

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»
В.о. завідувача кафедри
Ігор ШЕЛЕХОВ

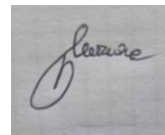
(підпис)

грудня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна система прослуховування аудіо-книг для людей з
вадами зору»
здобувача групи ІН – 01 Світайло Даніла Сергійовича

Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання на відповідне
джерело.



(підпис)

Даніл
СВІТАЙЛО

Керівник,
кандидат технічних наук, доцент

Віктор АВРАМЕНКО



(підпис)

Суми – 2024

Сумський державний університет
 Центр заочної, дистанційної та вечірньої форм навчання
 Кафедра комп'ютерних наук

«Затверджую»
 В.о. завідувача кафедри
 _____ Ігор ШЕЛЕХОВ
 (підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра
 зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
 здобувача групи ІН-01 Світайло Даніл Сергійович

1. Тема роботи: «Інформаційна система прослуховування аудіо-книг для людей з вадами зору»

затверджую наказом по СумДУ від «22» квітня 2024 р. № 0414-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 29 травня 2024 року

3. Вхідні дані до кваліфікаційної роботи _

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд розроблених рішень для людей з вадами зору, та інформаційних систем для прослуховування аудіокниг. 3) Розробка інформаційної система прослуховування аудіо-книг для людей з вадами зору. 4) Аналіз отриманих результатів.

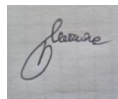
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання



(підпис)

Керівник



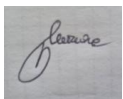
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>	20.04.2024	
2	<i>Огляд розроблених рішень для людей з вадами зору, та інформаційних систем для прослуховування аудіокниг</i>	25.04.2024	

3	<i>Розробка інформаційної система прослуховування аудіо-книг для людей з вадами зору</i>	20.05.2024	
4	<i>Аналіз отриманих результатів</i>	23.05.2024	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	1.05.2024	

Здобувач вищої освіти



(підпис)

Керівник



(підпис)

АНОТАЦІЯ

Записка:

Обґрунтування актуальності теми роботи - Тема кваліфікаційної роботи є актуальною, оскільки присвячена реалізації зручної інформаційної системи для слабоворих людей шляхом розробки відповідних елементів інтерфейсу та функціоналу.

Об'єкт дослідження – Інтерфейси для людей з вадами зору

Мета роботи – Розробка інформаційної системи прослуховування аудіо-книг для людей з вадами зору

Методи дослідження – Аналіз існуючих інформаційних систем для людей з вадами зору, аналіз існуючих сервісів для прослуховування аудіо книг

Результати – Проаналізовано сучасний стан інформаційних систем для людей з вадами зору і систем для прослуховування аудіо книг, на базі аналізу сформовано постановку задачі, обрано методи реалізації веб додатку, спроектовано веб додаток та програмно реалізовано з урахуванням поставлених вимог

Структура пояснювальної записки – пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 22 найменувань, додатку із програмним кодом.

Ключові слова:

ІНФОРМАЦІЙНА СИСТЕМА, ДОСТУПНІСТЬ, ВЕБ-ДОДАТОК, C#, ASP .NET.

ЗМІСТ

Вступ	6
1. Аналітичний огляд	7
1.1 Сучасний стан інформаційних систем.....	7
1.2 Аналіз існуючих інформаційних систем	7
1.3 Постановка задачі	22
2. Методи реалізації додатку.....	24
3. Проектування Web-додатку	26
3.1 Структурно-функціональне моделювання	26
3.2 Моделювання варіантів використання	28
3.3 Моделювання діаграми діяльності.....	31
3.4 Проектування бази даних web-додатку	32
4. Програмна реалізація	35
4.1 Розробка інтерфейсу	35
4.2 Структура проекту	37
4.3 Робота користувача з веб-додатком.....	38
4.4 Адміністрування додатку	40
4.5 Перевірка вимог до інтерфейсу	42
Висновок	44
Список літератури.....	46
ДОДАТОК А.....	49

ВСТУП

Сучасну епоху прийнято називати інформаційною ерою, або ерою комп'ютерів. Глобальний зсув до оцифрованої, комп'ютеризованої індустрії, на заміну традиційної, почався ще у середині минулого сторіччя і з тих пір лише набирає обертів.

Напрацювання інформаційної епохи спростили життя мільйонам людей, створили абсолютно нові професії та у корені змінили побут. Але є група людей, яким важко або неможливо користуватися комп'ютерами, а отже всі сучасні тенденції обходять їх стороною. Це люди з вадами зору. За даними ВООЗ, близько 300 мільйонів людей в світі страждає від порушень зору, з них майже 40 мільйонів незрячі.[1]

Історично склалося так, що слабозорі люди покладалися на переклади шрифтом Брайля або записані на плівку аудіоверсії текстів, від бестселерів до підручників і журналів. Цей формат читання є неймовірно цінним для незрячих або людей із вадами зору, адже слух для них є одним з основних джерел сприйняття інформації.

У комп'ютеризовану епоху, з усіма її можливостями і напрацюваннями для слабозорих людей є можливість розробити спеціальні інтерфейси, щоб спростити користування застосунками і сайтами, адже якщо сучасна епоха ставить на меті зробити повсякденне життя якомога простіше, то ці люди не мають залишатися осторонь.

1. АНАЛІТИЧНИЙ ОГЛЯД

1.1 Сучасний стан інформаційних систем

Для людей з вадами зору вже давно розробляються застосунки і сайти які ставлять на меті зробити можливим для них користування інтерфейсом. Також є велика кількість сайтів з аудіо книгами. Розглянемо деякі з існуючих.

1.2 Аналіз існуючих інформаційних систем

Be My Eyes [2]

Цей застосунок створює глобальну спільноту сліпих і слабоворих людей та зрячих волонтерів. Be My Eyes поєднує силу технологій та людського спілкування з метою допомогти особам із вадами зору. За допомогою відеодзвінка волонтери надають людям із порушенням зору візуальну допомогу із різними задачами, як-то порівняння кольорів, перевірка вимкненого світла або приготування вечері.



Рис. 1.1 - Інтерфейс застосунку "Be My Eyes"

У центрі інтерфейсу є велика кнопка «Подзвонити волонтеру», що є доцільним дизайнерським рішенням, адже такий елемент легко відрізнити людині з вадами зору від іншого контенту. Можемо побачити, що інтерфейс переважно складається з великих однотонних об'єктів синього кольору. Текст написаний білим кольором. Такий дизайн візуально краще сприймається людьми з вадами зору, які віддають перевагу тексту, що сильно контрастує з заднім фоном екрану або паперу.

Деякі люди з втратою зору люблять читати чорний текст на білому або жовтому фоні, оскільки це забезпечує гарний кольоровий контраст. Люди з захворюванням очей, що викликає проблеми з відблиском надають перевагу світлому тексту на темному фоні.

Зручність комбінації кольорів тексту і фону є дуже особистою і часто пов'язана з тяжкістю стану захворювання зору, або поєднання кількох захворювань. Двоє людей з однаковим відхиленням можуть бачити колір зовсім по-різному [3]. Враховуючи це, у інклюзивному інтерфейсі повинні бути присутні декілька варіантів кольорової палітри.

Ми можемо проаналізувати контрастність інтерфейсу за допомогою інструменту Colour Contrast Analyzer (ССА), щоб оцінити його зручність для використання людьми з вадами зору [4].

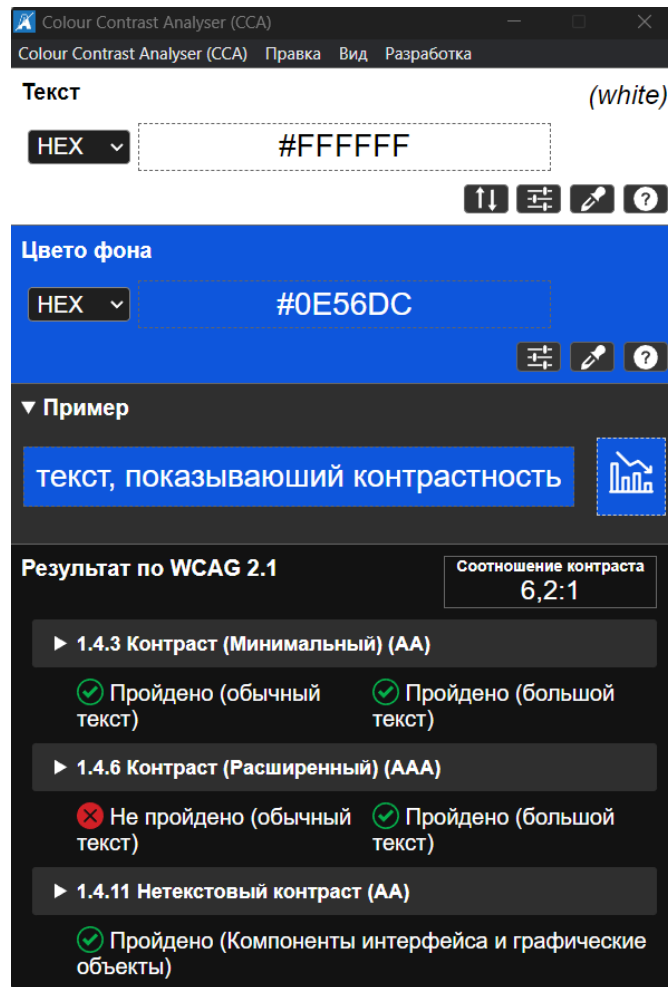


Рис. 1.2 - Аналіз контрасту інтерфейсу застосунку Ве Му Еєє

Контрастність аналізується за трьома категоріями, прописаними у Web Content Accessibility Guidelines (WCAG): контраст мінімальний, контраст розширений і нетекстовий контраст. Ці інструкції дозволяють оцінити та систематизувати доступність веб-вмісту, для створення інклюзивного дизайну [5][6].

Щоб задовольнити WCAG 2.0 рівня AA, текст(та зображення з текстом) повинні мати коефіцієнт контрастності не менше 4,5:1 для тексту «звичайного» розміру. Текст великого розміру (починаючи з 18pt/24px, жирний, починаючи з 14pt/18.5px) повинен мати контрастність як мінімум 3:1, якщо він не є елементом оформлення.

Для WCAG рівня AAA текст(та зображення з текстом) повинні мати коефіцієнт контрастності не менше 7:1 для тексту «звичайного» розміру. Текст

великого розміру (починаючи з 18pt/24px, жирний, починаючи з 14pt/18.5px) повинен мати контрастність як мінімум 4,5:1, якщо він не є елементом оформлення.

Для WCAG 2.1 рівня AA компоненти інтерфейсу (включаючи всі їх можливі стани та границі самих компонентів), а також графічні об'єкти повинні мати коефіцієнт контрастності не менше 3:1 до сусіднього(-их) кольору(-ів).

Як ми бачимо даний інтерфейс має відношення контрастності 6,2:1, що не задовольняє WCAG рівня AAA для звичайного тексту.

У застосунку відсутній режим для дальтоніків: кольорову схему неможливо змінити, що може ускладнити користування інтерфейсом людям, у яких присутні проблеми з розрізненням кольорів.

Недоліком також є, що у інтерфейсі присутні дрібні елементи, які можуть завадити цільовому користувачу зручно оперувати застосунком.

NonVisual Desktop Access [7]

У 2016 році Gov.uk опублікував результати опитування, під час якого Урядова цифрова служба запитала 712 користувачів про те, які пристрої, веб-браузери та допоміжні технології вони використовували для доступу до GOV.UK. [8]

Результати були наступні:

- Екранна лупа 30%;
- Зчитувач з екрана 29%;
- Розпізнавання мови 18%;
- Читальний асистент (для людей з дислексією) 15%;
- Інші 8%.

Згідно з опитуванням, 30% користувачів використовують екранну лупу, яка є програмним забезпеченням, що збільшує все, з чим користувачі

взаємодіють на екрані. 38% цих луп також мають можливості зчитування з екрана, причому ZoomText є найпоширенішим в обох категоріях. Інші популярні екранні лупи включали Supernova та MAGic.

Майже такий самий відсоток респондентів користувалися GOV.UK за допомогою програми зчитування з екрана. Це програми, які перетворюють екранний вміст на мову чи шрифт Брайля. Більшість програм зчитування з екрана складаються з двох аспектів: програмного забезпечення зчитувача з екрана та механізму синтезу мовлення (TTS), який диктує текст з екрана. Однією з найдоступніших програм зчитування з екрану є NonVisual Desktop Access.

NVDA дозволяє незрячим та людям зі слабким зором використовувати комп'ютер без додаткових витрат на придбання спеціальних програм. Застосунок дозволяє працювати з основними програмами Windows і стежити за подіями на екрані, використовуючи синтезатор мови в якості джерела виведення інформації. Програма озвучує все, що знаходиться і відбувається на робочому столі і у всіх програмах при наведенні курсору миші, є можливість читати текст по абзацам тощо.

NVDA дає змогу оцінити доступність веб-сайту для слабоворих людей. За допомогою нього можна зрозуміти, як користувачі програм зчитування з екрана переміщуються та читають веб-сайти. Розробники можуть отримати уявлення про те, як використовується веб-сайт, і краще зрозуміти вказівки у WCAG 2.0.

Ось низка проблем, які може вивести розробник після перевірки свого веб сайту за допомогою NVDA:

- Області, недосяжні програмою зчитування з екрана;
- Вміст сторінки не структурований заголовками;
- Заголовки не дотримуються логічної послідовності;
- Зображення без альтернативного тексту;

- Поля вводу без відповідних підписів;
- Посилання без доступного опису;
- Кнопки без доступного опису;
- Інтерактивні елементи не позначені належним чином за допомогою відповідного елемента HTML. [9]

MySight York [10]

Незалежна місцева благодійна організація міста Йорк у Великобританії, яка допомагає людям з проблемами зору стати незалежними та жити повноцінним життям.

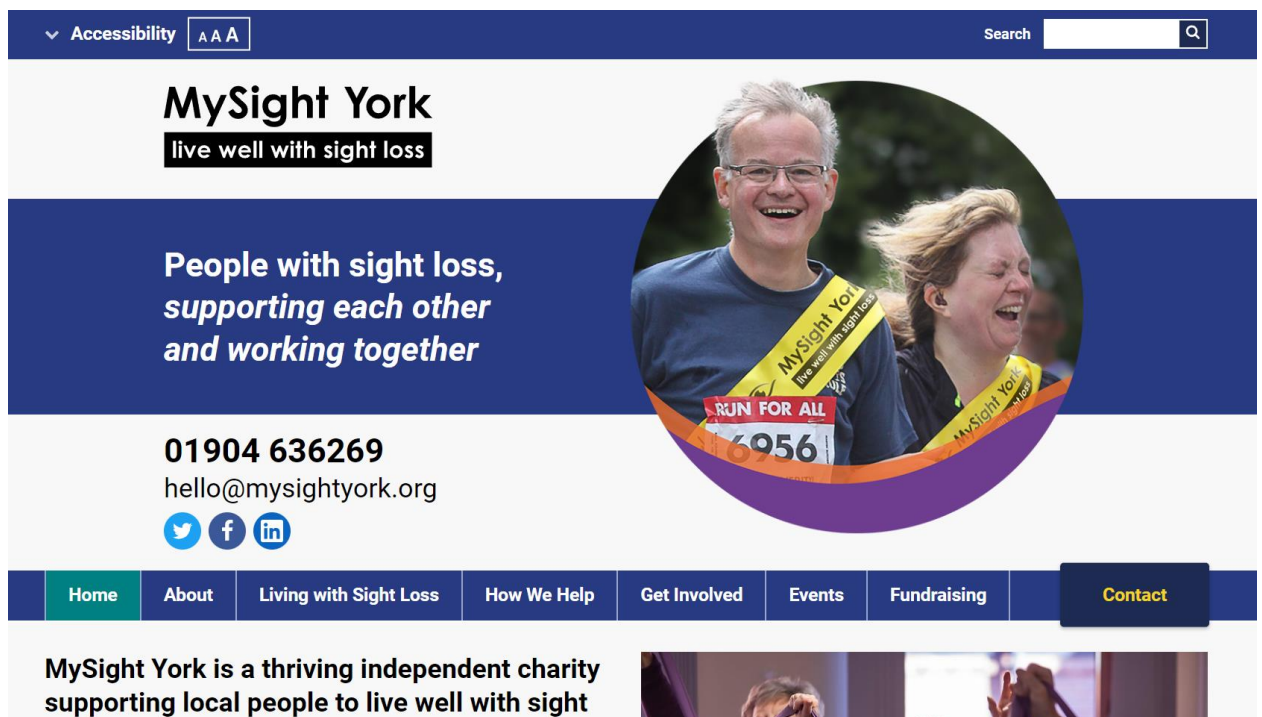


Рис. 1.3 - Інтерфейс сайту MySight York

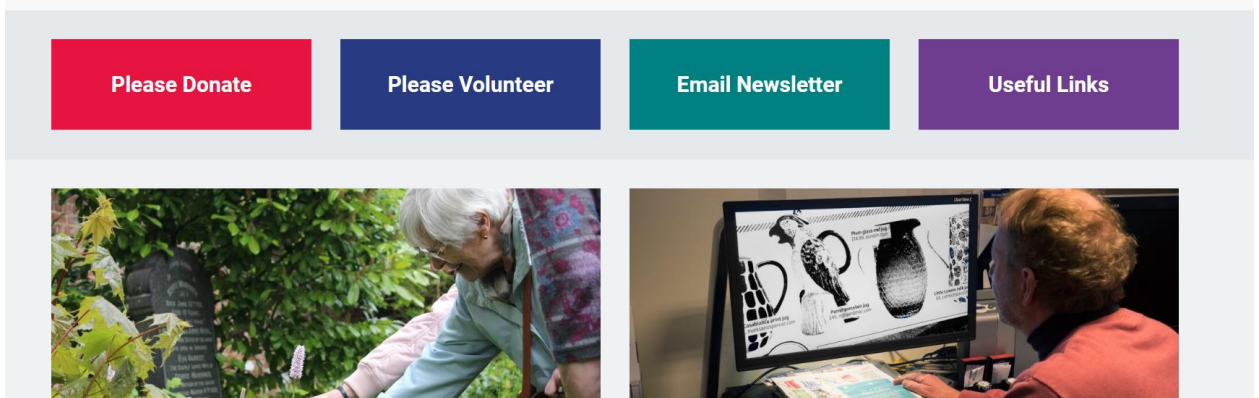


Рис. 4 - Оформлення кнопок на сайті

Можемо побачити, що інтерфейс переважно складається з великих однотонних об'єктів яскравого кольору, поверх яких текст написаний білим кольором.

Проаналізувавши контрастність тексту до кольору отримуємо відношення 10,4:1, що задовольняє всім критеріям WCAG рівня AA та AAA.

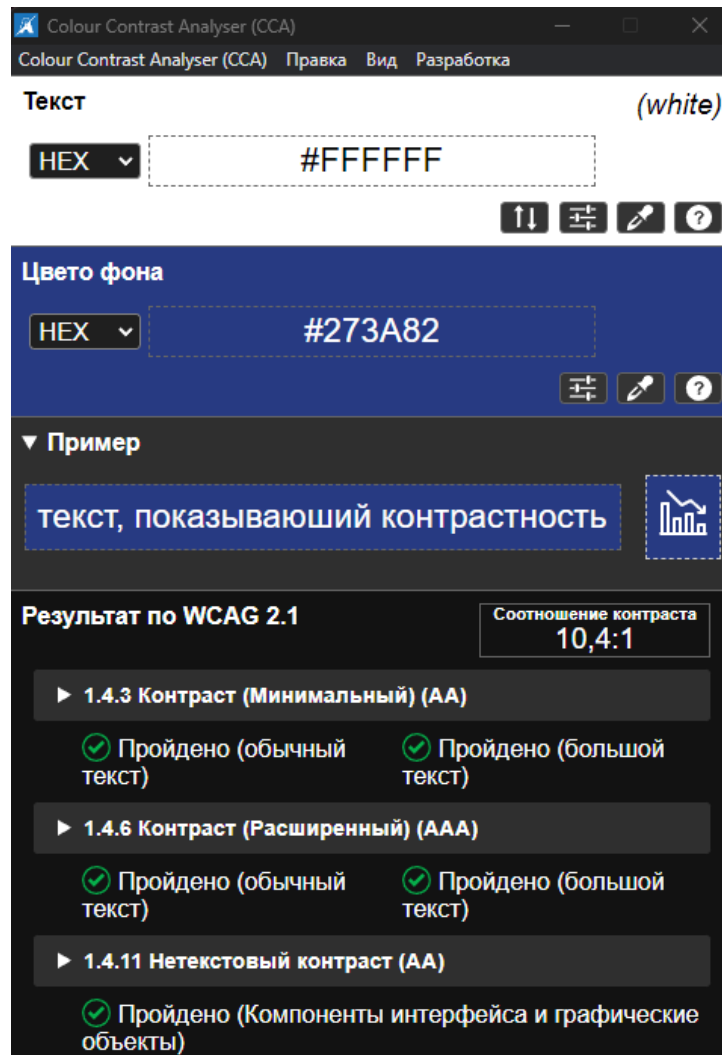


Рис. 1.5 - Аналізу контрастності інтерфейсу сайту MySight Yotk

На сайті присутні налаштування доступності тексту, де можна змінити розмір тексту, а також палітру кольорів під різні типи кольорової сліпоти.

Жовто-чорна палітра має відношення контрастності 14,2:1, що задовольняє всім критеріям WCAG рівня AA та AAA.



Рис. 1.6 - Режим кольорової сліпоти



Рис. 1.7 - Інвертовані кольори

Сайт не дуже вдало адаптується до розміру екрану для користування на мобільних пристроях, адже текст залишається малим, що ускладнює читання з екрану, але його можна змінити у вищезгаданих налаштуваннях доступності.



Рис. 1.8 - Адаптація сторінки MySight York

Недоліками також є, що на сайті присутні дрібні елементи, а також варіативна палітра кольорів, що може ускладнити читання для слабозорих людей.

Sumy GPS Inclusive [11]

Застосунок дозволяє відстежувати громадський транспорт у місті Суми для людей з вадами зору. Також, як заявляють розробники, застосунок зручний для людей з розладами руху, адже інтерфейс не має дрібних елементів.

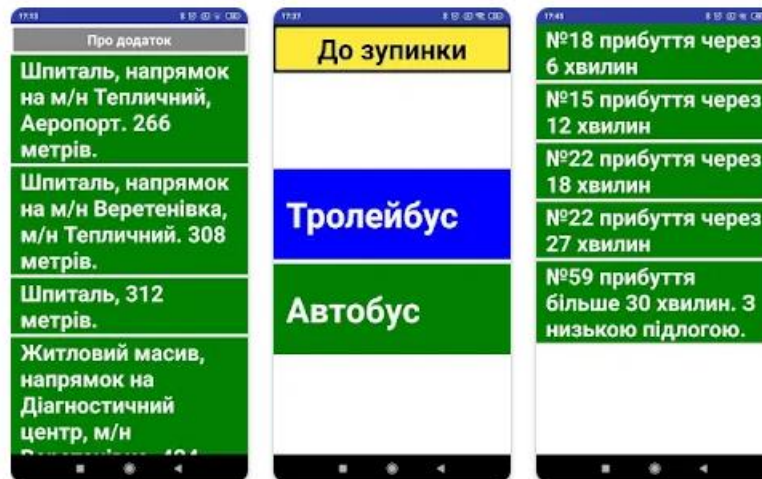


Рис. 1.9 - Інтерфейс застосунку "Sumy GPS Inclusive"

Можемо побачити, що інтерфейс переважно складається з великих однотонних об'єктів яскравого кольору, поверх яких текст написаний білим кольором.

У застосунку переважає зелений колір фону. Проаналізувавши його контрастність до білого тексту отримуємо відношення 5,1:1, що не задовольняє WCAG рівня AAA для звичайного тексту.

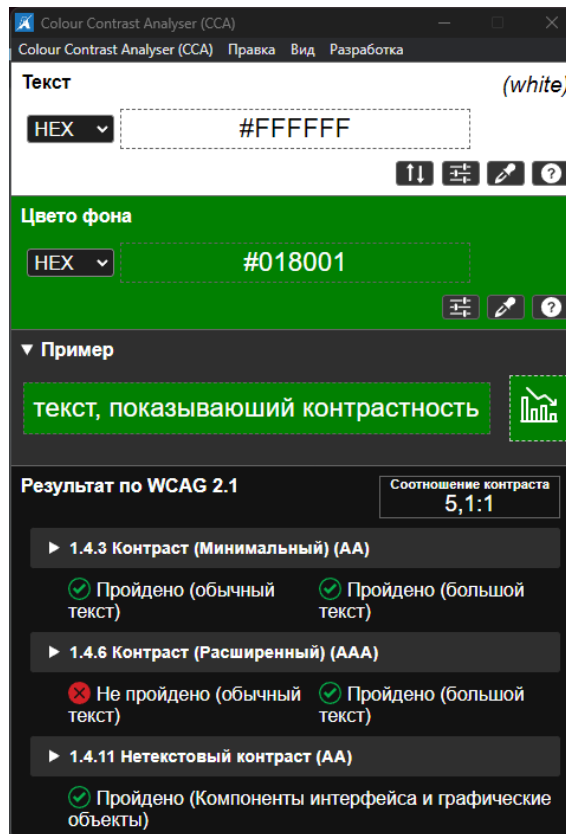


Рис. 1.10 – Аналіз інтерфейсу застосунку Sumy GPS Inclusive

Недоліком також є відсутність налаштування кольорів, що не дозволяє змінити палітру інтерфейсу під різні типи кольорової сліпоти.

Спеціальні можливості

На багатьох сучасних смартфонах в налаштуваннях присутній розділ «Спеціальні можливості», який з-поміж інших корисних функцій містить низку налаштувань для людей з вадами зору. Ці функції, по аналогії з вищезгаданим NVDA, дають змогу зрозуміти, як необхідно проектувати інтерфейси, щоб зробити їх доступними для слабоворих людей.

Розглянемо деякі з них на прикладі телефону компанії Самсунг. [12]

1 Покращення видимості

Пункт "Покращення видимості" в спеціальних можливостях дозволяє користувачу змінювати розмір та стиль шрифту в пристрої, контраст та колір відповідно до особистих вимог.

2 Підвищення чутності

Пункт "Підвищення чутності" налаштовує звучання пристрою для сприйняття користувачем або пропонує використання текстових альтернатив.

3 Моторика та взаємодія

Пункт "Моторика та взаємодія" дозволяє покращити або замінити сенсорний тип взаємодії з пристроєм та інші елементи керування.

4 Читання з екрана

Функція "Читання з екрана" надає користувачу аудіо супровід дій та особливе керування, для взаємодії з пристроєм без необхідності бачення екрана.


Для читання з екрану за допомогою цієї функції, текст повинен мати можливість виділення, тобто для веб-розмітки стиль елемента повинен містити атрибут «user-select : auto».

5 Спрощений режим

Спрощений режим це спрощений вигляд головного екрану, екрану додатків та налаштувань зі збільшеними елементами. При активованому спрощеному режимі налаштування головного екрану відмінюються та масштабування екрану встановлюється на максимум.


Audible [13]

Audible — це американський онлайн-сервіс аудіо книг і подкастів, який дозволяє користувачам транслювати та завантажувати аудіо книги та інший розмовний вміст. Цей сервіс є одним з найпопулярніших ресурсів для прослуховування аудіо книг.

Hi, Gretchen! | 1 Credit Available | Help | 

audible
an amazon company


Library | Wish List | Browse | Audible Originals | Gifts | Inside Audible

Find your next great listen 

Advanced Search

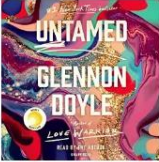
Library

Titles | Collections

Search your library 

All | Finished | Unfinished

Enhanced Format 4 | Date Added | Select






Untamed

By: Glennon Doyle

Narrated by: Glennon Doyle

☆☆☆☆☆ Write a review

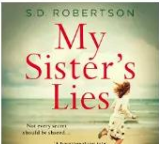
In her most revealing and powerful memoir yet, the activist, speaker, best-selling author, "patron saint of female empowerment" (*People*) explores the joy and peace we discover when we stop striving to meet others' expectations and start trusting the voice deep within us....

 Add to favorites |  Add to... |  Mark as Finished

8h 22m

[Listen now](#)

[Download](#)



My Sister's Lies

By: S.D. Robertson

Narrated by: Emma Pallant

☆☆☆☆☆ Write a review

For a decade, Hannah's life has been pretty close to perfect - she has a great job, she's married to Mark, and her child-free existence means she's free as a bird. The only

2h 1m left

[Listen now](#)

[Download](#)

Рис. 1.11 - Интерфейс веб-сайту Audible

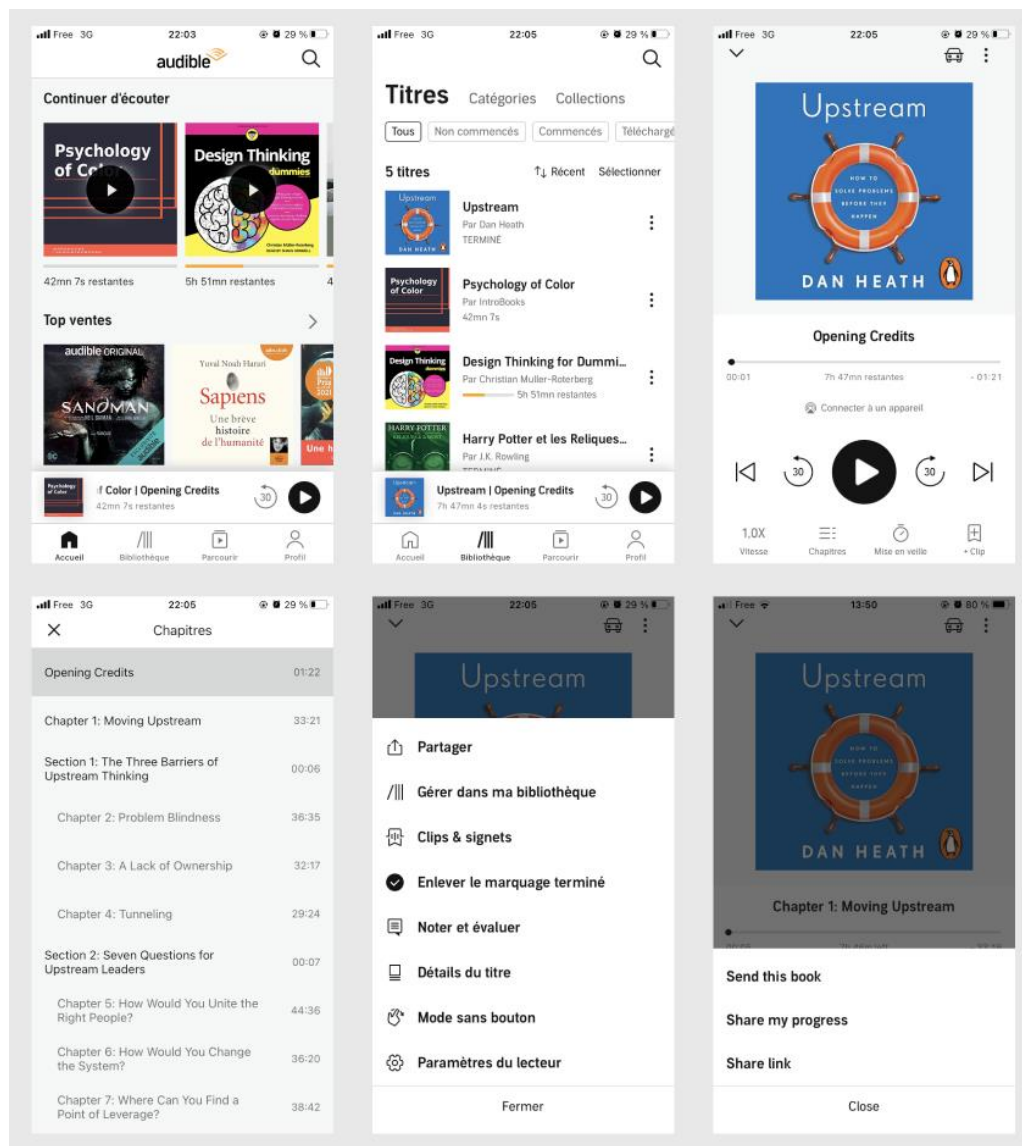


Рис. 1.12 - Інтерфейс застосунку Audible на мобільних пристроях

Як ми бачимо інтерфейс не пристосований для користування людьми з вадами зору. Все, починаючи від кольорів і закінчуючи піктограмами є важким для сприйняття слабозорій людині. Також сервіс є платним, що можна віднести до недоліків.

Audiobooks.com [14]

Подібно до Audible, цей сайт також пропонує послугу передплати в стилі Netflix з невеликою відмінністю. Замість того, щоб завантажувати книги, є можливість транслювати їх. Сервіс має функцію, яка дозволяє вам переходити

з одного пристрою на інший (з ноутбука на смартфон), не втрачаючи місця. Потокове передавання займає менше місця на пристрої, що не забиває пам'ять.

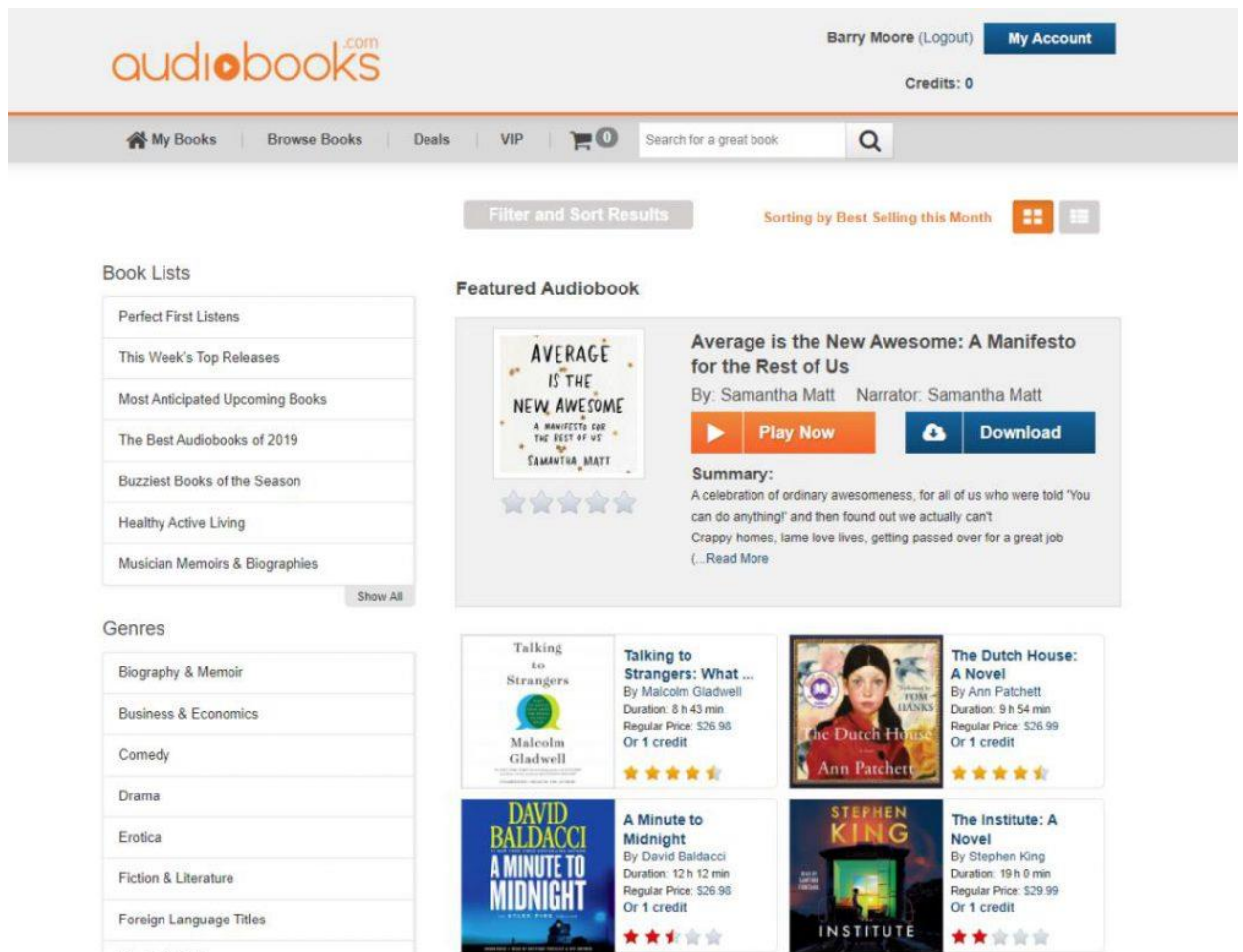


Рис. 1.13 - Інтерфейс веб-сайту Audiobooks.com

Інтерфейс цього сервісу також є не пристосованим для користування людьми з вадами зору.

1.3 Постановка задачі

Розробити інформаційну систему прослуховування аудіо-книг для людей з вадами зору.

Створити зручний інтерфейс з урахуванням потреб слабоворих людей та з оглядом на вже існуючі рішення.

- Інтерфейс інформаційної системи має задовольняти вимогам WCAG 2.1 по контрастності кольорів. Відношення контрастності повинне переважати 7:1.
- Інтерактивні елементи застосунку, як і текст в цілому, повинні бути великими
- Веб-сайт має вдало адаптуватися під мобільні пристрої.
- Текст має виділятися, та не мати обмежень з масштабування.

Інтерфейс має бути перевірений програмою зчитування з екрану та дотримуватися наступних вимог:

- Інтерфейс не повинен мати області, недосяжні програмою зчитування з екрана;
- Вміст сторінки має бути структурований заголовками;
- Заголовки мають дотримуватися логічної послідовності ;
- Зображення повинні мати альтернативний текст;
- Поля вводу повинні мати відповідні підписи;
- Посилання повинні мати доступний опис;
- Кнопки повинні мати доступний опис;
- Інтерактивні елементи повинні бути позначені належним чином за допомогою відповідного елемента HTML.

2. МЕТОДИ РЕАЛІЗАЦІЇ ДОДАТКУ

Оскільки на меті поставлено розробити інформаційну систему, що буде включати в себе інтерфейс (front-end) та серверну частину (back-end) а також взаємодію з СУБД необхідно обрати інструмент який дозволе працювати з усіма аспектами веб-додатку на достатньому рівні, а іншими словами - необхідний full-stack фреймворк. Для цього було обрано платформу розробки веб-додатків ASP.NET.

ASP.NET - це фреймворк веб-розробки на базі C#, який пропонує широкий спектр інструментів та компонентів для спрощення розробки веб-сайтів та веб-додатків. [15]

Для реалізації front-end частини було обрано сучасні інтерфейсні технології, такі як HTML, CSS та JavaScript, а також технологію RazorPages, яка дозволяє вбудовувати серверний код на мові C# прямо у веб сторінки ASP.NET. [16]

В якості СУБД було обрано MS SQL, а для взаємодії з нею буде використовуватися структура об'єктно-реляційного відображення (ORM) Entity Framework Core. [17]

Переваги використання зв'язки C#, ASP.NET та Entity Framework:

- **Потужність та масштабованість:** C# - це мова програмування загального призначення, яка підходить для створення складних та масштабованих веб-додатків. ASP.NET - це фреймворк веб-розробки на базі C#, який пропонує широкий спектр інструментів та компонентів для спрощення розробки веб-сайтів та веб-додатків. Entity Framework - це об'єктно-реляційний маппер, який спрощує роботу з базами даних з C#. MS SQL - це надійна та масштабована система управління базами даних, яка підходить для великих веб-додатків з високим трафіком. MS SQL сумісна з

C# та Entity Framework, що робить її простим вибором для розробників, які використовують ці технології. [15]

- Продуктивність: C# - це компільована мова програмування, що робить її більш продуктивною, ніж інтерпретовані мови, такі як JavaScript. ASP.NET та Entity Framework також оптимізовані для високої продуктивності.
- Безпека: C#, ASP.NET та Entity Framework мають вбудовані функції безпеки, які допомагають захистити веб-додатки від кібератак.

Web-додаток буде мати архітектуру MVC (model-view-controller) - архітектурна схема, яка складається з трьох компонентів Model, View та Controller, що ефективно відокремлює Business Logic від користувальницького інтерфейсу програми. [21]

- Модель (Model) зосереджена на логіці та обробці даних. Вона зберігає інформацію і визначає, як ця інформація має взаємодіяти з програмою.
- Вигляд (View) відповідає за відображення даних користувачеві. Це включає в себе графічний інтерфейс, веб-сторінки або інші способи представлення інформації користувачу.
- Контролер (Controller) взаємодіє з користувачем та визначає, які дії виконувати на основі взаємодії користувача з Виглядом. Він також відповідає за маршрутизацію і обробку запитів.

Так як передбачається, що проект буде громіздкий і, можливо, у майбутньому буде реалізовуватися та підтримуватися декількома розробниками, то до проекту було підключено розподілену систему контролю версій Git, а сам проект буде зберігатися на віддаленому репозиторії. [22]

3. ПРОЕКТУВАННЯ WEB-ДОДАТКУ

3.1 Структурно-функціональне моделювання

Структурно-функціональне моделювання (СФМ) - це метод дослідження та представлення систем, який описує їх як сукупність взаємопов'язаних елементів (компонентів) та функцій, які вони виконують.

Основна мета структурно-функціонального моделювання полягає в глибокому розумінні системи та її складових, а також вивченні взаємодій між ними. Це дозволяє аналізувати та прогнозувати різні аспекти функціонування системи, такі як ефективність, надійність, поведінка та можливість удосконалення. В рамках багатьох проектів важливо розуміти функціональну взаємодію між різними компонентами системи, що дозволяє швидко описати та відобразити функціональність всіх компонентів, які впливають на кінцевий продукт або послугу системи.

IDEF0 - це методологія створення функціональних моделей систем, яка використовується для аналізу, моделювання та документування бізнес-процесів. Ця методологія дозволяє систематизувати та візуалізувати функціональні аспекти системи за допомогою ієрархічної структури блоків і стрілок, що представляють взаємодії між різними елементами системи. [18]

Методологія надає засоби для ясного представлення функціональних зв'язків та процесів в системі. IDEF0 сприяє систематизації аналізу бізнес-процесів, дозволяючи крок за кроком розглядати та розуміти кожен елемент. Методологія базується на стандартах, що дозволяє забезпечити єдність та зрозумілість у роботі з функціональними моделями. За допомогою IDEF0 можна проводити аналіз функціональних зв'язків і процесів системи для виявлення можливостей їх оптимізації та покращення. Методологія дозволяє створювати чітку та структуровану документацію функціональних аспектів системи, що сприяє зрозумінню та спілкуванню між учасниками проекту.

Контекстна діаграма процесу прослуховування аудіо-книги з точки зору використання веб-додатку актором-користувачем зображена на рисунку 3.1

Для контекстної діаграми було визначено такі дані:

- вхідні дані: запит на прослуховування аудіо-книги;
- вихідні дані: відображення аудіо-книги для прослуховування;
- управління: наявність аудіо-книги, наявність реєстрації у користувача;
- механізми: веб-додаток, сервер, БД, користувач.



Рис. 3.1 - Контекстна діаграма

Декомпозиція IDEF0 для деталізації внутрішніх процесів зображена на рисунку 3.2.

Декомпозиція діаграми процесу прослуховування аудіо-книг представлена наступними підпроцесами:

- авторизація у систему;
- перегляд списку аудіо-книг;
- введення пошукового фільтру;
- вибір аудіо-книги зі списку.

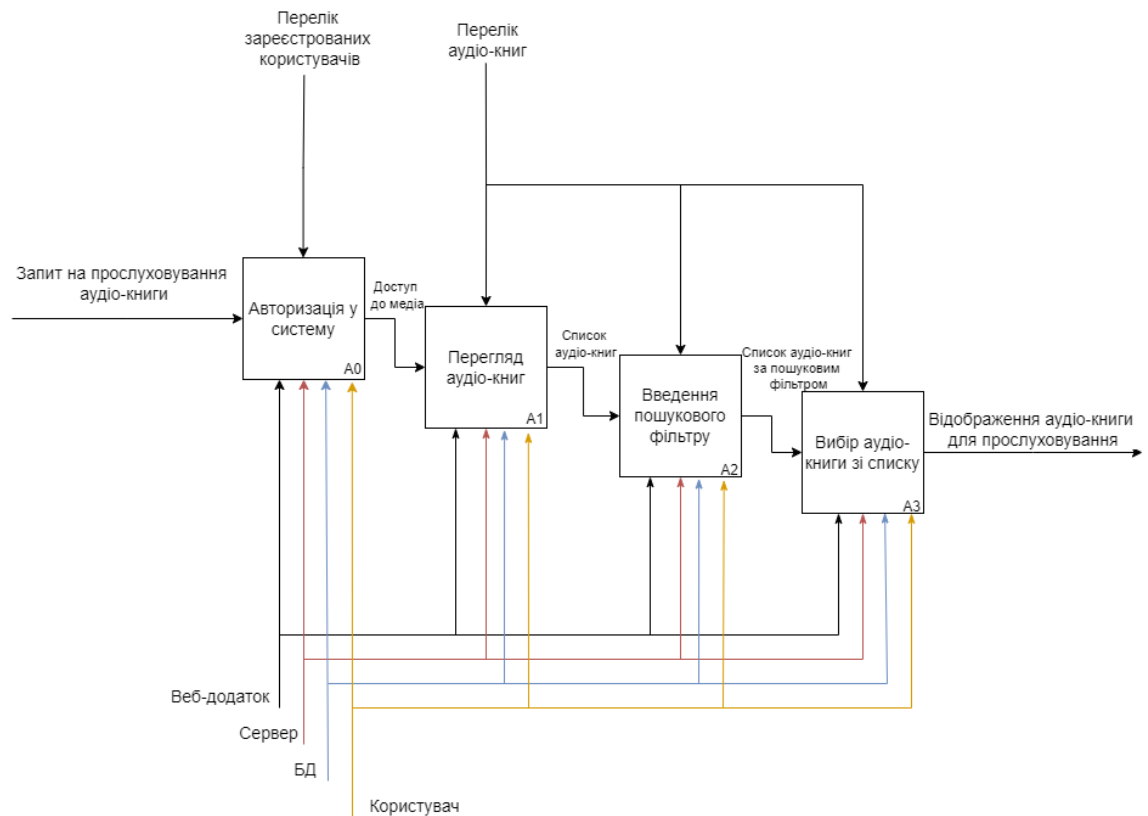


Рис. 3.2 - Діаграма першого рівня декомпозиції IDEF0

3.2 Моделювання варіантів використання

Моделювання варіантів використання (use cases) - це процес створення моделі або представлення, що дозволяє аналізувати і оцінювати різні можливі сценарії або варіанти використання продукту, системи або послуги акторами, тобто людьми в ролі користувачів, адміністраторів, модераторів тощо. Варіанти використання дозволяють визначити межі системи, а також відношення між системою та зовнішнім середовищем. Діаграми варіантів використання використовують для візуалізації вимог до системи. В результаті моделювання можуть бути зроблені прогнози щодо ефективності, вартості, впливу на користувачів та інші аспекти продукту. [19]

Діаграма варіантів використання не описує порядок, в якому виконуються варіанти використання. Для цього рекомендується використовувати інші діаграми, наприклад, діаграму дій.

На рисунку 3.3 показана діаграма варіанту використання додатку у ролі користувача.

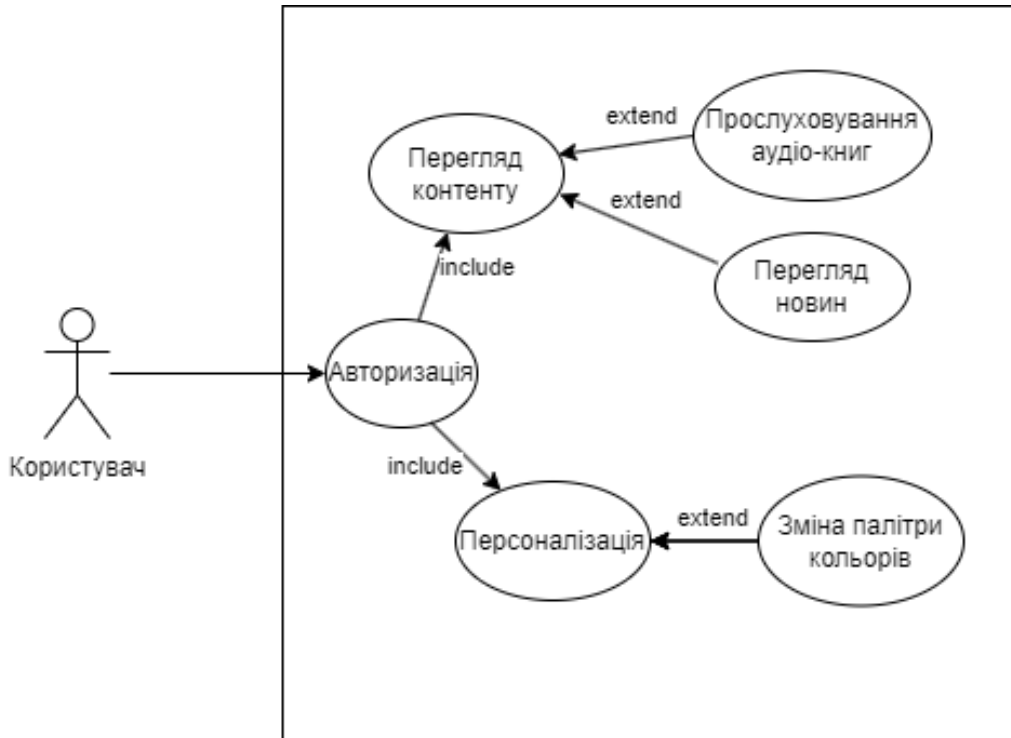


Рис. 3.3 - Діаграма варіантів використання у ролі користувача

На рисунку 3.4 показана діаграма варіанту використання додатку у ролі адміністратора.

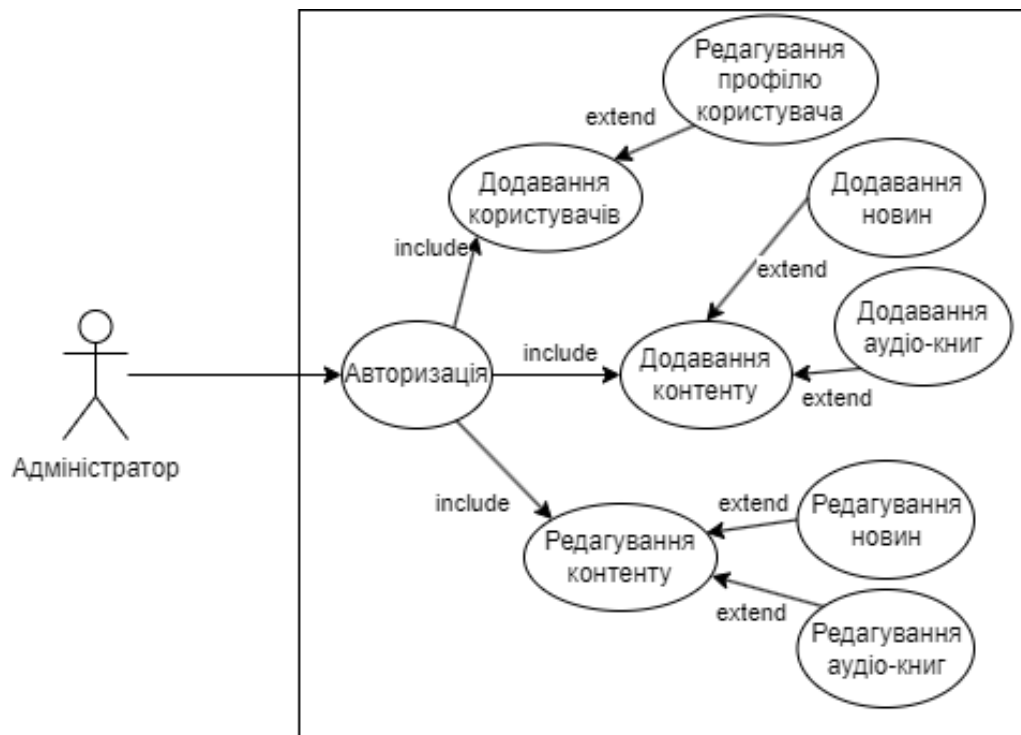


Рис. 3.4 - Діаграма варіантів використання у ролі адміністратора

Актори та описи варіантів використання наведені у таблиці 3.1.

Таблиця 3.1. Опис варіантів використання додатку

№	Актор	Опис
1	Користувач	Користувач web-додатку після авторизації має можливість: переглядати контент (переглядати новини, переглядати аудіо-книги), кастомізації (Зміна кольору палітри інтерфейсу).
2	Адміністратор	Адміністратор web-додатку після авторизації має можливість: додавати нових користувачів (редагувати профілі вже доданих користувачів), додавати контент (додавати новини, додавати аудіо-книги), редагувати контент (редагувати новини, редагувати аудіо-книги).

3.3 Моделювання діаграми діяльності

Діаграма діяльності - це тип діаграми в моделюванні систем, що використовується для візуалізації послідовності дій, процесів або операцій в системі або процесі. Вона демонструє крок-за-кроком послідовність виконання дій або операцій, які здійснюються в системі або в рамках певного процесу. Це допомагає зрозуміти та аналізувати процеси бізнесу чи програмного коду. [19]

На рисунку 3.5 зображено діаграму діяльності web-додатку.

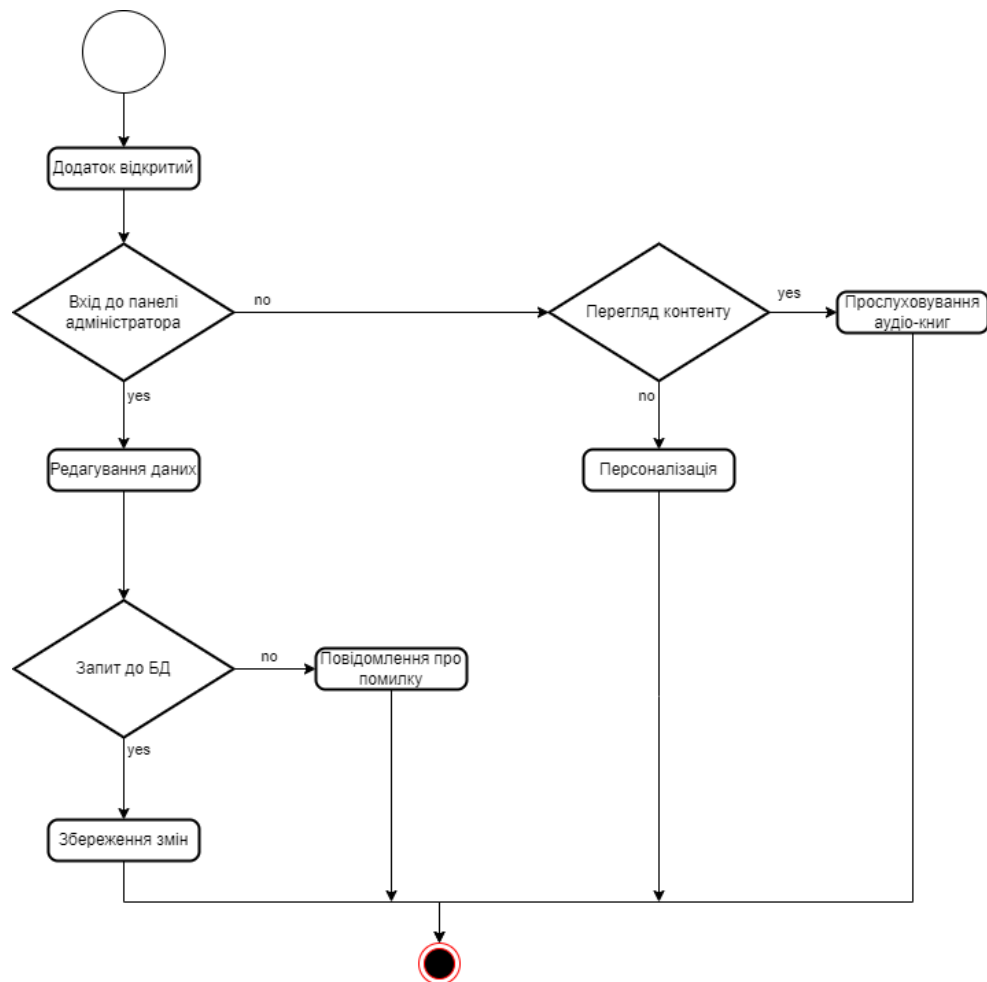


Рис. 3.5 - Діаграма діяльності

3.4 Проектування бази даних web-додатку

На рисунку 3.6 показано схему бази даних, яка використовується у web-додатку.

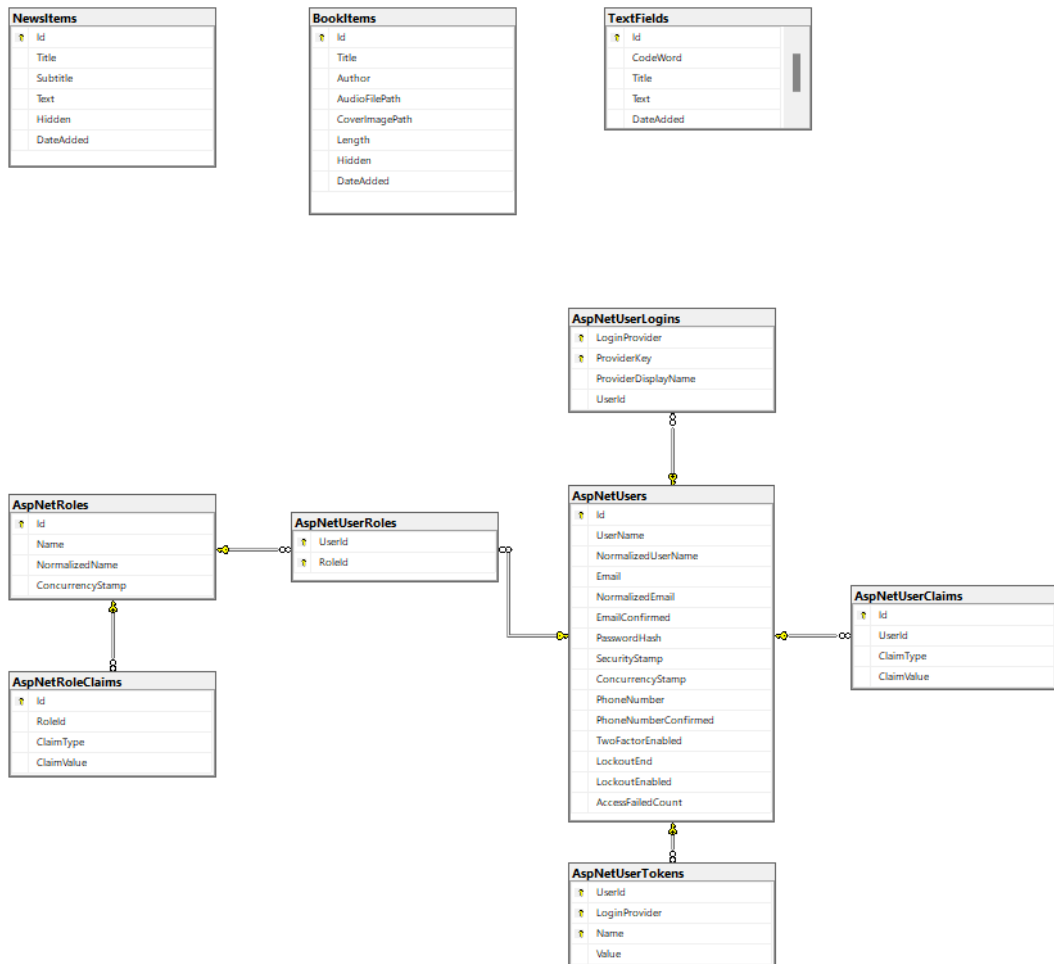


Рис. 14 - Діаграма бази даних

Сутність NewsItems створена для зберігання новин. Вона має поля: Id – ідентифікатор, Title – заголовок новини, Subtitle – опис новини, Text – текст новини, Hidden – флаг прихованості, DateAdded – дата додавання.

Сутність BookItems створена для зберігання книг. Вона має поля: Id – ідентифікатор, Title – заголовок книги, Author – автор книги, AudioFilePath – шлях до локального аудіо-файлу, CoverImagePath – обкладинка книги, Length – тривалість аудіо-файлу, Hidden – флаг прихованості, DateAdded – дата додавання.

Сутність `TextFields` створена для зберігання змісту сторінок, таких як «Медіа», «Новини» і інші, у випадку якщо їх потрібно буде редагувати, а також для індексації пошуковими системами. Вона має поля: `Id` – ідентифікатор сторінки, `CodeWord` – кодова назва сторінки для зручності посилань (наприклад `PageBooks` – для сторінки з аудіо-книгами), `Title` – заголовок сторінки, `Text` – вміст сторінки, `DateAdded` – дата додавання.

Наступні сутності є частиною системи `ASP.NET Identity` – API, яке дозволяє працювати з користувачами, паролями, ролями, кукі тощо. Саме цей інтерфейс гарантує безпеку та захист даних користувачів, перевірку справжності, зони доступу застосунку та інші аспекти роботи з обліковими записами. [20]

Сутність `AspNetUsers` зберігає користувача та його загальні атрибути, такі як ім'я, електронна пошта, телефонний номер, хешований пароль тощо.

Сутність `AspNetRoles` зберігає ролі для регулювання доступу до зон додатку.

Сутність `AspNetUserClaims` зберігає відношення `User` до `Claim`. Об'єкти `Claim` представляють певну інформацію про користувача, яку ми можемо використовувати для автентифікації в додатку. Наприклад, у користувача може бути визначений вік, місто, країна проживання, улюблена музична група тощо. І всі ці ознаки можуть представляти окремі об'єкти `claim`. Залежно від значення цих `claim` ми можемо надавати користувачу доступ до того чи іншого ресурсу. Таким чином, `claims` представляють більш загальний механізм автентифікації, ніж стандартні логіни або ролі, які прив'язані лише до однієї певної ознаки користувача. [20]

Сутність `AspNetUserToken` зберігає маркер перевірки справжності користувача.

Сутність `AspNetUserLogins` пов'язує користувача з логіном.

Сутність `AspNetRoleClaims` зберігає `Claim`, який надається всім користувачам певної ролі.

Сутність `AspNetUserRoles` пов'язує користувача з роллю.

4. ПРОГРАМНА РЕАЛІЗАЦІЯ

4.1 Розробка інтерфейсу

Головне меню було реалізовано як сітка 2x2 з кнопок, що заповнюють весь екран. Це полегшує навігацію і читання.

Головна сторінка (Рис.) має кнопку «Авторизація» для авторизації та переходу в особистий кабінет, «Інформація», при натисканні якої озвучується інформація про додаток, «Радіо», що транслює радіо в прямому ефірі та «Медіа», що переводить користувача до новин та аудіо книг.

Текст на кнопках реалізовано таким чином, щоб він міг бути прочитаний програмою синтезу мовлення.

Код розмітки Razor Page CSHTML головної сторінки наступний:

```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <div class="grid2x2 animate__animated animate__backInDown ">
        <div class="grid-row">
            <div id="login" style="cursor: pointer;" class=" grid-column rounded
concentrate blue" onclick="window.location='/Account/Index';">
                @ViewBag.Cabinet
            </div>
            <div id="radio" class="grid-column rounded concentrate blue"></div>
        </div>
        <div class="grid-row">
            <div id="info" class="grid-column rounded concentrate
blue">Інформація</div>
            <div id="media" style="cursor: pointer;" class="grid-column rounded
concentrate blue" onclick="window.location='/Media/Index';">
                @ViewBag.MediaText
            </div>
        </div>
    </div>
</div>

@section Scripts{

    <script>
        $('#radio').load("/Home/RadioPartial")

        var infoElement = document.getElementById('info');

        var voice = document.getElementById('voiceover');
```

```

        infoElement.addEventListener('click', function(){
            voice.src = "/voicelines/extended-info.mp3";
            voice.play()
        });
    </script>
}

```

Стилі передбачують динамічне масштабування інтерфейсу відповідно до розміру пристрою. Це гарантується CSS @-правилом:

```

@media only screen and (max-width: 700px) {
    .login-form{
        position: relative;
    }
    .grid-row {
        flex-wrap: wrap;
    }

    .grid-column {
        flex-basis: 100%;
    }

    .login-row {
        flex-wrap: wrap;
    }

    .login-column {
        flex-basis: 100%;
    }
}

```



Рис. 4.1 - Інтерфейс головної сторінки в розширенні 1920x1080

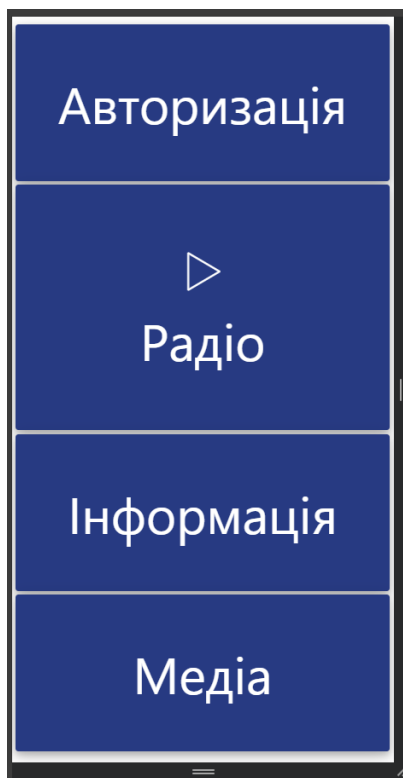


Рис. 4.2 - Інтерфейс головної сторінки на мобільному пристрої

4.2 Структура проекту

Проект реалізований за архітектурою MVC, що передбачає чітку файлову структуру (Рис. 4.3).

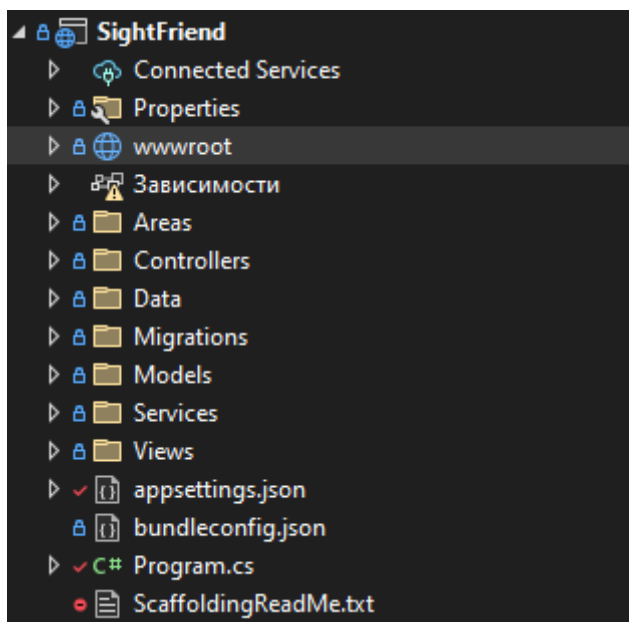


Рис. 4.3 - Структура проекту

В папці `wwwroot` розміщені медіа файли, такі як аудіо книги та зображення, стилі та скрипти.

В папці `Areas` знаходяться інструменти адміністрування, такі як адмін-панель, що регулює контроль до зон додатку окремим ролям користувачів.

В папці `Controllers`, `Models` та `Views` знаходиться основна структура проекту MVC.

В папці `Data` знаходяться класи для роботи з базою даних та `Entity Framework`.

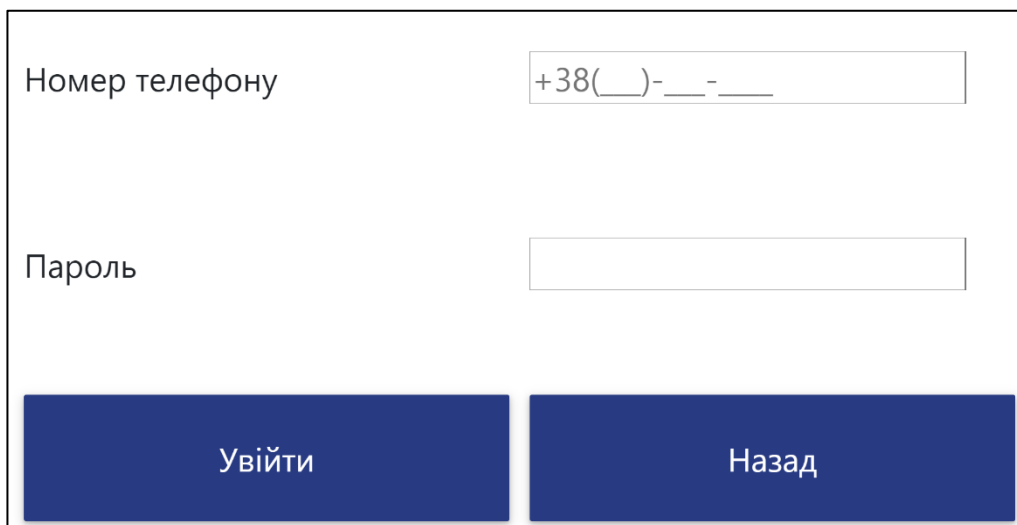
В папці `Services` знаходяться допоміжні класи для роботи з платформою ASP.NET.

Така структура проекту гарантує зручне масштабування додатку, спрощує орієнтацію по коду, підтримку та тестування.

4.3 Робота користувача з веб-додатком

В момент завантаження веб-додатку користувач потрапляє на головну сторінку (Рис.) з якої є можливість авторизуватися для використання основних функцій застосунку, прослуховувати онлайн-радіо, перейти до медіа, а також отримати інформацію про застосунок в озвученому вигляді.

При натисканні на кнопку «Авторизація» користувача переводить на сторінку логіну (Рис. 4.4), де він має можливість ввести свої дані для авторизації у систему та натиснути «Увійти» для їх обробки.



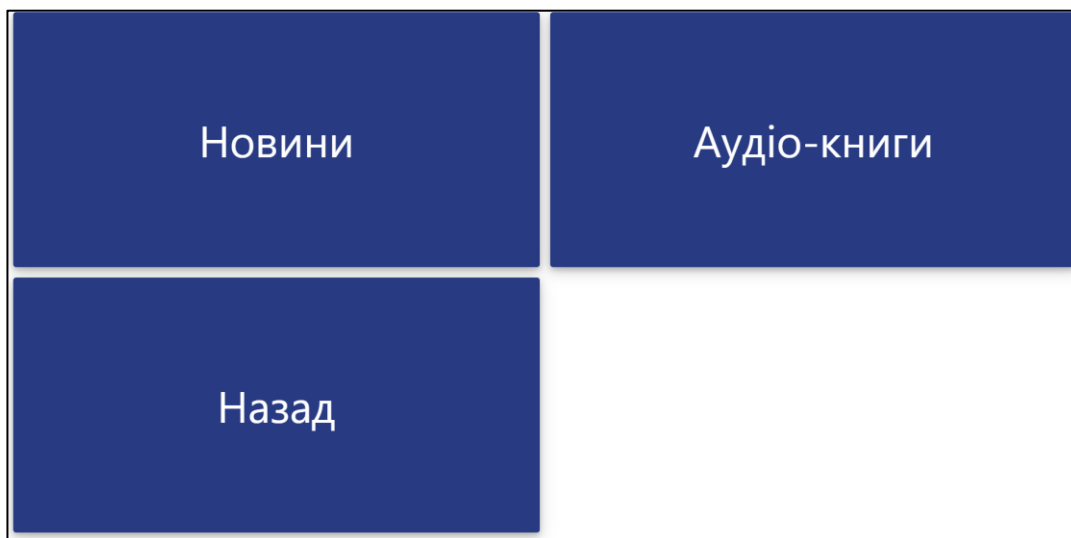
Номер телефону

Пароль

[Увійти](#) [Назад](#)

Рис. 4.4 - Сторінка логіну

При переходу на сторінку «Медіа» (Рис. 4.5) користувачу даються на вибір «Новини» та «Аудіо-книги».



[Новини](#) [Аудіо-книги](#)

[Назад](#)

Рис 4.5 - Сторінка "Медіа"

На сторінці «Аудіо-книги» (Рис. 4.6) присутній список аудіо книг доступних для прослуховування з яких користувач натисканням може перейти в аудіо плеєр (Рис. 4.7).

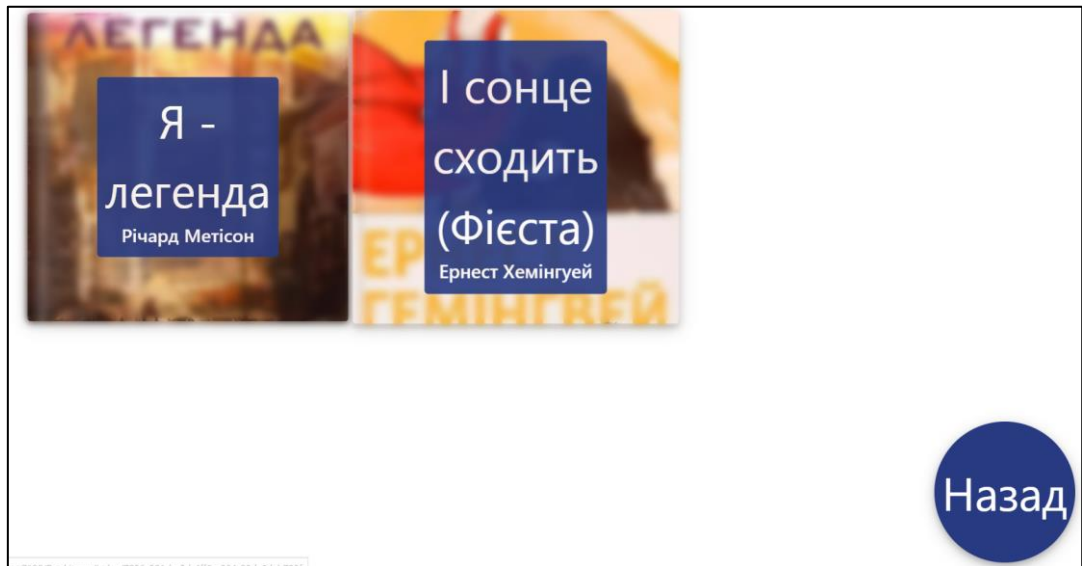


Рис. 4.6 - Сторінка "Аудіо-книги"

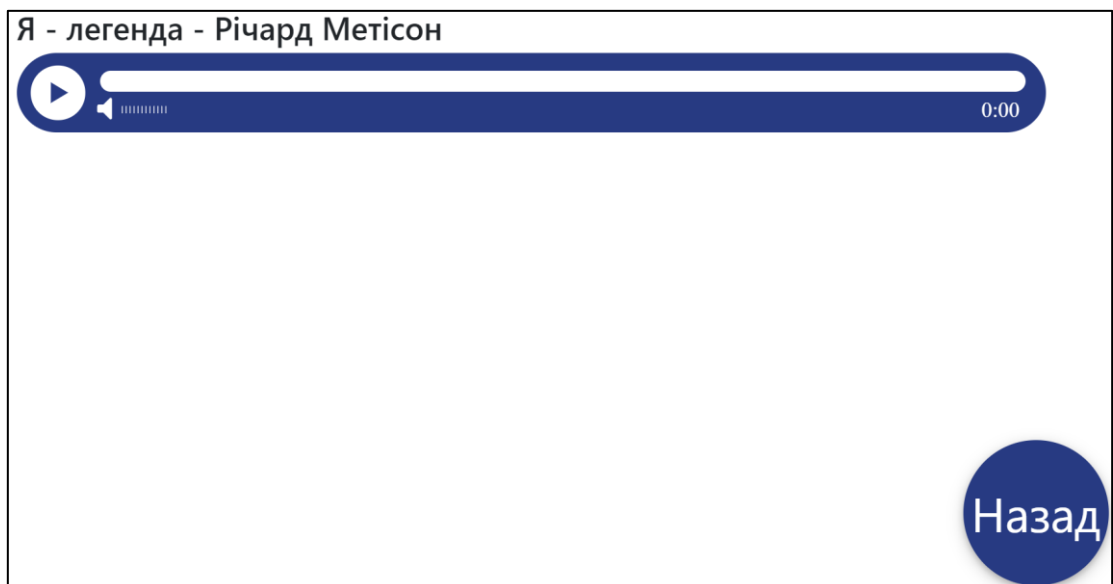


Рис. 4.7 - Сторінка аудіо книги

4.4 Адміністрування додатку

Після авторизації адміністратор має можливість перейти за посиланням /admin та потрапити у адмін-панель (Рис. 4.8), де він має можливість додавати та редагувати новини і аудіо книги (Рис. 4.9), редагувати сторінки сайту та користувачів додатку.

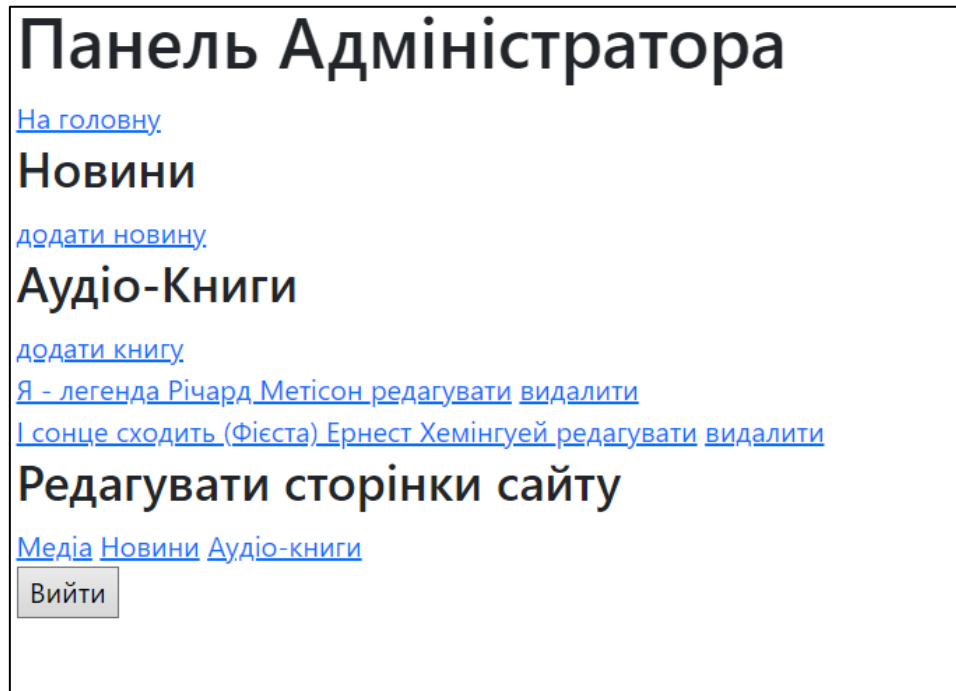


Рис. 4.8 - Адмін-панель

Редагувати книгу

Title

Я - легенда

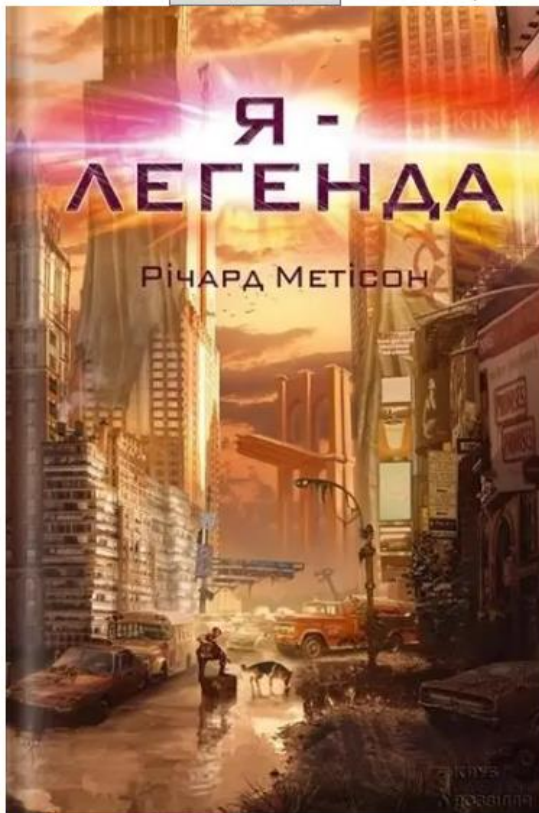
Author

Річард Метісон

AudioFilePath Файл не выбран

▶ 0:00 / 1:25:48 — 🔊 ⋮

CoverImagePath Файл не выбран



Hidden

Рис. 4.9 - Редагування аудіо книги

4.5 Перевірка вимог до інтерфейсу

В постановці задачі було визначено, що додаток має задовольняти вимогам WCAG 2.1 по контрастності кольорів, а відношення контрастності повинне переважати 7:1. Утиліта Colour Contrast Analyser показує, що додаток має відношення контрастності 10,4:1, що задовольняє вимоги WCAG 2.1 (Рис. 4.10).

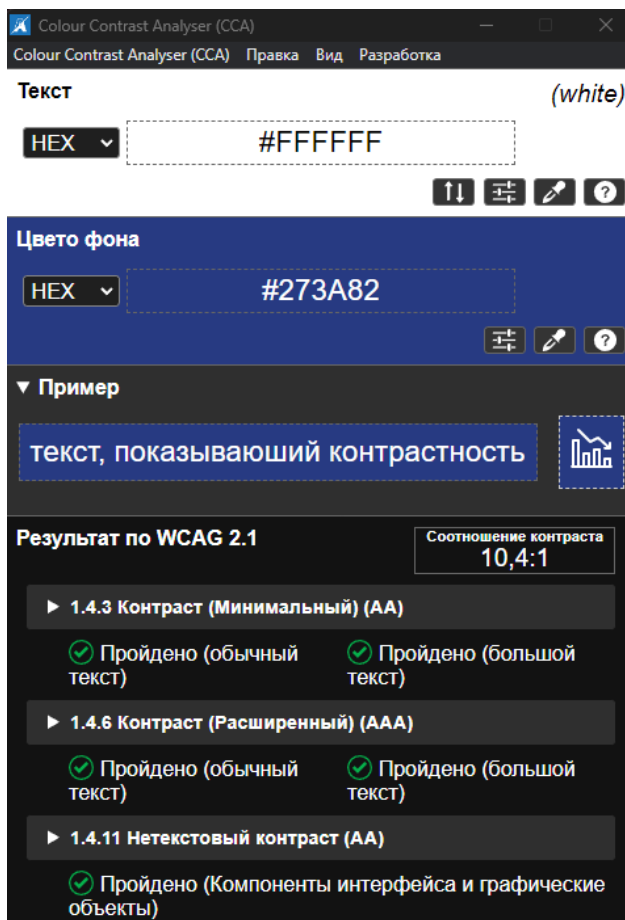


Рис. 4.10 - Перевірка контрастності

В постановці задачі також було визначено, що текст повинен не мати обмежень з масштабування.

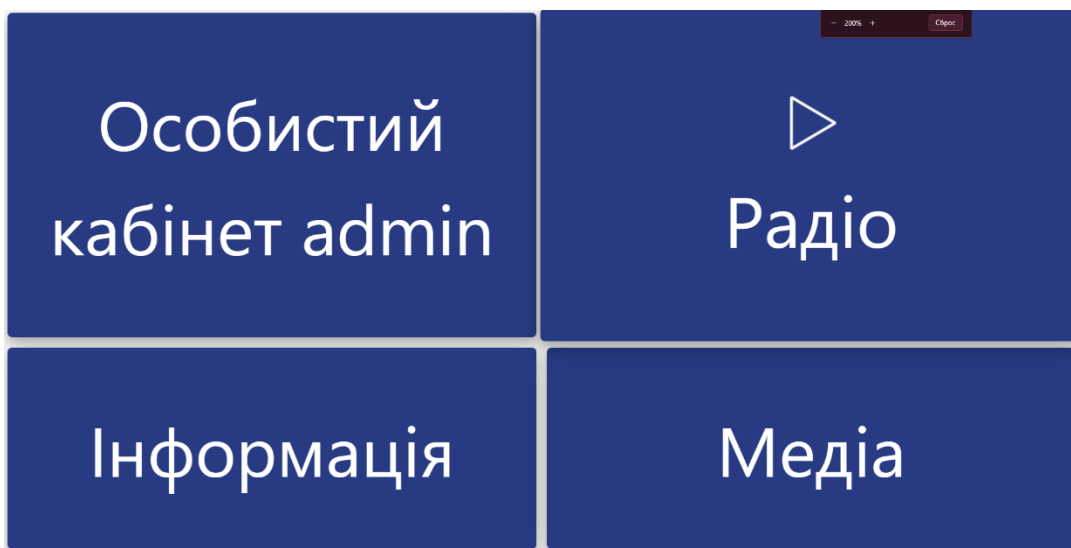


Рис. 4.11 - Текст веб-додатку у масштабі 200%

ВИСНОВОК

В результаті виконання кваліфікаційної роботи бакалавра було розроблено інформаційну систему для прослуховування аудіо книг людьми з вадами зору.

Було проведено аналітичний огляд в результаті якого було сформовано постановку задачі та визначено методи реалізації майбутньої інформаційної системи.

Було спроектовано майбутній веб-додаток, побудовано діаграму IDEF0 а також декомпозицію першого рівня для цієї діаграми. Було створено діаграму варіантів використання, яка визначає які дії може виконувати актор у додатку. Також було побудовано діаграми діяльності та спроектовано базу даних.

Було розроблено інтерфейс додатку, реалізовано масштабування інтерфейсу для різних пристроїв. Було закладено основну структуру проекту відповідно до архітектури MVC та реалізовано сторінки веб-додатку з огляду на постановку задачі.

Веб-додаток було перевірено програмою зчитування з екрану. Він дотримується наступних вимог:

- Інтерфейс не має області, недосяжні програмою зчитування з екрана;
- Вміст сторінки структурований заголовками;
- Заголовки дотримуються логічної послідовності ;
- Зображення мають альтернативний текст;
- Поля вводу мають відповідні підписи;
- Посилання мають доступний опис;
- Кнопки мають доступний опис;
- Інтерактивні позначені належним чином за допомогою відповідного елемента HTML.

Разом з цим інформаційна система дотримується наступних поставлених вимог:

- Інтерфейс інформаційної задовольняє вимогам WCAG 2.1 по контрастності кольорів. Відношення контрастності повинне переважати 7:1.
- Інтерактивні елементи застосунку, як і текст в цілому є великими
- Веб-сайт вдало адаптується під мобільні пристрої.
- Текст виділяється, та не має обмежень з масштабування.

СПИСОК ЛІТЕРАТУРИ

1. Міжнародний день незрячих людей : веб-сайт. URL: <http://labcentr.kr.ua/?p=14648#:~:text=За%20даними%20ВООЗ%2C%20Облизько%20300,них%20майже%2040%20мільйонів%20незрячі.>
(дата звернення: 20.08.2023).
2. Be My Eyes : веб-сайт. URL: <https://www.bemyeyes.com/language/ukrainian> (дата звернення: 20.08.2023).
3. Colour and contrast for people with sight loss. *RNIB* – С. 3-4. URL: https://media.rnib.org.uk/documents/Colour_and_contrast_for_people_with_sight_loss_2020.pdf (дата звернення: 20.08.2023).
4. Colour Contrast Analyzer (CCA) : веб-сайт. URL: <https://developer.paciellogroup.com/color-contrast-checker/> (дата звернення: 21.08.2023).
5. Web Content Accessibility Guidelines (WCAG) 2.1 : веб-сайт. URL: <https://www.w3.org/TR/WCAG21/> (дата звернення: 21.08.2023).
6. Інтерфейс для всіх. Чому варто зробити свій цифровий продукт інклюзивним : веб-сайт. URL: <https://prjctr.com/mag/why-inclusive>
(дата звернення: 21.08.2023).
7. About NonVisual Desktop Access : веб-сайт. URL: <https://www.nvaccess.org/about-nvda/> (дата звернення: 21.08.2023).
8. Results of the 2016 GOV.UK assistive technology survey : веб-сайт. URL: <https://accessibility.blog.gov.uk/2016/11/01/results-of-the-2016-gov-uk-assistive-technology-survey/> (дата звернення: 05.09.2023).
9. How to design websites for blind and partially sighted people : веб-сайт. URL: <https://www.usertesting.com/blog/website-design-for-blind-and-partially-sighted-people> (дата звернення: 05.09.2023).

10. MySight York : веб-сайт. URL: <https://www.mysightyork.org> (дата звернення: 30.08.2023).
11. Sumy GPS Inclusive : мобільний застосунок. URL: https://play.google.com/store/apps/details?id=appinventor.ai_serгей_svetaylo.SumyGPSInclusive&hl=ru&gl=US&pli=1 (дата звернення: 30.08.2023).
12. Спеціальні можливості в Samsung Galaxy : веб-сайт. URL: <https://www.samsung.com/ua/support/mobile-devices/accessibility/> (дата звернення: 31.08.2023).
13. Audible : веб-сайт. URL: <https://www.audible.com> (дата звернення: 03.08.2023).
14. Audiobooks.com : веб-сайт. URL: <https://www.audiobooks.com> (дата звернення: 03.08.2023).
15. Що таке ASP.NET і для чого використовується? : веб-сайт. URL: <https://hyperhost.ua/info/uk/shho-take-aspnet-i-dlya-cogo-vikoristovujetsya> (дата звернення: 28.04.2024)
16. Розмітка Razor : веб-сайт. URL: https://w3schoolsua.github.io/asp/webpages_razor.html#gsc.tab=0 (дата звернення 28.04.2024)
17. Entity Framework Core : веб-сайт. URL: <https://learn.microsoft.com/en-us/ef/core/> (дата звернення 28.04.2024)
18. Кондіус, В. В. (2022). Структурно-функціональне моделювання [Лекція]. Луцький національний університет імені Лесі Українки. https://elib.lntu.edu.ua/sites/default/files/elib_upload/Кондіус%202020готовва/page9.html
19. Іванов, О. О. (2005). Використання уніфікованої мови моделювання UML. В О. О. Іванов (Ред.), Об'єктно-орієнтоване програмування: Навчально-методичний посібник (с. 77-85). Київ: Національний

університет "Київська політехніка".

http://iwanoff.inf.ua/oop_kn/LabTraining05.html

20. Introduction to Identity on ASP.NET Core : веб-сайт. URL:

<https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-8.0&tabs=visual-studio> (дата звернення: 30.04.2024)

21. Побудова динамічних вебзастосунків за допомогою MVC : веб-сайт.

URL: <https://blog.ithillel.ua/articles/building-dynamic-web-applications-using-mvc> (дата звернення 30.04.2024)

22. GitHub Repository of SightFriend project : веб-сайт. URL:

<https://github.com/dellancelod/SightFriend>

ДОДАТОК А. ПРОГРАМНИЙ КОД

Program.cs

```

using Microsoft.AspNetCore.Http.Features;
using Microsoft.AspNetCore.Identity;
using Microsoft.EntityFrameworkCore;
using SightFriend.Data;
using SightFriend.Data.Repo.EntityFrameworkCore;
using SightFriend.Data.Repo.Interfaces;
using SightFriend.Services;
using System.Configuration;

var builder = WebApplication.CreateBuilder(args);

//Add services
builder.Services.AddTransient<ITextFieldRepository, EFExtFieldsRepository>();
builder.Services.AddTransient<INewsItemsRepository, EFNewsItemsRepository>();
builder.Services.AddTransient<IBooksItemsRepository, EFBooksItemsRepository>();
builder.Services.AddTransient<DataManager>();

//Configure Identity system
builder.Services.AddIdentity<IdentityUser, IdentityRole>(options =>
{
    options.Password.RequiredLength = 6;
    options.Password.RequireNonAlphanumeric = false;
    options.Password.RequireDigit = false;
    options.Password.RequireLowercase = false;
    options.Password.RequireUppercase = false;
}).AddEntityFrameworkStores<AppDbContext>()
.AddUserManager<CustomUserManager>()
.AddDefaultTokenProviders();

//Authentication cookies
builder.Services.ConfigureApplicationCookie(options =>
{
    options.Cookie.Name = "mriyaAuth";
    options.Cookie.HttpOnly = true;
    options.LoginPath = "/account/login";
    options.AccessDeniedPath = "/account/accessdenied";
    options.SlidingExpiration = true;
});

//Configure Authorization policy for admin area
builder.Services.AddAuthorization(x =>
{
    x.AddPolicy("AdminArea", policy => { policy.RequireRole("admin"); });
});

// Add services to the container.
builder.Services.AddControllersWithViews(x =>
{
    x.Conventions.Add(new AdminAreaAuthorization("Admin", "AdminArea"));
});

builder.Services.Configure<FormOptions>(options =>
{
    options.MultipartBodyLengthLimit = 512 * 1024 * 1024;
});

```

```

var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");

    // The default HSTS value is 30 days. You may want to change this for production
    scenarios, see https://aka.ms/aspnetcore-hsts.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseCookiePolicy();
app.UseAuthentication();
app.UseAuthorization();

app.UseEndpoints((endpoints) => {
    endpoints.MapControllerRoute(
        name: "admin",
        pattern: "{area:exists}/{controller=Home}/{action=Index}/{id?}"
    );
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}"
    );
});

app.Run();
    site.js

$(document).ready(function () {
    $("#phone-num-field").mask('+38(000)-000-0000', { placeholder: "+38(____)-____-____" });
});
    site.css

html {
    font-size: 14px;
}

@media (min-width: 768px) {
    html {
        font-size: 16px;
    }
}

html {
    position: relative;
    min-height: 100%;
}

body {
    margin-bottom: 60px;
}

```

```

}

/* Button Colors */
.red {
  color: white;
  background-color:red;
}

.blue {
  color: white;
  background-color: #273a82;
}

.green {
  color: white;
  background-color: green;
}

/* Main Grid */
.grid2x2 {
  position: absolute;
  top: 1%;
  bottom: 1%;
  right: 1%;
  left: 1%;
  font-size: 60px;
  margin-bottom: 0px;
  margin: 0px;
  display: grid;
}

.grid-row{
  display:flex;
  grid-column-gap: 1%;
}

.grid-column{
  display: flex;
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
  justify-content: center;
  align-items: center;
  flex: 1;
  margin-bottom: 1%;
}

.concentrate: hover {
  animation-name: grid-hover;
  animation-duration: 0.3s;
  animation-fill-mode: forwards;
}

/* Radio Button */
#radio-player {
  width: 100%;
  height: 100%;
  color: white;
  display: flex;
  justify-content: center;
  align-items: center;
}

```

```

}

#radio-player path{
  stroke-linejoin:bevel;
  stroke: white;
}

#radio-player button {
  padding: 0;
  border: 0;
  background: transparent;
  cursor: pointer;
  outline: none;
  width: 60px;
  height: 60px;
}

.bg-blink{
  animation: blink 3s infinite;
}

/* Login Form */
.login-form {
  position: absolute;
  top: 1%;
  bottom: 1%;
  right: 1%;
  left: 1%;
  font-size: 60px;
  margin: 0px;
  display: grid;
}

.login-row {
  display: flex;
}

.login-column {
  display: flex;
  padding: 1%;
  justify-content: left;
  align-items: center;
  flex: 1;
}

.login-column input {
  width:90%;
}

.login-button {
  border: none;
  margin: 1%;
  text-align: center;
  justify-content: center;
  user-select: text;
  text-decoration: none;
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
}

.login-button:hover{
  color: white;
}

/* Animations */
@-webkit-keyframes fadeIn {

```

```

    0% {
      opacity: 0;
    }

    75% {
      opacity: 0.6;
    }

    100% {
      opacity: 2;
    }
  }

  @keyframes blink {
    0% {
      background-color: #939cc0;
    }

    50% {
      background-color: #273a82;
    }

    100% {
      background-color: #939cc0;
    }
  }

  @keyframes grid-hover {

    0% {
      -webkit-transform: scale3d(1, 1, 1);
      transform: scale3d(1, 1, 1);
    }

    100% {
      -webkit-transform: scale3d(1.025, 1.025, 1.025);
      transform: scale3d(1.025, 1.025, 1.025);
    }
  }

  @keyframes fadeIn {
    0% {
      opacity: 0;
    }

    75% {
      opacity: 0.6;
    }

    100% {
      opacity: 2;
    }
  }

  @media only screen and (max-width: 700px) {
    .login-form{
      position: relative;
    }
    .grid-row {
      flex-wrap: wrap;
    }
  }

```

```

    .grid-column {
        flex-basis: 100%;
    }

    .login-row {
        flex-wrap: wrap;
    }

    .login-column {
        flex-basis: 100%;
    }
}

.admin-divbox{
    padding: 15px 0;
}

.admin-divbox label {
    display: block;
}

.admin-divbox a{
    display: block;
}

/* AudioBook list */
.grid-books {
    display: grid;
    margin: 1%;
    font-size: 60px;
    grid-template-columns: repeat(auto-fill, minmax(400px, 1fr)); /* Adjust column
width as needed */
    grid-auto-rows: 400px;
    grid-gap: 10px; /* Adjust gap between items as needed */
    justify-items: center;
}

.book-item {
    display: flex;
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
    justify-content: center;
    align-items: center;
    margin-bottom: 1%;
    width: 100%; /* Ensure each grid item spans the full width of the column */
    height: 100%; /* Ensure each grid item spans the full height of the row */
}

.book-cover {
    position: relative;
    background-size: cover; /* Cover the entire area with the background image */
    background-position: center; /* Center the background image */
}

.text-overlay {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    opacity: 90%;
    padding: 10px;
    border-radius: 5px;
}

```

```

    text-align: center;
}

.text-overlay h4 {
    margin-bottom: 5px;
}

.text-overlay p {
    margin: 0;
}

/* Apply blur effect */
.book-cover::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background: inherit;
    filter: blur(5px); /* Adjust blur strength as needed */
}

/* Styling for the "Go Back" button */
#back {
    z-index: 999; /* Ensure the button appears above other elements */
    min-width: auto; /* Set the width of the button */
    height: auto; /* Set the height of the button */
    display: flex;
    justify-content: center;
    align-items: center;
    font-size: 60px;
    padding: 10px;
    border-radius: 50%; /* Make the button round */
    box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
    aspect-ratio: 1 / 1; /* Set aspect ratio to 1:1 for a perfect circle */
}

```

BookItemsController.cs

```

using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.Extensions.Hosting;
using SightFriend.Data;
using SightFriend.Models;

namespace SightFriend.Areas.Admin.Controllers
{
    [Area("Admin")]
    public class BookItemsController : Controller
    {
        private readonly DataManager dataManager;
        private readonly IWebHostEnvironment hostEnvironment;
        public BookItemsController(DataManager dataManager, IWebHostEnvironment
hostEnvironment)
        {
            this.dataManager = dataManager;
            this.hostEnvironment = hostEnvironment;
        }

        //One view for Adding and Editing

```

```

    public IActionResult Edit(Guid id)
    {
        var entity = id == default ? new BookItem() :
dataManager.BooksItems.GetBookItemById(id);
        return View(entity);
    }
    [HttpPost]
    [RequestSizeLimit(512 * 1024 * 1024)]
    public IActionResult Edit(BookItem model, IFormFile audioBookFile, IFormFile
coverImageFile)
    {
        if (ModelState.IsValid)
        {
            if (audioBookFile != null)
            {
                if (audioBookFile.Length > 512 * 1024 * 1024)
                {
                    ModelState.AddModelError("audioBookFile", "Розмір аудіофайлу
перевищує 512Мб!");
                    return View(model);
                }

                model.AudioFilePath = audioBookFile.FileName;
                using (var stream = new
FileStream(Path.Combine(hostEnvironment.WebRootPath, "audio-books/",
audioBookFile.FileName), FileMode.Create))
                {
                    audioBookFile.CopyTo(stream);
                }
            }

            if (coverImageFile != null)
            {
                if (coverImageFile.Length > 512 * 1024 * 1024)
                {
                    ModelState.AddModelError("audioBookFile", "Розмір обкладинки
перевищує 512Мб!");
                    return View(model);
                }

                model.CoverImagePath = coverImageFile.FileName;
                using (var stream = new
FileStream(Path.Combine(hostEnvironment.WebRootPath, "bookcovers/",
coverImageFile.FileName), FileMode.Create))
                {
                    coverImageFile.CopyTo(stream);
                }
            }

            dataManager.BooksItems.SaveBookItem(model);
            return RedirectToAction(nameof(HomeController.Index),
nameof(HomeController).Replace("Controller", string.Empty));
        }

        return View(model);
    }
    [HttpPost]
    public IActionResult Delete(Guid id)
    {
        dataManager.BooksItems.DeleteBookItem(id);
        return RedirectToAction(nameof(HomeController.Index),
nameof(HomeController).Replace("Controller", string.Empty));
    }

```



```

    }
}
}

BookItem.cs

using Microsoft.AspNetCore.Mvc;
using System.ComponentModel.DataAnnotations;

namespace SightFriend.Models
{
    public class BookItem : EntityBase
    {
        [Required(ErrorMessage = "Заповніть назву")]
        [Remote(action: "IsTitleAvailable", controller: "BookItems", ErrorMessage =
"Назва вже існує")]
        public string Title { get; set; } = "Заголовок";
        [Required(ErrorMessage = "Заповніть автора")]
        public string Author { get; set; } = "Автор";
        [Required(ErrorMessage = "Вставте аудіофайл")]
        public string AudioFilePath { get; set; } = "Аудіо-Книга";
        public string CoverImagePath { get; set; } = "Обкладинка";
        public string Length { get; set; } = "Тривалість";
        public bool Hidden { get; set; }
    }
}

```

HomeController.cs

```

using Microsoft.AspNetCore.Mvc;
using SightFriend.Data;
using SightFriend.Models;
using System.Diagnostics;

namespace SightFriend.Controllers
{
    public class HomeController : Controller
    {
        private readonly DataManager dataManager;
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger, DataManager
dataManager)
        {
            _logger = logger;
            this.dataManager = dataManager;
        }

        public IActionResult Index()
        {
            if (HttpContext.User.Identity.IsAuthenticated)
            {
                ViewBag.Cabinet = "Особистий кабінет\n" +
HttpContext.User.Identity.Name;
            }
            else
            {
                ViewBag.Cabinet = "Авторизація";
            }
            ViewBag.MediaText =
dataManager.TextFields.GetTextFieldByCodeWord("PageMedia").Title;
            return View();
        }
    }
}

```

```

    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore
= true)]
    public IActionResult Error()
    {
        return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
    }

    public IActionResult RadioPartial()
    {
        return PartialView("_Radio");
    }

    public IActionResult InfoPartial()
    {
        return PartialView("_Info");
    }

    public IActionResult ColorBlindPartial()
    {
        return PartialView("_ColorBlind");
    }
}
}

```

DataManager.cs

```

using SightFriend.Data.Repo.Interfaces;

namespace SightFriend.Data
{
    public class DataManager
    {
        public ITextFieldsRepository TextFields { get; set; }
        public INewsItemsRepository NewsItems { get; set; }
        public IBooksItemsRepository BooksItems { get; set; }

        public DataManager(ITextFieldsRepository textFieldsRepository,
INewsItemsRepository newsItemsRepository, IBooksItemsRepository
booksItemsRepository)
        {
            NewsItems = newsItemsRepository;
            BooksItems = booksItemsRepository;
            TextFields = textFieldsRepository;
        }
    }
}

```

EntityBase.cs

```

using System.ComponentModel.DataAnnotations;

namespace SightFriend.Models
{
    public abstract class EntityBase
    {
        protected EntityBase() => DateAdded = DateTime.UtcNow;
        [Required]
        public virtual Guid Id { get; set; }
        [DataType(DataType.Time)]
        public DateTime DateAdded { get; set; }
    }
}

```

```

    }
}

```

ErrorViewModel.cs

```

namespace SightFriend.Models
{
    public class ErrorViewModel
    {
        public string? RequestId { get; set; }

        public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
    }
}

```

LoginViewModel.cs

```

using System.ComponentModel.DataAnnotations;

namespace SightFriend.Models
{
    public class LoginViewModel
    {
        [Required(ErrorMessage="Введіть номер телефону")]
        [Display(Name = "Номер телефону")]
        public string PhoneNumber { get; set; }

        [Required(ErrorMessage = "Введіть пароль")]
        [UIHint("Введіть пароль")]
        [Display(Name = "Пароль")]
        public string Password { get; set; }
    }
}

```

TextField.cs

```

using System.ComponentModel.DataAnnotations;

namespace SightFriend.Models
{
    public class TextField : EntityBase
    {
        [Required]
        public string CodeWord { get; set; }
        [Display(Name = "Назва розділу")]
        public string Title { get; set; } = "Інформаційна сторінка";
        [Display(Name = "Вміст сторінки")]
        public string Text { get; set; } = "Вміст заповнюється адміністратором";
    }
}

```

NewsItem.cs

```

using Microsoft.AspNetCore.Mvc;
using System.ComponentModel.DataAnnotations;

namespace SightFriend.Models
{
    public class NewsItem : EntityBase
    {
        [Required(ErrorMessage = "Заповніть заголовок новини")]
        [Remote(action: "IsTitleAvailable", controller: "NewsItems", ErrorMessage =
"Заголовок вже існує")]

```

```

        public string Title { get; set; } = "Заголовок";
        [Display(Name = "Опис")]
        public string Subtitle { get; set; }
        [Display(Name = "Зміст")]
        public string Text { get; set; } = "Вміст заповнюється адміністратором";
        public bool Hidden { get; set; }
    }
}

```

EFBookItemsRepository.cs

```

using Microsoft.EntityFrameworkCore;
using SightFriend.Data.Repo.Interfaces;
using SightFriend.Models;

namespace SightFriend.Data.Repo.EntityFramework
{
    public class EFBooksItemsRepository : IBooksItemsRepository
    {
        private readonly AppDbContext context;
        public EFBooksItemsRepository(AppDbContext context)
        {
            this.context = context;
        }
        public void DeleteBookItem(Guid id)
        {
            context.BookItems.Remove(new BookItem { Id = id });
            context.SaveChanges();
        }

        public void DeleteBookItem(BookItem BooksItem)
        {
            context.BookItems.Remove(BooksItem);
            context.SaveChanges();
        }

        public BookItem GetBookItemById(Guid id)
        {
            return context.BookItems.FirstOrDefault(x => x.Id == id);
        }

        public IQueryable<BookItem> GetBookItems()
        {
            return context.BookItems;
        }

        public void SaveBookItem(BookItem entity)
        {
            if (entity.Id == default)
            {
                context.Entry(entity).State = EntityState.Added;
            }
            else
            {
                context.Entry(entity).State = EntityState.Modified;
            }
            context.SaveChanges();
        }
    }
}

```

EFNewsItemsRepository.cs

```

using Microsoft.EntityFrameworkCore;
using SightFriend.Data.Repo.Interfaces;
using SightFriend.Models;

namespace SightFriend.Data.Repo.EntityFramework
{
    public class EFNewsItemsRepository : INewsItemsRepository
    {
        private readonly AppDbContext context;
        public EFNewsItemsRepository(AppDbContext context)
        {
            this.context = context;
        }
        public void DeleteNewsItem(Guid id)
        {
            context.NewsItems.Remove(new NewsItem { Id = id });
            context.SaveChanges();
        }

        public void DeleteNewsItem(NewsItem newsItem)
        {
            context.NewsItems.Remove(newsItem);
            context.SaveChanges();
        }

        public NewsItem GetNewsItemById(Guid id)
        {
            return context.NewsItems.FirstOrDefault(x => x.Id == id);
        }

        public IQueryable<NewsItem> GetNewsItems()
        {
            return context.NewsItems;
        }

        public void SaveNewsItem(NewsItem entity)
        {
            if (entity.Id == default)
            {
                context.Entry(entity).State = EntityState.Added;
            }
            else
            {
                context.Entry(entity).State = EntityState.Modified;
            }
            context.SaveChanges();
        }
    }
}

```

EfTextFieldsRepository.cs

```

using Microsoft.EntityFrameworkCore;
using SightFriend.Data.Repo.Interfaces;
using SightFriend.Models;

namespace SightFriend.Data.Repo.EntityFramework
{
    public class EFTextFieldsRepository : ITextFieldsRepository
    {
        private readonly AppDbContext context;
        public EFTextFieldsRepository(AppDbContext context)

```

```

    {
        this.context = context;
    }
    public void DeleteTextField(Guid id)
    {
        context.TextFields.Remove(new TextField { Id = id });
        context.SaveChanges();
    }

    public TextField GetTextFieldByCodeWord(string codeword)
    {
        return context.TextFields.FirstOrDefault(x => x.CodeWord == codeword);
    }

    public TextField GetTextFieldById(Guid id)
    {
        return context.TextFields.FirstOrDefault(x => x.Id == id);
    }

    public IQueryable<TextField> GetTextFields()
    {
        return context.TextFields;
    }

    public void SaveTextField(TextField entity)
    {
        if (entity.Id == default)
        {
            context.Entry(entity).State = EntityState.Added;
        }
        else
        {
            context.Entry(entity).State = EntityState.Modified;
        }
        context.SaveChanges();
    }
}
}

```

IBooksItemsRepository.cs

```

using SightFriend.Models;

namespace SightFriend.Data.Repo.Interfaces
{
    public interface IBooksItemsRepository
    {
        IQueryable<BookItem> GetBookItems();
        BookItem GetBookItemById(Guid id);
        void SaveBookItem(BookItem entity);
        void DeleteBookItem(Guid id);
        void DeleteBookItem(BookItem BooksItem);
    }
}

```

INewsItemsRepository.cs

```

using SightFriend.Models;

namespace SightFriend.Data.Repo.Interfaces
{
    public interface INewsItemsRepository
    {
        IQueryable<NewsItem> GetNewsItems();
    }
}

```



```

        Microsoft.AspNetCore.Identity.SignInResult result = await
signInManager.PasswordSignInAsync(user, model.Password, true, false);
        if (result.Succeeded)
        {
            return Redirect(returnUrl ?? "/");
        }
        ModelState.AddModelError(nameof(LoginViewModel.PhoneNumber),
"Невірний номер телефону або пароль");
    }
    return View(model);
}
[Authorize]
public async Task<IActionResult> Logout()
{
    await signInManager.SignOutAsync();
    return RedirectToAction("Index", "Home");
}
public IActionResult Index()
{
    if (!HttpContext.User.Identity.IsAuthenticated)
    {
        return Redirect("Login");
    }

    ViewBag.Username = HttpContext.User.Identity.Name;
    return View();
}
}
}

```

BooksItems->Edit.cshtml

```

@model BookItem
@{
    string strTitle = "Редагувати книгу";
    ViewBag.Title = strTitle;
}

<script src="~/js/ckeditor/ckeditor.js"></script>

<div>
    <h2>@strTitle</h2>
    <div>
        <form asp-area="Admin" asp-controller="BookItems" asp-action="Edit"
method="post" enctype="multipart/form-data">
            <input type="hidden" asp-for="Id" />
            <input type="hidden" asp-for="DateAdded" />

            <div asp-validation-summary="All"></div>
            <div class="admin-divbox">
                <label asp-for="Title"></label>
                <input asp-for="Title" />
                <span asp-validation-for="Title"></span>
            </div>
            <div class="admin-divbox">
                <label asp-for="Author"></label>
                <input asp-for="Author" />
                <span asp-validation-for="Author"></span>
            </div>
            <div>
                <label asp-for="AudioFilePath"></label>
                <input type="file" name="audioBookFile" id="audioBookFile" />
            </div>
        </form>
    </div>
</div>

```



```

        @if (!string.IsNullOrEmpty(Model.AudioFilePath))
        {
            <div>
                <audio controls>
                    <source src="~/audio-books/@Model.AudioFilePath"
type="audio/mp3">
                    Your browser does not support the audio element.
                </audio>
            </div>
        }
    </div>
    <div>
        <label asp-for="CoverImagePath"></label>
        <input type="file" name="coverImageFile" id="coverImageFile" />
        @if (!string.IsNullOrEmpty(Model.CoverImagePath))
        {
            <div>
                
            </div>
        }
    </div>
    <div>
        <label asp-for="Hidden"></label>
        <input asp-for="Hidden" type="checkbox" />
    </div>
    <input type="submit" value="Зберегти" />
</form>
<a asp-area="Admin" asp-controller="Home" asp-action="Index" asp-route-
id="">
    Відмінити редагування
</a>
</div>
</div>
    Home->Index.cshtml

@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <div class="grid2x2 animate__animated animate__backInDown ">
        <div class="grid-row">
            <div id="login" style="cursor: pointer;" class=" grid-column rounded
concentrate blue" onclick="window.location='/Account/Index';">
                @ViewBag.Cabinet
            </div>
            <div id="radio" class="grid-column rounded concentrate blue"></div>
        </div>
        <div class="grid-row">
            <div id="info" class="grid-column rounded concentrate
blue">Інформація</div>
            <div id="media" style="cursor: pointer;" class="grid-column rounded
concentrate blue" onclick="window.location='/Media/Index';">
                @ViewBag.MediaText
            </div>
        </div>
    </div>
</div>

@section Scripts{

```

```

<script>
    $('#radio').load("/Home/RadioPartial")

    var infoElement = document.getElementById('info');
    var voice = document.getElementById('voiceover');

    infoElement.addEventListener('click', function(){
        voice.src = "/voicelines/extended-info.mp3";
        voice.play()
    });
</script>
}

```

Areas->Admin->Home->Index.cshtml

```

@using SightFriend.Data
@model DataManager
@{
    string strTitle = "Панель Адміністратора";
    ViewBag.Title = strTitle;
    IQueryable<NewsItem> newsItems = Model.NewsItems.GetNewsItems();
    IQueryable<BookItem> bookItems = Model.BooksItems.GetBookItems();
}

<div>
    <h1>@strTitle</h1>
    <a asp-area="" asp-controller="Home" asp-action="Index" asp-route-id="">
        На головну
    </a>
    <div>
        <h3>Новини</h3>
        <div>
            <a asp-area="Admin" asp-controller="NewsItems" asp-action="Edit" asp-
route-id="">додати новину</a>
            <div>
                @if (newsItems.Any())
                {
                    <div>
                        @foreach (NewsItem entity in newsItems.OrderByDescending(x
=> x.DateAdded))
                        {
                            <div>
                                <a asp-area="" asp-controller="NewsItems" asp-
action="Index" asp-route-id="@entity.Id">
                                    @($"{entity.Title}")
                                </a>
                                <a asp-area="Admin" asp-controller="NewsItems" asp-
action="Edit" asp-route-id="@entity.Id">редагувати</a>
                                <form style="display: inline-block;" id="form-
@entity.Id" asp-area="Admin" asp-controller="NewsItems" asp-action="Delete">
                                    <input type="hidden" name="id" value="@entity.Id">
                                    <a href="#" onclick="document.getElementById('form-
@entity.Id').submit();">видалити</a>
                                </form>
                            </div>
                        }
                    </div>
                }
            </div>
        }
    </div>
</div>

```

```

</div>

<h3>Аудіо-Книги</h3>
<div>
  <a asp-area="Admin" asp-controller="BookItems" asp-action="Edit" asp-
route-id="">додати книгу</a>
  <div>
    @if (bookItems.Any())
    {
      <div>
        @foreach (BookItem entity in bookItems.OrderByDescending(x
=> x.DateAdded))
        {
          <div>
            <a asp-area="" asp-controller="BookItems" asp-
action="Index" asp-route-id="@entity.Id">
              @($"{entity.Title} {entity.Author}")
            </a>
            <a asp-area="Admin" asp-controller="BookItems" asp-
action="Edit" asp-route-id="@entity.Id">редагувати</a>
            <form style="display: inline-block;" id="form-
@entity.Id" asp-area="Admin" asp-controller="BookItems" asp-action="Delete">
              <input type="hidden" name="id"
value="@entity.Id">
              <a href="#"
onclick="document.getElementById('form-@entity.Id').submit();">видалити</a>
            </form>
          </div>
        }
      </div>
    }
  </div>
</div>

<div>
  <h3>Редагувати сторінки сайту</h3>
  <a asp-area="Admin" asp-controller="TextFields" asp-action="Edit" asp-
route-codeWord="PageMedia">@ViewBag.MediaText</a>
  <a asp-area="Admin" asp-controller="TextFields" asp-action="Edit" asp-
route-codeWord="PageNews">@ViewBag.NewsText</a>
  <a asp-area="Admin" asp-controller="TextFields" asp-action="Edit" asp-
route-codeWord="PageBooks">@ViewBag.BooksText</a>
</div>
<div>
  <form asp-area="" asp-controller="Account" asp-action="Logout"
method="post">
    <input type="submit" value="Вийти"/>
  </form>
</div>
</div>
</div>

```

Layout.cshtml

```

<!DOCTYPE html>
<html lang="en">
<head>
  @await Html.PartialAsync("MetatagsPartial")
  @await Html.PartialAsync("CssPartial")
  <script src='@Url.Content("~/js/player_blue.js")'
type="text/javascript"></script>
</head>
<body>

```

```

@*<header>
  Hello!
</header>*@
<div class="container">
  <audio id="voiceover">
    Your browser does not support the
    <code>audio</code> element.
  </audio>
  <main role="main" class="pb-3">
    @RenderBody()
  </main>

</div>

@await Html.PartialAsync("ScriptsPartial")
@await RenderSectionAsync("Scripts", required: false)

<div id="last-div"></div>

<script>
  $(document).ready(function(){
    $('#last-div').nextAll("center").remove();

    setTimeout(function(){
      $('#last-div').nextAll("div").remove();
    },100);

    setTimeout(function(){
      $('#last-div').nextAll("div").remove();
    },200);

    setTimeout(function () {
      $('#last-div').nextAll("div").remove();
    }, 400);
  });
</script>
</body>

</html>

```

BookItems->Show.cshtml

```

@model BookItem

<div>
  <h1>
    @Model.Title - @Model.Author
  </h1>

  <div id="player"></div>

</div>

<div id="back" style="cursor: pointer; position: absolute; bottom: 20px; right:
20px;" class="rounded-circle concentrate blue"
onclick="window.location='/BookItems/Index';">
  Назад
</div>

<script>

```

```

var audioFileName = "@Html.Raw(Model.AudioFilePath)";
var audioFilePath = "/audio-books/" + audioFileName; // Adjust the path as per
your folder structure

```

```

var player = new Playerjs({ id: "player", file: audioFilePath });
</script>

```

BookItems->Index.cshtml

```

@model IQueryable<BookItem>
<div class="text-center">
  <div id="booksContainer" class="animate__animated animate__backInDown grid-
books">
    @if (Model.Any())
    {
      @foreach (BookItem entity in Model)
      {
        if (entity.Hidden == false)
        {
          <a asp-area="" asp-controller="BookItems" asp-action="Index"
asp-route-id="@entity.Id" class="book-item rounded concentrate blue book-cover"
style="background-image: url('@Url.Content("~/bookcovers/" +
entity.CoverImagePath)');">
            <div class="book-cover">
              <div class="text-overlay blue">
                <p>@entity.Title</p>
                <h4>@entity.Author</h4>
              </div>
            </div>
          </a>
        }
      }
    }
  </div>
  <div id="back" style="cursor: pointer; position: absolute; bottom: 20px; right:
20px;" class="rounded-circle concentrate blue"
onclick="window.location='/Media/Index';">
    Назад
  </div>
</div>
<script>
var page = 1;
var loading = false;

function loadMoreBooks() {
  if (!loading) {
    loading = true;
    $.ajax({
      url: '@Url.Action("Index", "BooksItems")',
      type: 'GET',
      data: { page: page + 1 },
      success: function (data) {
        if (data.length > 0) {
          $('#booksContainer').append(data);
          page++;
        } else {
          $(window).off('scroll'); // No more items to load
        }
      }
    },
    complete: function () {

```

```
        loading = false;
    });
}

$(window).on('scroll', function () {
    if ($(window).scrollTop() + $(window).height() >= $(document).height()) {
        loadMoreBooks();
    }
});

// Load more books initially
loadMoreBooks();
</script>
```