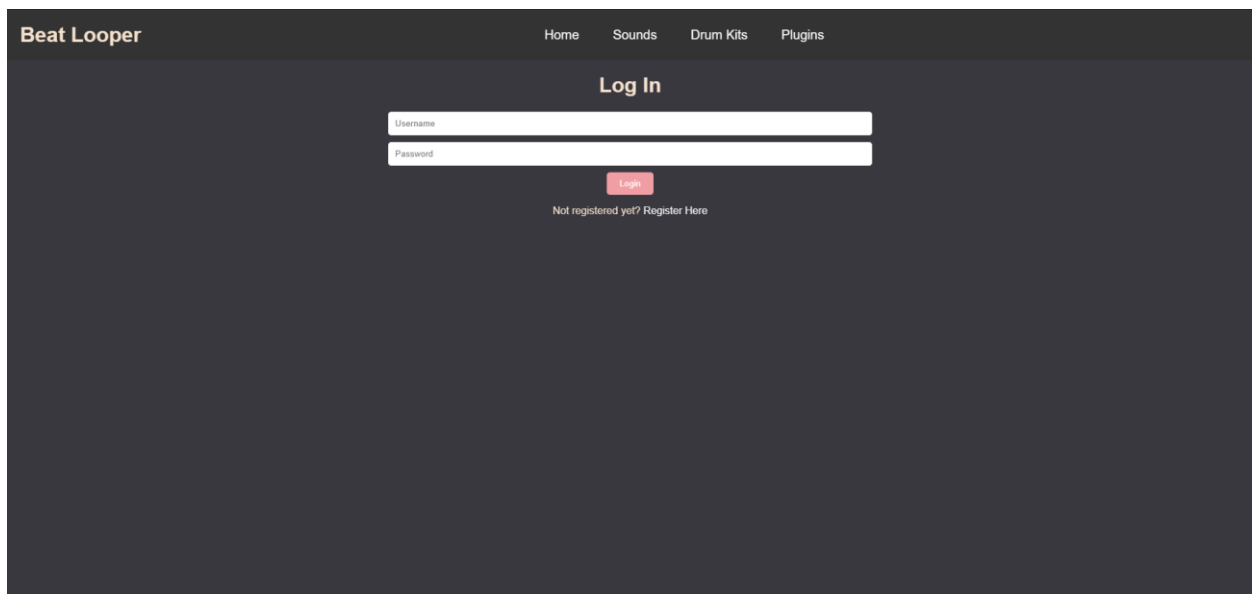


Рисунок 3.20 – Сторінка авторизації на вебсайт

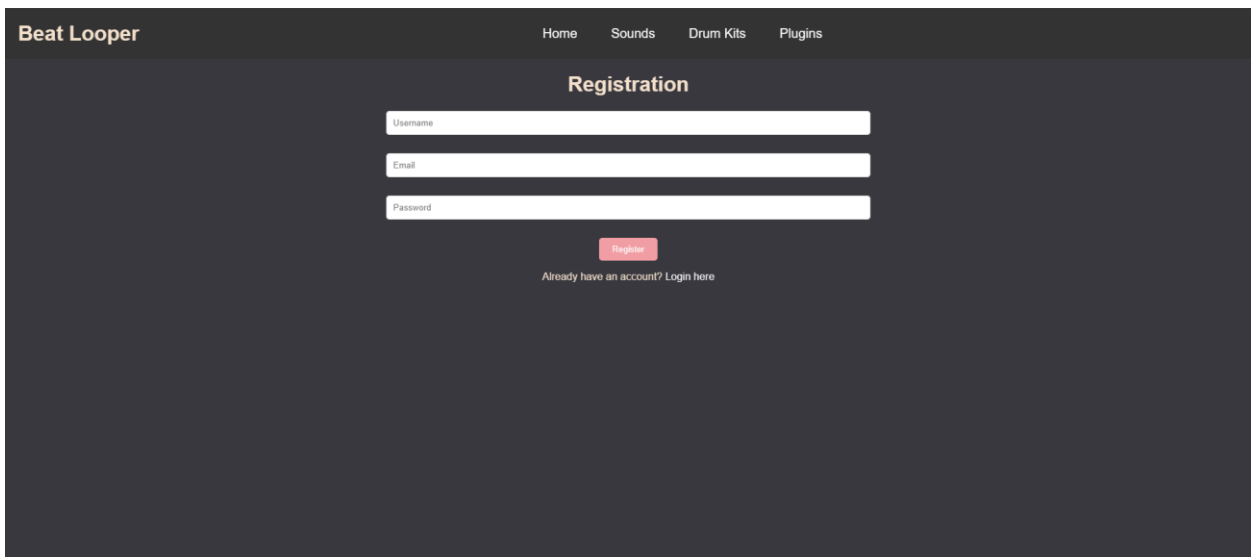


Рисунок 3.21 – Сторінка реєстрації на вебсайт

Сторінку профілю зображено на рисунку 3.22.

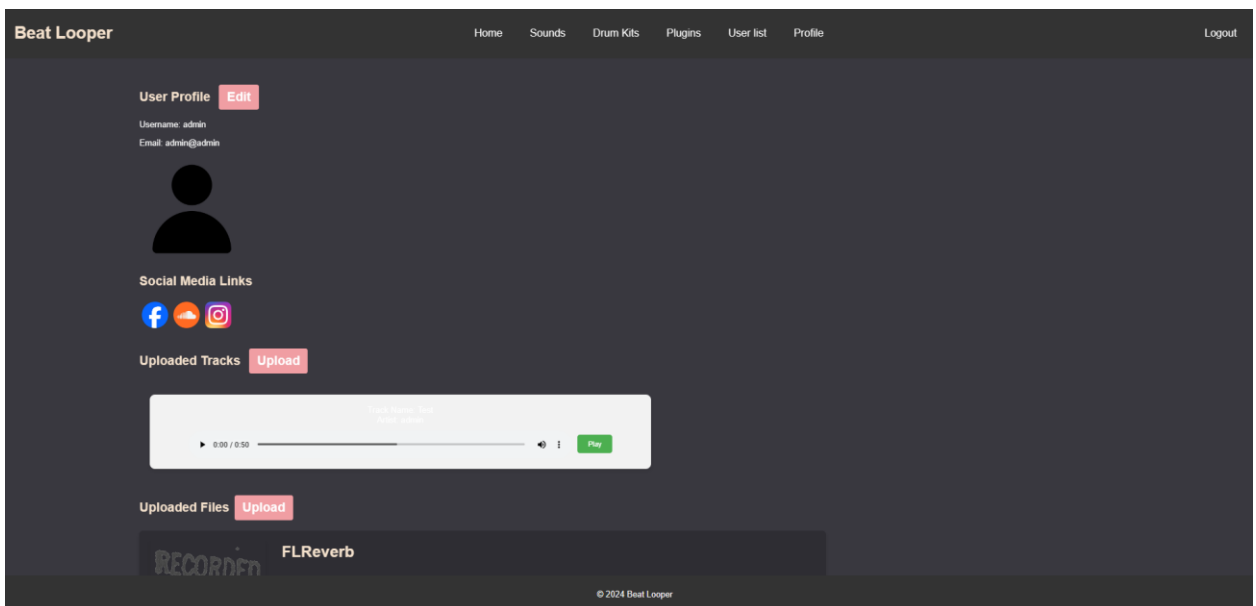


Рисунок 3.22 – Сторінка профілю користувача

Користувачі на вебсайті діляться на 3 типи:

- неавторизовані користувачі;
- авторизовані користувачі;
- користувачі з правами адміністратора.

Діаграма користування вебсайтом користувачами з різними типами прав зображено на рисунку 3.23.

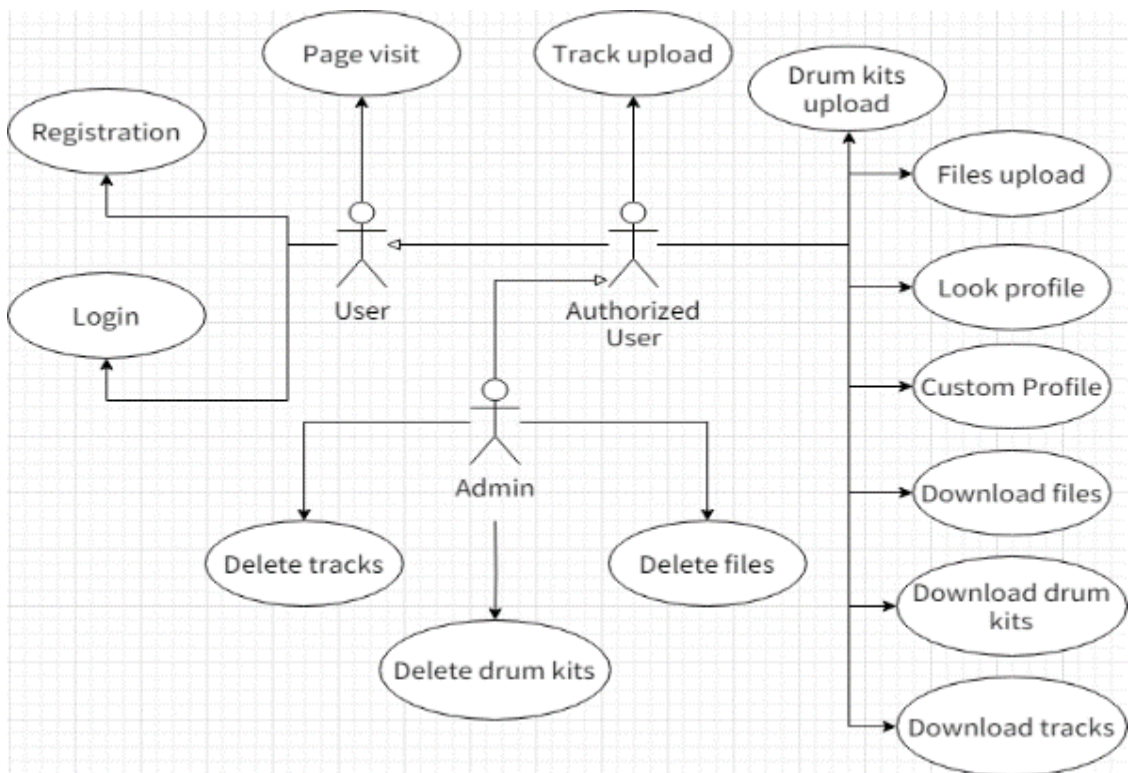


Рисунок 3.23 – Діаграма користування вебсайтом користувачами

Для редагування даних користувачу достатньо натиснути на кнопку Edit яка знаходиться біля поля User Profile, після чого його перенаправить на сторінку де він зможе відредагувати свої данні, сторінка редагування даних користувача зображена на рисунку 3.24.

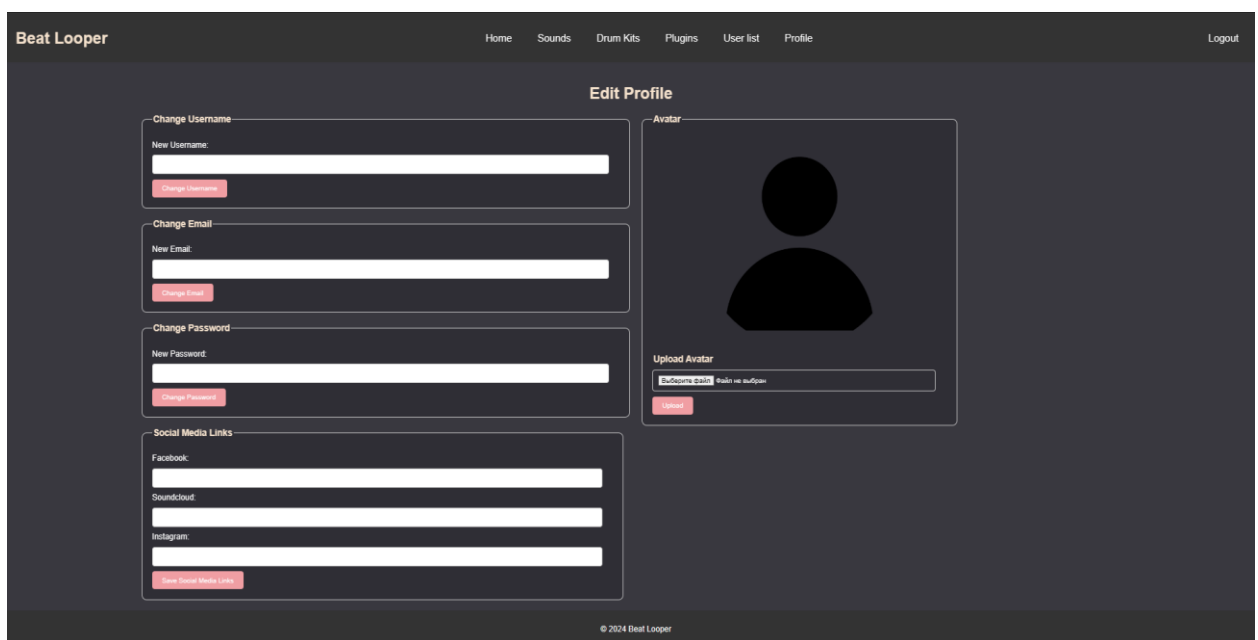


Рисунок 3.24 – Сторінка редагування профілю користувача

Також користувач після аторизації має змогу завантажити свої композиції, та файли в його профілі, натиснувши на кнопку Upload напроти поля Uploaded Tracks користувача перенаправить на сторінку для завантаження композиції, кнопка Upload для композицій зображена на рисунку 3.25, сторінка для завантаження композиції на вебсайт зображена на рисунку 3.26.

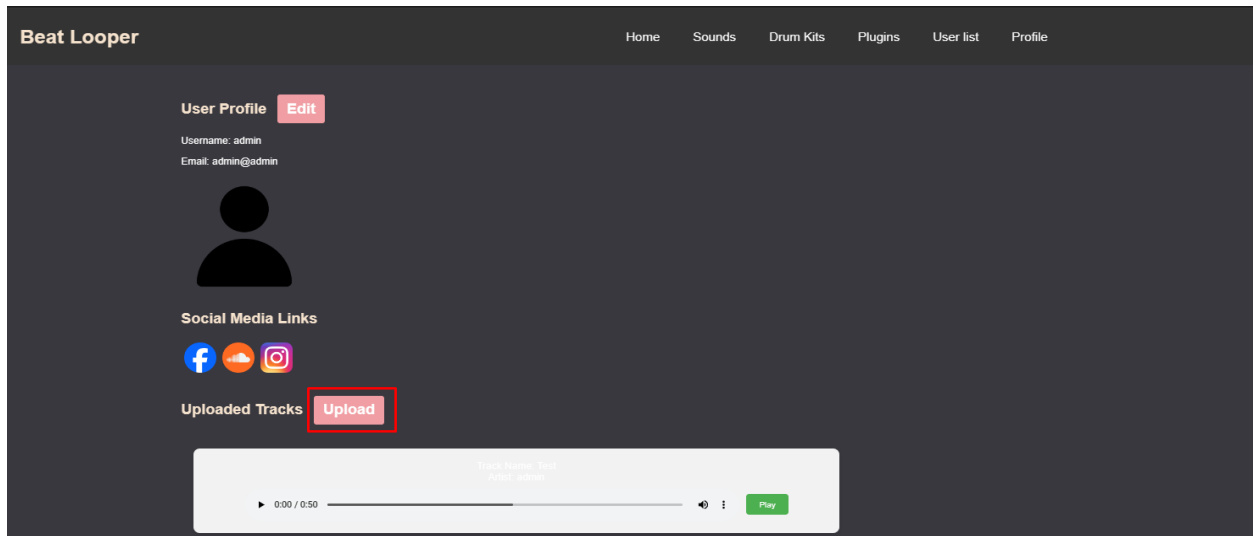


Рисунок 3.25 – Кнопка завантаження композицій

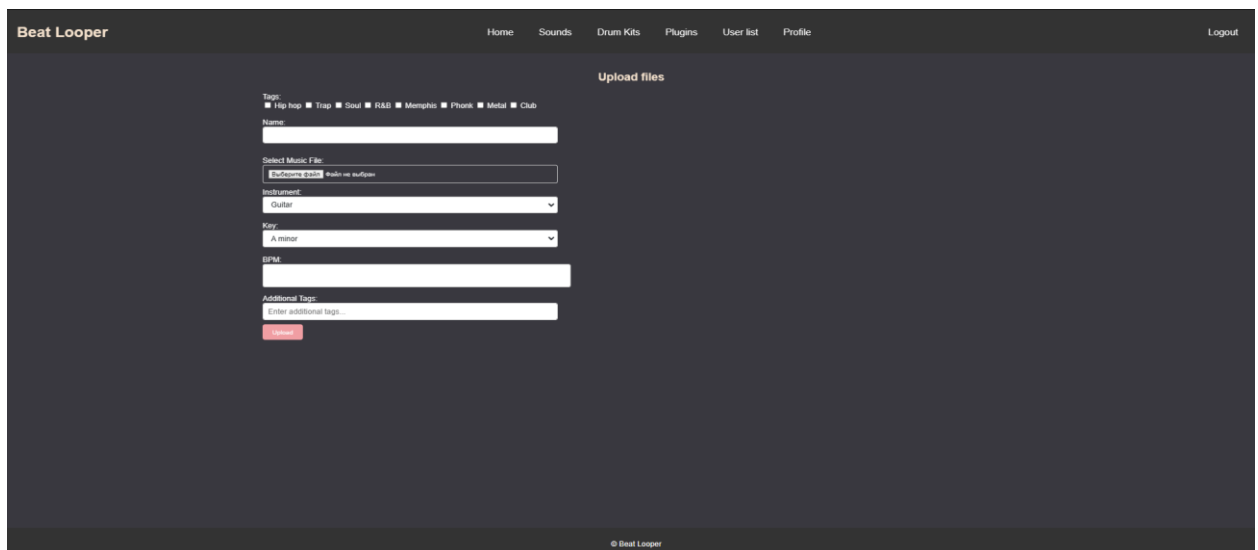


Рисунок 3.26 – Сторінка завантаження на вебсайт композицій

Для завантаження даних користувачу треба натиснути на кнопку Upload напроти поля Uploaded Files, після чого користувача перенаправить на

сторінку для завантаження файлів, кнопка Upload для композицій зображена на рисунку 3.27, сторінка для завантаження файлів зображена на рисунку 3.28.

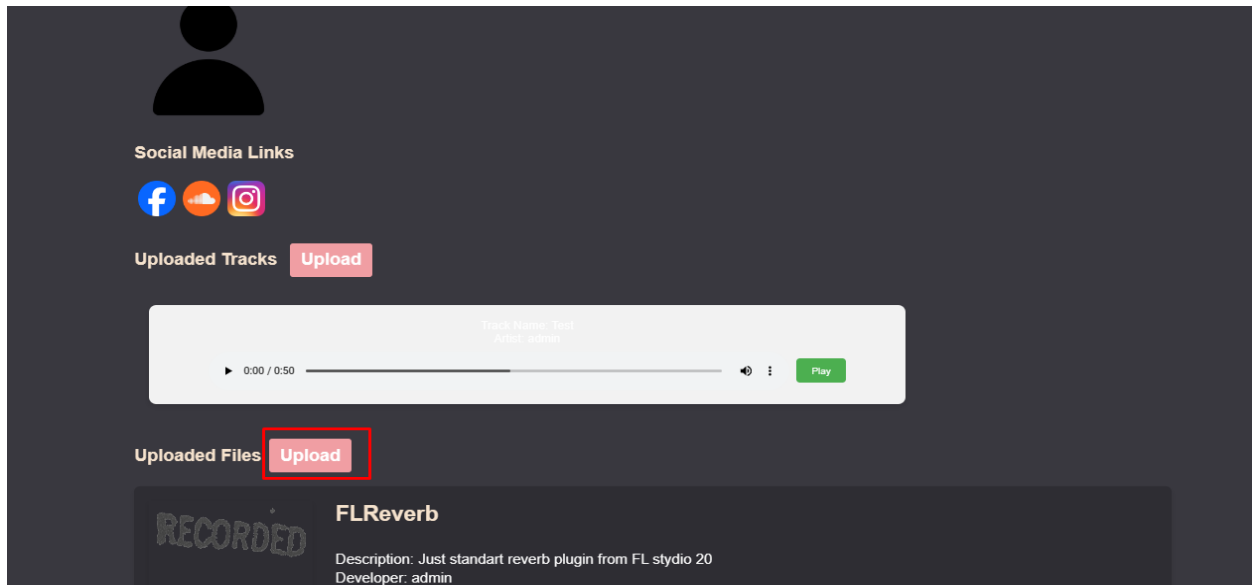


Рисунок 3.27 – Кнопка завантаження файлів

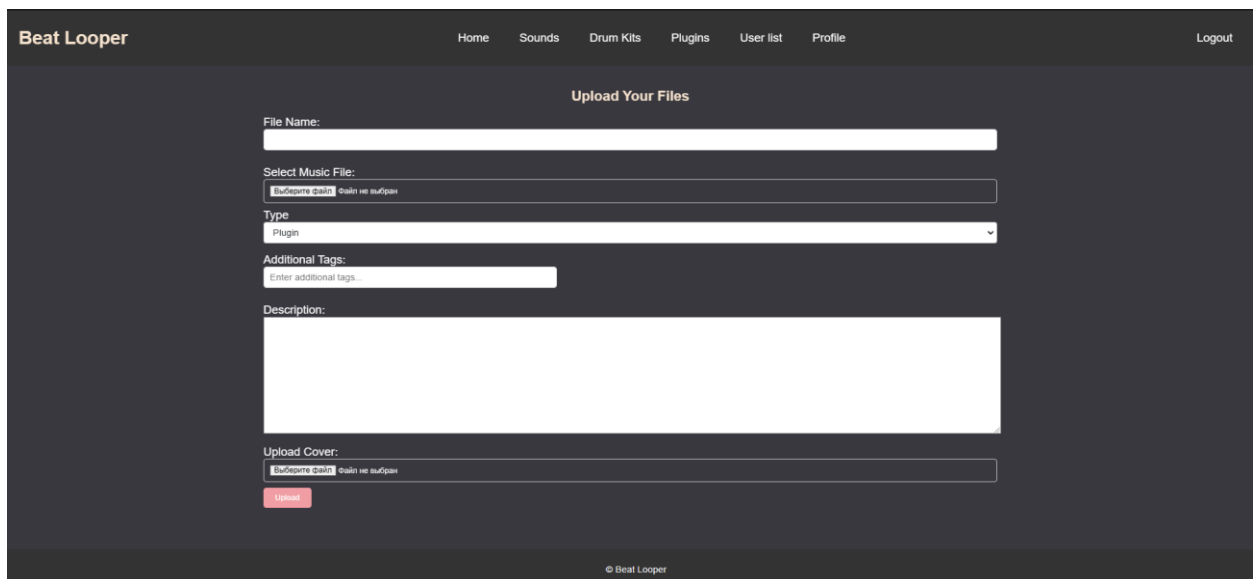


Рисунок 3.28 – Сторінка завантаження файлів

Після завантаження користувачем композиції вона з'явиться в нього в профілі та на сторінці звуків в плеєрі, де другі користувачі зможуть переглянути, послухати та завантажити цю композицію, приклад завантаженої композиції користувачем зображено на рисунку 3.29. Діаграма завантаження користувачем звуку з вебсайту зображено на рисунку 3.30.

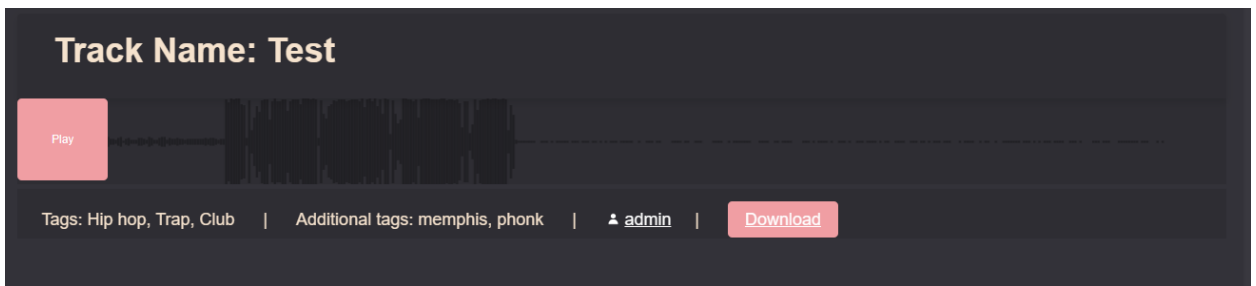


Рисунок 3.29 – Завантажена композиція користувачем на сторінці звуків

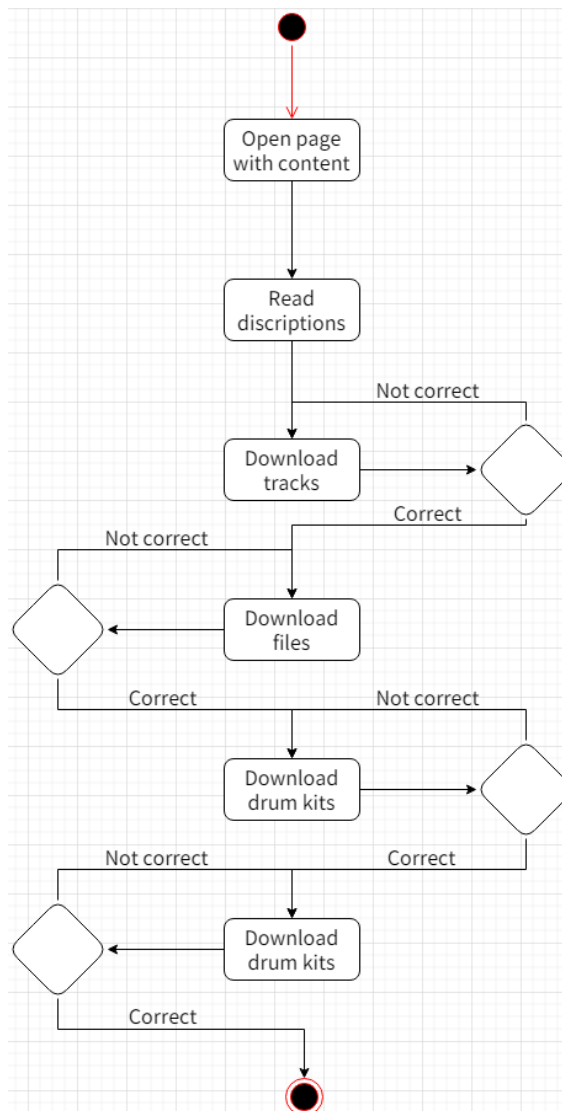


Рисунок 3.30 – Діаграма завантаження користувачем звуку

А після завантаження користувачем файлу він також з'явиться в нього в профілі, та на сторінці файлів, де другі користувачі також зможуть переглянути, послухати та завантажити цей файл для створення композицій, приклад завантаженого файлу користувачем зображено на рисунку 3.31.

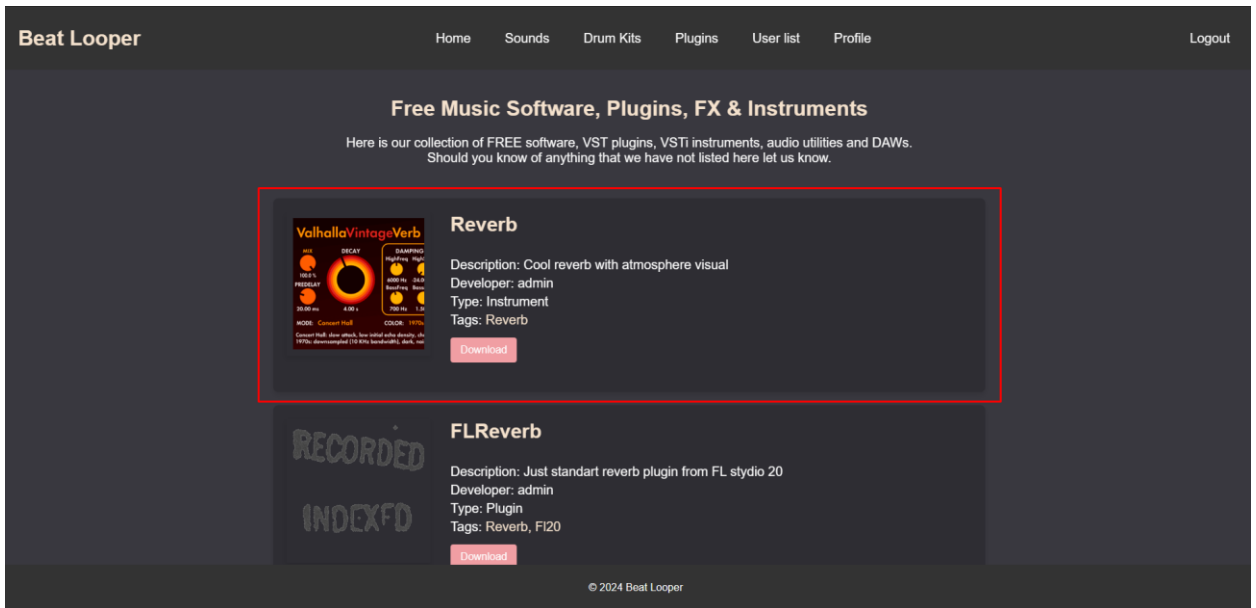


Рисунок 3.31 – Завантажена композиція користувачем на сторінці звуків

На сторінці з драмкітами користувач може переглянути інформацію про нього, послухати приклад композиції яку було створено на основі цього драмкіту, та завантажити його. Завантажити драмкіти може лише адміністратор вебсайту. Сторінка з завантаженими драмкітами зображена на рисунку 3.32.

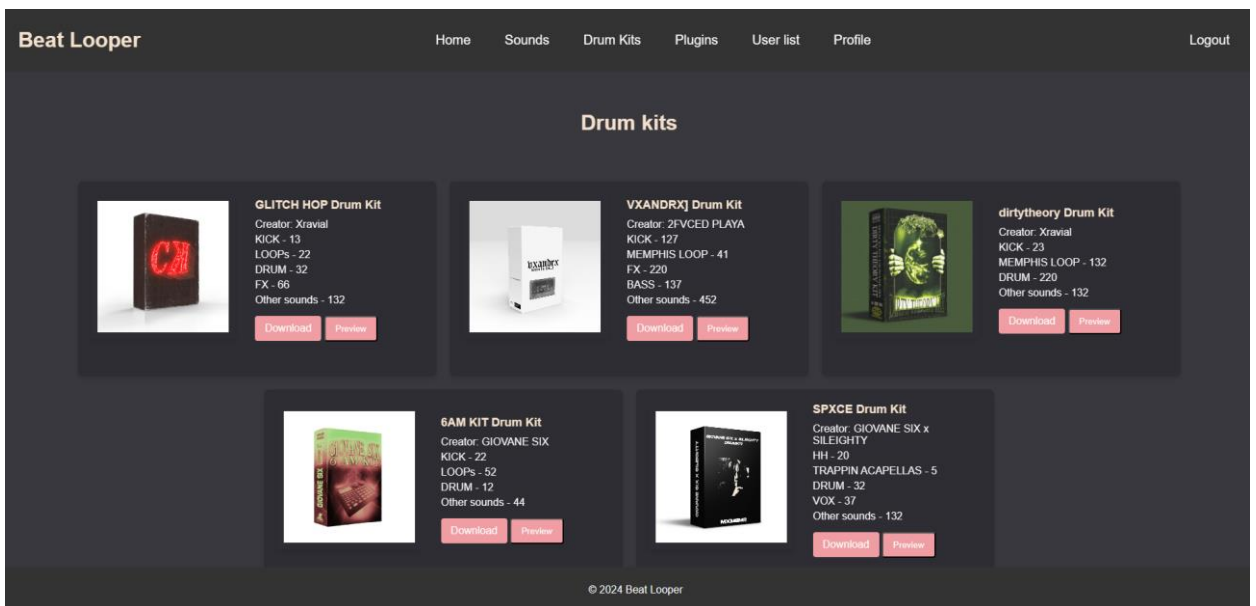


Рисунок 3.32 – Сторінка з завантаженими драмкітами

ВИСНОВКИ

У процесі виконання даної кваліфікаційної роботи було досягнуто мети – розроблено вебсайт для спільноти музикантів та бітмейкерів. Даний вебсайт надає користувачам можливість створювати профілі, завантажувати та прослуховувати музичні композиції, знаходити для себе самий підходящий матеріал, спілкуватися та об'єднуватися у групи для спільної роботи.

Для досягнення поставленої мети було виконано наступні завдання:

- аналіз існуючих вебтехнологій та платформ для створення соціальних мереж;
- розробка технічного завдання;
- розробка структури сайту та його дизайну;
- реалізація вебсайту;
- тестування та налагодження;
- оцінка результатів.

Було розглянуто різні підходи та технології для розробки вебсайтів, включаючи HTML, CSS, JavaScript та PHP. Вибір технологій обґрунтовано їхніми функціональними можливостями та придатністю для створення соціальної платформи.

На основі аналізу було складено технічне завдання, яке включало вимоги до функціоналу вебсайту, зокрема, можливість реєстрації та авторизації користувачів, створення та редагування профілів, завантаження музичних файлів, функціональність форуму та груп.

Було розроблено карту сайту, яка визначає основні розділи та підрозділи. Створено макети інтерфейсу користувача, що забезпечує зручність та інтуїтивну зрозумілість навігації по сайту.

Було реалізовано всі основні функції вебсайту за допомогою HTML для структури сторінок, CSS для стилізації, JavaScript для інтерактивних елементів, та PHP для серверної частини. MySQL використано як базу даних для зберігання інформації про користувачів та їх музичні твори.

Проведено юніт-тестування, інтеграційне тестування та юзабіліті-тестування для забезпечення високої якості кінцевого продукту. В процесі тестування було виявлено та виправлено ряд помилок, а також внесено поліпшення в функціонал та дизайн сайту.

В результаті тестування та аналізу відгуків користувачів було підтверджено, що розроблений вебсайт відповідає всім вимогам технічного завдання та забезпечує зручність використання.

Досягнуті результати:

- функціональність: сайт дозволяє користувачам створювати та редагувати профілі, завантажувати музичні файли, спілкуватися у форумах та об'єднуватися в групи;
- інтерфейс: забезпечено інтуїтивно зрозумілий і зручний для користувачів інтерфейс;
- продуктивність: сайт демонструє високу швидкість роботи та надійність завдяки використанню ефективних вебтехнологій;
- безпека: реалізовано заходи для забезпечення безпеки даних користувачів, включаючи захист від SQL-ін'єкцій та використання безпечних методів аутентифікації.

У цілому, розроблений вебсайт є готовим до використання та відповідає всім вимогам, що висувалися на початку проєкту. Сайт створює інтерактивний простір для спільноти музикантів та бітмейкерів, що сприяє розвитку творчих здібностей користувачів та поліпшенню їхніх навичок шляхом обміну досвідом та спільної роботи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Корпоративний сайт – створення веб сайту компанії. URL: <https://insite.com.ua/poslugi/stvorennya-saitu/korporativniy-sait> (дата звернення: 02.02.2024).
2. Створення сайту новин (ЗМІ) під ключ. URL: <https://digiants.agency/uk/development/razrobotka-sayta-dlya-biznesa/razrobotka-novostnogo-sayta> (дата звернення: 02.02.2024).
3. Сайт-візитка: практичне керівництво і приклади. URL: <https://webdevandseo.com.ua/site-business-card-practical-guide/> (дата звернення: 02.02.2024).
4. Що таке веб-дизайн і з чим його їдять? URL: <http://savelink.org.ua/shho-take-veb-dizajn-i-z-chim-jogo-yidyat/> (дата звернення: 02.02.2024).
5. Сайт-каталог – Webicom. URL: <https://webi.com.ua/ua/article/sajt-katalog/> (дата звернення: 02.02.2024).
6. Онлайн сервіс. Розробка, створення онлайн-сервісів. URL: <https://itmoko.com/onlajn-servis> (дата звернення: 20.03.2024).
7. Створення веб-сайтів – Ергономіка – Google Sites. URL: <https://sites.google.com/site/ergonomika55/stvorennya-veb-sajtiv> (дата звернення: 02.03.2024).
8. Конструювання сайтів. Використання онлайн-систем. URL: <https://vseosvita.ua/library/konstruuvannya-sajtiv-vikoristannya-onlajn-sistem-konstruuvannya-sajtiv-173632.html> (дата звернення: 02.03.2024).
9. Інтегрування бібліотеки Wavesurfer.js. URL: <https://wavesurfer.xyz/examples/> (дата звернення: 13.03.2024).
10. Найкращі Java-бібліотеки, які має знати кожен розробник. URL: <https://javarush.com/ua/groups/posts/uk.3447.kava-breyk-81-naykrajsh-java-bbloteki-jak-ma-znati-kozhen-rozrobnik-chomu-treba-pisati-htk> (дата звернення: 01.04.2024).

11. PHP Створення таблиці MySQL. URL: https://w3schoolsua.github.io/php/php_mysql_create_table#gsc.tab=0 (дата звернення: 02.04.2024).
12. Ключові знання для PHP розробника. URL: <https://webi.com.ua/ua/article/sajt-katalog/> (дата звернення: 09.04.2024).
13. Онлайн сервіс. Розробка, створення онлайн-сервісів. URL: <https://foxminded.ua/shcho-potribno-znaty-php-rozrobnyku/> (дата звернення: 20.04.2024).