

# КВАЛІФІКАЦІЙНА РОБОТА

Група ПЗс-2019

Волк А. В.

2023

**ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА**

**Факультет суспільних та прикладних наук**

**Кафедра інформаційних технологій**

на правах рукопису

**Волк Артем Віталійович**

УДК 004.582

**Розробка серверної частини веб-сайту для користувачів з обмеженими  
можливостями**

Спеціальність 121 – «Інженерія програмного забезпечення»

Кваліфікаційна робота на здобуття кваліфікації бакалавра

Нормоконтроль

Студент

\_\_\_\_\_ Стисло О.В.

(підпис, дата, розшифрування підпису)

\_\_\_\_\_ Волк А.В.

(підпис, дата, розшифрування підпису)

Допускається до захисту

Керівник роботи

Завідувач кафедри

\_\_\_\_\_ к.т.н., доц. Пашкевич О.П.

(підпис, дата, розшифрування підпису)

\_\_\_\_\_ к.т.н., доц. Зікратий С.В.

(підпис, дата, розшифрування підпису)

Івано-Франківськ – 2023

ЗВО УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА  
Факультет суспільних та прикладних наук  
Кафедра інформаційних технологій

Освітній ступінь: «бакалавр»  
Спеціальність: 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**  
**Завідувач кафедри**

« \_\_\_\_ » \_\_\_\_\_ 2023 року

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

**Волк Артем Віталійович**

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

Розробка серверної частини веб-сайту для користувачів з обмеженими  
можливостями

керівник роботи:

Зікратий Сергій Вікторович, кандидат технічних наук

затверджена наказом вищого навчального закладу від « 11 » листопада 2022 року  
№ 155/ІНВ

2. Термін подання студентом роботи 14.06.2023

3. Вихідні дані роботи: Мова програмування JavaScript

4. Зміст кваліфікаційної роботи (перелік питань, які потрібно розробити)

1. Розробка серверної частини веб-сайту для користувачів з обмеженими  
можливостями

2. Розробка моделі сайту та його функціоналу

3. Реалізація структури та функціоналу веб-сайту

5. Дата видачі завдання 10.10.2022

## КОНСУЛЬТАНТИ РОЗДІЛІВ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Розділ	Консультант (прізвище, ініціали та посада)	Позначка консультанта про виконання розділу	
		підпис	дата

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Основні теоретичні відомості	20.02.2023	Виконано
2.	Аналіз ринку та існуючих аналогів	11.03.2023	Виконано
3.	Розробка структури бази даних	10.04.2023	Виконано
4.	Розробка коннект-системи	18.04.2023	Виконано
5.	Розробка діаграм БД	02.05.2023	Виконано
6.	Оформлення пояснювальної записки	11.05.2023	Виконано
7.	Оформлення графічного матеріалу та підготовка до захисту роботи	01.06.2023	Виконано

Студент

\_\_\_\_\_

(підпис)

Волк А.В.

\_\_\_\_\_

(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_

(підпис)

Зікратий С.В.

\_\_\_\_\_

(прізвище та ініціали)

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

Сторінка	Опис графічного матеріалу	Сторінка	Опис графічного матеріалу
12	Діаграма компонентів розробки за принципом Accessibility	31	Шаблон-приклад Empathy map
13	Категорії фізичних вад за принципом доступності	31	Шаблон-приклад методології Design Thinking
16	Головна сторінка aamedia.org.uk/	33	Empathy map для першого типу користувачів
17	Вікно лендінг-пейджу	35	Empathy map другого типу користувачів
18	Головна сторінка сайту ablegamers.org	39	Dashboard застосунку MySQL Workbench
19	Презентаційний банер ablegamers.org	40	Інтерфейс робочого середовища
20	Навігаційне меню	44	Вайрфрейм форми авторизації
22	Dashboard-сторінка сайту Apple	45	Вайрфрейм каталогу товарів
23	Демонстрація основних інструментів доступності сайту	46	Вайрфрейм головної сторінки
24	Технологія текстового супроводу по фото	50	DB Navigator
28	Опис методу User Stories	51	Діаграма зв'язків бази даних

## АНОТАЦІЯ

Кваліфікаційна робота присвячена створенню та впровадженню функціонального та ефективної back-end частини веб-сайту, спрямованого на продаж музичних інструментів. Робота зосереджується на використанні сучасних технологій розробки, зокрема мови програмування Java та фреймворку Spring Boot для реалізації зв'язку бекенду та інших аспектів системи, а також бази даних MySQL для зберігання та керування інформацією про товари та замовлення.

В рамках дипломної роботи було проведено аналіз вимог та потреб потенційних користувачів, що дозволило врахувати їхні очікування та надати зручну та інтуїтивно зрозумілу функціональність. Застосування принципів Accesibility дозволило забезпечити доступність та використання сайту користувачами з обмеженнями зору.

Результатом дипломної роботи є функціональна бекенд-частина веб-сайту, яка надає користувачам усіх категорій функціонал для швидкого та зручного здійснення покупок музичних інструментів, отримання детальної інформації про товари, складання замовлення та здійснення оплати.

## ANNOTATION

The qualification work is dedicated to the development and implementation of a functional and efficient back-end part of a music instrument e-commerce website. The focus of the work is on utilizing modern development technologies, specifically the Java programming language and the Spring Boot framework, for implementing the backend logic and other aspects of the system, as well as the MySQL database for storing and managing information about products and orders.

Within the scope of the thesis, an analysis of requirements and user needs was conducted, allowing for the consideration of their expectations and the provision of convenient and intuitively understandable functionality. By applying Accessibility principles, the website's accessibility and usability for users with visual impairments were ensured.

The outcome of the thesis is a functional back-end component of the website that provides users of all categories with the necessary functionality for quick and convenient purchase of music instruments, obtaining detailed information about products, placing orders, and making payments.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	9
ВСТУП.....	10
РОЗДІЛ 1. ОПИС НАЯВНИХ АНАЛОГІВ.....	12
1.1 Загальна характеристика та опис застосування принципу Aaccessibility в Україні.....	1
2	
1.2 Загальний опис технологій Accessibility.....	15
1.3 Огляд та аналіз існуючих аналогів.....	15
1.4 Постановка задачі.....	27
Висновки до розділу 1 .....	27
РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ САЙТУ ТА ЙОГО ФУНКЦІОНАЛУ .....	29
2.1 Створення User Stories.....	29
2.2 Розробка Empathy map.....	32
2.3 Структура баз даних.....	39
2.4 Проектування User Interface.....	44
2.5 Інструменти за допомогою яких був створений сайт.....	48
Висновки до розділу 2 .....	50
РОЗДІЛ 3. РЕАЛІЗАЦІЯ СТРУКТУРИ ТА ФУНКЦІОНАЛУ ВЕБСАЙТУ .....	51
3.1 Розробка backend сайту .....	51
Висновки до розділу 3 .....	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- БД – бази даних;
- СУБД – система управління базою даних
- UI – інтерфейс користувача;
- JSON – текстовий формат обміну даними між комп'ютерами;
- API – прикладний програмний інтерфейс.



## ВСТУП

**Актуальність теми.** Web-site accessibility є важливою складовою розвитку ІТ-технологій в Україні. Для багатьох людей, зокрема тих що мають фізичні обмеження, доступність веб-сайтів є ключовою умовою для отримання інформації та взаємодії з онлайн-ресурсами. Однак, на сьогоднішній день, багато веб-сайтів в Україні не відповідають стандартам доступності, що ставить під загрозу права людей з обмеженнями на рівність у доступі до інформації та послуг.

Одним з основних проблем українського ринку веб-сайтів є низький рівень освіти та невідповідність законодавства щодо доступності веб-сайтів. Багато розробників веб-сайтів не мають достатнього знання про стандарти доступності та не розуміють важливості забезпечення доступності для всіх користувачів. Крім того, законодавство щодо доступності веб-сайтів в Україні є недостатнім, що призводить до того, що багато веб-сайтів не виконують навіть базові вимоги до доступності [1].

Однак, на щастя, зростає свідомість про важливість доступності веб-сайтів в Україні. Багато компаній та організацій звертають увагу на доступність своїх веб-сайтів та впроваджують стандарти доступності для поліпшення користувацького досвіду. Крім того, українські організації та громадські активісти працюють над вдосконаленням законодавства щодо доступності веб-сайтів, що дозволить забезпечити доступність для всіх користувачів.

Актуальність Web-site accessibility в Україні полягає в тому, що доступність веб-сайтів є важливим фактором для забезпечення рівних можливостей у доступі до інформації та послуг для всіх користувачів, в тому числі і тих з обмеженнями. Низький рівень освіти та невідповідність законодавства становлять головні проблеми на сьогоднішній день. Однак, зростає свідомість про важливість доступності веб-сайтів серед компаній, організацій та громадських активістів, що вказує на поступове покращення ситуації в цій сфері [2].

**Мета роботи.** Розроблення Web-сайту для продажу музикальних інструментів, з урахуванням та імплементацією UI-технологій для полегшення роботи з сайтом людей з порушеннями зору.

**Об'єкт роботи.** Підвищення доступності сайту та розширення його аудиторії, полегшення процесу користування Інтернетом людей з обмеженими можливостями.

**Предмет роботи:** Створення сайту для продажу музичних інструментів

**Завдання роботи:** Відповідно до вибраної теми в роботі покладені такі задачі:

- пошук та аналіз існуючих сайтів з аналогічними технологіями;
- вибір мови програмування, технологій та інших суміжних програм (при необхідності);
- пошук даних про технології Accessibility
- розробка архітектури сайту та якісної бекенд частини, відповідаючи принципам доступності;
- проведення тестування продукту.

**Методи роботи.** Для реалізації проекту та виконання поставлених задач, було використано мову програмування Java, фреймворк до мови Java – Spring Framework, а також СУБД MySQL Workbench 8.0

**Результати роботи.** В результаті виконання кваліфікаційної бакалаврської роботи був розроблений Web-сайт для продажу музикальних інструментів, з урахуванням та імплементацією UI-технологій для полегшення роботи з сайтом людей з порушеннями зору.

**Структура роботи.** Розділи – 4. Загальний обсяг основної частини – 46 сторінок. Список використаних джерел – 20.

## РОЗДІЛ 1. ОПИС НАЯВНИХ АНАЛОГІВ

### 1.1 Загальна характеристика та опис застосування принципу Accessibility України

Тема доступності веб-сайтів для людей з обмеженнями зору є надзвичайно актуальною і важливою в сучасному світі. Згідно з даними Всесвітньої організації охорони здоров'я (ВООЗ), у світі є близько 285 мільйонів людей з обмеженнями зору, з яких 39 мільйонів є незрячими. Ці люди мають право на повноцінний доступ до всіх інформаційних ресурсів, які є доступними для більшості користувачів Інтернету [1].

Тому, розробка веб-сайтів з використанням технологій доступності є дуже важливою. Це допомагає забезпечити рівний доступ до інформації для всіх користувачів, незалежно від їхніх фізичних можливостей. При цьому враховується також зростаюча популярність онлайн-магазинів з різними товарами, зокрема музичними інструментами. Для того, щоб користувачі з обмеженнями зору могли використовувати такі сайти, вони повинні мати доступ до адаптивного дизайну, який забезпечує зручне користування та навігацію.

Отже, важливо розуміти, що доступність веб-сайтів для людей з обмеженнями зору - це не просто вимога чи рекомендація, а це є необхідність, яка забезпечує рівні можливості для всіх користувачів Інтернету. Розробка сайту з використанням технологій доступності може стати додатковою перевагою, що забезпечить більшу кількість клієнтів та задоволеність їх потреб.

Нажаль, в Україні лише невелика частина веб-сайтів повноцінно відповідає принципам Доступності по всім параметрам. Але тенденція ситуації покращується і багато веб-розробників приділяють більше уваги користувачам з особливими потребами. [2]

Схема розробки принципу зображена на (рис. 1.1).

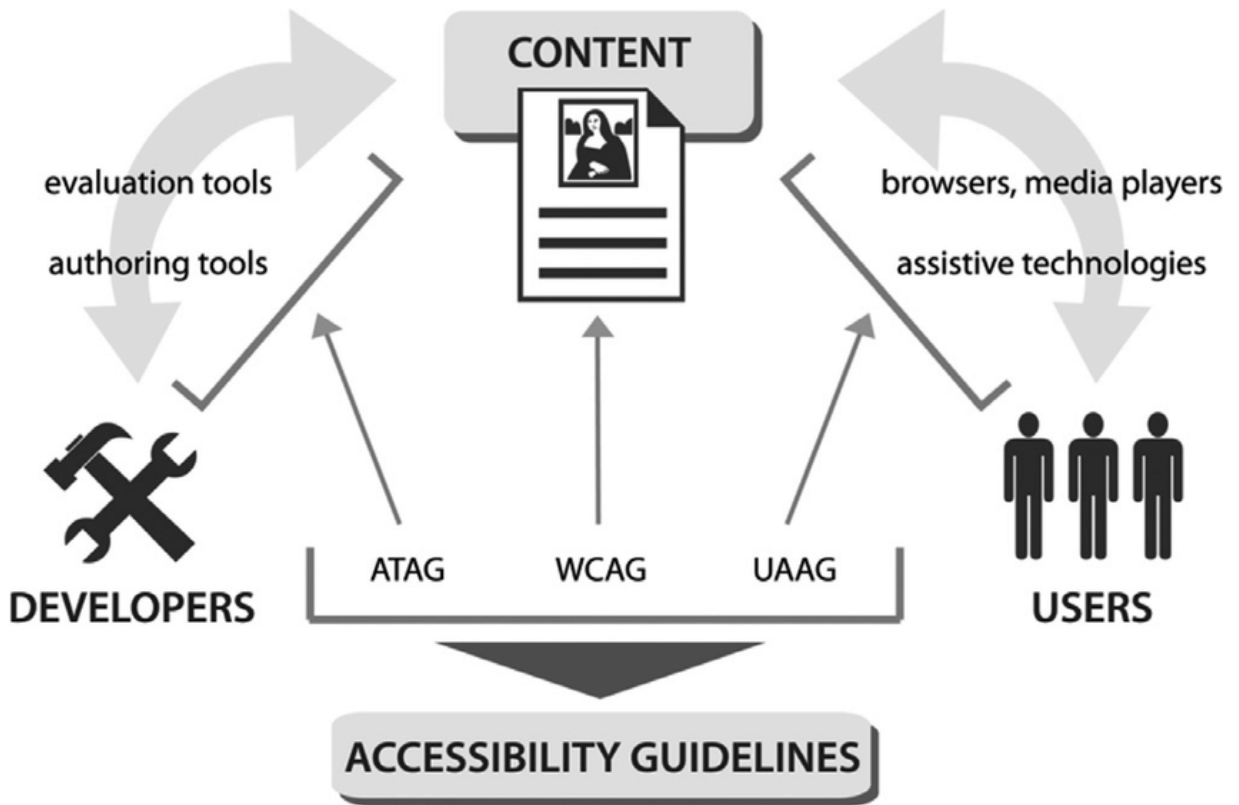


Рисунок 1.1 – Діаграма компонентів розробки за принципом Accessibility

Типи фізичних дефектів є різноманітними та можуть включати обмеження зору, слуху, рухової активності, чуттєвих відчуттів тощо. Для людей з обмеженнями зору можуть бути проблемними веб-сайти, які не забезпечують достатньої контрастності кольорів, мають недостатній розмір шрифту або погано розміщені зображення. Для людей з обмеженнями слуху можуть бути важкими сайти, які мають відео без субтитрів або з незрозумілим звуком. Для людей з обмеженнями рухової активності можуть бути проблемними сайти, які не мають можливості використовувати клавішні скорочення або мають недостатньою швидкість переходу між елементами. Розробка веб-сайтів з використанням технологій доступності може допомогти вирішити ці проблеми та зробити сайти доступними для всіх користувачів [3]. Ілюстрація категоризації фізичних вад користувачів за принципом Доступності зображена на (рис. 1.2).



Рисунок 1.2 – Категорії фізичних вад за принципом доступності

Web Accessibility (доступність веб-ресурсів) - це процес створення веб-сайтів та інших цифрових продуктів, що можуть бути використані людьми з різними здатностями. Основними принципами Web Accessibility є:

1. Перцептивність: першим принципом є можливість користувачів отримувати доступ до контенту через різні канали сприйняття, включаючи зоровий, слуховий та дотиковий. Це можна забезпечити шляхом використання доступних форматів медіа, наприклад, альтернативних текстів для зображень або підписів до відео.

2. Функціональність: цей принцип означає, що веб-ресурс повинен дозволити користувачам здійснювати інтерактивні дії за допомогою різних методів керування, включаючи клавіатуру, мишу, джойстик або голосові команди.

3. Розуміння та сприйняття: ще одним важливим принципом є забезпечення доступності веб-ресурсу для людей з різними рівнями когнітивних здібностей. Це може бути досягнуто, наприклад, за допомогою простих та легко зрозумілих інструкцій та текстів.

4. Робустність: цей принцип означає, що веб-ресурс повинен бути розроблений з урахуванням різних технічних можливостей і обмежень користувачів, а також з використанням стандартів, що дозволяють забезпечувати сумісність з різними браузерами та пристроями.

## 1.2 Загальний опис технологій Accessibility

Доступність веб-сайтів є важливою для забезпечення рівних можливостей та виключення будь-яких обмежень для користувачів з різними потребами. Деякі з основних веб-технологій, що використовуються для підвищення доступності сайтів, включають:

**HTML:** HTML є основним будівельним блоком веб-сторінок і використовується для структурування та вміщення контенту. Забезпечення правильного використання HTML-тегів, атрибутів та семантичної розмітки допомагає покращити доступність сайту.

**CSS:** CSS використовується для стилізації веб-сторінок. Застосування правильних CSS-стилів дозволяє забезпечити читабельність та навігацію для користувачів, особливо тих, хто має обмеження зору.

**JavaScript:** JavaScript використовується для створення інтерактивності на веб-сторінках. З правильним використанням JavaScript можна забезпечити доступність, зокрема реагування на дії користувачів з обмеженими можливостями та забезпечення альтернативних шляхів взаємодії.

**WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications):** WAI-ARIA є набором специфікацій, що надає стандарти для розширення можливостей доступності веб-додатків. Використання WAI-ARIA дозволяє позначати елементи та їхні функціональні властивості для поліпшення сприйняття та навігації користувачами з обмеженими можливостями.

**Резиновий дизайн та адаптивність:** Резиновий дизайн і адаптивність дозволяють адаптувати веб-сторінки до різних розмірів екранів і пристроїв.

## 1.2 Огляд та аналіз існуючих аналогів

Огляд та аналіз існуючих аналогів є важливою складовою процесу розробки будь-якого веб-сайту. Перед створенням власного сайту з використанням технологій Accessibility для людей з обмеженнями зору, необхідно ознайомитися з наявними аналогами на ринку та проаналізувати їх.

Під час огляду існуючих аналогів, можна дослідити їх функціональність, дизайн та зручність використання. Також варто звернути увагу на особливості використання технологій у кожному з аналогів, зокрема, які з них використовуються для поліпшення доступності для людей з обмеженнями зору.

Аналізуючи існуючі аналоги, можна виявити їхні переваги та недоліки, а також знайти можливість вдосконалення власного сайту. Наприклад, якщо в одному з аналогів знайдено добре реалізовану функцію збільшення розміру шрифту, можна спробувати інтегрувати її у власний сайт, покращуючи тим самим його доступність.

Загалом, огляд та аналіз існуючих аналогів є важливим етапом в розробці будь-якого веб-сайту, оскільки дозволяє збагнути найкращі практики та вдосконалити власний продукт. Нижче приведено декілька сайтів-аналогів на яких реалізовано технології підвищення доступності:

- <https://ablegamers.org/> [5];
- <https://www.aamedia.org.uk/> [6];
- Apple Accessibility [7];
- W3C Web Accessibility Initiative (WAI) [8].

Сайт <https://www.aamedia.org.uk/> належить до організації, що спеціалізується на створенні доступних продуктів для людей з різними обмеженнями. Основною метою сайту є надання інформації про доступні технології та послуги, які можуть полегшити життя людей з обмеженнями [5].

Дизайн сайту добре організований і забезпечує зручний доступ до інформації. Заголовки розділів і текстові блоки є чіткими і легко читаються. Колірна гамма сайту також є дуже приємною для очей, що особливо важливо для людей з візуальними обмеженнями.

Навігація на сайті дуже проста і інтуїтивно зрозуміла. На головній сторінці є меню, яке містить вкладки з розділами сайту, а також пошукову стрічку. Всі елементи на сайті мають якісні посилання, які дозволяють користувачам швидко і легко переходити між розділами.

Один з найкращих аспектів цього сайту полягає в тому, що він пропонує доступ до багатьох технологій і інструментів, які можуть полегшити життя людей з різними обмеженнями.

Узагалі, сайт <https://www.aamedia.org.uk/> дуже добре відповідає своїм цілям і надає корисну інформацію для людей з різними обмеженнями. Дизайн сайту (рис 1.3) є простим, але ефективним, і навігація на сайті є легкою та інтуїтивно зрозумілою.

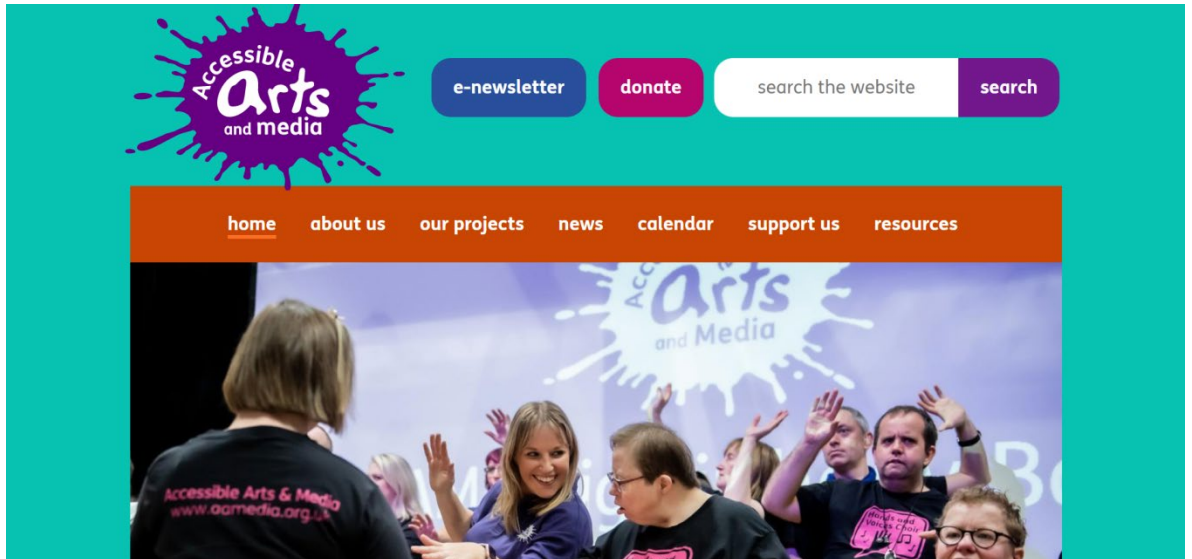


Рисунок 1.3 – Головна сторінка [aamedia.org.uk/](https://www.aamedia.org.uk/)

При розборі недоліків, хочеться зазначити відсутність локалізації сайту на будь-які інші мови, окрім англійської. При використанні сайту в темний час доби, яскравість сайту грає не на користь ще більше, але саме це забезпечує більш чітку та інформативну візуалізацію сайту для людей із слабким зором.

При огляді сайту з точки зору доступності для людей з обмеженнями зору:

Відсутність можливості збільшення тексту: на сайті немає можливості збільшення розміру тексту, що може створювати труднощі для людей з обмеженнями зору.

Невідповідність контрастності: у деяких випадках, контрастність між текстом та фоном не є достатньою, що може зробити його складним для читання.

Відсутність змісту альтернативного тексту для зображень: на сайті не вказано зміст альтернативного тексту для зображень, що може ускладнити сприйняття контенту людьми з відсутністю зору.

Використання виділення кольором для передачі інформації: деякі елементи сайту, такі як панель навігації, використовують виділення кольором для передачі



інформації, що може створювати труднощі для людей зі зниженою чутливістю до кольору.

Відсутність можливості переходу клавішами на головний вміст сайту: на сайті відсутня можливість переходу на головний вміст веб-сайту за допомогою клавіш замість миші.

Усі ці недоліки можуть ускладнити використання сайту людьми з обмеженнями зору, тому важливо їх виправити для покращення доступності сайту (рис 1.4).



Рисунок 1.4 – Вікно лендінг-пейджу

Далі, можна відзначити відсутність візуального контрасту між деякими елементами сайту, що може призвести до проблем з читанням для людей зі зниженою чутливістю до кольорів. Також, деякі елементи сайту можуть бути не очевидними для користувачів з обмеженнями зору.

Щодо веб-форм, на сайті вони реалізовані достатньо ефективно, однак було б корисним додати підказки до полів введення даних, що б допомогти користувачам з обмеженнями зору визначати призначення кожного поля.

Також, деякі зображення на сайті не мають альтернативних описів, що ускладнює їх інтерпретацію для людей з обмеженнями зору.

Проте, варто відзначити, що загальний рівень доступності сайту є достатнім для більшості користувачів, що відповідає мінімальним вимогам, але не забезпечує повної доступності для людей зі зниженим зором.

ablegamers.org – Сайт <https://ablegamers.org/> є веб-порталом, який присвячений розробці і просуванню доступної для ігрової громадськості взаємодії з відеоіграми. Загальний дизайн сайту доволі простий та зрозумілий, що допомагає користувачам швидко зорієнтуватися на сайті та знайти необхідну інформацію (рис. 1.5).

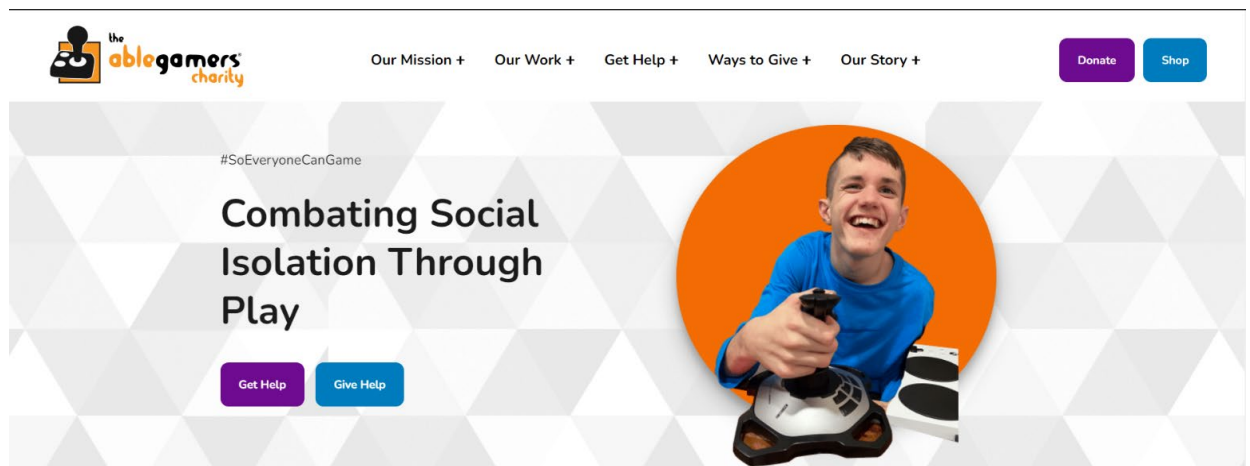


Рисунок 1.5 – Головна сторінка сайту ablegamers.org

Сайт містить безліч корисної інформації, яка охоплює різні аспекти інклюзивної геймінгової культури, такі як огляди гральних пристроїв, огляди відеоігор, новини та статті. Також, на сайті можна знайти відгуки та рекомендації від людей з обмеженнями та професіоналів у сфері ігрової індустрії.

Дизайн сайту відмінно підлаштований до потреб інклюзивної аудиторії, зокрема, у ньому зберігається контрастність між кольорами, що полегшує читання тексту для людей з обмеженнями зору. Також, сайт містить велику кількість мультимедійного контенту (відео, зображення), що дозволяє користувачам засвоювати інформацію різними способами.

Однак, на сайті можна знайти деякі недоліки щодо доступності. Наприклад, деякі кнопки на сайті не мають опису або занадто загальний опис, що може створювати проблеми для людей з обмеженнями зору, які використовують програми

для читання екрану (рис.1.6). Також, деякі відео не мають звукових описів, що є важливим аспектом доступності для людей з вадами слуху.

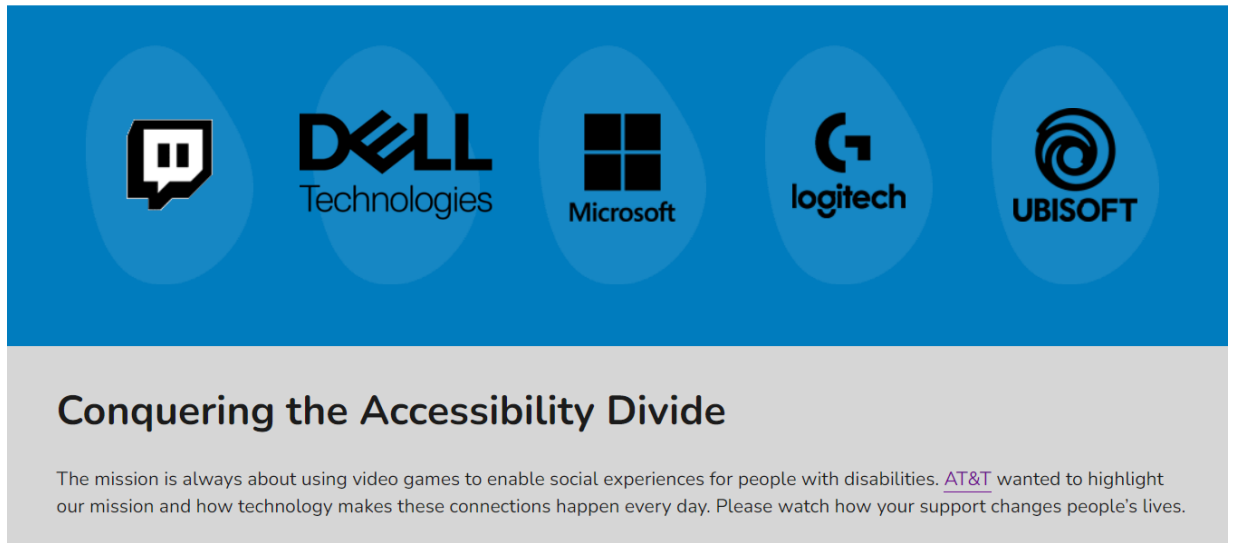


Рисунок 1.6 – Презентаційний банер ablegamers.org

За точкою зору доступності для людей з зоровими обмеженнями, обидва сайти - AbleGamers та AAMedia - мають свої переваги та недоліки [6].

Сайт AbleGamers має дуже високу доступність для людей з обмеженнями зору, оскільки він має яскравий контрастний дизайн та використовує великі шрифти, які легко читаються. Сайт також має добре організовану навігацію, що дозволяє швидко знайти потрібну інформацію. Крім того, на сайті є можливість змінювати розмір шрифту, що дозволяє користувачам зі знизеним зіром зручно переглядати контент.

Натомість, сайт AAMedia, хоч і забезпечує певний рівень доступності для людей з обмеженнями зору, проте має свої недоліки. Зокрема, дизайн сайту містить багато фотографій та графічних елементів, що може викликати проблеми зі сприйняттям інформації для користувачів з обмеженнями зору.

Отже, за точкою зору доступності для людей з обмеженнями зору, сайт AbleGamers є більш доступним та зручним для використання, ніж сайт AAMedia.

Навігаційне меню на сайті <https://ablegamers.org/> забезпечується за допомогою гамбургер-меню. Це меню розташоване в правому верхньому куті сторінки та складається з трьох горизонтальних ліній, які символізують випадаюче меню (рис.1.7).

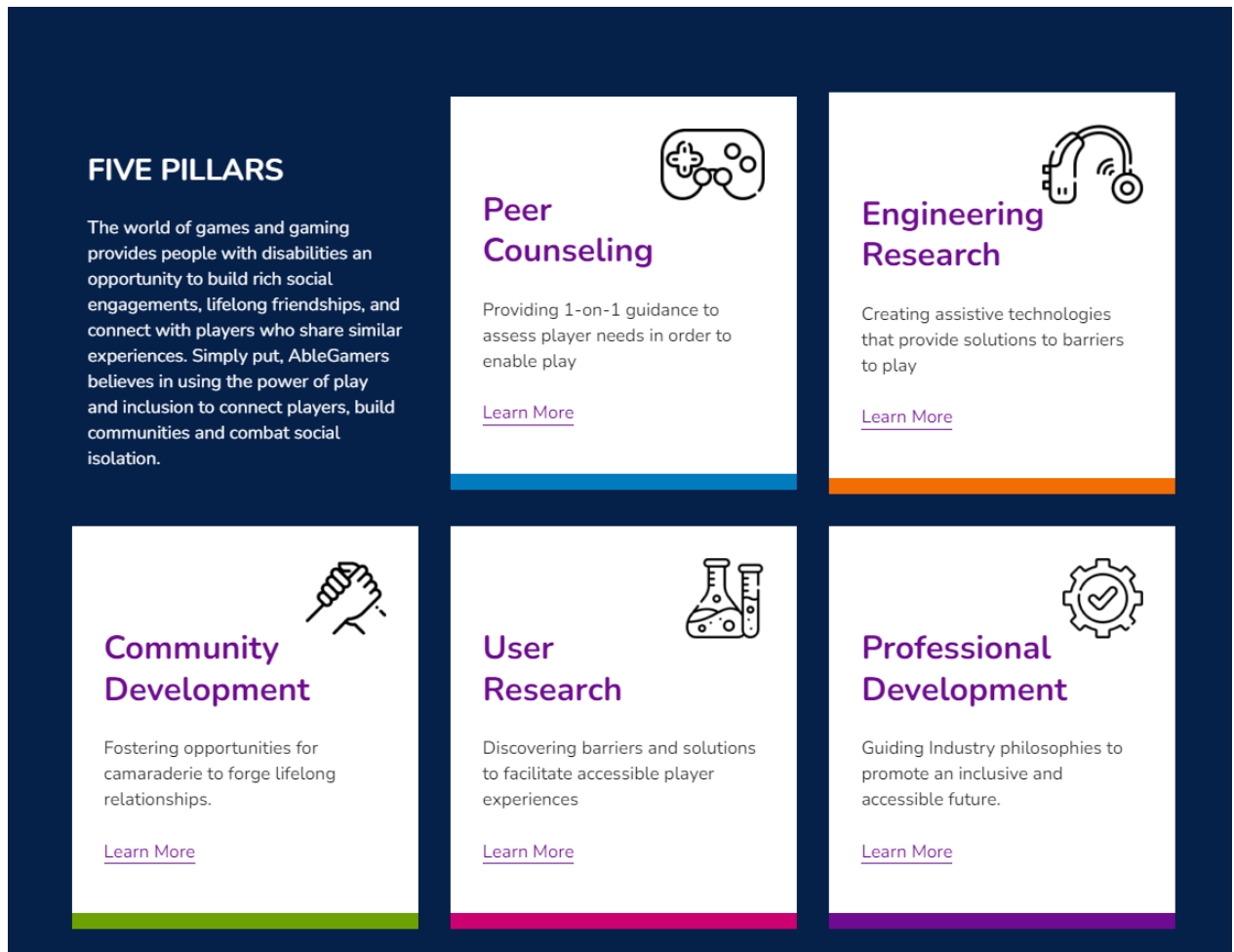


Рисунок 1.7 – Навігаційне меню

З точки зору доступності для людей з обмеженнями зору, таке меню може бути складним для сприйняття. Оскільки гамбургер-меню не має текстової мітки, користувачі з візуальними обмеженнями можуть мати проблеми з його розпізнаванням. Крім того, коли користувач натискає на гамбургер-меню, він не завжди може зрозуміти, що саме воно відкрило.

Щоб поліпшити доступність навігаційного меню для людей з обмеженнями зору, можна використовувати текстову мітку поруч з гамбургер-меню. Також можна використовувати іконку з більш розпізнаваним зображенням, що допоможе користувачам легше розпізнавати меню та розуміти, що саме воно відкрило.

Узагальнюючи, хоча навігаційне меню на сайті <https://ablegamers.org/> досить просте та легко користується, його доступність для людей з обмеженнями зору можна

покращити за допомогою текстової мітки поруч з гамбургер-меню та фіксованого блоку з головним меню.

Проблема поганої доступності навігаційного меню на сайті може бути вирішена за допомогою кількох кроків, що покращать його доступність для людей з різними видами обмежень зору.

1. Слід використовувати текстові мітки для посилань: Один з ключових моментів у поліпшенні доступності навігаційного меню - це використання текстових міток для кожного посилання. Використання текстових міток відображає на екрані користувачам з обмеженнями зору те, що знаходиться на посиланні.

2. Слід використовувати чіткі та контрастні кольори: Важливо використовувати контрастні кольори на кнопках, посиланнях та тексті, щоб зробити їх більш помітними для людей з обмеженнями зору.

3. Слід використовувати клавіші доступу до клавiш: Клавіші доступу до клавiш дозволяють користувачам навігуватися по сайту, використовуючи лише клавіатуру. Використання клавiш доступу до клавiш забезпечує користувачам з обмеженнями зору більш легкий та ефективний спосiб навігацiї по веб-сторiнках.

4. Слід використовувати назви меню та вкладок: Використовуйте короткі та прості назви для меню та вкладок, щоб дати користувачам з обмеженнями зору можливість легко зорієнтуватися на сайті та знайти потрібну інформацію.

5. Слід додати підказки та описи до кнопок та елементів меню: підказки та описи допомагають користувачам з обмеженнями зору зрозуміти, що відбувається при натисканні на кнопку або виборі елемента меню.

Сайт <https://www.apple.com/accessibility/> [7], є досить вражаючим прикладом веб-сайту, присвяченого технологіям доступності для людей з обмеженнями зору. Головна сторінка одразу ж демонструє великий рівень доступності і має декілька інструментів для полегшення взаємодії користувачів з обмеженнями зору (рис.1.8).

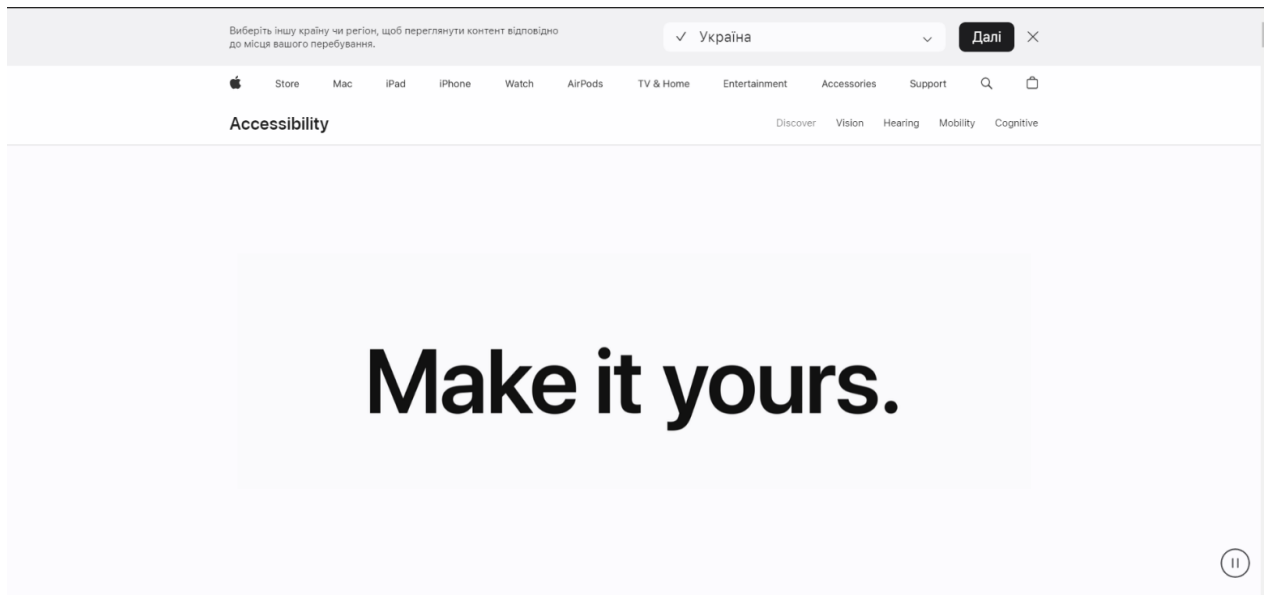


Рисунок 1.8 – Dashboard-сторінка сайту Apple

1. Візуальний дизайн: Сайт Apple Accessibility використовує сучасний і мінімалістичний дизайн, що сприяє зручному сприйняттю інформації. Користувачам з обмеженнями зору це може бути особливо корисним, оскільки спрощений дизайн полегшує навігацію та пошук потрібної інформації.

2. Зміст: Сайт містить багато корисних матеріалів про доступність продуктів Apple, включаючи описи функцій доступності та поради щодо їх використання. Інформація подана чітко та лаконічно, з фокусом на важливість доступності для всіх користувачів.

3. Альтернативний контент: Сайт Apple Accessibility використовує альтернативний текст для зображень, що дозволяє людям з обмеженнями зору отримувати описи цих зображень за допомогою екранних читачів. Це допомагає користувачам зрозуміти зміст сторінок, навіть якщо вони не можуть бачити зображення.

4. Відеоматеріали: Сайт містить відеоінструкції та демонстрації функцій доступності. Apple забезпечує наявність субтитрів і опису аудіо для цих відео, що робить їх доступними для користувачів з обмеженнями зору або слуху.

5. Клавіатурний доступ: Сайт Apple Accessibility підтримує навігацію за допомогою клавіатури, що дозволяє людям з обмеженнями зору використовувати веб-сайт без необхідності використання сторонніх програм та пристроїв.

6. VoiceOver: Apple пропонує власну технологію VoiceOver, яка є вбудованою у їхні пристрої, такі як iPhone, iPad та Mac. Сайт Apple Accessibility надає докладну інформацію про використання VoiceOver для навігації та взаємодії з веб-сайтом. Це забезпечує можливість повноцінного користування сайтом людьми з обмеженнями зору.

7. Доступність кольорів: Apple працює над забезпеченням високої доступності кольорів на своєму сайті. Вони використовують контрастність кольорів із зручною комбінацією, щоб забезпечити читабельність тексту та інших елементів для людей з обмеженнями зору.

8. Тестування доступності: Apple проводить регулярні тести на доступність свого веб-сайту, щоб переконатися, що він відповідає стандартам доступності.

9. Підтримка джерел даних: Сайт Apple Accessibility забезпечує можливість завантаження матеріалів у форматах, які є доступними для використання екранними читачами та іншими допоміжними технологіями. Це дозволяє людям з обмеженнями зору зручно отримувати інформацію, яку вони можуть використовувати на своїх пристроях.

Загалом, сайт Apple Accessibility демонструє високий рівень забезпечення доступності для людей з обмеженнями зору. Він надає розширені можливості, інформацію та підтримку для тих, хто використовує допоміжні технології (рис.1.9).

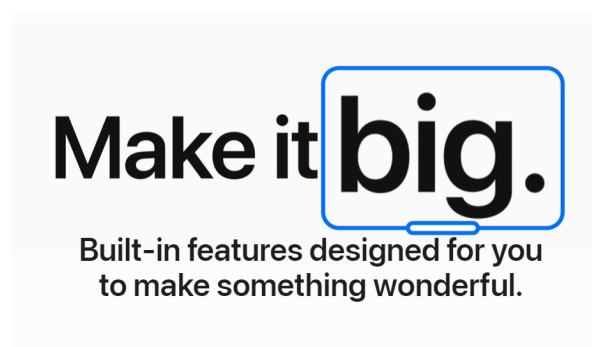


Рисунок 1.9 – Демонстрація основних інструментів доступності сайту

Незважаючи на високий рівень забезпечення доступності на сайті <https://www.apple.com/accessibility/>, є деякі недоліки, які можна зазначити. Навігація по сайту може викликати певну складність для деяких користувачів. Також можливо

використати технологію текстового супроводу по фото (рис 1.10), щоб забезпечити більш детальні описи для користувачів з обмеженнями зору .

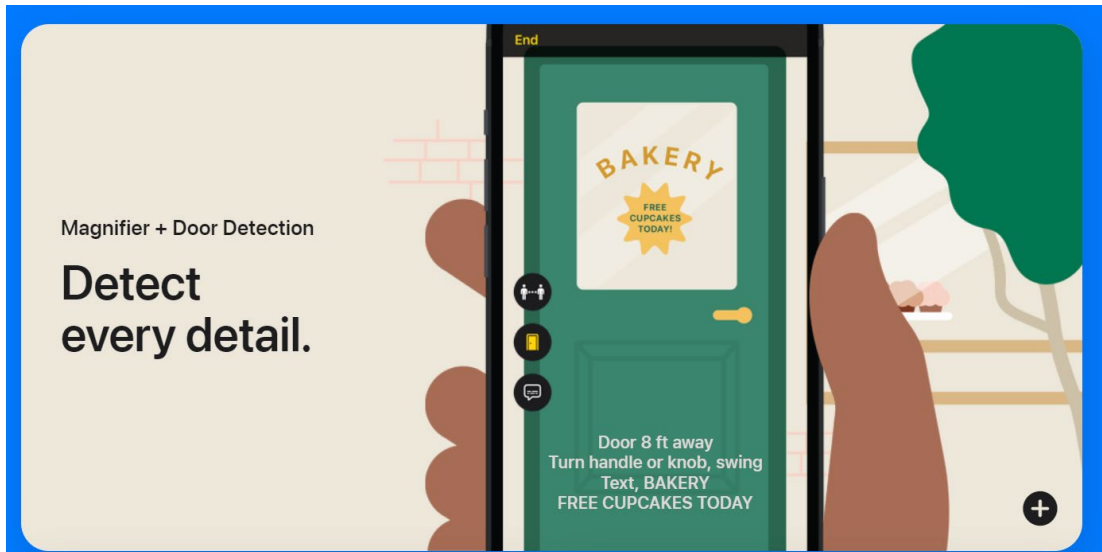


Рисунок 1.10 – Технологія текстового супроводу по фото

Після аналізу усіх існуючих сайтів-аналогів з технологіями доступності, можна виділити їхні спільні риси [8]:

1. Інклюзивний дизайн: Більшість з цих сайтів пропонують інклюзивний дизайн, що означає, що вони розроблені з урахуванням потреб користувачів з різними видами обмежень. Це означає використання чіткого та зрозумілого мовлення, консистентного дизайну, яскравого контрасту кольорів, розміщення контенту у логічних блоках та зручну навігацію.

2. Використання ARIA-атрибутів: Багато з цих сайтів використовують ARIA-атрибути (Accessible Rich Internet Applications) для покращення доступності. Ці атрибути дозволяють розробникам визначати ролі, стани та властивості елементів у веб-додатках, що полегшує їх інтерпретацію технологіями сприймання контенту.

3. Альтернативний текст для зображень: Всі ці сайти надають альтернативний текст для зображень. Це дозволяє користувачам з обмеженнями зору отримувати інформацію про зображення за допомогою програмного забезпечення для читання екрану.

4. Клавіатурна навігація: Більшість сайтів-аналогів забезпечують можливість повної навігації за допомогою клавіатури. Це дозволяє користувачам, які не можуть



використовувати мишу або тачпад, взаємодіяти з сайтом та отримувати доступ до всіх його функцій.

5. Сумісність з допоміжними технологіями: Сайти забезпечують сумісність з різними допоміжними технологіями, такими як програмне забезпечення для читання екрану, мовні вводи, допоміжні пристрої та інші. Це дає користувачам з обмеженнями зору можливість взаємодіяти з сайтом та отримувати доступ до інформації.

Загалом, сайти з технологіями доступності для людей з обмеженнями зору виявляють спільну уважливість до потреб цільової аудиторії. Вони забезпечують інклюзивне середовище, використовують відповідні технології та надають доступ до інформації та функціональності незалежно від користувачевого виду обмежень.

#### **1.4 Постановка задачі**

У відповідності до вищенаведених прикладів використання технологій Accessibility для розробки веб-сайтів, необхідно створити серверну частину веб-сайту онлайн-магазину для продажу музичних інструментів, з урахуванням принципів Доступності и відповідних технологій. Архітектура проекту будується наступним чином:

- провести аналіз переваг та недоліків уже існуючих сайтів-аналогів;
- вибрати мову та технології програмування;
- проаналізувати базові принципи Доступності;
- розробити бекенд для веб-сайту з функціоналом полегшуючим користування для людей з обмеженими можливостями;
- провести тестування продукту;
- розробити заходи з охорони праці.

## Висновки до розділу 1

У першому розділі було проведено дослідження технологій доступності для розробки веб-сайту з урахуванням потреб людей з обмеженнями зору. Були визначені основні принципи Web Accessibility.

Далі було проаналізовано кілька існуючих аналогів сайту для продажу музикальних інструментів з використанням технологій доступності. Було зазначено їхні переваги та недоліки з точки зору доступності для людей з обмеженнями зору.

Для покращення доступності навігаційного меню на сайті <https://ablegamers.org/> було запропоновано ряд рекомендацій, таких як використання підказок, покращення контрастності, забезпечення адекватного фокусування та використання доступних клавiшних шляхів.

Також було розглянуто веб-технологію звукового супроводу кнопок на сайті для людей з обмеженнями зору, яка може полегшити їхню взаємодію з сайтом та покращити їхнє враження від використання.

## РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ ФУНКЦІОНАЛЬНОСТІ ВЕБ-САЙТУ

### 2.1 Створення User Stories

User Stories - це короткі описи функціональності або вимог до програмного продукту, які зосереджені на потребах та перспективі користувачів. Вони є важливим інструментом Agile-методологій розробки програмного забезпечення, таких як Scrum [9].

User Stories формулюються від імені користувачів або представників користувачів і зазвичай включають трьохкомпонентну структуру: "Як (тип користувача), я хочу (функціональність), щоб (мета або користь)". Наприклад, "Як користувач-покупець, я хочу мати можливість додавати товари до кошика, щоб зручно зробити покупку".

User Stories допомагають команді розробки чітко розуміти потреби користувачів і спрямовують їхню увагу на важливі функціональність та цілі. Вони також дозволяють пріоритезувати роботу команди, встановлюючи пріоритети для кожної User Story.

У процесі розробки, User Stories можуть бути деталізовані, розширені і уточнені, а також використовуватися для планування, оцінки завдань, тестування та ітераційної розробки продукту. Вони сприяють зближенню команди розробки з потребами користувачів і покращують комунікацію всередині команди та зі зацікавленими сторонами.

На (рис.2.1) наведено схематичне зображення розробки ПЗ при використанні методології User Stories [10].

# HOW TO USER STORY

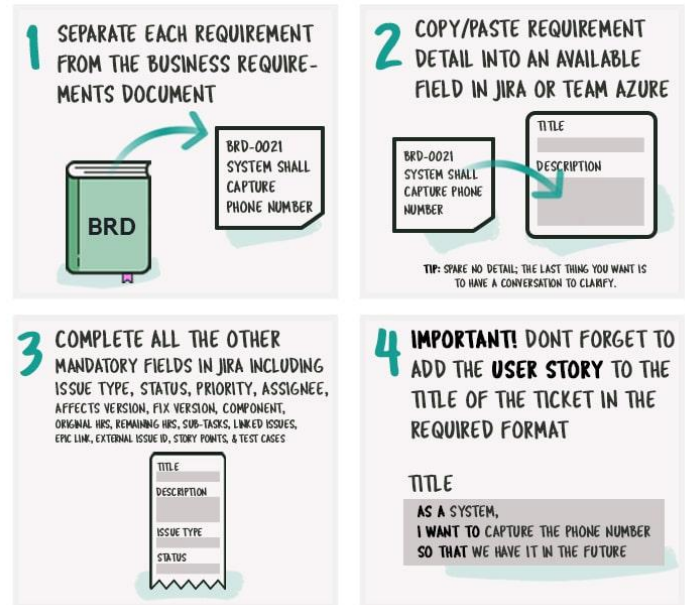


Рисунок 2.1 – Опис методу User Stories

Для реалізації сайту з продажу музикальних інструментів з підтримкою технологій доступності User Stories користувачів з обмеженнями зору буде виглядати наступним чином:

1. Як користувач з обмеженнями зору, я хочу мати можливість використовувати екранне читання для перегляду описів музичних інструментів, щоб отримати інформацію про їх характеристики та властивості.

2. Як користувач з обмеженнями зору, я хочу мати можливість використовувати клавіатуру для навігації по сайту та здійснення покупок, щоб легко переміщатися між сторінками та здійснювати операції.

3. Як користувач з обмеженнями зору, я хочу мати можливість налаштувати розмір шрифту та контрастність на сайті, щоб забезпечити зручну читабельність і поліпшити доступність.

4. Як користувач з обмеженнями зору, я хочу мати можливість слухати аудіоогляди музичних інструментів, щоб отримати додаткову інформацію про їх звукові характеристики та якість звучання.

5. Як користувач з обмеженнями зору, я хочу мати можливість використовувати схеми кольорів з високим контрастом та яскравими кольоровими позначками, щоб краще розрізняти елементи інтерфейсу та полегшити навігацію.

6. Як користувач з обмеженнями зору, я хочу мати можливість переглядати фотографії музичних інструментів з використанням альтернативного тексту, щоб отримати опис зображення та більше інформації про вигляд товару.

7. Як користувач з обмеженнями зору, я хочу мати можливість використовувати доступні формати файлів для завантаження інструкцій з використання музичних інструментів, щоб отримати доступ до текстової інформації в зручному форматі.

User Stories для звичайних користувачів:

1. Як звичайний користувач, я хочу мати можливість легко переглядати асортимент музичних інструментів на вашому сайті, щоб знайти той, який мені потрібний.

2. Як звичайний користувач, я хочу отримати докладну інформацію про кожен музичний інструмент, включаючи його характеристики, ціну, наявність та відгуки користувачів, щоб прийняти обґрунтоване рішення при покупці.

3. Як звичайний користувач, я хочу мати зручну кошик для покупок, щоб додавати обрані товари, керувати їх кількістю та легко оформляти замовлення.

4. Як звичайний користувач, я хочу мати можливість здійснювати безпечні та зручні платежі на вашому сайті, використовуючи різні методи оплати.

5. Як звичайний користувач, я хочу мати можливість зв'язатися з вас, використовуючи контактну форму або зворотний зв'язок, для отримання додаткової інформації, вирішення питань або надання відгуків.

6. Як звичайний користувач, я хочу мати зручну навігацію по сайту, зрозумілу структуру та логічне розташування основних елементів, щоб легко знаходити потрібну інформацію та товари.

Ці User Stories відображають основні потреби та очікування різних користувачів і можуть слугувати основою для подальшого розроблення функціональності та вдосконалення сайту

## 2.2 Розробка Empathy map

Empathy Map - це інструмент, що використовується для розуміння та сприйняття потреб, бажань, мотивацій та емоцій цільової аудиторії. Він допомагає створити більш глибоке розуміння користувачів, звернути увагу на їхні потреби та надати основу для розробки продукту, який задовольнить їх вимоги [11].

Empathy Map включає чотири основні області аналізу:

1. Зорова спостережливність (Seeing): Описує, що люди бачать і спостерігають. Це може включати фізичні аспекти, зовнішній вигляд, оточуюче середовище та взаємодію з іншими людьми або об'єктами.

2. Слухова сприйнятливність (Hearing): Розглядає, що люди чують або слухають. Це можуть бути слова, звуки, комунікація, відгуки або інші аспекти, які впливають на їх сприйняття.

3. Відчуття (Feeling): Відображає емоції, почуття та реакції користувачів. Це можуть бути позитивні або негативні емоції, задоволення або роздратування, радість або стрес і т.д.

4. Мислення (Thinking): Описує, що люди думають, їх потреби, цілі, мотивації та роздуми. Це можуть бути роздуми про рішення, пошук інформації, спосіб розв'язання проблеми та багато іншого.

Заповнення Empathy Map вимагає дослідження та спостереження цільової аудиторії, звернення уваги на їхні дії, емоції та мотивації [12]. Схема взаємодії методології Емпатії продемонстрована на (рис. 2.2).

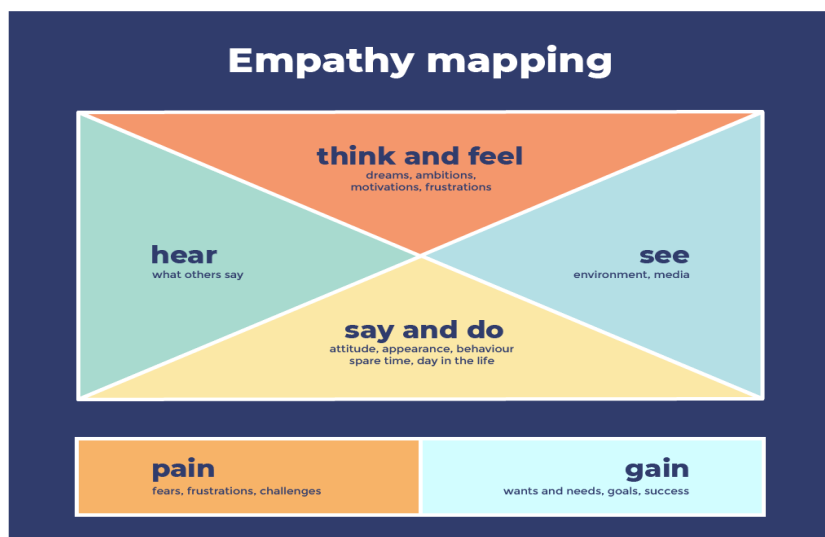


Рисунок 2.2 – Шаблон-приклад Empathy map

Empathy Map - це інструмент, який допомагає розробникам краще зрозуміти та співчувати користувачам. Він дозволяє отримати глибше розуміння їхніх потреб, бажань, мотивацій та емоцій. За допомогою Empathy Map розробники можуть отримати цінну інформацію про користувачів, їхні враження, реакції та переживання. Це сприяє розробці продукту, який краще відповідає потребам та вимогам цільової аудиторії. Заповнення Empathy Map допомагає встановити зв'язок з користувачами, сприяє емпатії та дозволяє зосередитися на створенні корисного та значущого продукту[13].

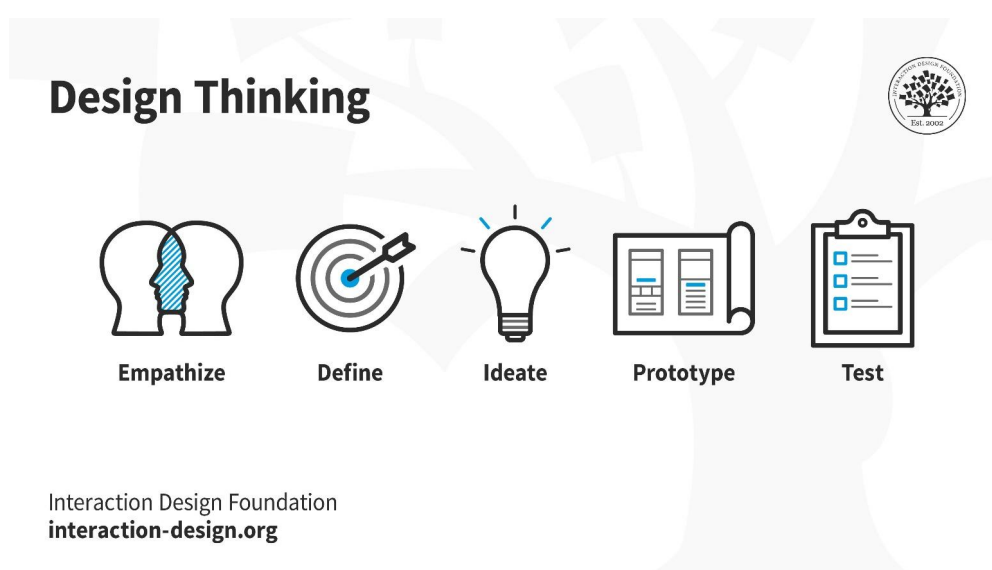


Рисунок 2.3 – Шаблон-приклад методології Design Thinking

В ході аналізу ринку і виявлення потенційної аудиторії сайту було знайдено два основних типи потенційних користувачів, чії потреби будуть враховуватись. В роботі розглянуто 2 типи потенційних клієнтів:

- звичайний покупець;
- покупець з фізичними обмеженнями зору;

Перший тип користувачів – це найбільш поширена ланка. Це звичайні користувачі, які хочуть придбати музичний інструмент.

Другий тип користувачів – це особи з фізичними вадами зору, які не можуть повноцінно користуватись звичайним UI (User Interface) і потребують додаткових інструментів та технологій для комфортного використання веб-ресурсів. На (рис 2.4, 2.5) зображено карти емпатії для двох основних типів користувачів.

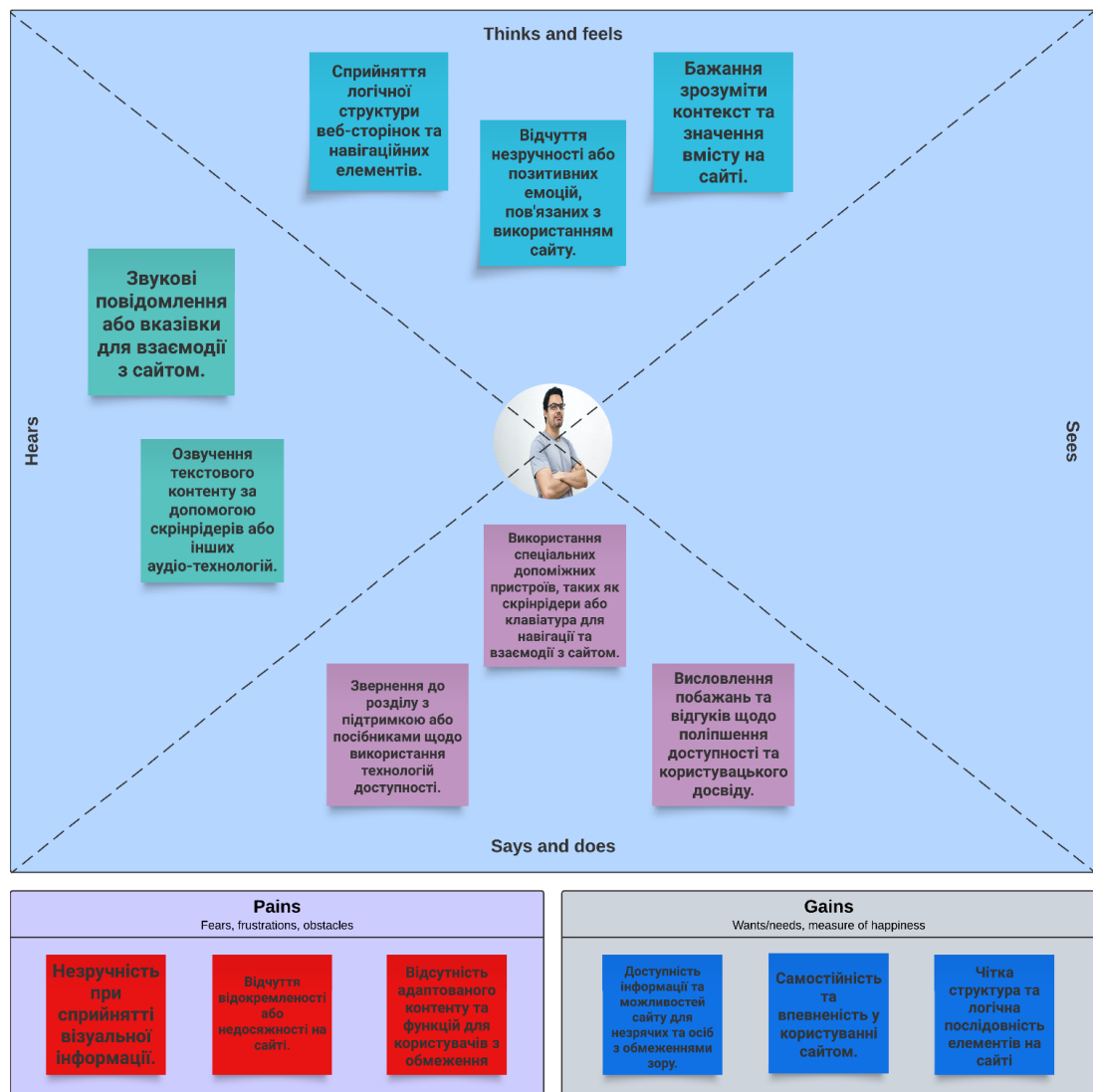


Рисунок 2.4 – Empathy map для першого типу користувачів



"Що бачить":

- Нічого

"Що чує":

- Звукові повідомлення або вказівки для взаємодії з сайтом.
- Озвучення текстового контенту за допомогою скрінрідерів або інших аудіо-технологій.

"Про що думає і що відчуває":

- Сприйняття логічної структури веб-сторінок та навігаційних елементів.
- Відчуття незручності або позитивних емоцій, пов'язаних з використанням сайту.
- Бажання зрозуміти контекст та значення вмісту на сайті.

"Що говорить і робить":

- Використання спеціальних допоміжних пристроїв, таких як скрінрідери або клавіатура для навігації та взаємодії з сайтом.
- Звернення до розділу з підтримкою або посібниками щодо використання технологій доступності.
- Висловлення побажань та відгуків щодо поліпшення доступності та користувацького досвіду.

Цінності:

- Доступність інформації та можливостей сайту для незрячих та осіб з обмеженнями зору.
- Самостійність та впевненість у користуванні сайтом.
- Чітка структура та логічна послідовність елементів на сайті.

Болі:

- Незручність при сприйнятті візуальної інформації.
  - Відчуття відокремленості або недосяжності на сайті.
  - Відсутність адаптованого контенту та функцій для користувачів з обмеження
- [14].



Рисунок 2.5 – Empathy map другого типу користувачів

"Що бачить":

- Веб-сторінки, включаючи текст, зображення, відео та графіку.
- Меню, кнопки та інші елементи навігації.
- Оформлення та дизайн сайту, включаючи кольори та шрифти.

"Що чує":

- Звукові ефекти, музика або аудіозаписи, які відтворюються на сайті.
- Аудіо-повідомлення або коментарі, які супроводжують взаємодію з елементами сайту.

"Про що думає і що відчуває":

- Враження від візуального оформлення та дизайну сайту.
- Розуміння контенту, його цінності та релевантності для користувача.
- Відчуття задоволення, цікавості або роздратування під час використання сайту.

"Що говорить і робить":

- Взаємодія з елементами сайту, які включають натискання кнопок, заповнення форм, переходи по посиланнях тощо.
- Звернення до функцій пошуку, фільтрування або сортування контенту.
- Перегляд та взаємодія зі сторінками, включаючи прокручування, читання, відповіді на коментарі тощо.

Цінності:

- Легкість використання та навігації по сайту.
- Зручність інтерфейсу та доступ до необхідної інформації.
- Позитивний користувацький досвід та задоволення від використання сайту.

Болі:

- Складнощі при пошуку необхідної інформації або здійсненні бажаних дій.
- Повільна швидкість завантаження сторінок або відгуків на дії користувача.
- Незручності в оформленні та розміщенні елементів, що ускладнюють взаємодію з сайтом [14].

Аналіз користувацького досвіду показав, що для успішної розробки веб-сайту з точки зору доступності для користувачів з обмеженнями зору необхідно враховувати їхні потреби та вимоги. Розробка такого сайту потребує уважного дослідження існуючих аналогів, вибору відповідних мов та технологій програмування, розробки зручного та сучасного дизайну та розрахунку наступних чинників: розуміння та врахування методик розрахунку, розробка максимально гнучкої архітектури сайту та проведення тестування продукту.

Крім того, для звичайних користувачів сайту також потрібно враховувати їхні потреби та наміри. Важливо забезпечити зручний та інтуїтивно зрозумілий інтерфейс, який дасть їм можливість легко взаємодіяти з сайтом, знаходити необхідну інформацію та виконувати бажані дії. Врахування їхніх цінностей та більш допоможе створити задоволення від використання сайту та позитивний користувацький досвід.

### **2.3 Структура баз даних**

З розвитком веб-технологій та збільшенням складності веб-проектів, належне проектування та організація баз даних стають ключовими аспектами успіху проекту.

Сучасні веб-проекти вимагають масштабованих та ефективних баз даних, які забезпечують швидкий доступ до інформації та зберігають дані в структурованому форматі. Основні складові структури баз даних включають таблиці, поля та відношення між ними. Це дозволяє нам організувати дані в логічну структуру та забезпечувати ефективний доступ до них.

Порівнюючи декілька систем управління базами даних (СУБД), можна визначити їх особливості та переваги.

MySQL: Це одна з найпопулярніших відкритих СУБД, відома своєю швидкодією та надійністю. MySQL є легким у використанні та має велику спільноту користувачів, що забезпечує підтримку та поширення знань [15].

PostgreSQL: Ця СУБД також відкрита, проте має більш розширені можливості порівняно з MySQL. Вона підтримує додаткові функції, такі як управління транзакціями, розширені можливості запитів та підтримку географічних даних [16].

Oracle: Це комерційна СУБД, яка відома своєю потужністю та масштабованістю. Oracle підтримує великі обсяги даних та має багатий набір функцій, таких як аналітика, реплікація даних та безпека [17].

Microsoft SQL Server: Це комерційна СУБД, яка інтегрована з екосистемою Microsoft. Вона підтримує різні типи даних, включаючи структуровані та неструктуровані дані. SQL Server має добру інтеграцію з іншими продуктами Microsoft та надає зручний інтерфейс управління [18].

MongoDB: Це документоорієнтована NoSQL СУБД, яка підтримує горизонтальне масштабування та роботу з недокументованими даними. MongoDB дозволяє гнучко зберігати дані та швидко виконувати запити [19].

У кожної СУБД є свої особливості та варіанти використання. Вибір залежить від конкретних потреб проекту та вимог до надійності, масштабованості, швидкодії та функціональності.

Для реалізації даного проекту була використана Система Управління Базами Даних (СУБД) MySQL. База даних відіграє важливу роль у зберіганні, організації та управлінні великими обсягами даних.

Однією з головних функцій бази даних є забезпечення постійного доступу до інформації. Вона зберігає дані в структурованому форматі, дозволяючи здійснювати ефективні запити та операції з даними. База даних MySQL, зокрема, пропонує широкі можливості управління даними, включаючи створення, зміну та видалення таблиць, індексів, засоби для резервного копіювання та відновлення даних.

Ще однією важливою функцією бази даних є забезпечення безпеки даних. Вона може здійснювати аутентифікацію та авторизацію користувачів, контролювати рівні доступу до різних частин даних і забезпечувати конфіденційність і цілісність інформації.

Також база даних забезпечує можливість здійснювати швидкий пошук та фільтрацію даних, що є важливим для оптимальної роботи веб-проекту. Запити до бази даних дозволяють отримувати необхідну інформацію та виконувати різні операції з даними, що сприяє ефективній роботі веб-додатку.

Усі ці функції бази даних сприяють покращенню продуктивності, надійності та швидкодії веб-проекту. Використання MySQL у проекті дозволяє зберігати, керувати та отримувати доступ до даних зручним та ефективним способом, сприяючи успішній реалізації завдань та досягненню поставлених цілей.

Для роботи з базою даних використовується середовище MySQL Workbench 8.0 (рис. 2.6).

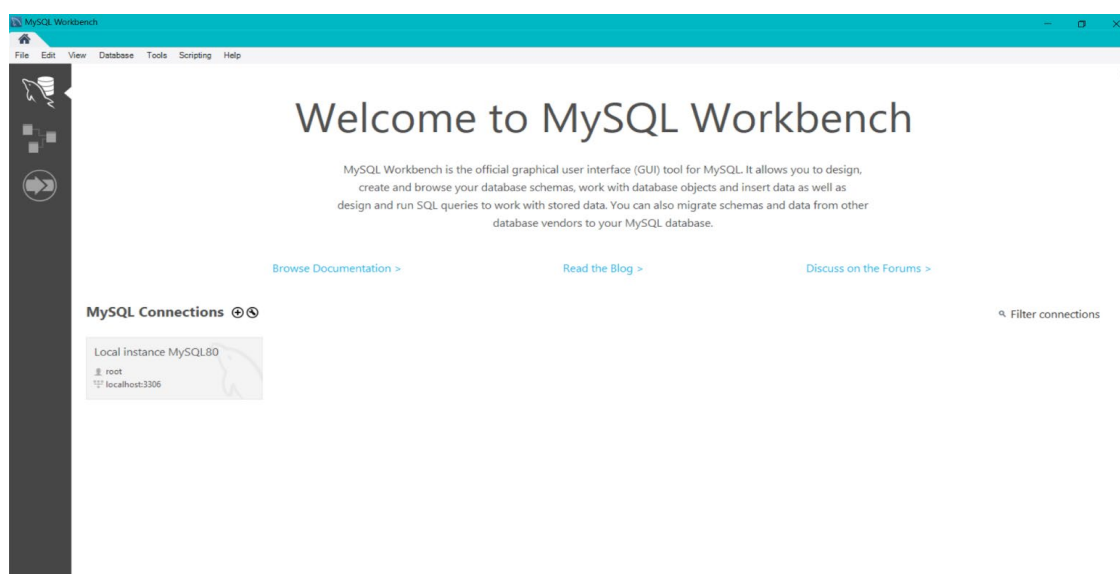


Рисунок 2.6 – Dashboard застосунку MySQL Workbench

MySQL Workbench - це інструмент для адміністрування баз даних MySQL, який має декілька переваг, що сприяють зручності та продуктивності роботи.

По-перше, інтерфейс MySQL Workbench має інтуїтивно зрозумілу та добре організовану структуру, що спрощує навігацію та взаємодію з різними компонентами програми. Він надає зручний доступ до основних функцій, таких як створення та редагування таблиць, запитів SQL, резервне копіювання та відновлення даних.

По-друге, MySQL Workbench надає широкі можливості для візуального проектування баз даних. Він має інструменти для створення схем баз даних, визначення зв'язків між таблицями, налаштування індексів та обмежень. Це дозволяє розробникам візуально представляти структуру даних та зручно налаштовувати її параметри.

По-третє, MySQL Workbench забезпечує потужний інструментарій для написання та виконання запитів SQL. Він підтримує автодоповнення коду, синтаксичне виділення, перевірку помилок та інші функції, що полегшують розробку та налагодження запитів.

В цілому, інтерфейс MySQL Workbench є потужним та зручним інструментом для роботи з базами даних MySQL. Він спрощує процес розробки, адміністрування та оптимізації баз даних, сприяючи покращенню продуктивності та ефективності роботи розробників. Інтерфейс робочого простіру є простим і зрозумілим, без зайвих елементів (рис. 2.7).

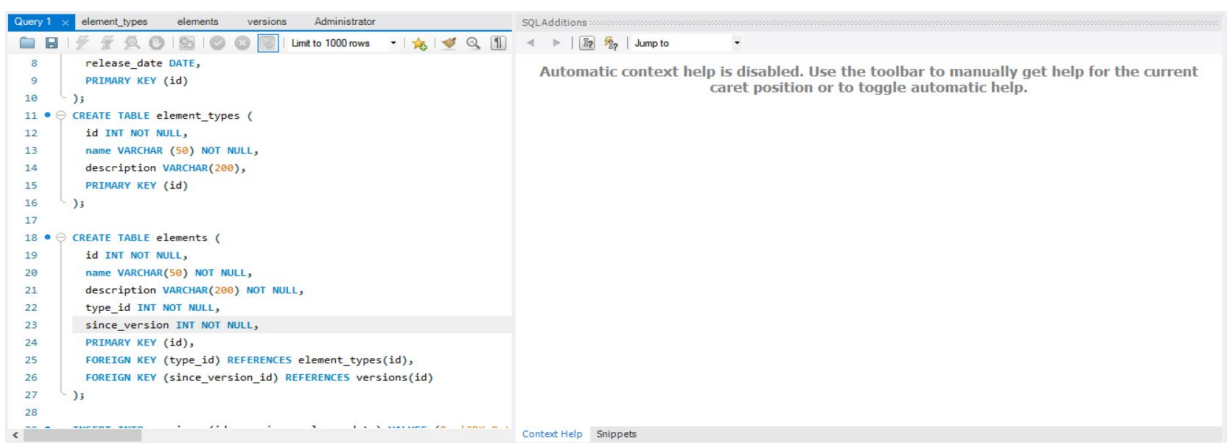


Рисунок 2.7 – Інтерфейс робочого середовища

Загальна структура бази даних для коду нижче має такі таблиці і відповідні зв'язки:

Таблиця "users" (Користувачі):

- id: INT (первинний ключ, автоінкремент), розмір не вказаний
- name: VARCHAR(255) NOT NULL, розмір поля 255 символів
- email: VARCHAR(255) NOT NULL, розмір поля 255 символів
- password: VARCHAR(255) NOT NULL, розмір поля 255 символів

Таблиця "products" (Товари):

- id: INT (первинний ключ, автоінкремент), розмір не вказаний
- name: VARCHAR(255) NOT NULL, розмір поля 255 символів
- description: TEXT, тип поля текстовий, розмір не обмежений
- price: DECIMAL(10, 2) NOT NULL, числовий тип з розмірністю 10, 2

десяткових знаків

- image: VARCHAR(255), розмір поля 255 символів

Таблиця "orders" (Замовлення):

- id: INT (первинний ключ, автоінкремент), розмір не вказаний
- user\_id: INT NOT NULL, розмір не вказаний
- order\_date: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP, тип поля дата-час зі значенням за замовчуванням поточний час

дата-час зі значенням за замовчуванням поточний час

- status: ENUM('В обробці', 'Виконано', 'Скасовано') DEFAULT 'В обробці',

перерахування зі значенням за замовчуванням "В обробці"

Таблиця "order\_details" (Деталі замовлення):

- id: INT (первинний ключ, автоінкремент), розмір не вказаний
- order\_id: INT NOT NULL, розмір не вказаний
- product\_id: INT NOT NULL, розмір не вказаний
- quantity: INT NOT NULL, розмір не вказаний

Таблиця "categories" (Категорії):

- id: INT (первинний ключ, автоінкремент), розмір не вказаний
- name: VARCHAR(255) NOT NULL, розмір поля 255 символів
- parent\_id: INT, розмір не вказаний

Таблиця "reviews":

- id: INT (первинний ключ, автоінкремент), розмір не вказаний
- product\_id: INT NOT NULL, розмір не вказаний
- user\_id: INT NOT NULL, розмір не вказаний
- rating: INT NOT NULL, розмір не вказаний
- comment: TEXT, тип поля текстовий, розмір не обмежений

Таблиця "cart" (Кошик):

- id: INT (первинний ключ, автоінкремент), розмір не вказаний
- user\_id: INT NOT NULL, розмір не вказаний
- product\_id: INT NOT NULL, розмір не вказаний
- quantity: INT NOT NULL, розмір не вказаний

Зв'язки між таблицями:

- Таблиця "orders" має зв'язок багато-до-одного з таблицею "users" через поле "user\_id".

- Таблиця "order\_details" має зв'язок багато-до-одного з таблицями "orders" та "products" через поля "order\_id" і "product\_id" відповідно.

- Таблиця "categories" має зв'язок багато-до-одного сама з собою через поле "parent\_id", що вказує на батьківську категорію.

- Таблиця "reviews" має зв'язки багато-до-одного з таблицями "products" та "users" через поля "product\_id" і "user\_id" відповідно.

- Таблиця "cart" має зв'язки багато-до-одного з таблицями "users" та "products" через поля "user\_id" і "product\_id" відповідно.

Ця структура даних дозволяє зберігати та організовувати інформацію про користувачів, товари, замовлення, категорії, відгуки та кошик.

## 2.4 Проектування User Interface

User Interface (UI) - це взаємодіючий шар між користувачем і комп'ютерною системою, що дозволяє користувачу взаємодіяти з програмним забезпеченням чи пристроєм. Інтерфейс користувача включає у себе всі елементи, з якими користувач може взаємодіяти, такі як кнопки, меню, поля введення, графічні елементи і т.д.



Головна мета інтерфейсу користувача - забезпечити зручну та інтуїтивно зрозумілу взаємодію між користувачем і системою. Він повинен бути приємним у використанні, ефективним та логічно організованим. Гарний інтерфейс дозволяє користувачам легко виконувати завдання, знаходити потрібну інформацію і контролювати систему без зайвих зусиль.

Основні принципи доброго інтерфейсу користувача включають простоту, зручність, зрозумілість, послідовність та естетичний дизайн. Він повинен враховувати потреби та очікування цільової аудиторії, а також забезпечувати доступність для користувачів з різними навичками та обмеженнями.

Для створення інтерфейсу користувача використовуються різні інструменти та технології, такі як графічні редактори, фреймворки, макетні засоби та інші. Розробники стараються створювати інтерфейси, які будуть ефективними, ергономічними та приємними у використанні, спрощуючи завдання користувачів і поліпшуючи їх досвід взаємодії з технологією.

Мокап, вайрфрейм і прототип - це найбільш поширені інструменти, які використовуються в процесі розробки програмного забезпечення або веб-дизайну для візуалізації і тестування ідей та концепцій перед фактичною реалізацією проекту.

Мокап є статичним зображенням або моделлю, яка відображає вигляд і розташування елементів інтерфейсу. Він дозволяє відтворити загальну структуру та компоненти, такі як кнопки, поля вводу, меню і т. д. Мокапи зазвичай створюються за допомогою графічних редакторів або спеціальних онлайн-інструментів і використовуються для візуалізації концепцій та отримання затвердження від замовника чи команди проекту.

Вайрфрейм є більш детальною версією мокапу, відображаючи не тільки компоненти інтерфейсу, але й їхні взаємозв'язки та функціональні можливості. Вайрфрейми зазвичай мають простий макет без дизайнерських елементів, фокусуючись на структурі, навігації та розміщенні контенту. Вони допомагають визначити логіку і взаємодію елементів інтерфейсу перед подальшою розробкою.

Прототип - це інтерактивна модель, що відтворює функціональність і поведінку фактичного продукту або системи. Він може бути реалізований як простий прототип з обмеженими функціями або повноцінний функціональний прототип з усіма

основними функціями. Прототипи використовуються для тестування та збору відгуків від користувачів, виявлення проблем та вдосконалення дизайну перед фінальною реалізацією проекту.

Усі ці інструменти допомагають командам розробників та дизайнерів узгодити ідеї, відобразити концепції та протестувати функціональність перед фактичним впровадженням проекту. Вони сприяють покращенню розуміння вимог користувачів і забезпечують більш точну і ефективну реалізацію продукту.

В процесі розробки даного проекту було використано модель «вайрфрейм» для створення схематичного шаблону майбутнього сайту. На (рис. 2.8, 2.9, 2.10) зображено ці вайрфрейми.

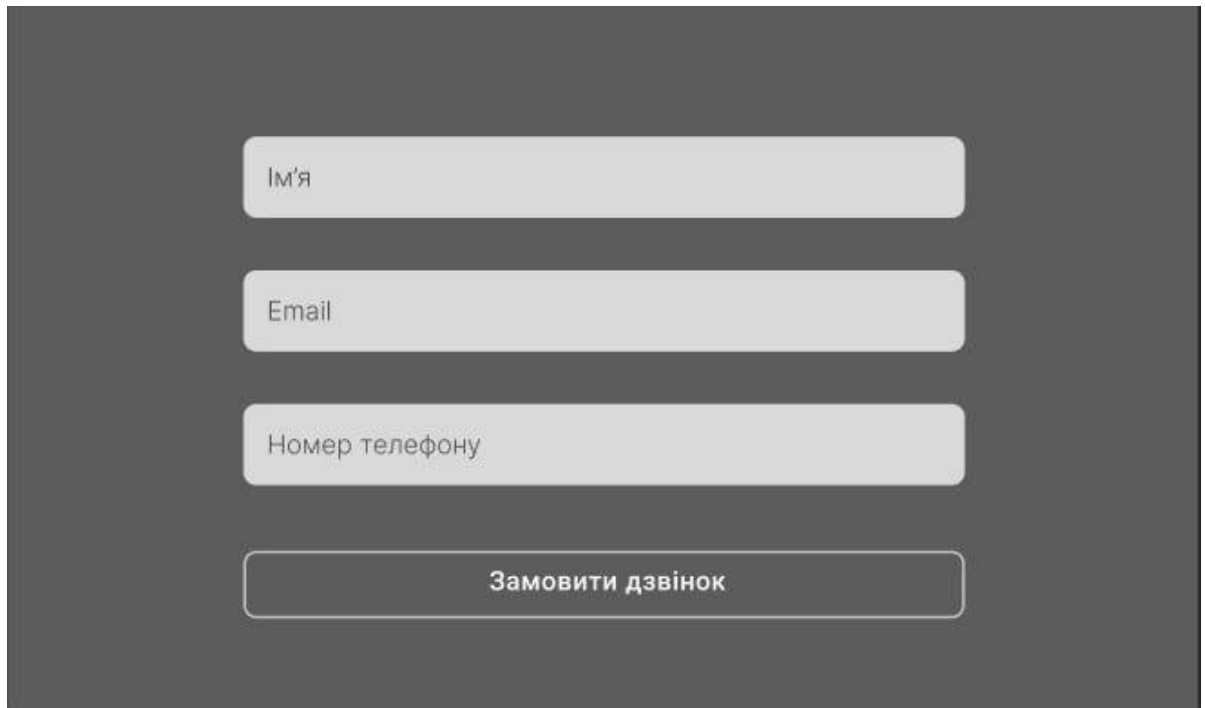
The image shows a wireframe of a login form on a dark gray background. It consists of four horizontal input fields stacked vertically. The first field is labeled 'Ім'я' (Name), the second 'Email', and the third 'Номер телефону' (Phone number). Below these fields is a button labeled 'Замовити дзвінок' (Call). All elements are rendered in a light gray color with rounded corners and a subtle shadow effect.

Рисунок 2.8 – Вайрфрейм форми авторизації

Форма авторизації на сайті є важливим елементом, який дозволяє користувачам отримати доступ до своїх особистих облікових записів або ресурсів на сайті. Форма авторизації зазвичай складається з двох основних полів: поле для введення електронної пошти або ідентифікатора користувача та поле для введення пароля. Важливо, щоб форма авторизації була простою, зрозумілою та зручною для користувачів. Вона повинна мати чітку вказівку про необхідні обов'язкові поля і

надавати помилки в разі неправильного введення даних. Також варто враховувати заходи безпеки, наприклад, обмеження кількості спроб авторизації і застосування механізмів шифрування паролів.

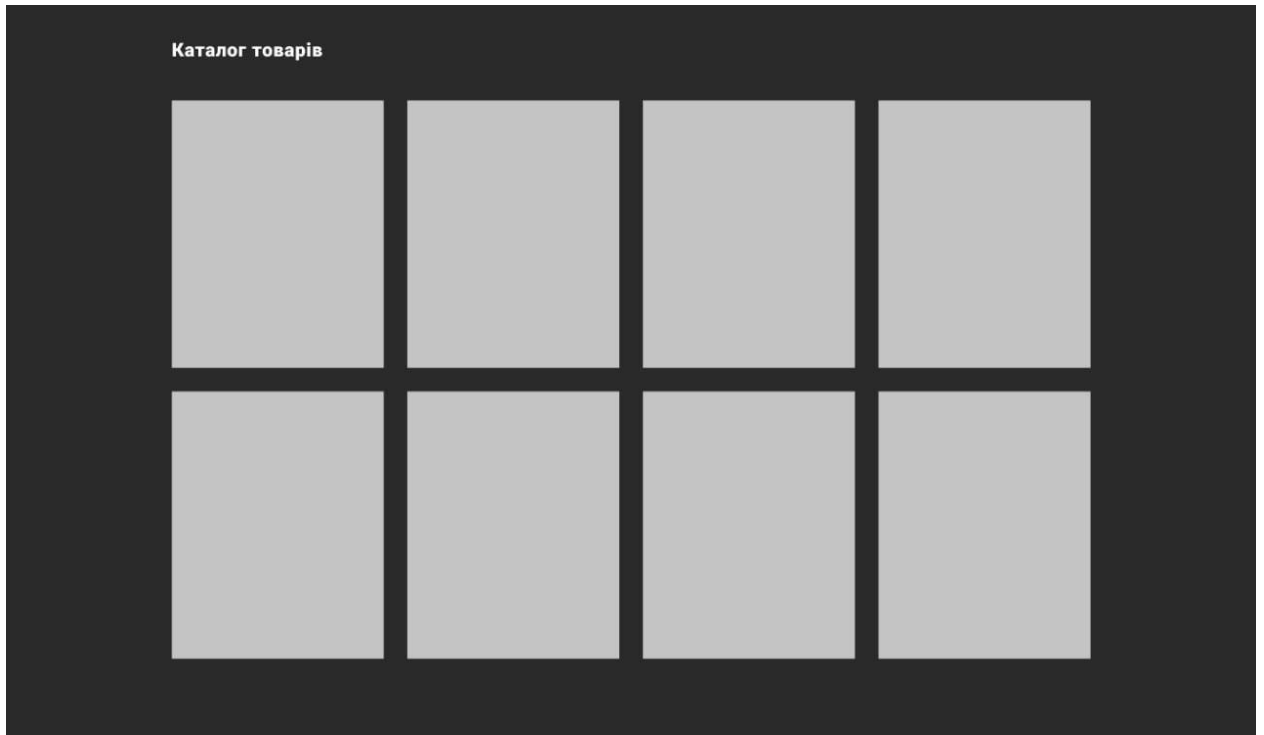


Рисунок 2.9 – Вайрфрейм каталогу товарів

Каталог товарів на сайті є центральним місцем, де користувачі можуть ознайомитися з асортиментом продукції, яку пропонує магазин. Каталог товарів зазвичай організований у вигляді списку або сітки, де кожен товар представлений окремим блоком з інформацією про назву, зображення, опис, ціну та можливість додавання до кошика або покупки.

Користувачі можуть переглядати товари за категоріями, використовуючи фільтри або пошукову функцію. Важливо, щоб каталог був логічно структурованим, зрозумілим та зручним для навігації. Товари можуть бути розташовані в підкатегоріях або мати теги, що допомагають знаходити потрібні товари швидко і ефективно.



Рисунок 2.10 – Вайрфрейм головної сторінки

Головна сторінка сайту продажу музичних інструментів є центральним місцем, яке вітає відвідувачів і надає їм огляд основного асортименту та можливостей магазину.

На головній сторінці можуть бути представлені вибрані продукти або акційні пропозиції, що привертають увагу користувачів. Зображення високої якості, відповідні назви та ціни товарів допомагають створити привабливий вигляд сторінки.

Також на головній сторінці можуть бути посилання на розділи каталогу, де користувачі можуть переглянути повний асортимент музичних інструментів, аксесуарів та інших продуктів. Навігаційне меню або панель категорій допомагають легко орієнтуватись на сайті і знайти потрібні товари.

Головна сторінка також може містити корисну інформацію про ваш магазин, наприклад, про команду, доставку, гарантію та контактні дані. Це дозволяє забезпечити довіру та впевненість покупців у нашому бренді. [20]

## 2.5 Інструменти за допомогою яких був створений сайт

Бекенд-частина проекту для магазину продажу музичних інструментів з підтримкою принципів Accessibility був розроблений з використанням таких технологій як: мова програмування Java та фреймворк Spring Boot. База даних проектувалась та створювалась за допомогою MySQL та робочого середовища MySQL Workbench. Для написання коду на Java використовувалось робоче середовище IntelliJ Idea.

Мова програмування Java: Використовується для розробки бізнес-логіки та функціональності сайту. Java є потужною та універсальною мовою програмування, яка дозволяє реалізувати складні операції та інтерактивні функції.

Фреймворк Spring Boot: Використовується для швидкої розробки веб-додатків на основі платформи Spring. Spring Boot дозволяє ефективно налаштовувати та розгортати додатки, а також надає зручні інструменти для роботи з базою даних та іншими компонентами.

База даних MySQL: Використовується для зберігання та управління даними, пов'язаними з магазином музичних інструментів. MySQL є потужною та популярною системою керування базами даних, яка забезпечує надійність та ефективність роботи з даними.

Робоче середовище MySQL Workbench: Використовується для моделювання та адміністрування бази даних MySQL. Workbench надає зручний графічний інтерфейс для створення таблиць, запитів та інших операцій з базою даних.

Основні переваги фреймворку Spring Boot:

Швидкість розробки: Spring Boot надає шаблони та автоматичну конфігурацію, що спрощує процес розробки веб-додатків. Завдяки цьому, програмісти можуть швидко створювати функціональність, не затрачаючи багато часу на налаштування.

Стислість коду: Spring Boot пропонує конвенції, які дозволяють написати менше коду для досягнення бажаної функціональності. Це сприяє зменшенню кількості написаного коду та збільшенню його зрозумілості.

Автоматична конфігурація: Spring Boot надає можливість автоматично налаштувати багато компонентів і бібліотек, що значно спрощує управління залежностями та конфігурацією додатку.

Вбудований сервер додатків: Spring Boot має вбудований сервер додатків, такий як Tomcat або Jetty, що дозволяє запускати додаток без необхідності окремої настройки сервера.

Велика спільнота: Spring Boot має велику та активну спільноту розробників, яка надає підтримку, документацію та різноманітні розширення. Це робить фреймворк надійним та стабільним рішенням для розробки веб-додатків.

Інтеграція з іншими технологіями: Spring Boot добре інтегрується з іншими технологіями та бібліотеками, що дозволяє розширювати його функціональність та використовувати різні інструменти для розробки.

## **Висновки до розділу 2**

Результатом розділу є написання та складання різних додаткових знарядь які допомагають при розробці. Зокрема для того щоб краще зрозуміти потреби клієнта та користувача були складені декілька карт емпатії. Було написано User Story для того щоб краще розуміти що потрібно розробити. Описано структуру бази даних та вказано яка база використовувалася. Важливим та об'ємним пунктом було складання wireframe. Завдяки цьому в майбутньому, при написанні UI було витрачено значно менше часу. В кінці розділу описано які інструменти було використано під час розробки проекту.

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ СТРУКТУРИ ТА ФУНКЦІОНАЛУ ВЕБСАЙТУ

### 3.1 Розробка backend сайту

Backend частина системи управління контентом для сайту продажу музичних інструментів відповідає за обробку та керування даними, логікою бізнес-процесів і комунікацією з базою даних. Вона забезпечує підтримку функцій, необхідних для ефективного управління контентом сайту.

Backend реалізує серверну сторону додатку та забезпечує взаємодію з фронтендом, який відповідає за відображення даних та взаємодію з користувачем. Основні компоненти backend частини системи управління контентом включають:

**API (Application Programming Interface):** Він надає спосіб взаємодії між фронтендом та backendом, дозволяючи передавати дані та виконувати запити до сервера. API визначає доступні операції та формати даних для комунікації з системою.

**Управління базою даних:** Backend використовує базу даних для зберігання та організації контенту, такого як інформація про продукти, замовлення, користувачі тощо. Це дозволяє ефективно керувати даними, здійснювати пошук та фільтрацію, зберігати історію змін тощо.

**Бізнес-логіка:** Backend виконує логіку бізнес-процесів, пов'язаних з управлінням контентом. Він обробляє запити користувачів, перевіряє права доступу, проводить валідацію даних, розраховує ціни, керує запасами тощо. Це забезпечує правильне та безперебійне функціонування сайту.

**Автентифікація та авторизація:** Backend відповідає за ідентифікацію та перевірку прав доступу користувачів. Він забезпечує можливість реєстрації та входу в систему, контролює доступ до певних функцій та ресурсів, забезпечує захист даних користувачів.

**Інтеграція з платіжними системами:** Backend може включати інтеграцію з платіжними системами, щоб забезпечити можливість онлайн-оплати за товари та

послуги. Він обробляє запити на оплату, здійснює перевірку платежів, взаємодіє з платіжними шлюзами та забезпечує безпеку транзакцій.

Backend частина системи управління контентом є важливою складовою ефективного та безперебійного функціонування сайту. Вона дозволяє керувати даними, забезпечувати безпеку, здійснювати функції бізнес-логіки та взаємодію з іншими системами. Для спрощення взаємодії програмного коду і зв'язку з базою даних використовувався інтегрований у середовище розробки інструмент DB Navigator. На (рис.3.1) зображено вікно створення зв'язку (connection) для бази даних.

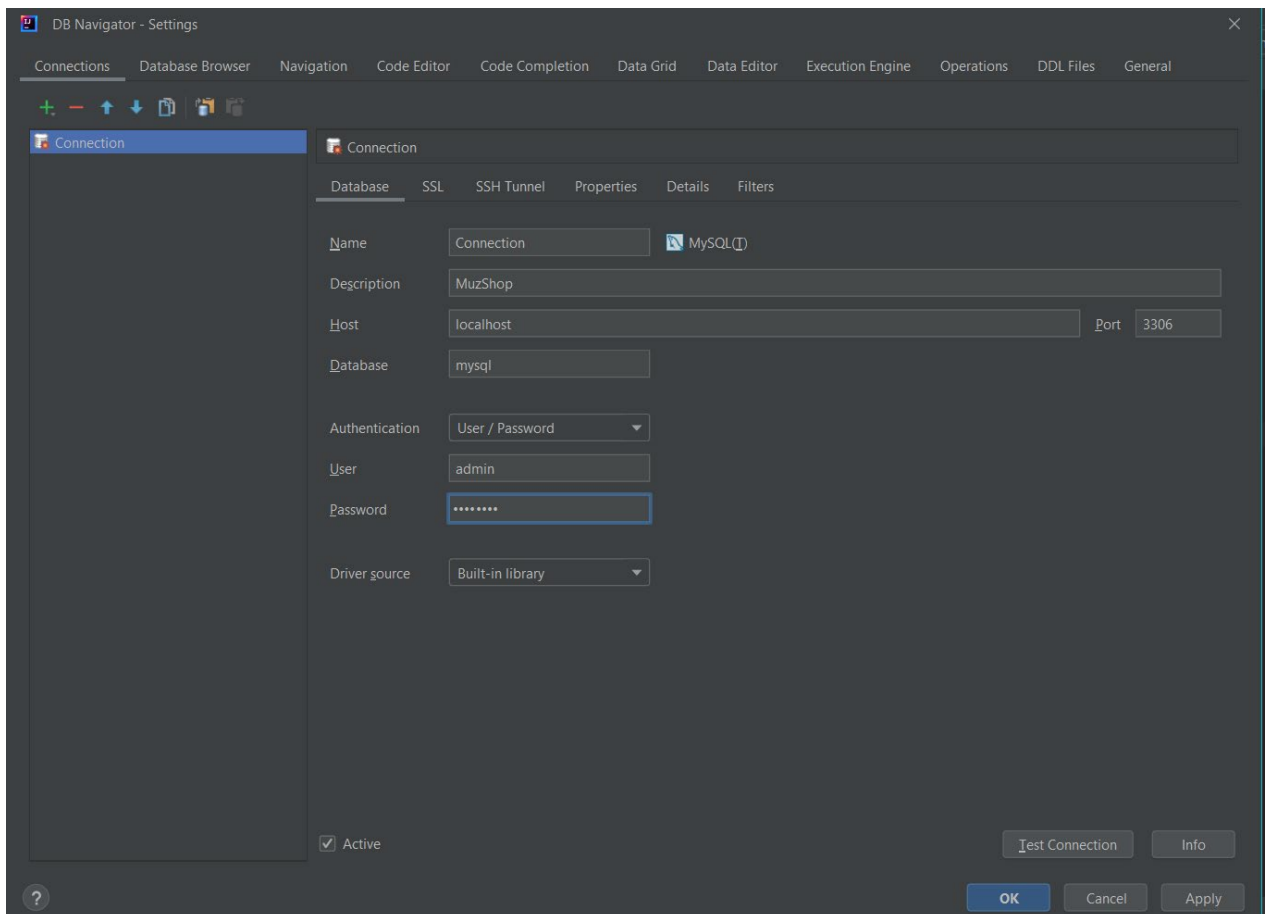


Рисунок 3.1 – DB Navigator

У програмному середовищі MySQL Workbench було створено таблиці та списки для бази даних. На (рис.3.2) зображена діаграма зв'язків таблиць у базі даних.



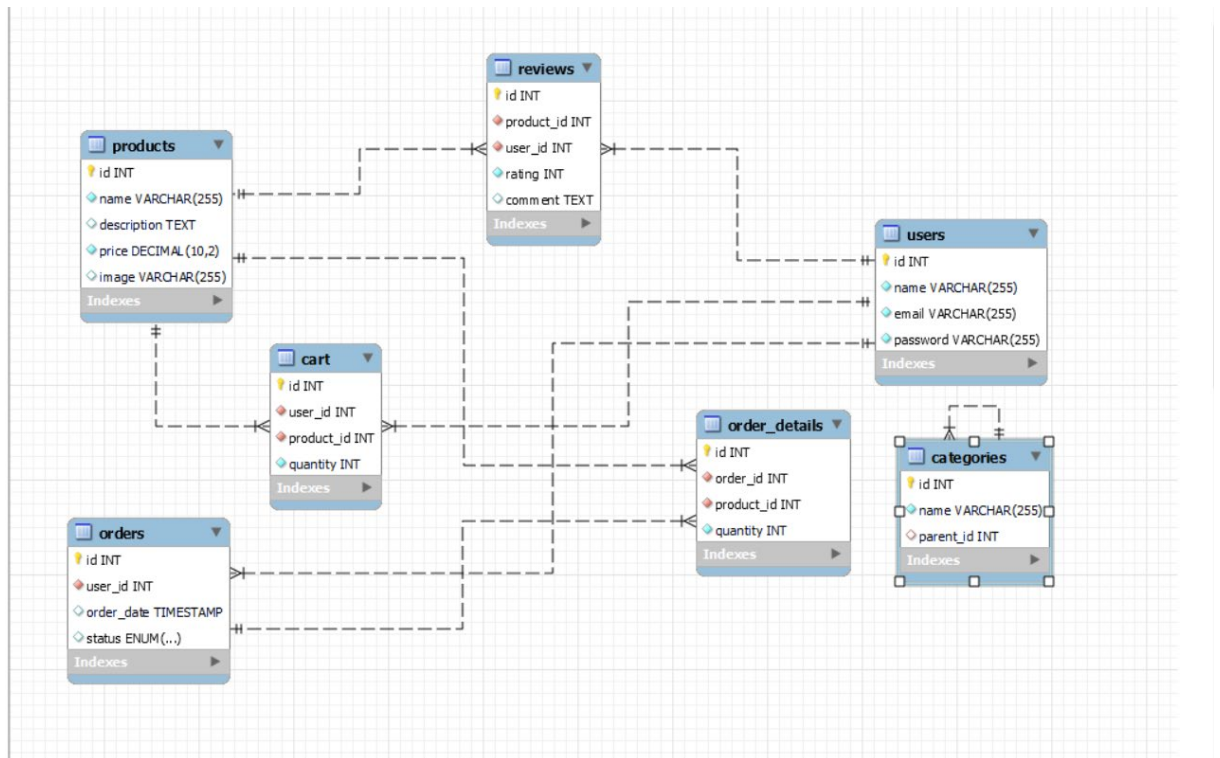


Рисунок 3.2 – Діаграма зв'язків бази даних

Створена база даних для сайту продажу містить кілька таблиць, які використовуються для зберігання різних типів інформації. Таблиця "Користувачі" містить дані про зареєстрованих користувачів, включаючи їх ідентифікатор, ім'я, електронну пошту та пароль. Таблиця "Товари" містить інформацію про доступні продукти, такі як назва, опис, ціна та зображення. Таблиця "Замовлення" використовується для зберігання деталей про замовлення, включаючи ідентифікатор замовлення, ідентифікатор користувача, дату замовлення та статус. Таблиця "Деталі замовлення" зберігає інформацію про кожен товар у замовленні, включаючи ідентифікатор замовлення, ідентифікатор товару та кількість. Таблиця "Категорії" використовується для категоризації товарів і має поля для назви категорії та батьківського ідентифікатора. Таблиця "Відгуки" зберігає відгуки користувачів про певний товар, включаючи оцінку та коментар. Нарешті, таблиця "Кошик" містить інформацію про товари, які користувач додав до свого кошика перед оформленням замовлення.

Код форми реєстрації:

```

@Controller
public class RegistrationController {
    @Autowired
    private UserService userService;
    @GetMapping("/registration")
    public String showRegistrationForm(Model model) {
        model.addAttribute("user", new User());
        return "registration";
    }
    @PostMapping("/registration")
    public String registerUser(@ModelAttribute("user") @Valid User user, BindingResult
bindingResult) {
        if (bindingResult.hasErrors()) {
            return "registration";
        }
        if (userService.isUserExists(user.getUsername())) {
            bindingResult.rejectValue("username", "error.user", "Це ім'я користувача вже
використовується");
            return "registration";
        }
        userService.saveUser(user);
        return "redirect:/login";
    }
}

```

У данному фрагменті коду реалізовано `RegistrationController` відповідає за обробку запитів пов'язаних з реєстрацією користувачів. При GET-запиті на шлях `/registration` метод `showRegistrationForm` повертає відповідний шаблон для відображення форми реєстрації, передаючи порожній об'єкт `User` у модель.

При POST-запиті на шлях `/registration` метод `registerUser` отримує дані, введені користувачем у форму реєстрації, як об'єкт `User`. Застосовуються правила валідації, які описані у моделі `User` за допомогою анотацій `@Valid`. Якщо дані не валідні, повертається та відображається той самий шаблон форми реєстрації з помітками про помилки. Якщо користувач з таким ім'ям вже існує, також повертається той самий шаблон з відповідною помилкою. Якщо все пройшло успішно, користувач зберігається за допомогою `userService`, і користувач перенаправляється на сторінку входу `/login`.

## Код форми авторизації:

```

@Controller
public class AuthController {
    @Autowired
    private UserService userService;
    @GetMapping("/login")
    public String showLoginForm() {
        return "login";
    }
    @PostMapping("/login")
    public String login(@RequestParam("username") String username,
        @RequestParam("password") String password,
        Model model) {
        // Перевірка введених даних авторизації
        if (userService.authenticate(username, password)) {
            // Успішна авторизація
            return "redirect:/dashboard";
        } else {
            // Невірні дані авторизації
            model.addAttribute("error", "Invalid username or password");
            return "login";
        }
    }
}

```

У данному фрагменті коду реалізовано `AuthController`, який відповідає за обробку запитів пов'язаних з авторизацією. У методі `showLoginForm` ми просто відображаємо шаблон для форми авторизації. У методі `login` ми отримуємо дані з форми (ім'я користувача та пароль), перевіряємо їх правильність за допомогою служби `UserService` і, в залежності від результату, перенаправляємо користувача на сторінку управління (`/dashboard`) або відображаємо повідомлення про невірні дані авторизації на сторінці авторизації.

## Код форми каталогу:

```

@Controller
public class ProductController {
    @Autowired
    private ProductService productService;
    @GetMapping("/products")

```

```

public String showProductCatalog(Model model) {
    List<Product> products = productService.getAllProducts();
    model.addAttribute("products", products);
    return "productCatalog";
}
@GetMapping("/products/{id}")
public String showProductDetails(@PathVariable("id") Long productId, Model model) {
    Product product = productService.getProductById(productId);
    model.addAttribute("product", product);
    return "productDetails";
}
}
}

```

У даному фрагменті коду реалізовано контролер `ProductController`, який відповідає за обробку запитів пов'язаних з каталогом товарів. У методі `showProductCatalog` ми отримуємо всі товари з допомогою сервісу `ProductService` і передаємо їх на сторінку `productCatalog` за допомогою моделі. У методі `showProductDetails` ми отримуємо ідентифікатор товару з URL, отримуємо деталі цього товару з сервісу `ProductService` і передаємо його на сторінку `productDetails`.

Контроллер використовує відповідні шаблони `productCatalog.html` та `productDetails.html` для відображення каталогу товарів та деталей товару відповідно.

Код кошику:

```

@Controller
public class CartController {
    @Autowired
    private CartService cartService;
    @GetMapping("/cart")
    public String showCart(Model model) {
        Cart cart = cartService.getCart();
        model.addAttribute("cart", cart);
        return "cart";
    }
    @PostMapping("/cart/add")
    public String addToCart(@RequestParam("productId") Long productId,
    @RequestParam("quantity") int quantity) {
        cartService.addProductToCart(productId, quantity);
        return "redirect:/cart";
    }
}

```

```

@PostMapping("/cart/remove")
public String removeFromCart(@RequestParam("productId") Long productId) {
    cartService.removeProductFromCart(productId);
    return "redirect:/cart";
}
@PostMapping("/cart/update")
public String updateCart(@RequestParam("productId") Long productId,
@RequestParam("quantity") int quantity) {
    cartService.updateProductQuantity(productId, quantity);
    return "redirect:/cart";
}
}
}

```

У даному фрагменті коду реалізовано `CartController`, який відповідає за обробку запитів пов'язаних з кошиком товарів. У методі `showCart` ми отримуємо кошик з допомогою сервісу `CartService` і передаємо його на сторінку `cart` за допомогою моделі. У методі `addToCart` ми отримуємо ідентифікатор товару та кількість, які користувач хоче додати до кошика, і передаємо ці дані до сервісу `CartService` для додавання товару до кошика. Методи `removeFromCart` та `updateCart` відповідають за видалення товару з кошика та оновлення кількості товару в кошику відповідно.

Для звукового супроводу текстових записів, у відповідності з принципом Доступності для людей з обмеженнями зору, на сайті `MuzShop`, використовується `HTML5 Audio API` разом з `Java` та фреймворком `Spring Boot`. Нижче наведено код, який відтворює звуковий файл при завантаженні сторінки:

Код `JavaAudio`:

```

@Controller
public class HomeController {
    @GetMapping("/")
    public String home(Model model) {
        // Отримуємо посилання на звуковий файл
        String audioUrl = "/path/to/audio/file.mp3";
        model.addAttribute("audioUrl", audioUrl);
        return "home";
    }
}

```

Шаблон `HTML Api` може виглядати наступним чином:

```
<!DOCTYPE html>
<html>
<head>
  <title>MuzShop</title>
</head>
<body>
  <h1>Ласкаво просимо до MuzShop!</h1>
  <audio autoplay>
    <source th:src="{audioUrl}" type="audio/mpeg">
  </audio>
  <p>Home page</p>
</body>
</html>
```

В цьому кодi ми отримуємо посилання на звуковий файл, яке може бути збережено в базi даних. У контролерi HomeController ми додаємо це посилання до моделi та передаємо його на сторiнку home.html. У шаблонi ми використовуємо HTML5 Audio елемент для вiдтворення звуку, вказуючи джерело звукового файлу за допомогою `{audioUrl}`

### **Висновки до роздiлу 3**

У цьому роздiлi було описано структуру розробки проекту, приведено ключовi фрагменти коду та продемонстровано реалiзацiю основних Backend-функцiй сайту. Також наведена структура бази даних, створена її модель.

Реалiзованi функцiї реєстрацiї нових користувачiв на сайтi та авторизацiї вже зареєстрованих, розроблено функцiонал каталогу товарiв, який дозволяє додавати та видаляти новi найменування з адмiн-панелi сайту, а також переглядати асортимент товарiв та робити замовлення для користувачiв.

Наведено код реалiзацiї принципу Доступностi за допомогою HTML5 Audio API створено функцiю звукового супроводу для тексту, яка значно полегшує процес використання сайту людьми з обмеженнями зору. Також реалiзована функцiя високої контрастностi. За допомогою бiблiотеки Java Sound API створена iмплементацiя

збереження звукових файлів у базі даних та виклику по особистому ідентифікатору для зв'язку з бібліотекою HTML.

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи були розглянуті та вирішені такі задачі проектування:

Пошук та аналіз існуючих сайтів з аналогічними технологіями: Виконано пошук та аналіз різноманітних веб-сайтів, що підтримують технології Доступності. Були визначені сильні та слабкі сторони кожного з сайтів, враховані їхня функціональність та дизайн для подальшого використання найкращих практик у розробці веб-сайту MuzShop.

Вибір мови програмування, технологій та інших суміжних програм: Після аналізу різних альтернатив було прийнято рішення використовувати мову програмування Java та фреймворк Spring Boot для реалізації backend частини веб-сайту. При створенні стабільної системи менеджменту даних була використана база даних MySQL для зберігання та керування інформацією про товари та замовлення.

Пошук даних про технології Accessibility: Виконано пошук та аналіз різних технологій та практик Accessibility, спрямованих на забезпечення доступності веб-сайтів для користувачів з обмеженими можливостями. Були враховані рекомендації WCAG (Web Content Accessibility Guidelines) для створення доступного веб-сайту MuzShop.

Розробка архітектури сайту, відповідаючи принципам доступності: При проектуванні архітектури сайту MuzShop були враховані принципи доступності, такі як зручність навігації, використання чіткого та зрозумілого контенту, альтернативні текстові описи для зображень, технологія звукового супроводу для текстових позначень та інші.

Проведення тестування продукту: Було проведено тестування функціональності та коректності роботи веб-сайту MuzShop. Різні сценарії були протестовані, включаючи реєстрацію користувача, пошук товарів, складання замовлення та оплату. Були виявлені та виправлені помилки, що гарантує належну роботу веб-сайту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ



1. Офіційний сайт Всесвітньої організації охорони здоров'я в Україні (ВООЗ)  
URL: <https://geneva.mfa.gov.ua/posolstvo/2612-who> (дата звернення: 21.02.2023)
2. “ Веб-доступність у 2020 році: чим керуватись? ” URL:  
<https://internetdevels.ua/blog/website-accessibility-in-2020> (дата звернення:  
23.02.2023)
3. “ How to Do Website Accessibility the right way ” URL:  
<https://adasitecompliance.com/how-do-website-accessibility-right-way/> (дата  
звернення: 25.02.2023)
4. “ Which Type of Navigation is Best to Improve the Accessibility of Your Website  
”URL: <https://accessibilityspark.com/best-to-improve-the-accessibility-of-your-website/>  
(дата звернення: 28.02.2023)
5. AbleGamers Charity Website URL:  
[https://ablegamers.org/?gclid=CjwKCAjwyqWkBhBMEiwAp2yUFlqilsz1H914nJoXH  
snzgzJGfaXdB2qjhqKjLF0fsx0nokFHwWqZshoCNgwQAvD\\_BwE](https://ablegamers.org/?gclid=CjwKCAjwyqWkBhBMEiwAp2yUFlqilsz1H914nJoXHsnzgzJGfaXdB2qjhqKjLF0fsx0nokFHwWqZshoCNgwQAvD_BwE) (дата звернення:  
29.02.2023)
6. Accessible Arts and Media URL: <https://www.aamedia.org.uk/> (дата звернення:  
04.03.2023)
7. Apple - Accessibility Presentation Page URL:  
<https://www.apple.com/accessibility/> (дата звернення: 16.03.2023)
8. Web Accessibility Initiative - Make the Web Accessible URL:  
<https://www.w3.org/WAI/> (дата звернення: 24.03.2023)
9. Що таке User Story і як її писати URL:  
<https://training.qatestlab.com/blog/technical-articles/user-story/> (дата звернення:  
01.04.2023)
10. How to User Story URL: <https://vitalitychicago.com/blog/how-to-user-story/>  
(дата звернення: 03.04.2023)
11. Як зрозуміти що важливо вашим клієнтам? URL:  
<https://leosvit.com/art/empathy-map> (дата звернення: 05.04.2023)

12. Empathy Mapping: The First Step to Design Thinking URL: <https://www.nngroup.com/articles/empathy-mapping/#:~:text=Definition%3A%20An%20empathy%20map%20is,empathy%20mapping%20and%20its%20uses>. (дата звернення: 11.04.2023)
13. What is Design Thinking? URL: <https://www.interaction-design.org/literature/topics/design-thinking> (дата звернення: 13.04.2023)
14. Lucid App URL: <https://lucid.app/documents> (дата звернення: 19.04.2023)
15. MySQL Workbench URL: <https://www.mysql.com/> (дата звернення: 24.04.2023)
16. PostgreSQL: The world's most advanced open source system URL: <https://www.postgresql.org/> (дата звернення: 26.04.2023)
17. Oracle Ukraine - Cloud Application URL: <https://www.oracle.com/ua/> (дата звернення: 02.05.2023)
18. SQL Server Downloads URL: <https://www.microsoft.com/en-us/sql-server/sql-server-downloads> (дата звернення: 05.05.2023)
19. MongoDB: Build the next big thing URL: <https://www.mongodb.com/> (дата звернення: 10.05.2023)
20. Figma: The collaborative interface URL: <https://www.figma.com/> (дата звернення: 25.05.2023)