

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра мережевих та інтернет технологій

ЗАТВЕРДЖУЮ

завідувач кафедри

мережевих та інтернет технологій

_____ **Юрій КРАВЧЕНКО**

«__» _____ 2023 року

КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА

галузі знань 17 «Електроніка та телекомунікації»
за спеціальністю 172 «Телекомунікації та радіотехніка»
освітньо-професійна програма «Мережеві та інтернет технології»

на тему:

РОЗРОБКА УНІВЕРСАЛЬНОГО МУЗИЧНОГО ВЕБ-ДОДАТКУ
НА МОВІ C#

Виконав: студент групи МІТ-41

Андрій ГРАБОВСЬКИЙ _____

Керівник:

доцент **Олександр ПЛЮЩ** _____

Київ 2023

Міністерство освіти і науки України
«Київський національний університет імені Тараса Шевченка»

Факультет інформаційних технологій
Кафедра мережевих та інтернет технологій

ЗАТВЕРДЖУЮ
завідувач кафедри
мережевих та інтернет технологій
Юрій КРАВЧЕНКО

«_____» _____ 2023 року

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ

Здобувачу вищої освіти Грабовському Андрію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи:

Розробка універсального музичного веб-додатку на мові с#

затверджена на засіданні кафедри МІТ «07» грудня 2022 р. протокол №5

2. Термін здачі закінченої роботи «31» травня 2023 р.

3. Вихідні дані до проекту
(роботи)

Сучасні технології створення додатків

4. Зміст пояснювальної записки (перелік питань, що їх потрібно розробити, обсяг – 35-40 сторінок)

Вступ

1. Дослідження методів та алгоритмів побудови додатків з графічною складовою. Постановка задачі

1.1 Мета дослідження

1.2 Об'єкт та предмет дослідження

1.3 Завдання та методи дослідження

1.4 Новизна роботи

2.. Огляд технологій С# для розробки веб-додатків

2.1 Веб-розробка та її особливості

2.2 Мова програмування С#

2.3 Технології ASP.NET, HTML, CSS та JavaScript для розробки веб-додатків

3. Аналіз вимог до додатку

3.1 Визначення функціональних та нефункціональних вимог до додатку

3.2 Визначення цільової аудиторії

4. Розробка додатку

4.1 Аналіз архітектури та проектування додатку

4.2 Створення об'єктів (MVC)

4.3 Створення веб-додатку в Visual Studio

Дата видачі завдання _____

Керівник роботи _____

Олександр ПЛЮЩ

(підпис)

(посада, прізвище, ім'я, по батькові)

Завдання прийняв до виконання _____

Андрій ГРАБОВСЬКИЙ

(підпис)

КАЛЕНДАРНИЙ ПЛАН ВИКОНАННЯ РОБОТИ

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Підготовчий	20.01.2023	
2	Розділ 1	15.02.2023	
3	Розділ 2	15.03.2023	
4	Розділ 3	15.04.2023	
5	Розділ 4	20.04.2023	
6	Доповідь та слайди	25.05.2023	
7	Пояснювальна записка	31.05.2023	

Здобувач вищої освіти _____ Андрій ГРАБОВСЬКИЙ
(підпис)

Керівник _____ Олександр ПЛЮЩ
(підпис)

РЕФЕРАТ

Пояснювальна записка: 39 с, 18 рис, 0 таблиць, 0 додатків, 7 джерел.

Об'єкт дослідження: Технології розробки універсальних музичного веб-додатку на мові С#.

Предмет дослідження: Різні веб-застосування для забезпечення потреб інтернет користувачів .

Мета роботи (дослідження): Створити працюючий універсальний музичний веб-додаток з використанням мови програмування С# та супутніх технологій. Розробити функціональний та естетично привабливий додаток, який забезпечуватиме користувачам зручний спосіб прослуховування та взаємодії з музичним вмістом через веб-інтерфейс.

Методи дослідження: Аналіз вимог до додатку, проектування структури та функціоналу, розробка додатку з використанням мови програмування С# та веб-технологій, тестування та вдосконалення.

В роботі розроблено універсальний музичний веб-додаток, який надає можливість користувачам знайти, прослухати та зберігати музичні треки, створювати власні плейлисти, ділитися музикою з іншими користувачами. Використано мову програмування С# для реалізації серверної логіки, використовуючи фреймворк ASP.NET та технології веб-розробки, такі як HTML, CSS та JavaScript.

Практичне значення роботи полягає у створенні функціонального музичного веб-додатку, який може бути використаний у різних сферах, таких як розваги, освіта або комерційні потреби. Додаток дозволяє користувачам зручно насолоджуватися музикою, взаємодіяти з музичним вмістом та встановлювати особисті налаштування. Результати дослідження можуть бути використані для покращення та розширення функціоналу подібних веб-додатків у майбутньому.

Ключові слова: Універсальний музичний веб-додаток, С#, ASP.NET, HTML, CSS, JavaScript, музичний веб-контент.

Abstract

Explanatory Note: 39 p, 18 photos, 0 tables, 0 appendices, 7 references.

Research Object: Development of a Universal Music Web Application using C#.

Research Goal: To explore the possibilities of developing a universal music web application using the C# programming language and related technologies. To create a functional and aesthetically appealing application that provides users with a convenient way to listen to and interact with music content through a web interface.

Research Methods: Analysis of application requirements, design of structure and functionality, development of the application using C# programming language and web technologies, such as ASP.NET, HTML, CSS, and JavaScript, testing and refinement.

Developed: A universal music web application that allows users to find, listen to, and save music tracks, create their own playlists, and share music with other users. The C# programming language was utilized for implementing the server-side logic, using the ASP.NET framework and web development technologies such as HTML, CSS, and JavaScript.

Practical Significance: The significance of this work lies in the creation of a functional music web application that can be used in various domains such as entertainment, education, or commercial needs. The application enables users to conveniently enjoy music, interact with music content, and customize their preferences. The research findings can be applied to enhance and expand the functionality of similar web applications in the future.

Keywords: Universal music web-application, C#, ASP.NET, HTML, CSS, JavaScript, music content.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ОБ’ЄКТ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	9
1.1 Мета дослідження.....	9
1.2 Об’єкт та предмет дослідження.....	9
1.3 Завдання та методи дослідження.....	10
1.4 Новизна роботи.....	10
РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ВЕБ-ДОДАТКІВ...12	
2.1 Веб-розробка та її особливості.....	12
2.2 Мова програмування C#.....	14
2.3 Технології ASP.Net, HTML, CSS, Javascript для розробки веб-додатків.....	16
2.4 Microsoft Visual Studio.....	18
РОЗДІЛ 3. АНАЛІЗ ВИМОГ ДО ДОДАТКУ.....19	
3.1 Визначення функціональних та нефункціональних вимог до додатку.....	19
3.2 Визначення цільової аудиторії.....	20
РОЗДІЛ 4. РОЗРОБКА ДОДАТКУ.....21	
4.1 Створення проекту в Visual Studio та його налаштування.....	21
4.2 Створення об’єктів (MVC).....	31
4.3 Створення веб-додатку в Visual Studio.....	32
ВИСНОВКИ.....	37
ПЕРЕЛІК ПОСИЛАНЬ.....	39

ВСТУП

У сучасному світі музика є невід'ємною частиною нашого життя. Вона викликає емоції, надихає, об'єднує людей та створює неповторну атмосферу. Завдяки стрімкому розвитку технологій, веб-додатки стали новим зручним засобом споживання музики та отримання актуальної інформації.

Метою цієї дипломної роботи є розробка універсального музичного веб-додатку на мові C#, який надасть користувачам можливість насолоджуватись музикою та отримувати оновлення з новин музичного світу. Головна мета полягає у створенні веб-додатку зі зручним інтерфейсом, що дозволить користувачам прослуховувати музику, переглядати тексти пісень та отримувати свіжі новини про улюблених виконавців та музичні події.

У даному вступі будуть представлені об'єкт дослідження, мета роботи та структура дипломної роботи для кращого розуміння контексту та завдань, поставлених перед дослідником. Зазначені завдання та методи дослідження сприятимуть досягненню мети та розробці універсального музичного веб-додатку зі зручним інтерфейсом та актуальним контентом.

Результати цієї дипломної роботи можуть бути корисні для розробників веб-додатків, які прагнуть створити зручні та функціональні інструменти для музичного споживання та отримання новин музичного світу. Крім того, дані дослідження можуть бути використані в розвитку інших веб-додатків, що спрямовані на надання інформації про різні аспекти музики та її виконавців.

Огляд технологій та аналіз вимог до додатку дозволять визначити найефективніші та найзручніші інструменти для розробки універсального музичного веб-додатку. Розділ 1 розкриє об'єкт та методи дослідження, а розділ 2 присвятиться огляду технологій, що використовуються для розробки веб-додатків з музичною тематикою. Розділ 3 аналізує вимоги до додатку та визначає цільову аудиторію. Розділ 4 буде присвячений розробці самого додатку, включаючи моделювання інтерфейсу, реалізацію функціональності та тестування.

Усі ці кроки спрямовані на досягнення мети розробки універсального музичного веб-додатку зі зручним інтерфейсом, який надасть користувачам можливість насолоджуватись музикою та отримувати актуальну інформацію з музичного світу. В результаті виконання дипломної роботи користувачі матимуть доступ до зручного та привабливого веб-додатку, який поєднає функції прослуховування музики та перегляду новин, забезпечуючи незабутній досвід взаємодії зі світом музики.

РОЗДІЛ 1. ОБ'ЄКТ ТА МЕТОДИ ДОСЛІДЖЕННЯ

1.1 Мета дослідження

Метою дослідження є розробка універсального музичного веб-додатку на мові C#, спрямованого на надання користувачам зручного інтерфейсу для прослуховування музики та отримання актуальної інформації про новини музичного світу. Основна мета полягає у створенні функціонального та привабливого додатку, що задовольнятиме потреби користувачів у зручному споживанні музики та отриманні оновлень про улюблених виконавців та музичні події.

1.2 Об'єкт та предмет дослідження

Об'єктом дослідження є музичний веб-додаток, розроблений на мові C#. Предметом дослідження є функціональність та ефективність додатку, його інтерфейс та можливості для прослуховування музики та отримання актуальних новин музичного світу. Дослідження також спрямоване на вивчення технологій, що використовуються для розробки веб-додатків, зокрема мови програмування C#.

1.3 Завдання та методи дослідження

Для досягнення мети дослідження визначено наступні завдання:

- Аналіз вимог до музичного веб-додатку: оцінка потреб користувачів у функціональних можливостях, інтерфейсі та способах взаємодії з музикою та новинами.
- Огляд технологій: дослідження та аналіз доступних технологій та інструментів для розробки веб-додатків на мові C#, зокрема для роботи зі звуком, медіа та веб-інтерфейсом.
- Проектування інтерфейсу додатку: розробка зручного та естетичного користувацького інтерфейсу для прослуховування музики та отримання новин, з урахуванням потреб та звичок цільової аудиторії.

- Розробка функціональності додатку: імплементація основних функцій, таких як пошук і відтворення музики, перегляд новин та музичних подій, збереження вибраних треків та інше.

- Тестування та оптимізація додатку: проведення тестів для перевірки працездатності та стабільності додатку, а також виявлення та виправлення можливих помилок та недоліків.

1.4 Новизна роботи

Новизною роботи є поєднання функціональності прослуховування музики та отримання новин музичного світу у зручному веб-додатку, розробленому на мові програмування C#. Використання цієї технології та створення інтуїтивно зрозумілого інтерфейсу дозволяють користувачам насолоджуватись музикою та отримувати оновлення про музичні новини з одного зручного джерела.

В цьому розділі було розглянуто об'єкт та методи дослідження, а також визначено мету дослідження. Метою дослідження є розробка універсального музичного веб-додатку на мові C# з врахуванням особливостей веб-розробки та використання технологій ASP.NET, HTML, CSS та JavaScript.

Об'єктом дослідження є сам додаток, який має надавати користувачам зручний інтерфейс для прослуховування музики, перегляду новин світу та отримання текстів пісень. Предметом дослідження є вивчення технологій C# та веб-розробки для реалізації функцій додатку.

У роботі були поставлені конкретні завдання, які включають розробку архітектури додатку, імплементацію функцій реєстрації та авторизації користувачів, інтеграцію з зовнішніми сервісами для отримання новин, пошук та програвання музики, а також відображення текстів пісень.

Для досягнення мети дослідження та виконання поставлених завдань були використані методи аналізу літературних джерел, вивчення документації технологій, розробка та тестування програмного коду. Ці методи дозволили здійснити повноцінне дослідження та розробку музичного веб-додатку.

Однією з ключових особливостей роботи є новизна додатку, який поєднує можливість прослуховування музики, перегляду новин світу та отримання текстів

пісень в одному зручному інтерфейсі. Розроблений додаток відповідає сучасним вимогам і може забезпечити задоволення потреб користувачів.

РОЗДІЛ 2. ОГЛЯД ТЕХНОЛОГІЙ C# ДЛЯ РОЗРОБКИ ВЕБ-ДОДАТКІВ

2.1 Веб-розробка та її особливості

Веб-розробка - це широке поле, що охоплює процес створення веб-додатків, які доступні через Інтернет. Веб-додатки забезпечують користувачам можливість спілкуватися, отримувати інформацію, виконувати різноманітні завдання та взаємодіяти з веб-сервером.

Однією з особливостей веб-розробки є клієнт-серверна архітектура. У цій моделі веб-додаток складається з двох основних компонентів: клієнтської сторони, яка відображається на браузері або веб-клієнті, і серверної сторони, яка забезпечує обробку запитів та надсилання відповідей.

Ще одна важлива особливість - це HTTP-протокол, який використовується для обміну даними між клієнтом і сервером. Веб-розробники повинні мати розуміння принципів цього протоколу та його методів, таких як GET і POST, для ефективної комунікації між клієнтом і сервером.

Різні браузери також мають свої особливості та підтримують різні стандарти. Веб-розробники повинні бути враховувати ці особливості та забезпечувати сумісність своїх веб-додатків з різними браузерами, щоб користувачі могли однаково комфортно використовувати додаток на будь-якому браузері.

Респонсивний дизайн є ще однією важливою складовою веб-розробки. Завдяки респонсивному дизайну веб-додаток адаптується до різних пристроїв та розмірів екранів, забезпечуючи оптимальне відображення і зручну взаємодію для користувачів на комп'ютерах, планшетах та мобільних пристроях.

2.2 Мова програмування C#

Мова програмування C# (C-Sharp) є однією з основних мов для розробки веб-додатків на платформі .NET. Вона була розроблена компанією Microsoft і має багато переваг, що роблять її привабливою для веб-розробників.

Однією з головних переваг C# є його простота і легкість вивчення. Вона базується на синтаксисі мови програмування C++, але має більш просту і зрозумілу структуру. Це робить C# доступною для новачків і дозволяє розробникам швидко оволодіти мовою.

C# є типізованою мовою програмування, що означає, що змінні мають типи даних, які повинні бути визначені перед використанням. Це допомагає забезпечити безпеку даних і полегшує виявлення помилок на ранніх етапах розробки.

Великою перевагою C# є його інтеграція з платформою .NET. .NET Framework надає широкий набір бібліотек і інструментів для розробки веб-додатків, що полегшує роботу з базами даних, мережевими операціями, безпекою та іншими аспектами веб-розробки.

C# також підтримує об'єктно-орієнтований підхід до програмування, що дозволяє розробникам створювати модульний і розширюваний код. Об'єктно-орієнтоване програмування дозволяє розбити програму на окремі класи, що забезпечує зручність управління кодом і полегшує його повторне використання.

Крім того, C# підтримує асинхронне програмування, що дозволяє ефективно взаємодіяти з асинхронними операціями, такими як мережеві запити, що поліпшує продуктивність та відзивчивість веб-додатків.

-Технологія ASP.NET є фреймворком розробки веб-додатків, розробленим компанією Microsoft. Вона надає розробникам потужні та ефективні інструменти для створення високоякісних веб-додатків з використанням мови програмування C#.

Основні особливості технології ASP.NET включають:

Мова програмування C#: ASP.NET базується на мові програмування C#, що дозволяє розробникам використовувати потужну та експресивну мову для створення веб-додатків. C# є однією з найпоширеніших мов програмування, що має сильну типізацію та широкі можливості для роботи з об'єктами, керування пам'яттю, операціями з базами даних та багато іншого.

Модель програмування на основі подій (Event-driven programming): ASP.NET використовує модель програмування на основі подій, що дозволяє розробникам реагувати на події, які відбуваються в рамках веб-додатку, наприклад, натискання кнопок, введення даних користувачем тощо. Це дозволяє створювати інтерактивні та реактивні веб-додатки з багатофункціональними можливостями.

Розділення логіки та представлення (Separation of Concerns): ASP.NET сприяє розділенню логіки додатку та його представлення. Це дає можливість створювати шаровану архітектуру, де бізнес-логіка та представлення веб-сторінок знаходяться в окремих компонентах. Такий підхід спрощує розробку, тестування та підтримку додатків.

Інтеграція зі середовищем Visual Studio: ASP.NET повністю інтегрований з середовищем розробки Visual Studio, що надає розробникам потужні інструменти для швидкої розробки, налагодження та розгортання веб-додатків. Visual Studio надає широкі можливості автоматизації, рефакторингу коду, візуального дизайну інтерфейсу користувача та багато іншого, що сприяє покращенню продуктивності розробника.

Підтримка безпеки: ASP.NET надає механізми для забезпечення безпеки веб-додатків, включаючи автентифікацію, авторизацію, захист від атак та кросс-сайтового скриптування (XSS). Це дозволяє розробникам захистити дані та функціональність своїх додатків та забезпечити безпеку користувачів.

Використання технології ASP.NET разом з HTML, CSS та JavaScript дозволяє створювати потужні та інтерактивні веб-додатки зі зручним інтерфейсом та багатофункціональними можливостями. Ці технології популярні серед розробників та забезпечують широкі можливості для створення як невеликих веб-сайтів, так і складних корпоративних додатків.

-HTML (HyperText Markup Language) є основним мовою розмітки для створення веб-сторінок та веб-додатків. Вона використовується для структурування та оформлення вмісту веб-сторінки, такого як текст, зображення, посилання, таблиці, форми та багато іншого. HTML базується на тегах, що визначають роль та вигляд різних елементів на сторінці.

Розмітка HTML надає можливість створювати структуровану розмітку для веб-сторінок. Використовуючи HTML-теги, розробники можуть вказувати, які елементи є заголовками, абзацами, списками, таблицями тощо. Це дозволяє організувати вміст сторінки зрозумілим способом і полегшує подальшу стилізацію та маніпулювання елементами за допомогою CSS та JavaScript.

Теги: HTML використовує теги для визначення ролі та вигляду елементів. Кожен тег починається з відкриваючого символу < та закривається закриваючим символом >. Наприклад, <p> використовується для визначення абзаців, - для вставки зображень, <a> - для створення посилань тощо. Теги можуть мати атрибути, що додають додаткові властивості елементам.

Вкладеність та ієрархія: HTML дозволяє вкладати елементи один в одного, утворюючи ієрархію. Наприклад, абзаци можуть містити посилання або зображення всередині себе. Це дає можливість створювати складніші структури сторінок та організувати їх зрозумілим способом.

Форми: HTML надає можливість створювати форми для збору даних від користувачів. За допомогою різних типів полів вводу, таких як текстові поля, флажки, кнопки, випадаючі списки тощо, розробники можуть отримувати дані від користувачів і використовувати їх для подальшої обробки.

Семантична розмітка: HTML також надає можливість семантичної розмітки, що дозволяє визначати смислову роль елементів. Наприклад, використання тега <header> для заголовка сторінки або <nav> для навігаційного меню допомагає пошуковим системам та іншим програмам краще розуміти структуру сторінки та забезпечує кращу доступність та інтерпретацію контенту.

HTML є фундаментальною складовою веб-розробки і необхідним інструментом для створення веб-додатків. Розробка на основі HTML дозволяє створювати доступні, структуровані та зручні для використання веб-сторінки та додатки.

-CSS (Cascading Style Sheets) є мовою стилізації, яка використовується для задання вигляду та оформлення елементів веб-сторінок, створених з використанням HTML. CSS визначає, які кольори, шрифти, розміри, відступи, рамки та інші властивості повинні мати різні елементи на сторінці.

Основні особливості CSS:

Розділення змісту та вигляду: CSS відокремлює структуру та вміст веб-сторінки від її вигляду. Це означає, що HTML використовується для розмітки та організації елементів, а CSS використовується для визначення їх зовнішнього

вигляду. Це спрощує управління стилями та забезпечує однорідний вигляд для багатьох сторінок на сайті.

Використання селекторів: CSS використовує селектори для вибору елементів, до яких будуть застосовуватися стилі. Селектори можуть бути базовими (наприклад, вибір за назвою тега або класом) або більш складними (з використанням ієрархії елементів, атрибутів, псевдокласів тощо). Це дозволяє точно налаштувати, які елементи будуть стилізовані.

Каскадність та спадковість: CSS використовує концепцію каскадності, що дозволяє визначати пріоритетність стилів. Якщо одному елементу задаються декілька правил, вони можуть бути виконані в порядку визначеної специфічності або пріоритету. Крім того, CSS використовує спадковість, що дозволяє наслідувати стилі від батьківських елементів. Наприклад, стиль, заданий для загального контейнера, може автоматично застосовуватися до всіх вкладених елементів.

Можливості оформлення: CSS надає широкий спектр можливостей для оформлення елементів. Ви можете задавати кольори, фони, шрифти, розміри, відступи, рамки, тіні, анімацію та інші властивості. CSS також підтримує медіа-запити, які дозволяють задавати різні стилі для різних пристроїв та розмірів екранів.

Підтримка розташування та розміщення: CSS надає різні методи для розташування та розміщення елементів на сторінці. Ви можете використовувати плаваючі елементи, гнучкі контейнери, сітки, вирівнювання та інші техніки для створення бажаного макету сторінки.

CSS є потужним інструментом для стилізації та оформлення веб-додатків. Використання CSS разом з HTML та JavaScript дозволяє створювати зручні, естетично привабливі та користувачам зрозумілі веб-інтерфейси.

-JavaScript є мовою програмування, що використовується для розробки динамічних та інтерактивних веб-додатків. Вона працює у браузері користувача та надає можливість взаємодії зі сторінкою, змінюючи її вміст, реагуючи на події та виконуючи скрипти.

Основні особливості JavaScript:

Клієнтська взаємодія: JavaScript дозволяє створювати динамічні елементи на веб-сторінці та реагувати на дії користувача. Ви можете створювати валідацію форм, взаємодію з кнопками, створювати анімацію, маніпулювати елементами сторінки та багато іншого.

Маніпуляція з DOM: JavaScript дозволяє змінювати структуру та вміст веб-сторінки, використовуючи DOM (Document Object Model). Ви можете динамічно створювати, змінювати або видаляти елементи сторінки, змінювати їх атрибути та стилі, маніпулювати текстом та іншими властивостями елементів.

Обробка подій: JavaScript дозволяє реагувати на різні події, такі як клік миші, натискання клавіші, завантаження сторінки та інші. Ви можете призначати обробники подій, які виконують певні дії при виникненні певної події, такі як відправка форми, зміна значення поля вводу, перехід по посиланню тощо.

Робота з AJAX: JavaScript дозволяє виконувати асинхронні HTTP-запити до сервера без перезавантаження сторінки. Це дозволяє завантажувати або відправляти дані на сервер без втрати поточного стану сторінки, що забезпечує швидку та плавну взаємодію з користувачем.

Валідація та перевірка даних: JavaScript надає можливості для валідації введених даних на стороні клієнта, перевірки правильності заповнення форм, обробки помилок та надання зрозумілих повідомлень користувачеві.

JavaScript є однією з ключових технологій для розробки веб-додатків, оскільки вона дозволяє зробити сторінку більш динамічною, інтерактивною та користувачам зрозумілою. Інтеграція JavaScript разом з HTML та CSS дозволяє створювати повноцінні веб-додатки зі зручним інтерфейсом та багатофункціональними можливостями.

Веб-розробка є складним процесом, який вимагає знань різних технологій та інструментів. Вона включає розробку функціональних та дизайнерських елементів, взаємодію з базами даних, роботу з серверною та клієнтською частинами додатка. Мова програмування C# відіграє важливу роль у веб-розробці,

оскільки вона надає потужні можливості для створення динамічних та масштабованих веб-додатків.

Технології ASP.NET, HTML, CSS та JavaScript є ключовими компонентами веб-розробки з використанням C#. ASP.NET надає фреймворк для створення веб-додатків, забезпечуючи потужність, швидкодію та безпеку. HTML і CSS використовуються для створення структури та оформлення веб-сторінок, а JavaScript дозволяє реалізувати інтерактивність та динамічні функції на клієнтській стороні.

Огляд технологій C# для розробки веб-додатків допоміг визначити потрібні компоненти та інструменти для подальшої реалізації проекту музичного веб-додатку.

У цьому розділі було проведено огляд технологій C#, пов'язаних з розробкою веб-додатків. Було розглянуто особливості веб-розробки та визначено роль мови програмування C# у цьому контексті. Також було представлено технології ASP.NET, HTML, CSS та JavaScript, які є основними компонентами для розробки веб-додатків з використанням C#.

2.4 Microsoft Visual Studio

Visual Studio є однією з найпопулярніших інтегрованих середовищ розробки (IDE) для програмістів. Ось кілька переваг, які він надає:

Багатофункціональність: Visual Studio підтримує широкий спектр мов програмування, включаючи C++, C#, Visual Basic, F#, Python та багато інших. Він також надає розширення для роботи з різними технологіями, такими як веб-розробка, мобільна розробка, хмарні послуги та штучний інтелект.

Багатоінструментальність: Visual Studio має велику кількість інструментів для покращення продуктивності розробника, таких як автодоповнення коду, налагоджувач, система контролю версій, підтримка тестування, аналіз коду та багато інших. Ці інструменти сприяють швидкій і ефективній розробці програмного забезпечення.

Інтеграція з платформами Microsoft: Visual Studio щільно інтегрований з платформами Microsoft, такими як Windows, Azure, Office, SQL Server та інші. Це

дозволяє розробникам легко створювати програми, які взаємодіють з цими платформами і використовують їх функціонал.

Екосистема розширень: Visual Studio має велику і активну спільноту розробників, яка надає безліч розширень для покращення функціональності IDE. Розширення дозволяють додати нові інструменти, шаблони проектів, підтримку інших мов програмування та багато іншого.

Висока продуктивність: Visual Studio відомий своєю продуктивністю та ефективністю роботи навіть з великими проектами. Він пропонує оптимізації, які допомагають прискорити компіляцію та виконання програм.

Розробка для різних платформ: Visual Studio підтримує розробку програмного забезпечення для різних платформ, включаючи Windows, macOS, iOS, Android, Linux та інші. Це дозволяє розробникам створювати кросплатформенні додатки та сервіси.

Інструменти для командної розробки: Visual Studio надає рішення для командної розробки, які дозволяють спільно працювати над проектами, керувати версіями коду, вирішувати конфлікти та забезпечувати спільний доступ до ресурсів.

Враховуючи ці переваги, Visual Studio є популярним вибором серед програмістів для розробки різноманітних програм та додатків, через це ми його й будемо використовувати.

РОЗДІЛ 3. АНАЛІЗ ВИМОГ ДО ДОДАТКУ

3.1 Визначення функціональних та нефункціональних вимог до додатку.

Для успішної розробки універсального музичного веб-додатку мовою C# необхідно проаналізувати та визначити функціональні та нефункціональні вимоги до додатку. Цей аналіз допомагає визначити ключові функціональні можливості, які повинен надавати додаток, а також необхідні параметри і вимоги до його продуктивності, надійності та інших характеристик.

Функціональні вимоги описують конкретні функції та можливості, які повинен мати музичний веб-додаток:

-Відображення світових новин: Додаток повинен надавати доступ до актуальних новин з різних джерел і відображати їх користувачам.

-Відтворення музики: Додаток повинен дозволяти користувачам відтворювати їх за допомогою зручного аудіоплеєра.

-Відображення текстів пісень: Додаток повинен надавати користувачам тексти пісень для відображення під час прослуховування музики.

Нефункціональні вимоги описують якісні характеристики та обмеження, яким повинен відповідати додаток. Деякі приклади нефункціональних вимог можуть включати:

-Продуктивність та ефективність: Додаток повинен працювати швидко і ефективно, навіть з великим обсягом музичних треків і новинного контенту.

-Користувацький інтерфейс та досвід користувача: Додаток повинен мати зручний і візуально привабливий інтерфейс, який забезпечує легку навігацію і зручність використання для користувачів.

-Безпека даних: Додаток повинен забезпечувати захист приватності користувачів, персональних даних та безпечне зберігання музичних треків.

3.2 Визначення цільової аудиторії

З огляду на різноманітність пісень для всіх вікових груп та різних жанрів, а також відсутність географічних обмежень, визначення цільової аудиторії виявляється ключовим кроком для успішної розробки та позиціонування додатку на ринку.

Вікова група: В універсальному музичному веб-додатку, який має пісні для всіх вікових груп, цільовою аудиторією можуть бути діти, підлітки, молодь, дорослі та літні люди. Додаток має надавати можливість відібрати пісні відповідно до вікових уподобань користувача.

Музичні уподобання: Оскільки додаток пропонує пісні різних жанрів, цільова аудиторія може включати любителів популярних жанрів, таких як поп, рок, електронна музика, а також фанатів специфічних жанрів, наприклад, джазу, класичної музики, хіп-хопу. Додаток повинен забезпечувати можливість знаходження та прослуховування пісень усіх жанрів.

Технічна грамотність: Оскільки додаток має веб-інтерфейс, цільова аудиторія може включати користувачів з різним рівнем технічної грамотності. Додаток має мати простий та зрозумілий інтерфейс, що дозволить зручно користуватися навіть людям з обмеженим досвідом роботи з веб-додатками.

Географічні обмеження: Зважаючи на відсутність географічних обмежень, цільова аудиторія може бути глобальною. Додаток має бути доступним для користувачів з будь-якої країни та надавати можливість вибору мови та локалізації контенту.

Визначення цільової аудиторії допоможе зорієнтувати розробку додатку та створити зручні умови для різних категорій користувачів. Враховуючи потреби та вподобання цільової аудиторії, додаток зможе ефективно задовольняти їх очікування та стати популярним серед широкого кола користувачів.

РОЗДІЛ 4. РОЗРОБКА ДОДАТКУ

4.1 Створення проекту в Visual Studio та його налаштування.

Створення проекту в Microsoft Visual Studio є першим кроком у розробці універсального музичного веб-додатку на мові C#. Microsoft Visual Studio є інтегрованою середовищем розробки (IDE), яке надає розширені інструменти, ресурси та функціональні можливості для створення програмного забезпечення.

Для створення проекту в Microsoft Visual Studio, виконуються наступні кроки:

Відкриваємо Microsoft Visual Studio :

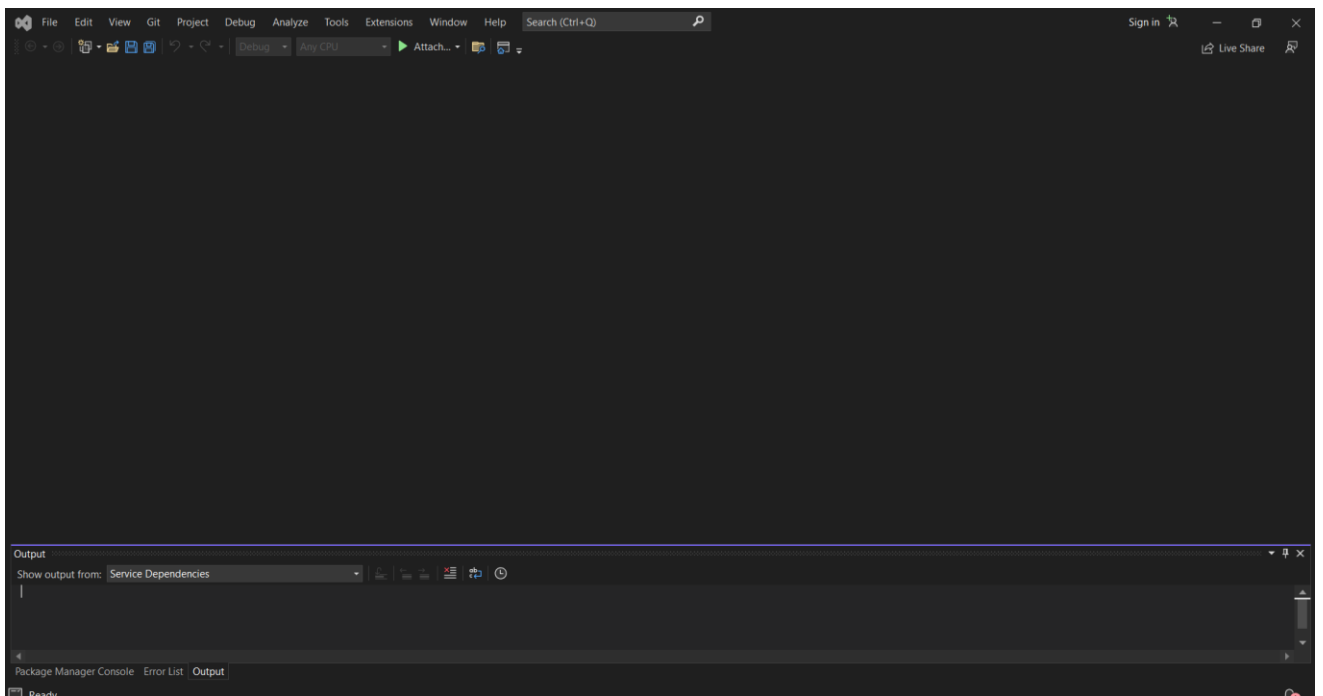


Рисунок 4.1 – відкритий Visual Studio

Створюємо новий проект: У головному меню обираємо "File" (Файл) > "New" (Новий) > "Project" (Проект) як на рисунку 4.2.

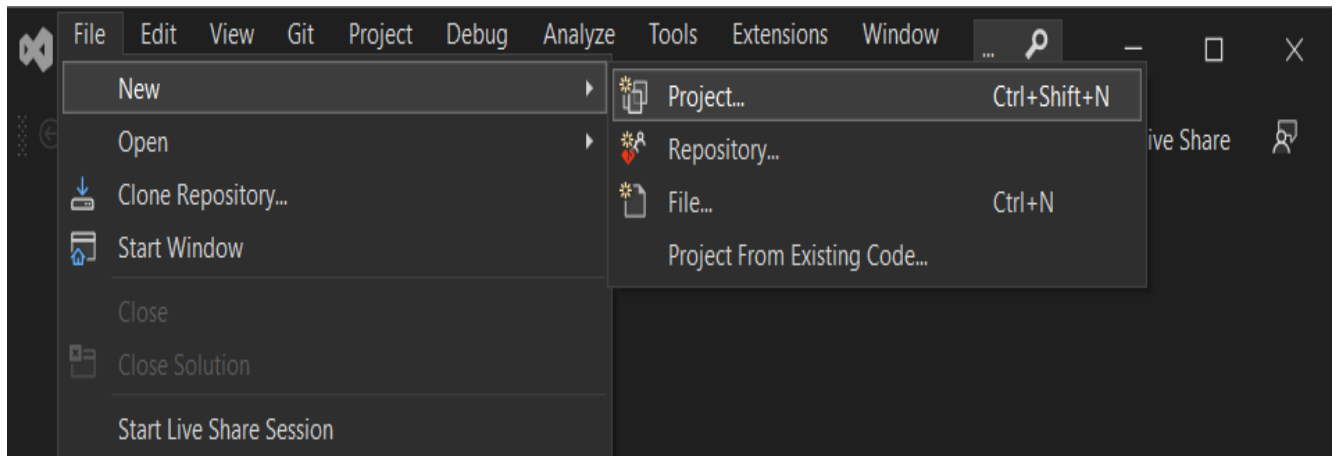


Рисунок 4.2 – Створення нового проекту

Відкриється діалогове вікно "New Project" (Створити проект), де ми обираємо шаблон проекту, який відповідає нашій потребі так як показано на рис.4.3

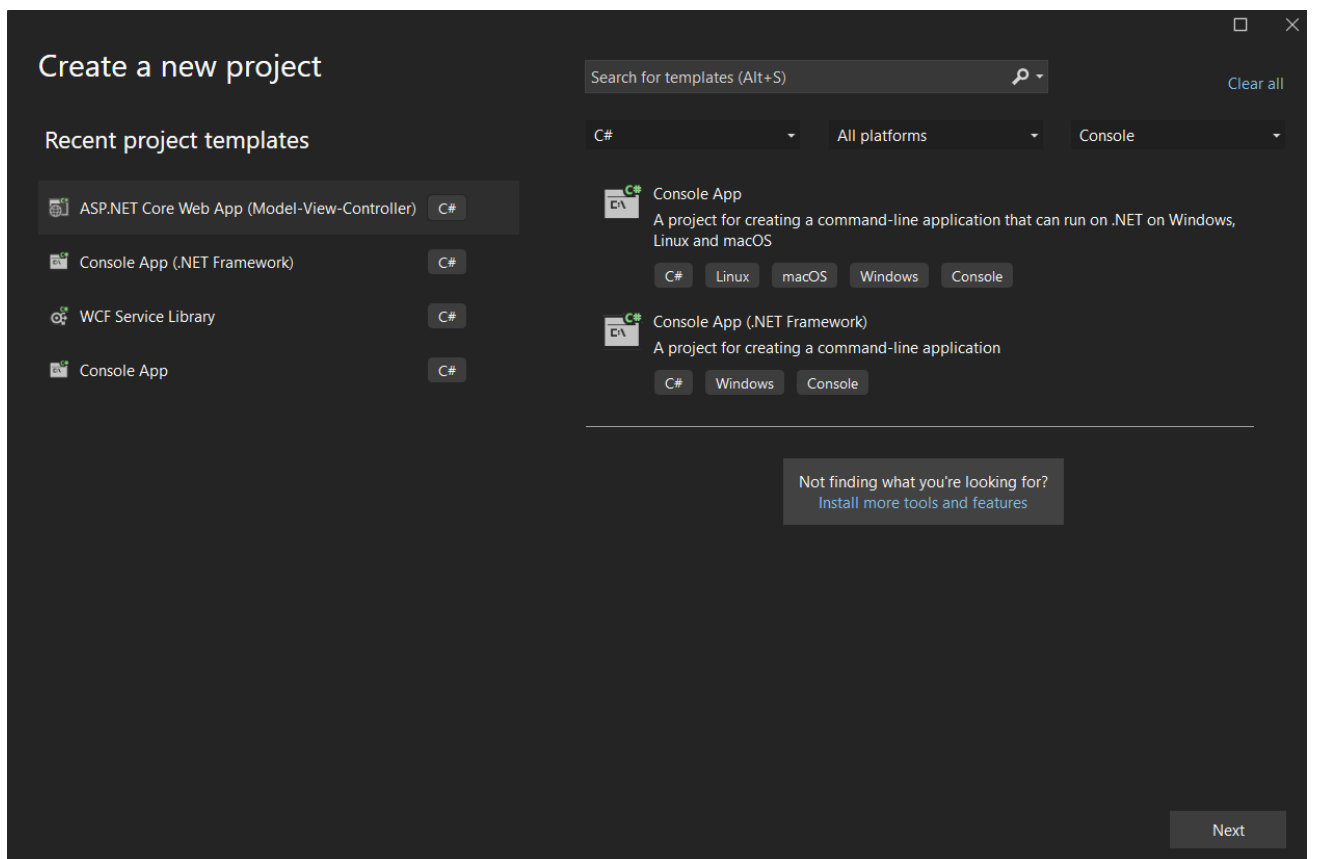


Рисунок 4.3 – вікно вибору шаблону

Для розробки веб-додатку на C# вибираємо "ASP.NET Web Application" (Веб-додаток ASP.NET).

Налаштування проекту: Після вибору шаблону проекту, ми налаштовуємо його параметри, такі як ім'я проекту, розташування, налаштування веб-сервера та інші параметри відповідно до вашої роботи як на рисунку 4.4

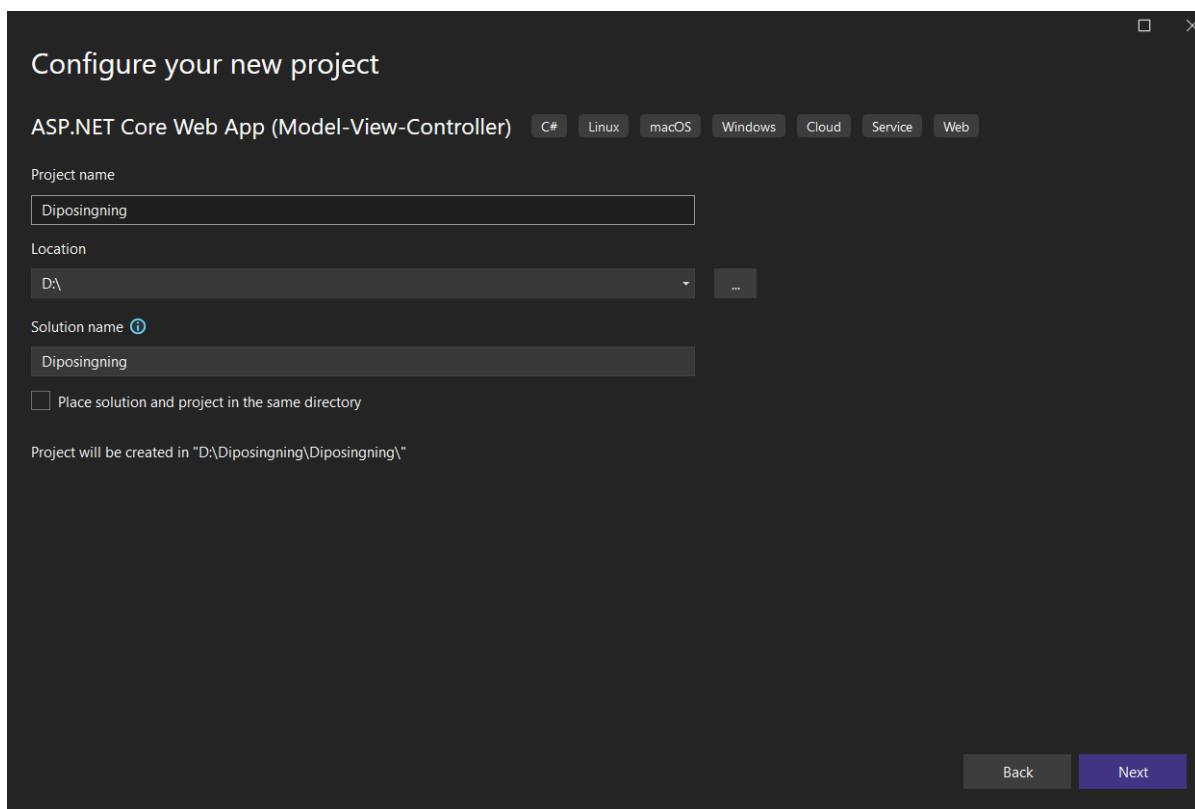


Рисунок 4.4 – головне меню налаштування нашого проекту

Також , після того як ми натиснули “Next”, ми вибираємо версію фреймворку та інші додаткові відомості як на рис. 4.5.

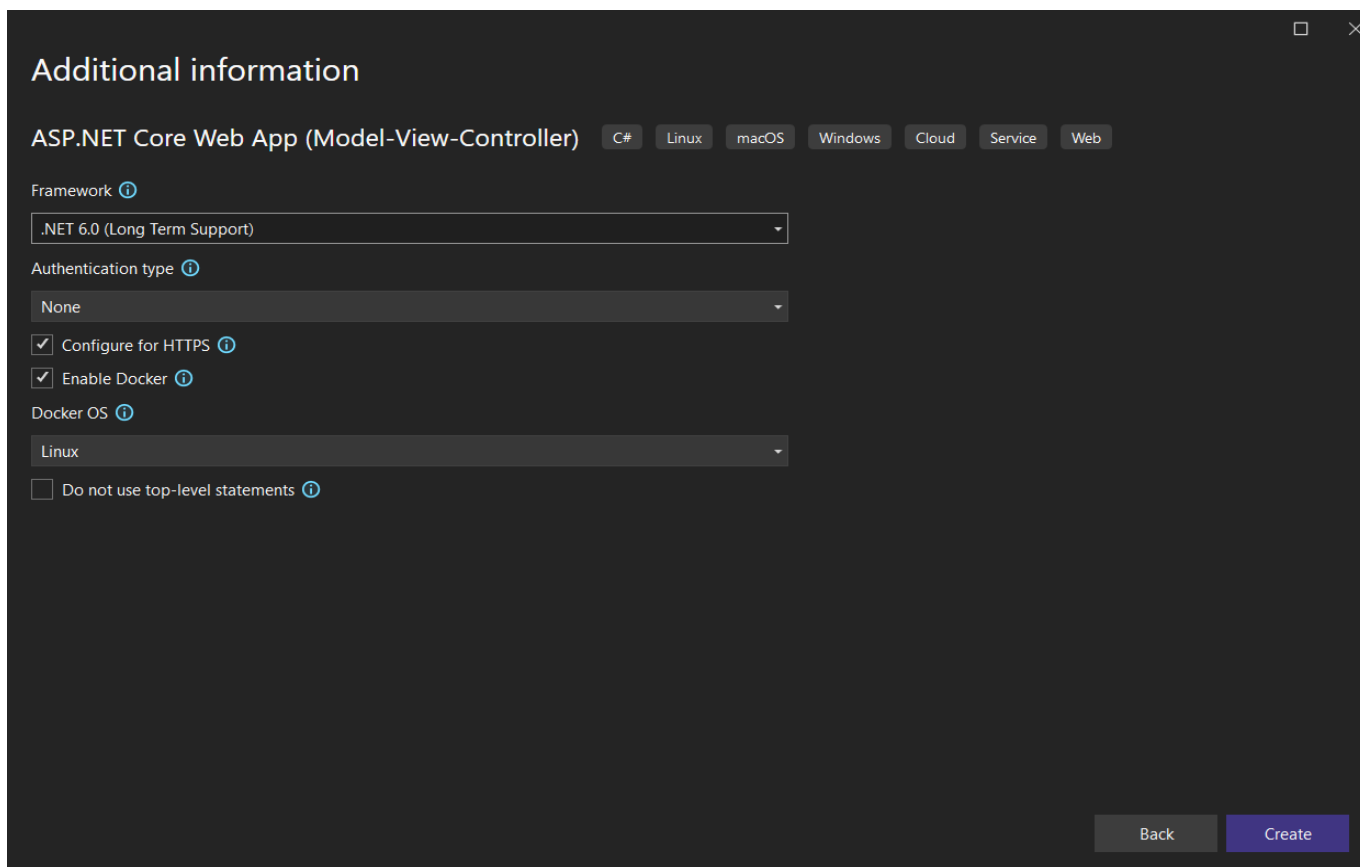


Рисунок 4.5- додаткова інформація

Виберіть тип веб-додатку: У процесі створення проекту, вам буде запропоновано вибрати тип веб-додатку, наприклад, "Empty" (Порожній), "MVC" (Модель-Вид-Контролер), "Web Forms" (Веб-форми) тощо. Вибираємо MVC.

Також нам потрібно встановити додаткові фреймворки для коректної роботи нашої програми тому натискаємо пкм на наш проект та вибираємо Manage NuGet Packages так як показано на рисунку 4.6.

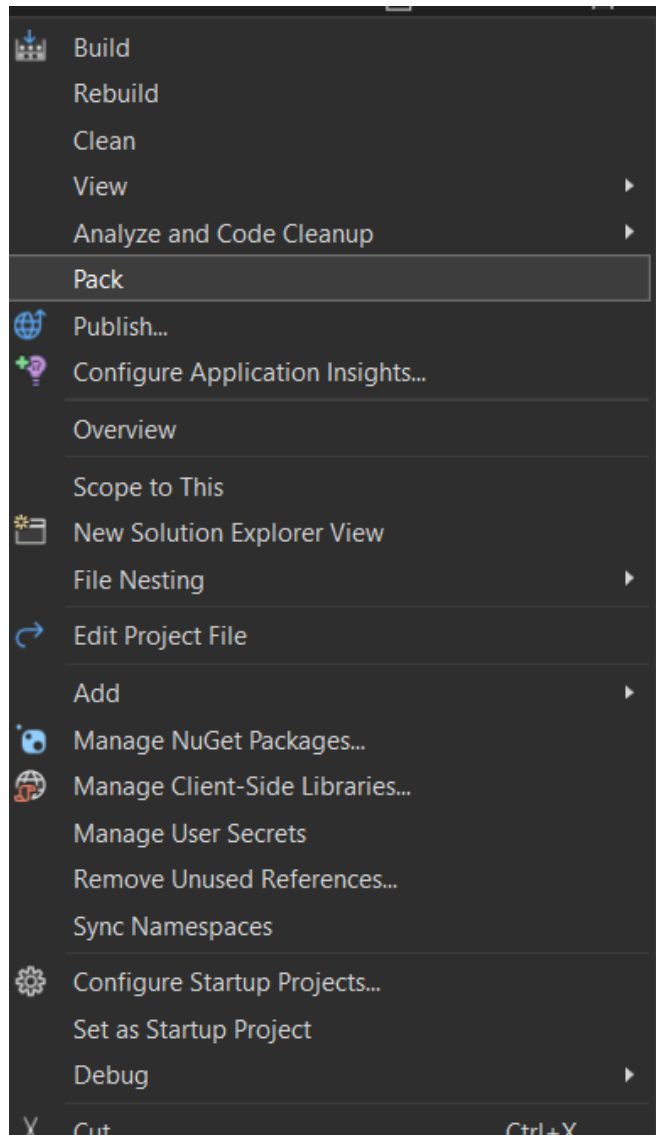


Рисунок 4.6 – додаткові налаштування проекту

Після натискання , в нас висувається вікно в якому ми можемо знайти ті фреймворкі які нам потрібні та завантажити найстабільніші версії, для стабільної роботи.

Нам потрібні три додаткові фреймворкі, Bootstrap, jQuery та Microsoft.EntityFrameworkCore так як зазначено на рисунку 4.7.

Bootstrap - це популярний фреймворк для розробки веб-додатків, який використовується для створення адаптивних і мобільних сайтів. Він надає набір CSS-стилів, компонентів JavaScript та готових шаблонів, які можна легко налаштувати для створення сучасних і привабливих веб-сайтів.

Ось основні функції та переваги використання Bootstrap у веб-розробці:

Адаптивний дизайн: Bootstrap побудований з орієнтацією на мобільні пристрої, що забезпечує зручний вигляд і функціонування веб-сайту на різних пристроях і розмірах екранів. Він надає адаптивну систему сітки та класи CSS, які автоматично адаптують розміщення і вміст до розміру екрану.

Проста система сітки: Система сітки Bootstrap є потужним інструментом, що дозволяє створювати гнучкі і адаптивні макети. Вона використовує 12-колонкову систему сітки, що дозволяє легко розділити веб-сторінку на декілька секцій і розмістити вміст візуально привабливим способом.

Перевикористовувані компоненти: Bootstrap пропонує великий набір перевикористовуваних компонентів інтерфейсу, таких як кнопки, форми, навігаційні панелі, випадаючі списки, сповіщення, модальні вікна та інші. Ці компоненти мають готовий стиль і можуть бути легко використані, що економить час і зусилля при проектуванні та кодуванні з нуля.

Налаштовувані стилі: Використовуючи Bootstrap, ви можете налаштувати зовнішній вигляд свого веб-сайту, замінивши або додавши власні CSS-стилі. Ви можете змінювати кольори, шрифти, розміри та інші стилізації, щоб відповідати вимогам вашого проекту.

Кросбраузерна сумісність: Bootstrap підтримує більшість сучасних браузерів і забезпечує сумісність з різними платформами, що дозволяє вашому веб-сайту працювати на різних браузерах без проблем.

Використання Bootstrap спрощує розробку веб-додатків, прискорює процес розробки та забезпечує стабільну основу для створення сучасних та адаптивних веб-сайтів.

jQuery - це широко використовувана бібліотека JavaScript, яка спрощує взаємодію з HTML-документами, обробку подій, анімацію, виконання AJAX-запитів і багато іншого. Вона забезпечує простий і зручний спосіб роботи з веб-сторінками, розширюючи можливості стандартного JavaScript.

Ось основні характеристики та переваги використання jQuery:

Маніпуляція DOM: jQuery надає потужні засоби для зручної маніпуляції HTML-документами. Ви можете швидко знаходити, змінювати, видаляти та

додавати елементи DOM, змінювати їх атрибути, стилі, класи та багато іншого. Це спрощує взаємодію зі сторінкою і дозволяє швидко змінювати вміст та вигляд веб-сторінки.

Обробка подій: З допомогою jQuery ви можете легко прив'язувати обробники подій до елементів сторінки. Ви можете реагувати на кліки, наведення, введення тексту та багато інших подій, що дозволяє створювати взаємодію з користувачем та реалізовувати різноманітні функціональні можливості.

AJAX-запити: jQuery надає простий спосіб виконання асинхронних запитів до сервера з використанням технології AJAX. Ви можете взаємодіяти з сервером без перезавантаження сторінки, відправляти дані, отримувати відповіді та оновлювати вміст сторінки динамічно. Це дозволяє створювати більшість сучасних веб-додатків, що працюють в реальному часі.

Анімація та ефекти: jQuery має багатий набір функцій для створення анімації та візуальних ефектів на сторінці. Ви можете анімувати рух, зміну розміру, прозорість, кольори та інші властивості елементів DOM. Це дозволяє створювати привабливу та інтерактивну веб-графіку.

Розширення функціональності: Існує велика кількість розширень (плагінів) для jQuery, які додають додаткові функціональні можливості до бібліотеки. Ви можете легко використовувати ці плагіни для реалізації конкретних функцій, таких як каруселі, перетягування елементів, валідація форм та багато іншого.

Microsoft.EntityFrameworkCore - це платформа для розробки програмного забезпечення, яка надає можливості для роботи з базами даних в середовищі .NET. Вона є частиною фреймворку Entity Framework, який забезпечує об'єктно-орієнтований підхід до роботи з даними і забезпечує доступ до реляційних баз даних з використанням об'єктів та LINQ-запитів.

Основні можливості Microsoft.EntityFrameworkCore включають:

Моделювання даних: Entity Framework дозволяє створювати модель даних за допомогою класів .NET, які представляють таблиці бази даних і взаємозв'язки між ними. Ви можете визначити сутності (Entity), відношення між ними і атрибути для збереження та виконання операцій з даними.

Міграції баз даних: Entity Framework надає механізм міграцій, що дозволяє змінювати схему бази даних під час розвитку проекту. Ви можете створювати міграції, які автоматично змінюють структуру бази даних, забезпечуючи відповідність до моделі даних.

Виконання запитів: За допомогою Entity Framework ви можете створювати LINQ-запити для отримання даних з бази даних. LINQ (Language Integrated Query) дозволяє зручно формулювати запити до даних, використовуючи властивості і методи класів .NET.

Контекст бази даних: Ви можете створити контекст бази даних, який представляє з'єднання з базою даних і відповідає за взаємодію з нею. Контекст дозволяє вам виконувати операції з даними, такі як додавання, видалення та оновлення записів.

Підтримка різних провайдерів баз даних: Entity Framework підтримує різні провайдери баз даних, такі як SQL Server, MySQL, PostgreSQL, SQLite та інші. Це дозволяє вам працювати з різними системами управління базами даних, використовуючи єдиний підхід до роботи з даними.

Microsoft.EntityFrameworkCore є потужним інструментом для роботи з базами даних у .NET-проектах. Вона спрощує розробку, забезпечує широкий функціонал для роботи з даними та дозволяє підключатися до різних провайдерів баз даних.

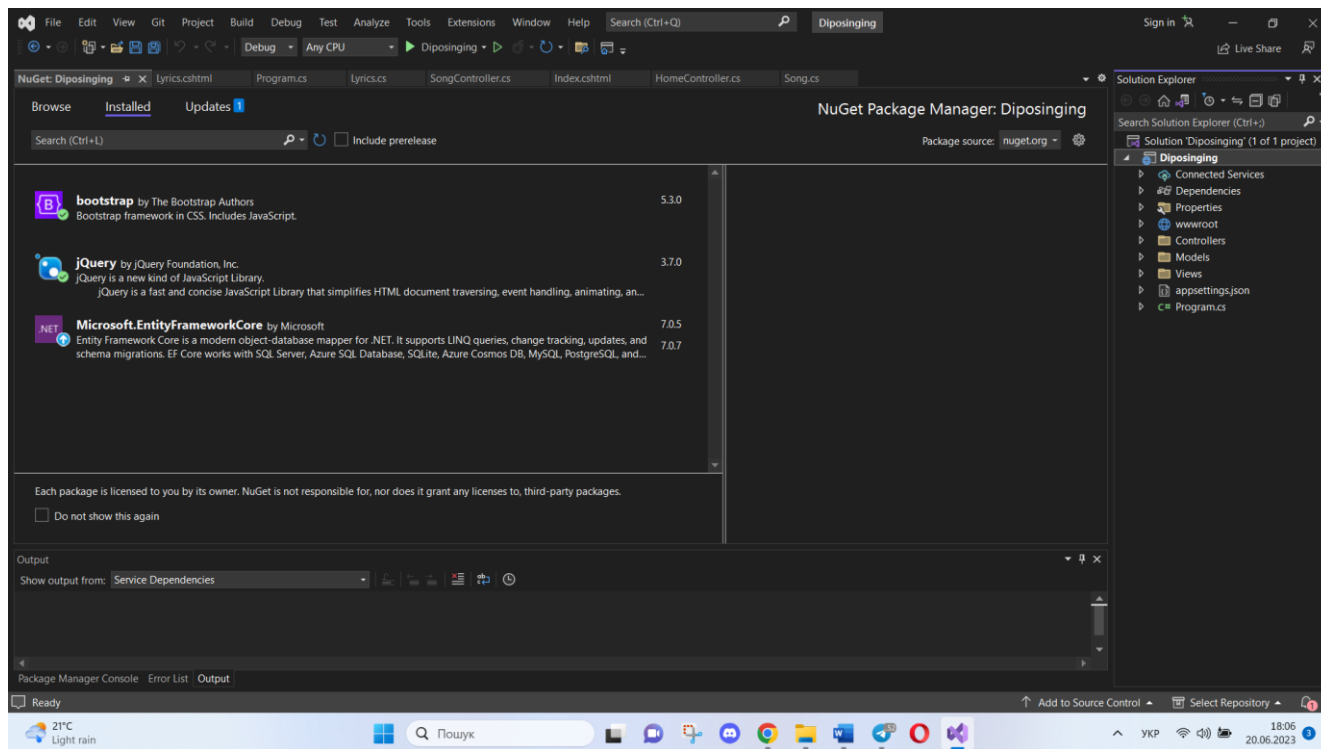


Рисунок 4.7 – завантаження додаткових фреймворків

4.2 Створення об'єктів (MVC)

Тепер ми можемо розпочати програмувати наш веб-додаток.

Так як ми використовуємо MVC(model, view, controller), MVC (Model-View-Controller) є підходом до проектування програмного забезпечення, який дозволяє розділити додаток на три основні компоненти: модель, представлення та контролер. Цей підхід покращує організацію коду, забезпечує його повторне використання та спрощує розвиток і супровід додатків.

Основні концепції MVC:

Модель (Model): Модель відповідає за представлення даних та бізнес-логіку додатку. Вона зберігає дані та визначає способи їх обробки. Модель може включати класи, структури, методи доступу до даних, правила валідації і багато іншого.

Представлення (View): Представлення відповідає за візуальне відображення даних користувачу. Воно відповідає за генерацію HTML-сторінок, відображення даних, форм та інших елементів інтерфейсу. Представлення отримує дані від моделі та відображає їх користувачу.

Контролер (Controller): Контролер відповідає за обробку запитів користувача та керування потоком додатку. Він приймає запити від користувача, взаємодіє з моделлю для отримання потрібних даних та вибирає відповідне представлення для їх відображення. Контролер також відповідає за взаємодію з іншими компонентами, такими як сервіси, бази даних тощо.

Нам потрібно додати нові моделі з якими ми будемо взаємодіяти, я додав 2 моделі це пісні та тексти пісень, а також в моделях знаходяться дві дуже важливі моделі які вже були згенеровані і вони допомагають в деяких складних ситуаціях. Всі моделі позначено на рис. 4.8.

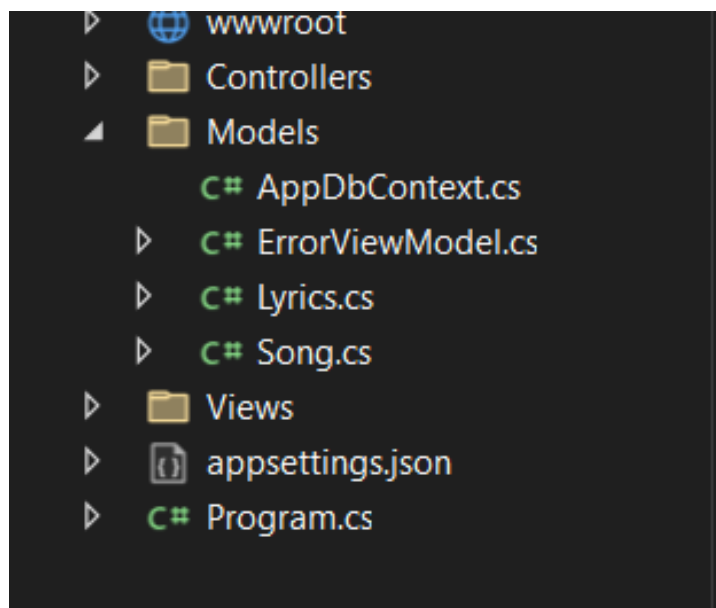


Рисунок 4.8 – моделі

Ось як ми задаємо модель пісень, ми можемо це побачити на рисунку 4.9

```
1 using Microsoft.AspNetCore.Mvc;
2
3 39 references
4 public class Song
5 {
6     31 references
7     public int Id { get; set; }
8     29 references
9     public string Title { get; set; }
10    28 references
11    public string Artist { get; set; }
12
13    28 references
14    public string FilePath { get; set; }
15
16    29 references
17    public string Lyrics { get; set; }
18 }
```

Рисунок 4.9 - клас пісень

Ось як ми задаємо наші тексти пісень на рисунку 4.10

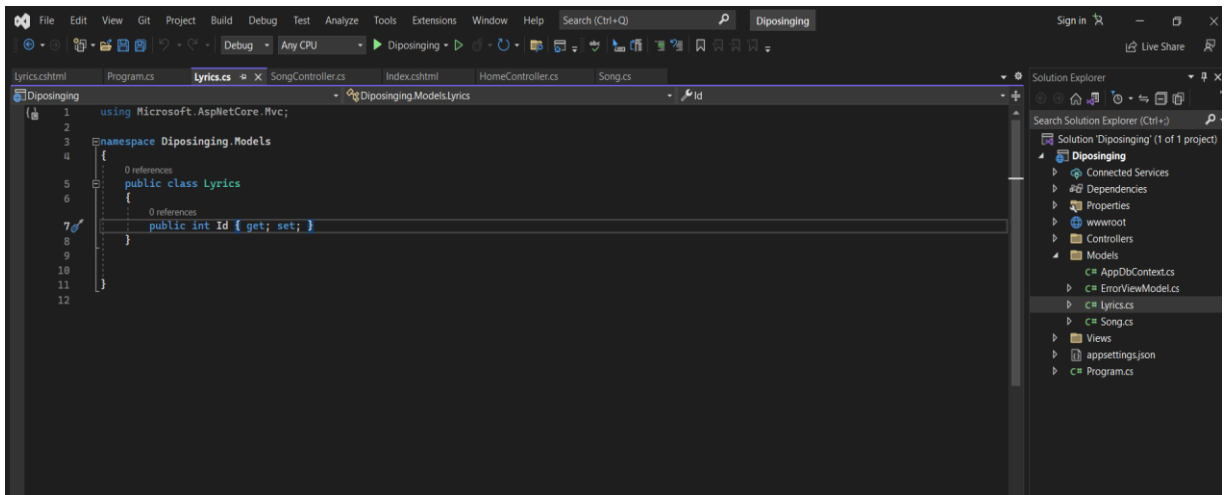


Рисунок 4.10 - Клас тексту до пісень

Тепер давайте перейдемо до view , всі наші views ми можемо переглянути на рисунку 4.11.

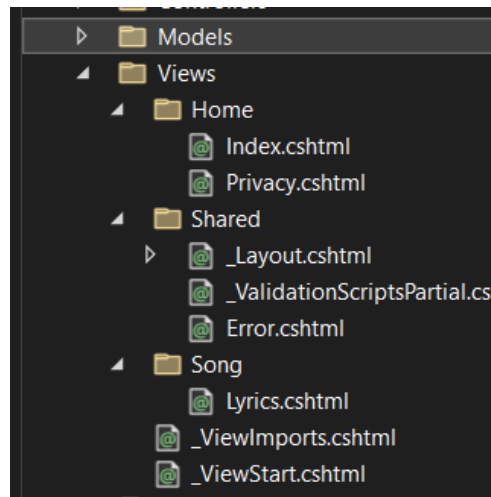


Рисунок 4.11- views

Тепер переглянемо наші контролери на рисунку 4.12

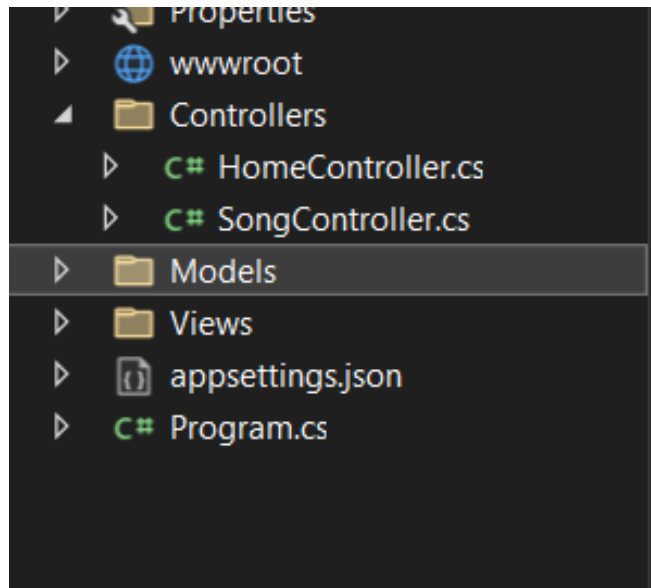


Рисунок 4.12 контролери

Тепер ми можемо зрозуміти , що в нас є контролери які взаємодіють з моделями та views і оформлюють нам гарну складність програми.

4.3 Огляд фінальної версії додатку.

Основний код програми написаний в файлі Home(index.cshtml) , там прописано всі кнопки які є на сторінці, як буде виглядати сторінка, її функціонал і так далі.

Тому перейдемо до функціоналу самого веб-додатку .На рисунку 4.13 ви можете побачити як виглядає сторінка нашого веб-додатку.

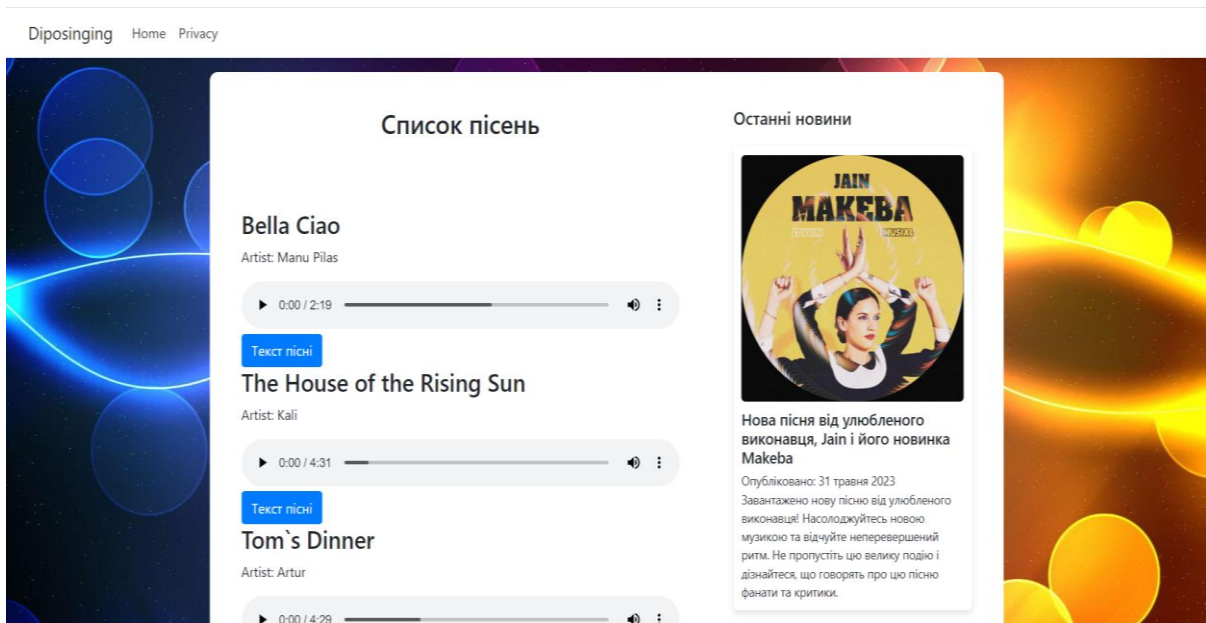


Рисунок 4.13 веб-додаток

Як ви бачите сторінка складається з двох головних блоків , як ‘Список пісень’ та ‘Останні новини’. Це виконано для того , щоб , коли слухаєш музику ти міг опускатися по сторінці і паралельно шукаючи якійсь цікавий трек, або ж прослуховування ти міг читати найсвіжіші новини які сталися в музичному світі за останній час.

Можна прослуховувати пісні , після закінчення пісні автоматично буде вмикатися наступна, також пісню можна завантажити прямо під час того як її прослуховуєш, є можливість швидкого прослуховування, зміни гучність пісні та перемотки треку на певний момент пісні. Це все показано на рисунку 4.14 та 4.15 та 4.16.

Bella Ciao

Artist: Manu Pilas

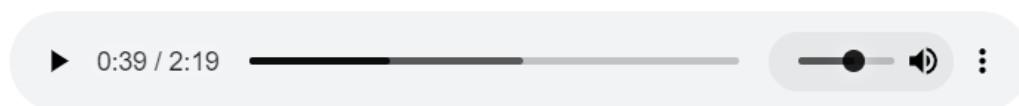


Рисунок 4.14 – можливість відображення прогресу пісні

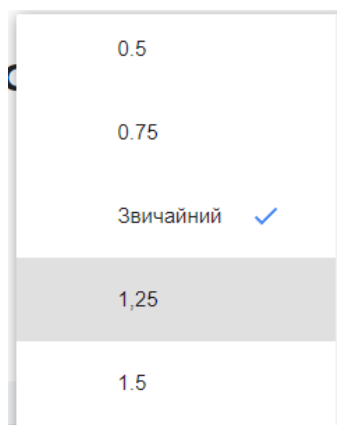


Рисунок 4.15 – швидкість програвання пісні

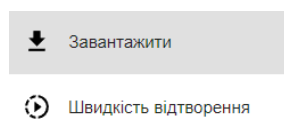


Рисунок 4.16 – можливість завантаження пісні

Також є можливість переглядання тексту пісні, після натискання синьої кнопки ‘Текст пісні’ на рисунку 4.17,



Рисунок 4.17 – можливість перевірки тексту

Після натискання в вас з’являється вікно в якому і знаходиться тексти всіх пісень, в залежності під яким треком ви натиснете ‘текст пісні’, вам видасть належний кожній пісні її ж текст. Можемо це побачити на рисунку 4.18

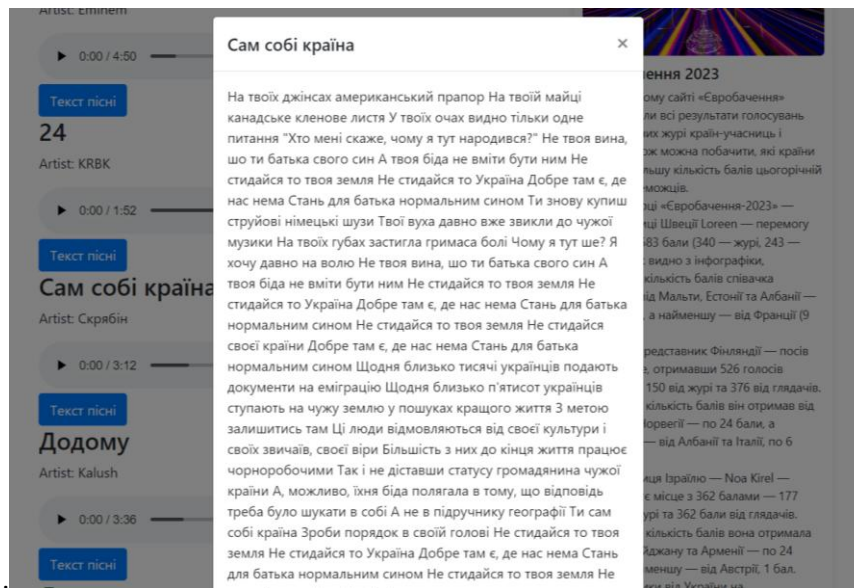


Рисунок 4.18 – текст пісні

У цьому розділі було проведено розробку додатку, що базується на мові програмування C# та використовує технологію ASP.NET. Розпочавши зі створення проекту в Visual Studio та його налаштування, були встановлені необхідні залежності та конфігурації для подальшої розробки.

Далі, було проведено створення об'єктів в рамках архітектури MVC (Model-View-Controller). Були визначені моделі даних, контролери та представлення, що дозволяють реалізувати логіку додатку, його взаємодію з користувачем та збереження даних.

Останнім етапом було створення веб-додатку в Visual Studio. За допомогою інструментів та функціоналу Visual Studio, було розроблено інтерфейс користувача, додано стилізацію за допомогою CSS, підключено скрипти JavaScript для додаткової функціональності та взаємодії з елементами сторінки.

Результатом розділу є створення прототипу веб-додатку, який вже має базовий функціонал та взаємодію з користувачем.

ВИСНОВКИ

У даній дипломній роботі була проведена розробка універсального музичного веб-додатку з використанням технологій C#, ASP.NET, HTML, CSS та JavaScript. Метою дослідження було створення функціонального та естетично привабливого додатку, який забезпечує користувачам можливість пошуку та програвання музики, перегляду новин та відображення текстів пісень.

У розділі "Об'єкт та методи дослідження" було визначено об'єкт дослідження - музичний веб-додаток, та визначено мету, завдання та методи дослідження, які використовувалися для реалізації додатку. Розділ "Огляд технологій C# для розробки веб-додатків" надав загальний огляд використаних технологій, їх особливостей та можливостей.

Аналіз функціональних та нефункціональних вимог до додатку в розділі "Аналіз вимог до додатку" дозволив визначити основні функціональні можливості додатку, такі як реєстрація та авторизація користувачів, пошук та програвання музики, відображення текстів пісень, а також нефункціональні вимоги, зокрема щодо швидкодії, інтерфейсу та безпеки даних.

Дослідження цільової аудиторії дозволило визначити, що додаток розрахований на широкий спектр користувачів різних вікових груп, із різними музичними уподобаннями, без географічних обмежень. Це дозволило побудувати додаток, який надає доступ до музики для різних смаків та вподобань.

У розділі "Розробка додатку" було проведено створення проекту в Visual Studio, налаштування середовища розробки та створення об'єктів за архітектурою MVC. Крім того, було реалізовано веб-додаток з використанням необхідних технологій, таких як HTML, CSS та JavaScript, що забезпечують ефективну взаємодію з користувачем та забезпечують зручний інтерфейс.

В результаті розробки додатку було створено прототип, який вже має базовий функціонал та забезпечує основні можливості пошуку та

програвання музики. Продовження роботи над додатком передбачає розширення його функціональності, оптимізацію та вдосконалення інтерфейсу, а також реалізацію додаткових функцій.

Загалом, розробка даного музичного веб-додатку виявилася цікавим та важливим завданням, що сприяє розвитку навичок у веб-розробці та програмуванні. Результати роботи можуть бути використані для подальшого розвитку та удосконалення додатку, а також як основа для подальших наукових досліджень у галузі веб-розробки та музичних технологій.

ПЕРЕЛІК ПОСИЛАНЬ

Документація C# URL: <https://learn.microsoft.com/en-us/dotnet/csharp/>
(електронний ресурс)

Документація MVC URL: <https://learn.microsoft.com/ru-ru/aspnet/mvc/>
(електронний ресурс)