

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ПУСТОВОЙТЕНКО ВЛАДИСЛАВ ВІКТОРОВИЧ

Допускається до захисту:

завідувач кафедри

інформаційних технологій,

доктор технічних наук, доцент

_____ Т. В. Нескородева

« ____ » _____ 20__ р.

**РОЗРОБКА ІНФОРМАЦІЙНО-ПОШУКОВОГО САЙТУ З ФУНКЦІЄЮ
ОПЛАТИ МІСЬКИХ СЕРВІСІВ**

Спеціальність 122 «Комп'ютерні науки»

Кваліфікаційна (бакалаврська) робота

Керівник:

Нескородева Т.В., завідувач кафедри

інформаційних технологій,

д.т.н., доцент

Оцінка: ____ / ____ / ____

(бали за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

(підпис)

Анотація

Пустовойтенко В.В. Розробка веб-сайту для замовлення та оплати послуг в місті з використанням фреймворку ASP .NET MVC. Спеціальність 122 «Комп'ютерні науки», ОП «Сучасні інформаційні технології та програмування». Донецький національний університет імені Василя Стуса, Вінниця, 2022.

Ціллю виконання бакалаврської роботи є розробка веб-сайту для замовлення та оплати послуг в місті онлайн.

У вступі показано актуальність і переваги розробки веб-сайту саме за допомогою платформи .NET Core, а також наведені аналоги даного проекту або його частин.

У другому та третьому розділі розглянуто технології та інструменти, за допомогою яких реалізований мій веб-сайт. Розписані кроки налаштування та запуску сайту за допомогою ASP .NET Core та Angular.

Третій розділ посвячено опису користувацького інтерфейсу, розписана детальна архітектура сайту та продемонстровано його функціонал , а саме: оплату послуг, використання Google карт, реєстрація та вхід до аканту , CRUD та інші функції.

Ключові слова: веб-сайт, надання послуг онлайн, NET Core, C#, ASP .NET Core, Microsoft SQL, Entity Framework.

55 с., 44 рис., 43 джерел.

ABSTRACT

Pustovoitenko VV Development of a website for ordering and paying for services in the city using the ASP .NET MVC framework. Specialty 122 "Computer Science". OP "Modern Information Technologies and Programming". Vasyl Stus Donetsk National University, Vinnytsia, 2022.

The purpose of the bachelor's thesis is to develop a website for ordering and paying for services in the city online.

The introduction shows the relevance and benefits of website development using the .NET Core platform, as well as analogues of this project or its parts.

У другому та третьому розділі розглянуто технології та інструменти, за допомогою яких реалізований мій веб-сайт. Розписані кроки налаштування та запуску сайту за допомогою ASP .NET Core та Angular.

The third section is devoted to the description of the user interface, describes the detailed architecture of the site and demonstrates its functionality, namely: payment for services, use of Google Maps, registration and login, CRUD and other features.

Keywords: website, online services, .NET Core, C #, ASP .NET Core, Microsoft SQL, Entity Framework.

55 p., 44 pict., 43 sources.

ЗМІСТ

ВСТУП	
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	
1.1 Аналіз предметної області	
1.2 Постановка задачі	
1.3 Огляд існуючих аналогів.....	
2 ОПИС ВИКОРИСТАНИХ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ ДЛЯ НАПИСАННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ	
2.1 Загальні відомості про проект і використані технології.....	
2.2 Вибір середовища розробки	
2.3 Створення та перший запуск проекту	
2.4 Розробка клієнтської частини.....	
3 ОПИС ВИКОРИСТАНИХ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ ДЛЯ НАПИСАННЯ СЕРВЕРНОЇ ЧАСТИНИ.....	
3.1 Інструменти	
3.2 Вибір стеку технологій.....	
3.3 Платформа .NET	
3.4 Технологія ASP.NET MVC Core	
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКА	
4.1 Опис призначеного для користувача інтерфейсу	
4.2 Автентифікація та особистий кабінет користувача.....	
4.3 Автентифікація та особистий кабінет користувача.....	
4.4 Опис спеціальних функцій сайту	
ВИСНОВКИ	
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	
ДЕКЛАРАЦІЯ	

ВСТУП

Ще двадцять років тому більша частина людства навіть не знала про існування інтернету, а в наш час сучасне суспільство не уявляє свого життя без нього. Сьогодні все більше сфер життєдіяльності переходить в онлайн формат. Це пов'язано не тільки з розвитком цифрових технологій, а й з всеохоплюючою пандемією коронавірусу, яка торкнулася мабуть що кожного і змінила сучасний світ і спосіб ведення бізнесу на до і після.

Сьогодні більшість компаній, у сферах в яких це можливо, стараються перемістити свій бізнес в онлайн формат для подальшого його функціонування і зменшення втрати доходів. Тому зараз буде як ніколи доцільним створення платформи для надання та отримання звичних для нас послуг та сервісів онлайн, що планується реалізувати в результаті цієї кваліфікаційної роботи.

У рамках бакалаврської роботи було досягнуто таких цілей:

1. Поглиблення своїх знань у веб-розробці, а саме вивчення таких технологій як ASP.NET Core, Angular, Entity Framework, мов програмування C# та TypeScript.
2. Освоєння використання npm пакетів.
3. Огляді існуючих веб-сайтів, які мають схоже призначення.
4. Написання власного веб-сайту з серверною та клієнтською частиною.
5. Можливість реєстрації, входу до персонального кабінету, кастомізації та зміни персональних даних користувачем.
6. Розділ користувачів на ролі: адміна, користувача або менеджера.
7. Можливість переглядати здійснені користувачем замовлення за певний проміжок часу.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

Веб-сайти або веб-додатки є найбільш поширеним і найбільш популярним продуктом в сфері розробки програмного забезпечення. В основному це зумовлено тим, що веб-сайт є найдоступнішим видом ПЗ для будь-якого користувача, так як його можна відвідати з будь-якого девайсу, починаючи з смартфона і планшета та закінчуючи персональним комп'ютером і ноутбуком. На відміну від різноманітних десктопних або мобільних додатків чи ігор, сайт не потребує затрат часу для його завантаження, місця на жорсткому диску чи сумісність операційної системи і відповідних системних вимог. Для доступу і взаємодії з веб-ресурсом потрібно мати лише стабільне інтернет з'єднання, який завгодно браузер – програма для пошуку інформації в інтернеті та гіперпосилання на потрібний нам веб-сайт. З кожним днем кількість веб-сайтів стає все більшою і більшою, так само як і росте кількість розробників і замовників цього продукту.

З кінця 2019 року епідемія COVID-19 поширилась на весь світ, на жаль ця страшна хвороба не оминула і Україну. Пандемія нанесла сильного удару маже по всім сферам життя. Світова економіка зазнала великої шкоди, деякі види бізнесу були на грані вимирання, особливо це торкнулося індустрії з великим потоком людей, такі як кінотеатри, ресторани, спортзали тощо. Деякі компанії змогли адаптуватись до карантинних умов і перешли в онлайн режим. В основному це стосувалося офісних працівників, учнів, студентів які були вимушені перейти на дистанційну форму роботи та навчання відповідно. Також збільшився попит на замовлення послуг через інтернет – оплата рахунків, покупка і доставка продуктів і готових страв, на даний момент це не тільки заощаджує купу часу, а ще й рятує життя, так як не потрібно відвідувати місця з великим скупченням людей, стояти в чергах тощо. Так як не кожен магазин, ресторан, ЖКХ мав свій веб-сайт спричинило небувалий ріст замовлень сайтів

даного типу. Тому розробка таких сайтів є дуже актуальною і доволі прибутковою справою, як і сьогодні, в період пандемії, так і після її завершення, так як люди швидко звикнуть до такого способу отримання послуг і з великою ймовірністю не заохочують повертатися до колишнього.

1.2 Постановка задачі

Необхідно реалізувати веб-сайт для надання різних послуг в певному місті, де серверна частина буде написана мовою C# з використанням середовища розробки ASP .NET, а клієнтська частина за допомогою фреймворка Angular. Додаток повинен виконувати наступні функції:

1. Реєстрація вхід на сайт під особистим email.
2. Персональний кабінет користувача.
3. Реалізація наступних сервісів:
 - a. Доставки.
 - b. Оплати комунальних послуг.
 - c. Замовлення таксі.
 - d. Бронювання готелю.
 - e. Купівлі квитків на різні події (кіно, фестивалі тощо).
4. Додавання, редагування, видалення нових компаній, які будуть розміщені на сайті та їхніх товарів/послуг.
5. Можливість робити покупку/замовлення товару/послуги.
6. Перегляд раніше зроблених замовлень в особистому кабінеті.
7. Можливість оцінити будь-яку компанію/сервіс.
8. Пошук компанії/продукту/послуги за повною або частковою назвою.
9. Сортування компаній/продуктів/послуг за рейтингом.
10. Надання прав адміністратора користувачу.
11. Створення інформаційної панелі новин.

1.4 Огляд існуючих аналогів

Сервіс надання послуг, резервування білетів, замовлення їжі чи оплати рахунків онлайн не є чимось новим чи революційним. В часи до пандемії багато людей вже користувались чимось подібним, що неабияк економило їм час, а зараз в умовах карантину попит на дані послуги значно зростає, так як крім часу, в цей не легкий період, може й вберегти ваше здоров'я.

Дивлячись на поставлені цілі, проект повної схожості у функціоналі з існуючими аналогами знайти важко, так як основною ціллю є створення гнучкої платформи, де будуть представлені різні компанії, які в свою чергу зможуть надавати свої послуги, аналогій до кожного наявного сервісу, буде представлено далі. Дана платформа була б дуже корисною для тих корпорацій, що поки не мають власного сайту через брак коштів або часу, так як для розробки нового сайту з нуля потрібен тривалий період, а в сучасному світі час – це гроші. Завдяки гнучкості платформи, можна швидко і без жодних зусиль додати компанію з її сервісами.

Найближчим аналогом є інтернет-магазин Rozetka:

1. Rozetka



Рисунок 1.1 – Логотип інтернет-магазину Rozetka

Rozetka — один з найпопулярніших українських інтернет-магазинів та маркетплейсів, що з'явився 2005 року. Станом на серпень 2020 року сайт посідає 7 місце серед найвідвідуваніших в Україні. Сайт дозволяє іншим компаніям виставляти, крім низку товарів власного виробництва. Окрім інтернет-магазину також має фізичні магазини, точки видачі і склади. На відміну від нашої платформи працює на всій території України, а не лише в одному місті. Але в майбутньому також можна буде розширити свій ринок.

Як було зазначено раніше, платформа складається з різних сервісів, а саме:

- ❖ Сервіс продажу і доставки певних товарів
- ❖ Сервіс резервування готельних номерів
- ❖ Сервіс замовлення таксі або каршерінгу
- ❖ Сервіс оплати комунальних платежів

Розглянемо до кожного з них схожі за призначенням веб-сайти або додатки.

Starwood



Рисунок 1.2 – Логотип інтернет-магазину Starwood

Starwood - це локальна Вінницька компанія, що спеціалізується на випічці та доставці піци та суші у місті Вінниця. Особливість цього закладу полягає у

швидкій доставці протягом 33 хвилин, якщо дана умова не буде виконана клієнт отримує купон на безкоштовну піцу при наступному замовленні, а також те що дана компанія не має фізичного місця, де можна було б спробувати їх продукцію і працює лише онлайн і доставляє їжу клієнтові. Якщо поглянути на технічну складову сайти мають схожу класифікацію продуктів і можливість додавати їх до корзини.

Uber



Рисунок 1.3 – Логотип сервісу таксі Uber

Uber - американська компанія, що створила однойменний мобільний застосунок для пошуку, виклику та оплати таксі або приватних водіїв. Застосунок доступний у більш ніж 200 великих містах в 67 країнах світу.

Першою і основною відмінністю є те що Uber – це додаток на смартфон, а наш проект є веб-сайтом, що полегшує співпрацю з ним, так як не потрібно

додатково завантажувати програму на смартфон. Другою відмінністю є те що наш веб-сайт призначений для функціонування лише в одному місті, а Uber розповсюджений по всьому світу, що є його великою перевагою. Що ж до спільного в обох продуктах є можливість співпраці з картою, вибору маршруту, автомобіля тощо.

Висновок до розділу 1

У даному розділі розглянуто та проаналізовано предметну область в якій застосування нашого сайту була б найкращою. Також була чітко визначена постановка задачі та технології, які будуть застосовуватись для розробки сайту. Також було досліджено схожі за функціоналом та призначенням існуючі аналоги.

РОЗДІЛ 2

ОПИС ВИКОРИСТАНИХ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ ДЛЯ НАПИСАННЯ КЛІЄНТСЬКОЇ ЧАСТИНИ

2.1 Загальні відомості про проект і використані технології

Перш за все метою дипломної роботи є створення веб-сайту для надання послуг з підключеною базою даних для зберігання інформації про користувачів, компаній їх операцій між собою тощо.

Для цієї цілі добре підходить технологія корпорації Microsoft, а саме – ASP .NET Core. Ця технологія не є мовою програмування, а являє собою каркас для комфортної взаємодії між собою серверної та клієнтської частини. Для реалізації бек-енду була обрана мова програмування C#, що також є розробкою Майкрософту. Що ж до фронт-ендної частини, була надана перевага фреймворку Angular і мові TypeScript для роботи з ним.

Даний набір інструментів зумовлений не лише їхньою популярністю і продуктивністю роботи, а ще й тому що я маю певний досвід використання саме цих технологій, так як я завершив курс з C# і ASP.NET Core в університеті на дисципліні «Розробка клієнт-серверних систем».

Далі перейдемо до розбору процесу розробки клієнтської частини і всіх використаних технологій окремо і більш детально.

2.2 Вибір середовища розробки

Для створення і запуску проекту була обрана IDE «Microsoft Visual Studio» 2022-го року (остання версія).



Рисунок 2.1 – Логотип програми Visual Studio 2022

IDE(Integrated development environment або ж українською Інтегроване середовище розробки) – це комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Інтегровані середовища розробки створені для того, щоб максимізувати продуктивність програміста, надавши йому пов'язані інструменти розробки зі схожими інтерфейсами як одну програму, в якій відбуватиметься весь процес розробки й яка надає необхідні функції для модифікації, компілювання, розгортання та налагодження програмного забезпечення.

Microsoft Visual Studio

Microsoft Visual Studio — серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також вебсайти, вебзастосунки, вебслужби як у рідному, так і в керованому кодах для всіх платформ.

Microsoft Visual Studio була обрана не спроста. По-перше, як можна зрозуміти з назви і як було сказано раніше, вона є розробкою і власністю корпорації Microsoft, а це значить, що вона має найкращу сумісність з мовою C# і платформою ASP .NET Core. По-друге, ця IDE являється найпопулярнішою і найчастіше використовуваною для розробки на мові C# :

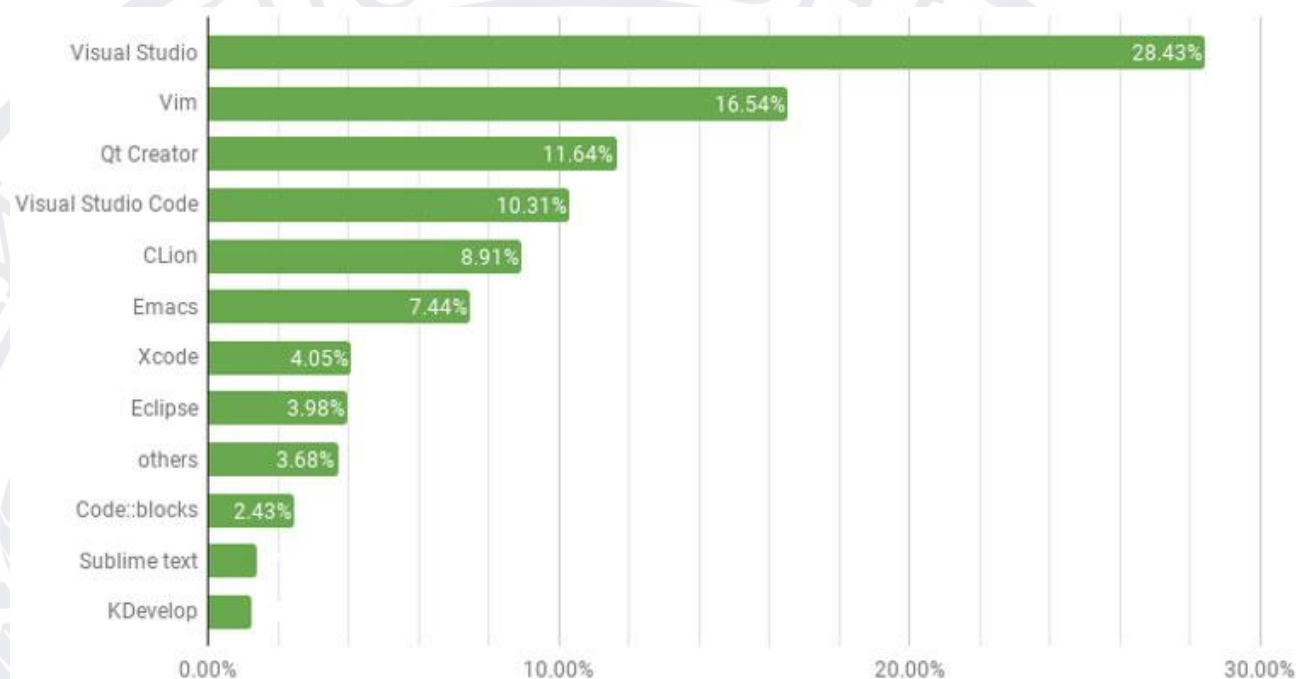


Рисунок 2.2 – Найпопулярніші IDE для мови C#

Можливості Visual Studio 2022:

- ❖ Швидка навігація, написання та виправлення коду;
- ❖ Проста налагодження, профілювання і діагностика коду;
- ❖ Комплексні інструменти тестування допомагають писати якісний код;
- ❖ Використовуйте систему управління версіями Git для гнучкої та ефективної спільної роботи;

- ❖ Безліч розширень, які дозволять вам налаштувати середовище розробки під себе.

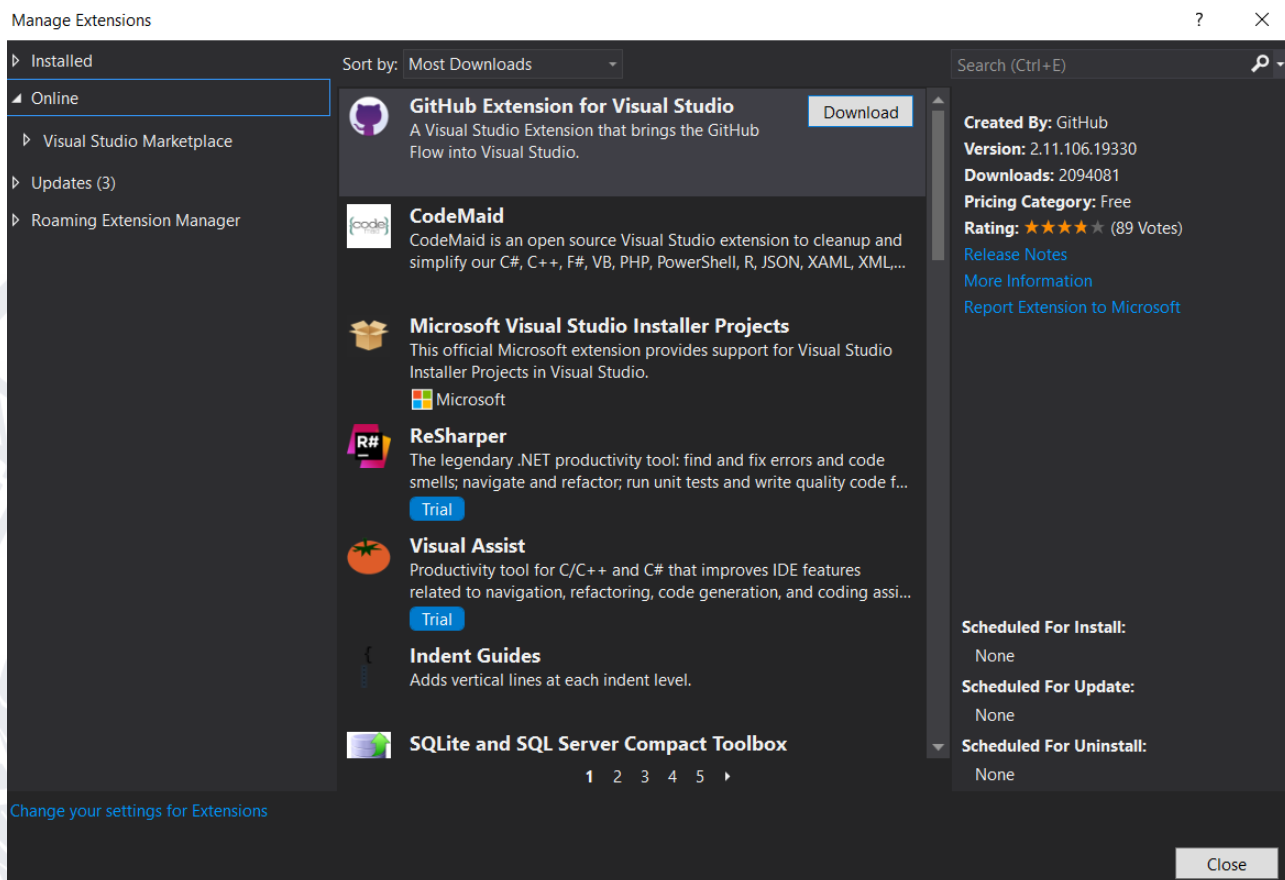


Рисунок 2.3 – Менеджер розширень Visual Studio 2022

Visual Studio 2022 найкраще підходить для:

- ❖ Розробки додатків та ігор для всіх пристроїв під управлінням Windows;
- ❖ Створення власних або гібридних додатків для Android, iOS і Windows;
- ❖ Проста збірка, розгортання масштабованих хмарних додатків і управління ними в Azure;
- ❖ Розробка сучасних веб-додатків з гнучкими і багатифункціональними засобами роботи з відкритим вихідним кодом;
- ❖ Багатифункціональні засоби для всіх типів розробки в Microsoft Office;

- ❖ Розробка, написання коду та налагодження ігор з використанням сучасних інструментів графіки і скриптів;

- ❖ Написання власних розширень для Visual Studio;

Проста розробка та розгортання баз даних SQL Server і SQL Azure.

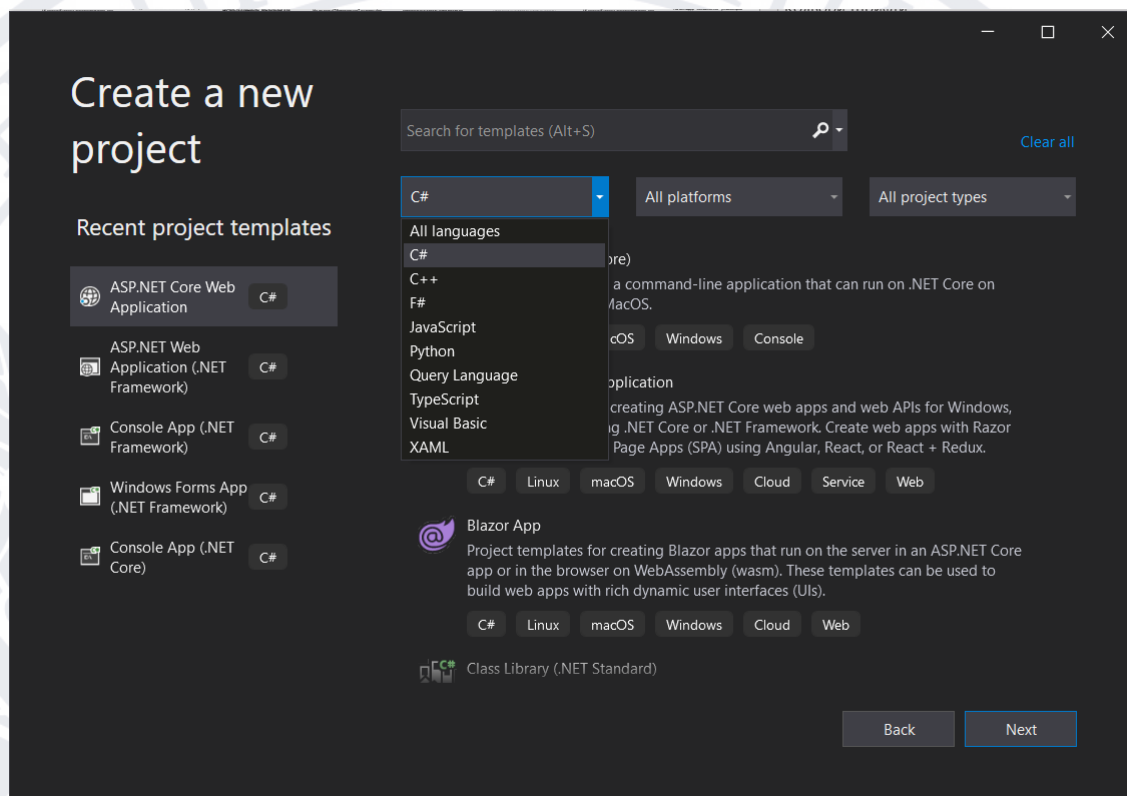


Рисунок 2.4 –Можливість вибору мови Visual Studio 2022 перед створенням проекту

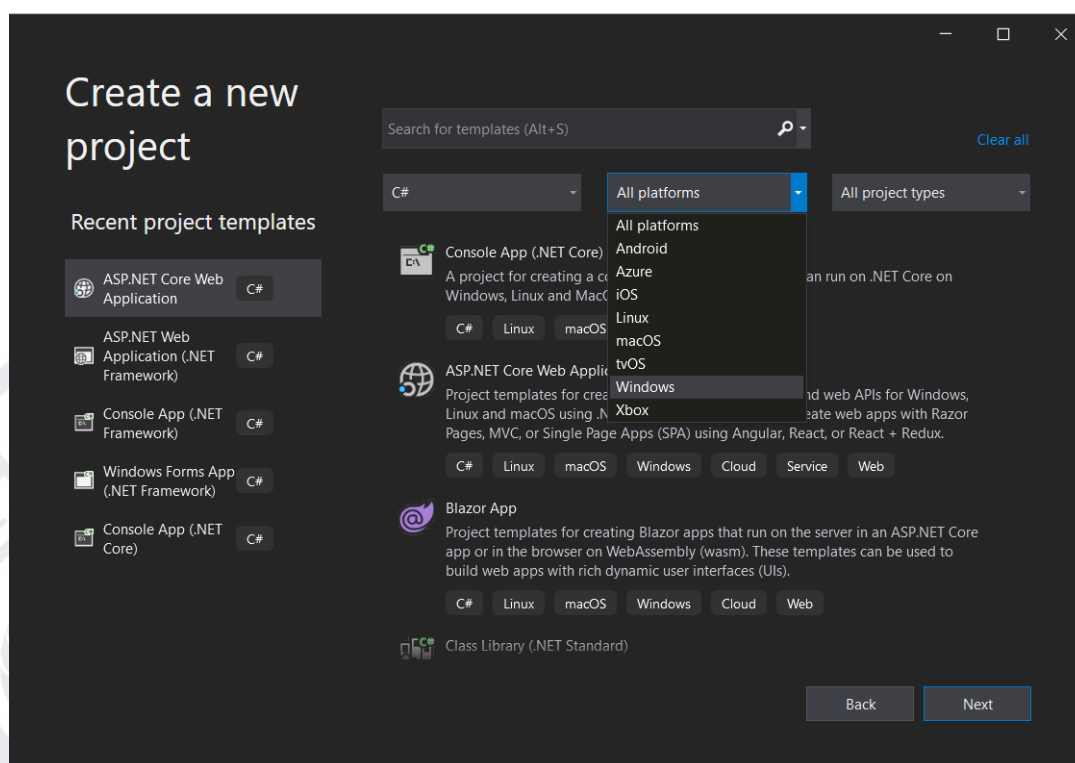


Рисунок 2.5 – Можливість вибору платформи в Visual Studio 2022

Різноманітність вибору технологій у Visual Studio 2022:

- ❖ Використовуйте C++, щоб домогтися швидкості, високої продуктивності і сумісності на широкому спектрі пристроїв;
- ❖ Збірка швидких масштабованих кроссплатформених додатків на JavaScript за допомогою Node.js;
- ❖ Вирішуйте завдання для створення кроссплатформених скриптів, веб-служб, Інтернету речей та вивчення даних за допомогою Python;
- ❖ Створення додатків і служб за допомогою .NET для будь-якого пристрою і ОС;
- ❖ Підвищення продуктивності, якості і гнучкості в додатках JavaScript.

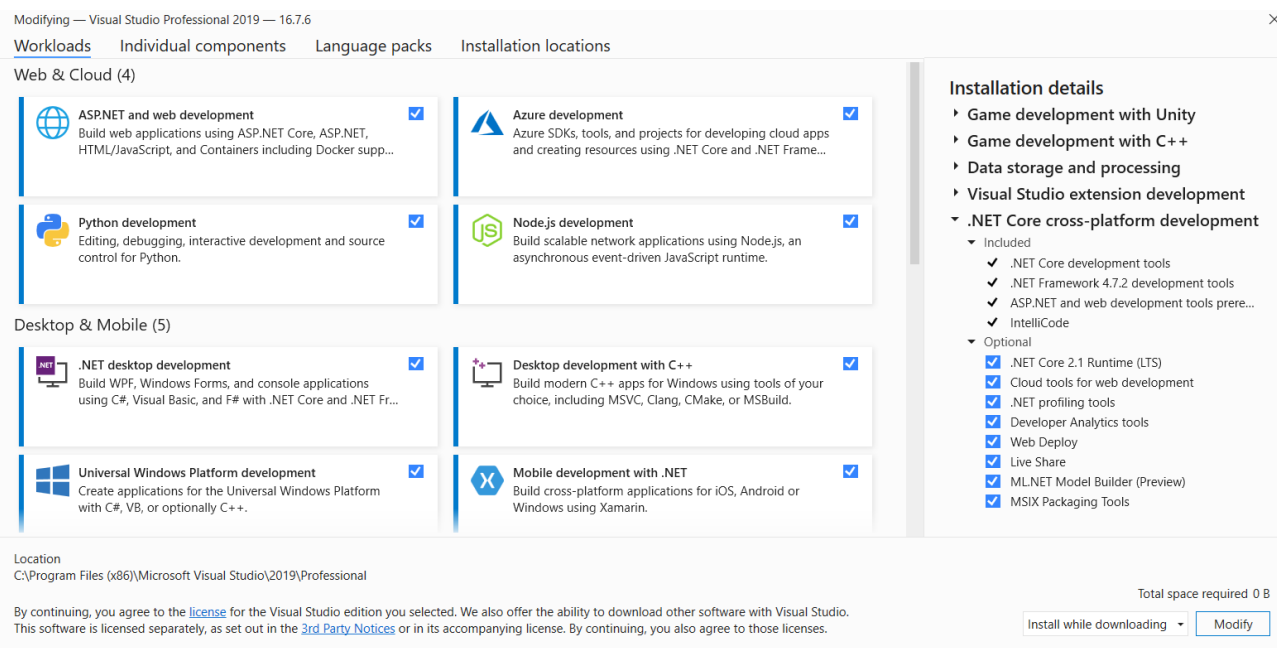


Рисунок 2.6 – Інструменти та функції Visual Studio (Частина 1)

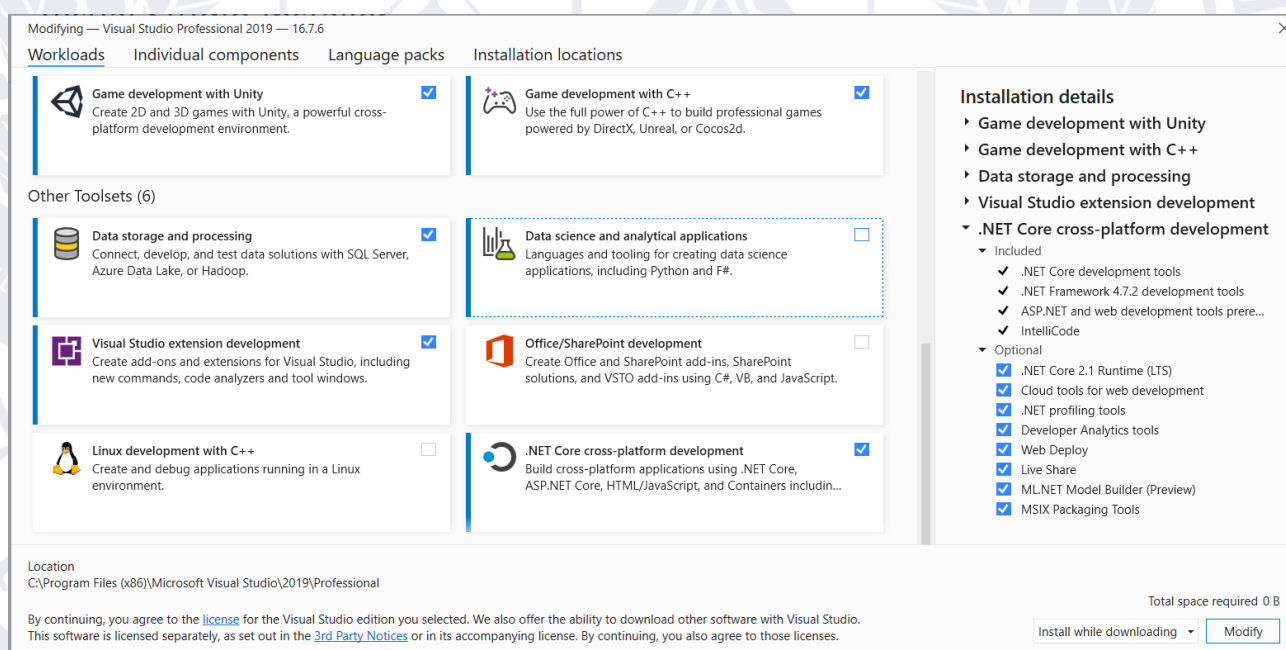


Рисунок 2.7 – Інструменти та функції Visual Studio (Частина 2)

2.3 Створення та перший запуск проекту

Перш за все потрібно завантажити середовище розробки MS Visual Studio 2022. Завдяки тому, що я студент Донецького національного університету мені доступна Professional версія даного IDE, що значно розширить функціонал

порівняно з загальнодоступною Community версією. Після чого потрібно інсталиювати інструменти зображені на рисунку 2.8.

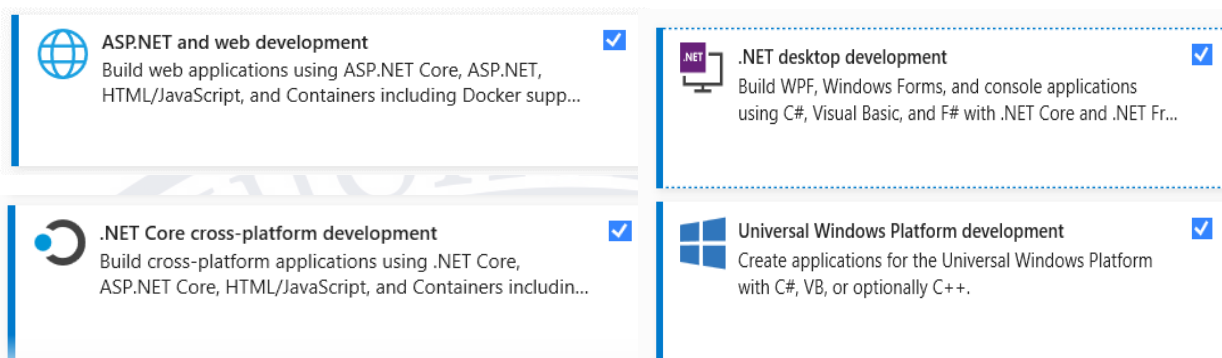


Рисунок 2.8 – Необхідні інструменти для створення веб-сайту за допомогою ASP .NET Core

Після успішного встановлення вище перелічених інструментів, в меню створення нового проекту потрібно пункт ASP .NET Core Web Application. Щоб полегшити пошук в пошуковий рядок можна ввести запит: «ASP .NET»

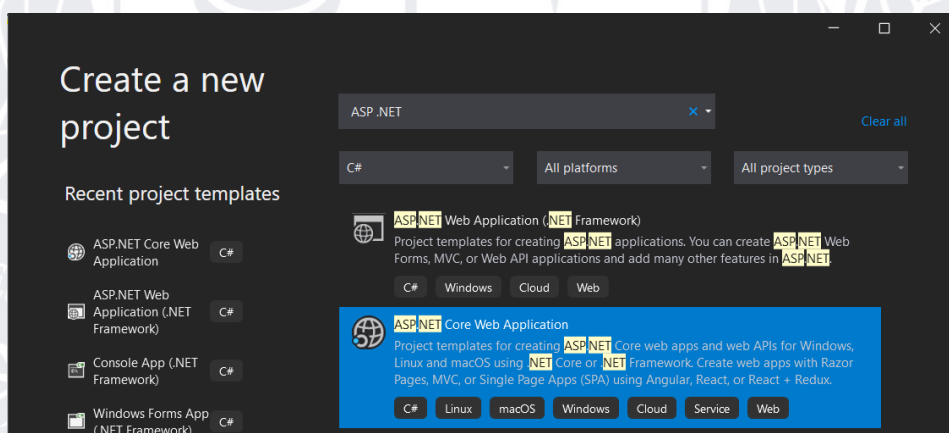


Рисунок 2.9 – Вибір платформи для майбутнього проекту

Створивши проект він вже має основні файли для успішного запуску його на локальному сервері. До числа цих файлів входять:

- ❖ Бібліотеку Bootstrap, яка полегшує створення адаптивного веб-інтерфейсу CSS та JavaScript шаблонів;

- ❖ HomeController.cs який відповідає за backend двох Представлень , які являють собою HTML-сторінки з розширенням .cshtml, що в свою чергу дає змогу інтегрувати C# код напряму в HTML шаблон.
- ❖ Program.cs та Startup.cs що відповідають за запуск всього проекту , підключення сервісів та інших налаштувань;
- ❖ _Layout.cshtml - даний файл стандартним шаблоном всього сайту в якому знаходяться статичні компоненти дизайну такі як шапка та футер сайту, який буде незмінний та присутній на кожній сторінці сайту, що дає змогу значно зменшити об'єм повторюваного коду на Представленнях.

Запуск додатку виконується за допомогою інструменту IIS Express. IIS - веб-сервер, який дозволяє розміщувати в Інтернеті сайти. IIS підтримує протоколи HTTP, HTTPS, FTP, POP3, SMTP, NNTP.

IIS Express — це легка автономна версія IIS, оптимізована для розробників. IIS Express дозволяє легко використовувати найновішу версію IIS для розробки та тестування веб-сайтів. Він має всі основні можливості IIS 7 і вище, а також додаткові функції, призначені для полегшення розробки веб-сайтів, зокрема:

- ❖ IIS Express добре працює з додатками ASP.NET і PHP.
- ❖ Кілька користувачів IIS Express можуть працювати незалежно на одному комп'ютері.
- ❖ Він не працює як служба і не вимагає прав адміністратора для виконання більшості завдань.
- ❖ Не потрібно додаткове встановлення, так як даний інструмент міститься в Visual Studio, починаючи з версії 2010 року.

Після першого запуску маємо такий початковий вигляд нашого веб-сайту:

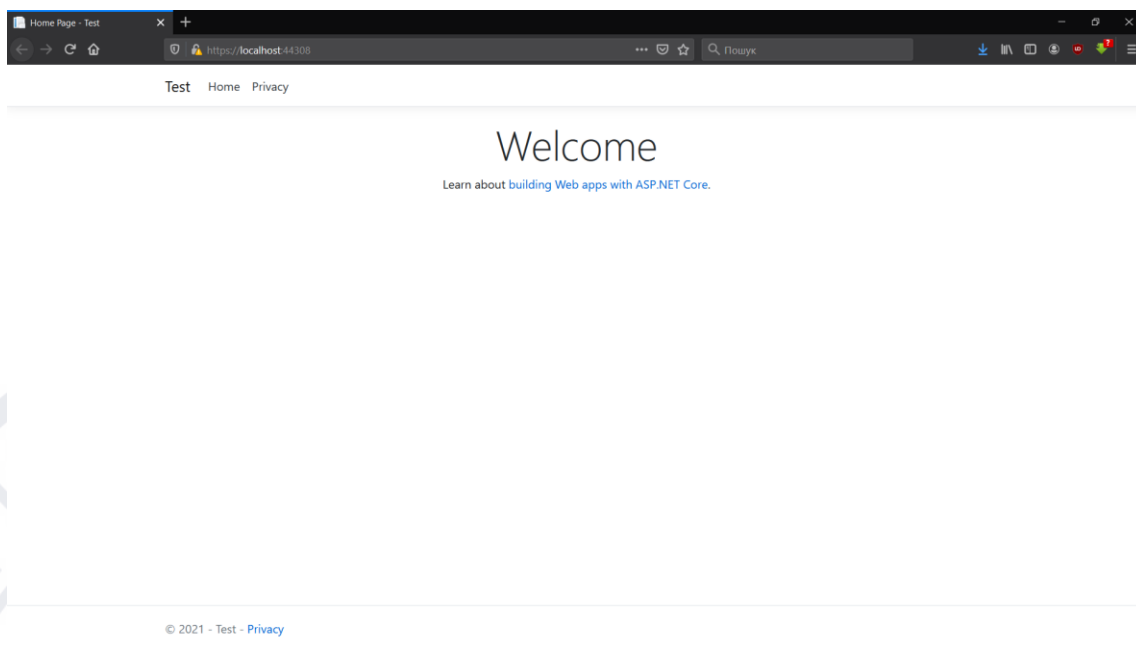


Рисунок 2.10 – Вигляд сайту після першого запуску

2.4 Розробка клієнтської частини

Хоч Visual Studio являється самодостатньою програмою для розробки веб-додатків і не тільки, але зазвичай веб-розробники не використовують IDE для роботи над клієнтською частиною, так як це не є доцільно занадто ресурсозатратно, тому що зміни на клієнтській частині робляться доволі часто, а у IDE потрібно кожен раз перезавантажувати проект, що займає не малий проміжок часу. Тому для цієї справи текстовий редактор коду. Таких редакторів існує безліч, наприклад: Sublime Text, WebStorm або PhpStorm. Але я надаю перевагу використанню Visual Studio Code.



Рисунок 2.11 – Логотип Visual Studio Code

Visual Studio Code - засіб для створення, редагування та зневадження сучасних веб-застосунків і додатків. Позиціонується як «легкий» редактор коду для кросплатформної розробки веб- та хмарних програм. Включає в себе відладчик, інструменти для роботи з Git, підсвічування синтаксису, IntelliSense та засоби для рефакторингу.

Visual Studio Code має такий суттєвий ряд переваг:

- ❖ На відміну від WebStorm і PhpStorm, а також Visual Studio є безкоштовним для використання.
- ❖ Є кросплатформним редактором, тобто використовується на таких операційних системах як Windows, Linux, macOS.
- ❖ Має широкі можливості для кастомізації: теми користувача, поєднання клавіш і файли конфігурації.
- ❖ Має підсвічування синтаксису та автозаповнення за допомогою встроєного розширення IntelliSense, який надає розумні завершення на основі типів змінних, визначень функцій та імпортованих модулів.
- ❖ Надає можливість налагоджувати код прямо з редактора. Запускати або приєднувати до запущених програм і налагоджувати їх за допомогою точок зупинки, стеків викликів та інтерактивної консолі.
- ❖ Має велетенську бібліотеку розширень, що в свою чергу полегшують процес написання коду. Також можна створювати і використовувати свої розширення.
- ❖ Робота з хмарними репозиторіями такими як Git або Microsoft Azure, що полегшує роботу над проектами в команді. Також має особливість при зміні гілки репозиторію в Visual Studio, гілка автоматично змінюється і в текстовому редакторі, що дуже сильно пришвидшує процес розробки.
- ❖ Вбудована консоль Windows PowerShell, яка дає змогу встановлювати так експлуатувати npm, yarn та інші види пакетів.

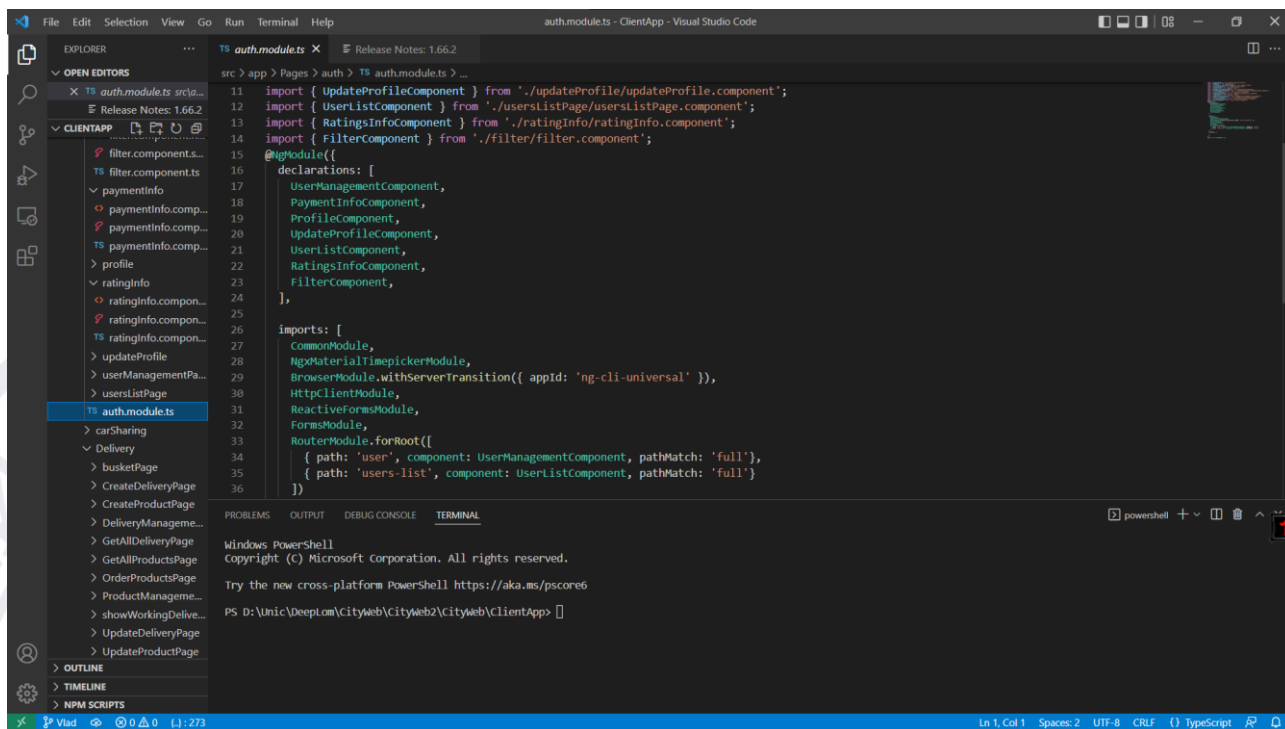


Рисунок 2.12 – Зовнішній вигляд Visual Studio Code

Для розробки клієнтської частини було використано фреймворк Angular та мову TypeScript.

TypeScript

TypeScript — це строго типізована мова програмування, яка базується на JavaScript, що дає розробникам набагато кращі інструменти в будь-якому масштабі та більш комфортну роботу з даними ніж в JavaScript через строгу типізацію. Однак код TypeScript перетворюється на JavaScript, який запускається скрізь, де виконується JavaScript: у браузері, на Node.js або Deno і у ваших програмах.

Angular

Angular — front-end фреймворк написаний на TypeScript з відкритим кодом. Цей фреймворк розроблений компанією Google і Призначений для розробки односторінкових або багатосторінкових додатків, що складаються з одної HTML сторінки, файлу стилів CSS і TypeScript файлу де міститься логічна частина додатку. Його основна ціль— розширення браузерних застосунків на основі

шаблону MVC(Модель-вид-контролер), а також спрощення їх тестування та розробки.



Рисунок 2.13 – Логотип фреймворку Angular

Для установки і коректної роботи Angular на вашому комп'ютері потрібно мати установлену програму Node.js , вона нам потрібна для запуску проекту на локальному сервері та для керування та встановлення модулів, потрібних для функціонування Angular. Для цього використовується пакетний менеджер npm (node package manager). Це можна зробити з офіційного сайту Node.js - <https://nodejs.org/uk/> .



Рисунок 2.14 – Логотип Node.js

Node або Node.js - програмна платформа, з відкритим кодом для виконання високопродуктивних мережевих застосунків. Node.js додає можливість

взаємодіяти з пристроями вводу-виводу через свій API, підключати інші зовнішні бібліотеки, написані різними мовами, забезпечуючи виклики до них із JavaScript-коду. Node.js застосовується переважно на сервері, виконуючи роль веб-сервера, крім цього є можливість розробляти на Node.js також десктопні віконні програми і навіть програмувати мікроконтролери.

Node.js має наступні властивості:

- ❖ неблокуючий ввід/вивід;
- ❖ система модулів CommonJS;
- ❖ рушій JavaScript Google V8;
- ❖ асинхронна одно-нитева модель виконання запитів;

Для компіляції програми ми будемо використовувати інфраструктуру Angular CLI. Angular CLI спрощує створення програми, її компіляцію. Angular CLI поширюється як пакет npm, тому для його використання необхідно спочатку встановити. Для встановлення відкриємо консоль/командний рядок, також для цього можна використати термінал в Visual Studio Code. Виконаємо в ньому наступну команду, після чого можна перейти безпосередньо до розробки:

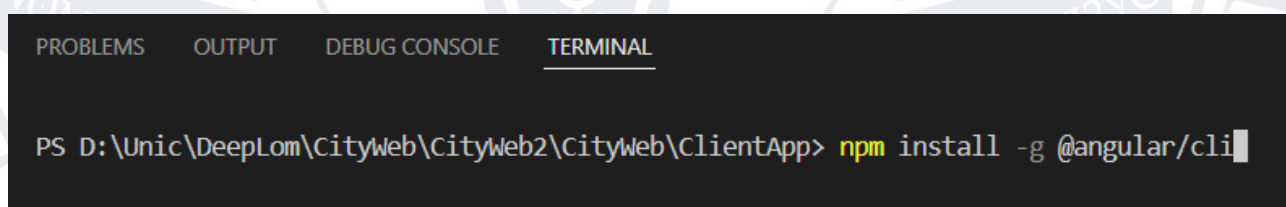


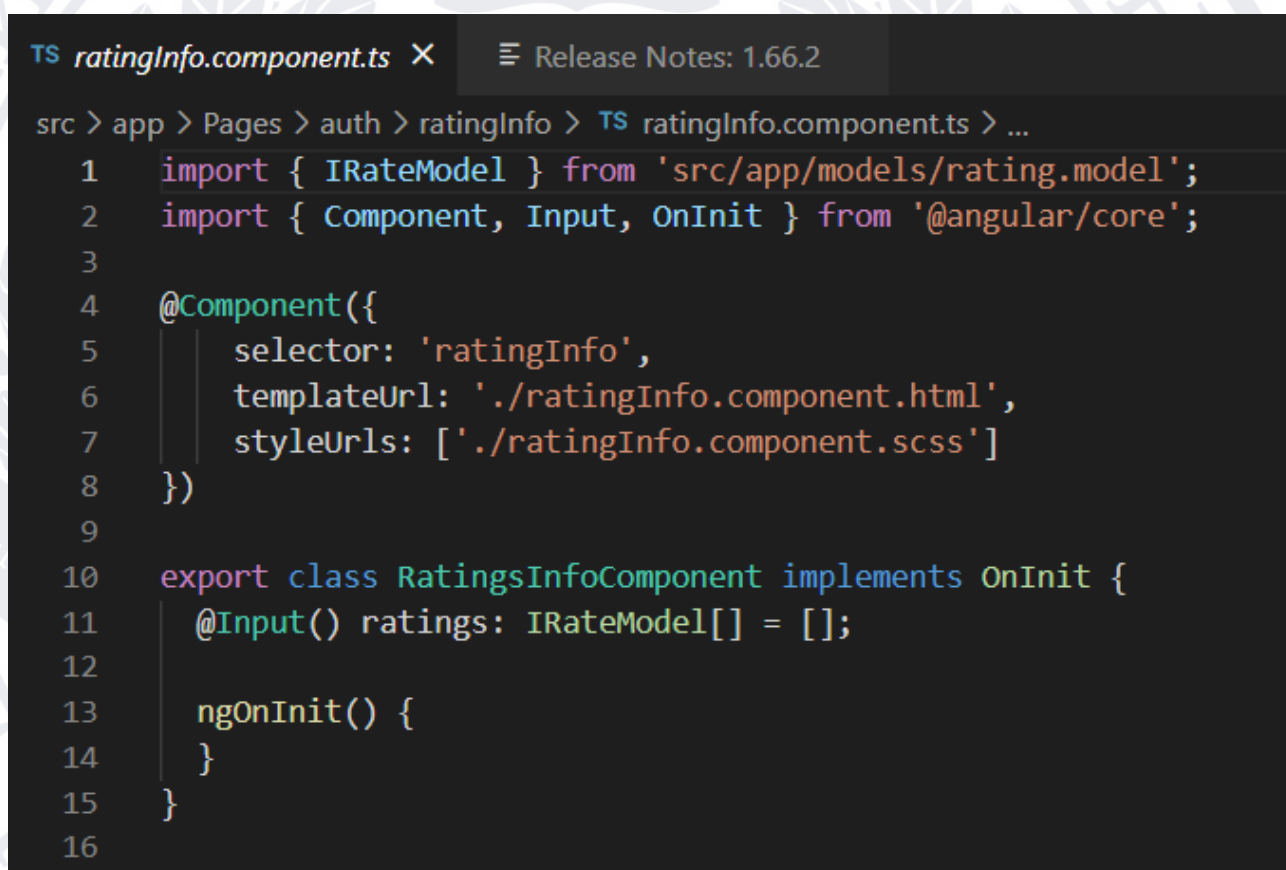
Рисунок 2.15 – Установка Angular CLI

Дехто плутає або вважає одним і тим самим Angular і AngularJS, але це не так. Angular є спадкоємцем іншого фреймворку AngularJS. У той же час Angular це не нова версія AngularJS, а новий фреймворк. Основна відмінність полягає в мові програмування, яка застосовується при розробці. В AngularJS це JavaScript, а в Angular – TypeScript. Хоча в Angular також можна застосовувати JavaScript та

мову Dart, але застосування TypeScript є найбільш розповсюдженою практикою і є найбільш доречним рішенням.

Одним із ключових елементів програм на Angular є компоненти. Компонент керує відображенням уявлення на екрані.

Компонент (Angular component) - відокремлена частина функціоналу зі своєю логікою, HTML-шаблоном та CSS-стилями. Клас стає Angular компонентом, якщо його оголошення передуює декоратор `@Component()` з об'єктом конфігурації.



The screenshot shows a code editor with a file named `ratingInfo.component.ts`. The code defines an Angular component with the following structure:

```

src > app > Pages > auth > ratingInfo > TS ratingInfo.component.ts > ...
1  import { IRateModel } from 'src/app/models/rating.model';
2  import { Component, Input, OnInit } from '@angular/core';
3
4  @Component({
5      selector: 'ratingInfo',
6      templateUrl: './ratingInfo.component.html',
7      styleUrls: ['./ratingInfo.component.scss']
8  })
9
10 export class RatingsInfoComponent implements OnInit {
11     @Input() ratings: IRateModel[] = [];
12
13     ngOnInit() {
14     }
15 }
16

```

Рисунок 2.15 – Код найпростішої компоненти в проекті

За рисунком 2.15 розібрати певні елементи з яких складається компонента в Angular. Перш за все це конвенція найменувань. Кожна компонента повинна закінчуватись ключивим словом “Component”, що полегшує роботу з проектом як і розробнику для подальшого рефакторингу коду, так самому фреймворку , сигналізуючи , що це саме компонента , а не якийсь інший об’єкт. Зверху ми

бачимо підключення сторонніх бібліотек , інших компонентів , моделей тощо. Для цього використовують блок “import”:

```
1  import { IRateModel } from 'src/app/models/rating.model';
2  import { Component, Input, OnInit } from '@angular/core';
```

Рисунок 2.16 – Підключення файлів в компоненті

Далі в декораторі @Component() можна побачити підключення html-сторінки та файлу з стилями CSS – “ratingInfo.component.html” та “ratingInfo.component.scss” відповідно.

```
4  @Component({
5      selector: 'ratingInfo',
6      templateUrl: './ratingInfo.component.html',
7      styleUrls: ['./ratingInfo.component.scss']
8  })
```

Рисунок 2.17 – Підключення стилів та html-сторінки

Після чого йде саме тіло компоненти. Компонента являє собою клас і при її оголошенні має стояти ключове слово export , завдяки якому можна використовувати дану компоненту в інших файлах. Для цього використовують декоратори @Output() – для передачі в інший файл та @Input() – для отримання даних компоненти.

```
10  export class RatingsInfoComponent implements OnInit {
11      @Input() ratings: IRateModel[] = [];
12
13      ngOnInit() {
14      }
15  }
16
```

Рисунок 2.18 – Тіло компонента

Переваги компонентів над використанням звичайного підходу розробки веб-сайту:

- ❖ Можливість перевикористання коду, тобто компоненту можна використовувати багаторазово в різних місцях з використанням різних даних
- ❖ Вкладеність, тобто використання компонента в компоненті
- ❖ Централізація коду, якщо потрібно щось змінити не потрібно лізти в купу файлів, а можна все змінити лише в одному.

Висновок до розділу 2

У цьому розділі були представлені технології та допоміжні програми, які було застосовано для комфортної та ефективної розробки клієнтської частини проекту. Було детально описано як правильно налаштувати і запустити проект. Також детально описано використані фреймворки та чому саме вони були обрані.

Розділ 3

ОПИС ВИКОРИСТАНИХ ІНСТРУМЕНТІВ ТА ТЕХНОЛОГІЙ ДЛЯ НАПИСАННЯ СЕРВЕРНОЇ ЧАСТИНИ

3.1 Інструменти

Для розробки серверної частини я обрав .NET Framework, а точніше мову програмування C# і платформу під неї ASP.NET Core. Якщо говорити про C#, ASP.NET Core, та в цілому про .NET то однозначним вибором є IDE Microsoft Visual Studio, переваги якої було детально розписано на початку другого розділу, тому перейдемо відразу до опису використаних технологій.

3.2 Вибір стеку технологій

Зазвичай більшість розробників для написання веб-додатків та сайтів надають перевагу таким мовам як JavaScript або PHP так як дані мови в основному призначені лише для роботи з WEB розробкою. Незважаючи на це я вирішив обрати C# , яка є більш функціональною та має ширшу сферу застосування ніж JavaScript та PHP.

Застосування C# для розробки:

- ❖ Консольних програм;
- ❖ Десктопних додатків за допомогою WinForms та іншими технологіями;
- ❖ За допомогою Xamarin можна робити мобільні додатки , як на Android так і на iOS;
- ❖ Web-сайтів та додатків використовуючи ASP.NET, ASP.NET Core та інші;
- ❖ Скриптів та додаткових кодів для відеоігор на двигуні Unity;

C# заслужено належить до топу найкращих мов, які інтенсивно розвиваються та використовують більший попит на ринку IT. C# застосовує

навчальну програму від найсучасніших програм і до величезних веб-сервісів, через які кожен день проходять мільйони клієнтів.

Мова програмування C# використовує синтаксис, сильно схожий на Cі та наближений до C++ і Java. Його перевагою є те що , якщо ти знайомий з одною з цих мов, то вивчити C# буде значно легше.

3.3 Платформа .NET

Якщо мова йдеться про C#, то часто розуміються технології, пов'язані з платформою .NET (WPF, ASP.NET). З іншого боку так само справедливо і в оберненому напрямку, говорячи про .NET, також мають на увазі C#. Безсумнівно, поняття мають дуже щільний взаємозв'язок, але не є синонімами. C# - це мова, створена для взаємодії з фреймворком .NET.

Ключові риси платформи:

- ❖ Працює паралельно з різними мовами. Зараз платформа здатна працювати з C#, VB.NET, C++, F#, але і на цьому список не закінчується, адже вона працює з діалектами, що прив'язані до .NET (на зразок Delphi.NET). Після компіляції коду з будь-якого з перерахованих мов, вся інтерпретується в спільну мову CIL - це своєрідний асемблер для .NET. Такий підхід дозволяє використовувати кілька мов для створення модулів програми;
- ❖ Кросплатформеність. Дану платформу реально переносити, хоч і є окремі обмеження. Сьогодні актуальна версія фреймворку працює на всіх підтримуваних Віндовс: від Windows Vista до «десятки». За рахунок проекту Mono з'явилася можливість розробляти програми під Linux (різні дистрибутиви), Android і iOS;
- ❖ Велика бібліотека класів. .NET Framework володіє єдиною, уніфікованою бібліотекою класів, з якої працюють всі підтримувані мови. Бібліотека класів стане в нагоді при створенні будь-яких програм: від блокнота до величезного веб-сайту;

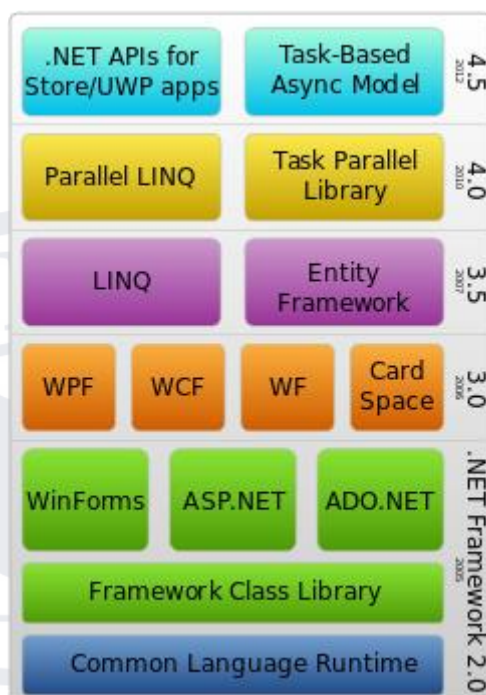


Рисунок 3.1 – Стек технологій .NET Framework

3.4 Технологія ASP.NET MVC Core

ASP.NET - це технологія для розробки сайтів та веб-додатків. Вона підтримує роботу з декількома мовами програмування, що входять в збірку фреймворка: Basic NET, C #, J # і до ряду інших. За допомогою даної платформи розробник отримує можливість створювати як прості веб-ресурси, так і дуже складні сайти, здатні до обробки багатотисячного потоку користувачів.

ASP.NET за швидкістю роботи значно перевершує інші скриптові мови програмування. Причина швидкої обробки полягає в тому, що основна частина коду компілюється при першому підключенні користувача і відразу додається в кеш комп'ютера. Всі наступні переходи по сайту використовують вже збережений в кеші код, який просто витягується з пам'яті, а не завантажується з сервера повторно з кожним візитом на сайт. Такий підхід сильно заощаджує час на парсингу, завантаженні і обробці файлів.

Framework Usage Statistics

Statistics for websites using Framework technologies

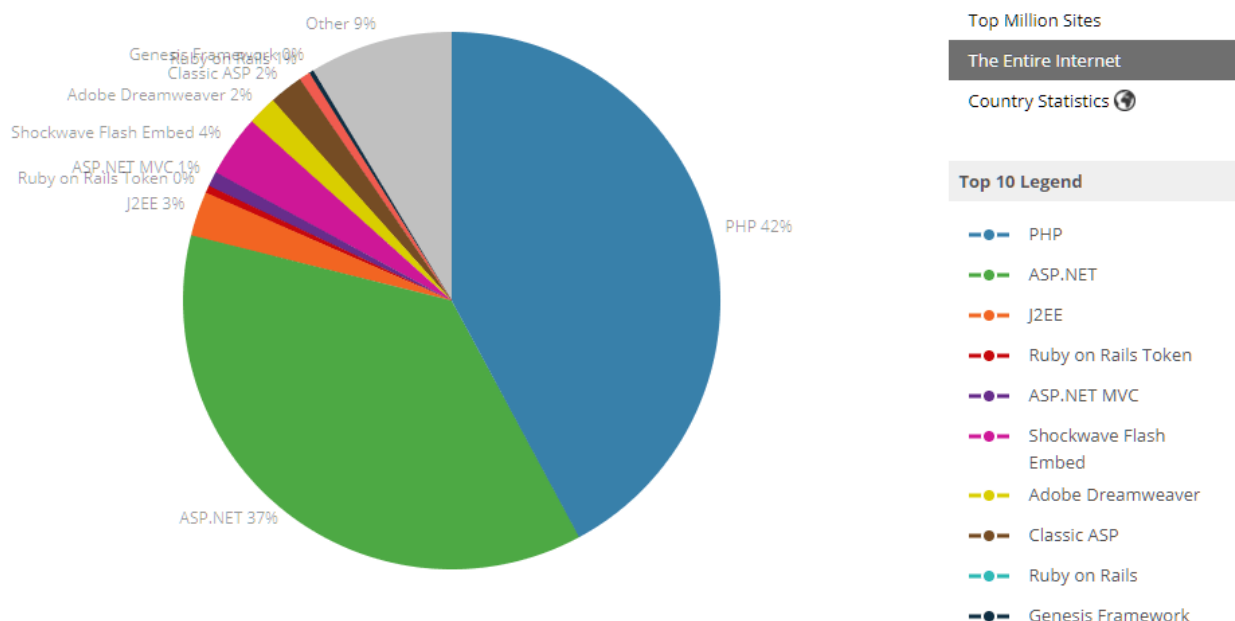


Рисунок 3.2 – ASP.NET посідає друге місце серед найпопулярніших технологій та мов програмування для розробки веб-ресурсів

У нинішніх реаліях платформа ASP .NET неймовірно популярна, вона лежить в основі величезних інтернет-порталів. Мені вдалося знайти хороший матеріал в статті на англomовному веб-сайті www.danylkoweb.com, в якому освітлено 10 кращих веб-ресурсів, створених на цій платформі.

У цьому списку присутні:

- ❖ StackOverflow - це один з найбільших форумів для вирішення різноманітних проблем та задач пов'язаних з програмуванням на будь-якій мові, фреймворкові тощо. З його допомогою тисячі людей щодня отримують відповіді на нагальні питання, що були задані іншими людьми раніше ;

- ❖ Основний портал корпорації Microsoft та більшість його програмних продуктів таких як Visual Studio, Visual Studio Code, Windows та інших. Вибір компанією Microsoft саме платформи ASP .NET є доволі

зрозумілим так як дана корпорація являється творцем мови C# та платформи .NET, до якої входить і сам ASP .NET, що можна пробачити на Рисунку 2.5;

- ❖ Найбільший сайт для реєстрації доменів GoDaddy. Сьогодні займає лідируючі місця в сфері веб-хостингу;

- ❖ Офіційний веб-сайт Dell – американської компанії, що є однією з провідних світових лідерів в галузі розробки і виробництва комп'ютерних систем.

У ASP .NET застосовується традиційна схема MVC - Модель-Представлення-Контролер (ASP NET Core MVC). Ця ж схема часто зустрічається в C# (C# MVC). Всі елементи відповідають за конкретні дії. Для прикладу, користувач запускає процес реєстрації і відправляє на сервер реєстраційні дані. Контролер інтерпретує дії людини і передає моделі інформацію про внесені зміни в статус користувача. Модель реагує на дії контролера і працює з поставляються даними. Вид відповідає за відображення інформації з моделі.

Концепція паттерна MVC передбачає поділ додатка на три компоненти:

- ❖ **Модель (model):** описує використовувані в додатку дані, а також логіку, яка пов'язана безпосередньо з даними, наприклад, логіку валідації даних. Як правило, об'єкти моделей зберігаються в базі даних. У MVC моделі представлені двома основними типами: моделі Представлень, які використовуються для відображення і передачі даних, і моделі домену, які описують логіку управління даними. Модель може містити дані, зберігати логіку управління цими даними. У той же час модель не повинна містити логіку взаємодії з користувачем і не має визначати механізм обробки запиту. Крім того, модель не повинна містити логіку відображення даних в Представленні;

- ❖ **Представлення (view):** відповідають за візуальну частину або призначений для користувача інтерфейс, нерідко html-сторінка, через який користувач взаємодіє з додатком. Також уявлення може містити логіку, пов'язану з відображенням даних. У той же час уявлення не повинно містити логіку обробки запиту користувача або управління даними;

❖ **Контролер (controller):** відповідає за центральний компонент MVC, який забезпечує зв'язок між користувачем та програмою, Представленням і сховищем даних. Він містить логіку обробки запиту користувача. Контролер отримує вводяться користувачем дані і обробляє їх. І в залежності від результатів обробки відправляє користувачеві певний висновок, наприклад, у вигляді Представлення, наповненого даними моделей.

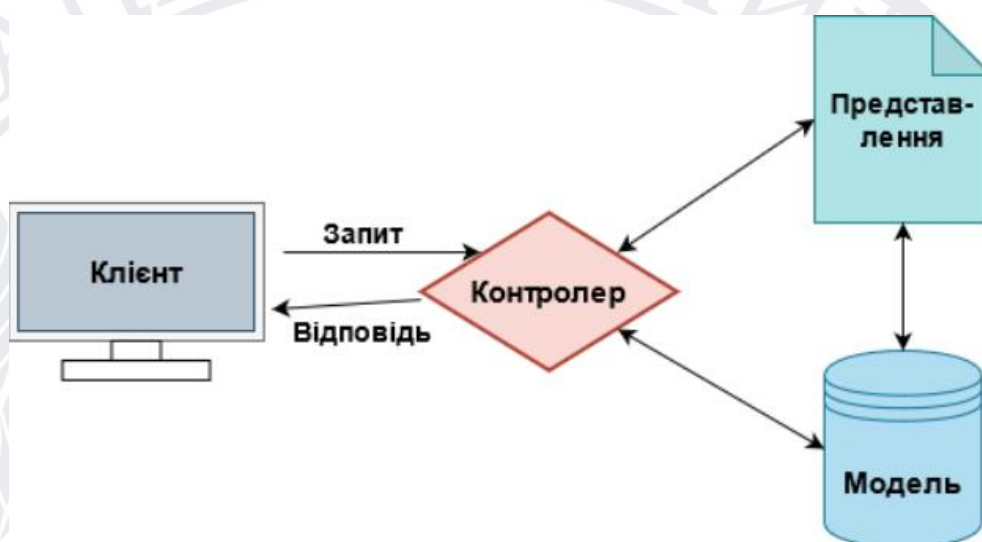


Рисунок 3.4 – Приклад стандартної MVC C#. Точно така ж використовується в ASP.NET

2.6 Різниця між ASP .NET та ASP .NET Core

.NET Core - це кросплатформене середовище для створення додатків для інтернету або консольних програм. Програмні продукти, розроблені на даний платформі, можуть успішно виконуватися на Linux, Windows, MacOS.

Основні плюси:

- ❖ Мультиплатформеність;
- ❖ Відкритий вихідний код.

.NET Framework - середовище виконання, яка призначена виключно для операційної системи Windows. Він допомагає в розробці десктопних програм під Windows і веб-додатків ASP .NET під IIS.

Основні плюси:

- ❖ Величезний асортимент готових бібліотек;
- ❖ Значно більше можливостей у порівнянні з минулим платформою .NET Core.

Сьогодні Microsoft посилено просуває і модернізує .NET Core, постійно додаючи корисний функціонал. Такими темпами це середовище в найближчий час може зрівняється з .NET Framework.

ASP .NET Core - краще застосовувати, якщо ви:

- ❖ Бажаєте встановити таргетинг в додатку на всі популярні ОС;
- ❖ Не маєте страху до вивчення нового матеріалу;
- ❖ Не боїтеся і маєте можливість приділяти достатньо часу для виправлень і доопрацюванню, адже ASP .NET Core не дійшов до статичної точки, а періодично змінюється і поповнюється новим функціоналом.

ASP .NET - ідеальне рішення, якщо ви:

- ❖ Не відчуваєте необхідність організовувати кроссплатформенну підтримку веб-додатки;
- ❖ Потребуйте стабільному середовищі розробки;
- ❖ Не маєте великої кількості часу для розробки, тобто дедлайни підтискають;
- ❖ Уже займаєтеся розробкою або модернізацією існуючої програми;
- ❖ Входите до складу команди, яка має досвід роботи з ASP.NET.

2.7 Використання бази даних

У сучасному світі більшість проектів як десктопних так і веб повинні мати місце для зберігання даних. Для цього в програмуванні застосовують бази даних. Для управління, додавання, видалення та створення даних в базі даних

використовують Системи управління базами даних (далі СУБД) та спеціальна мова для роботи з базами даних SQL.

Система управління базами даних — набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

Найпопулярніші СУБД:

- ❖ Microsoft SQL Server;
- ❖ Oracle;
- ❖ MySQL;
- ❖ PostgreSQL;
- ❖ Firebird;
- ❖ SQLite.

Так як курсова робота написана на ASP .NET MVC Core та C# я не довго думаючи обрав в ролі СУБД саме Microsoft SQL Server так як в нього найкраща сумісність з даним стеком технологій та його можна використовувати в середовищі розробки Microsoft Visual Studio 2019 , а не потрібно завантажувати додаткову програму як для інших вище перерахованих СУБД.

Microsoft SQL Server - система управління базами даних, яка розробляється корпорацією Microsoft. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі.

Для роботи з базами даних зазвичай використовують мову SQL , але в ASP .NET можна легко обійтися без знання даної мови . Це стає можливим завдяки установці офіційного плагіна від Microsoft - Entity Framework. Є два варіанта даного фреймворку : для роботи .NET Framework та .NET Core. Так як мій проект розроблений на архітектурі .NET Core , то і використовувати потрібно версію Entity Framework Core.

Entity Framework Core (EF Core) - являє собою об'єктно-орієнтовану, легковажну і здатну до розширення технологію від компанії Microsoft для доступу до даних. EF Core є ORM-інструментом (object-relational mapping - відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати базами даних, але є більш високий рівень абстракції: EF Core дозволяє абстрагуватися від самої бази даних і її таблиць і працювати з даними незалежно від типу сховища. Якщо на фізичному рівні ми оперуємо таблицями, індексами, первинними і зовнішніми ключами, то на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами певного класу.

Висновок до розділу 3

Третій розділ за своїм вмістом і структурою дечим схожий на попередній, другий розділ, але на відміну від нього в цьому розділі описуються технології для розробки серверної частини, що відповідає за логіку проекту. Крім цього було пояснено роботу з базою даних, та чому було обрано саме цю СУБД, а не іншу.

Розділ 4

ПРОГРАМНА РЕАЛІЗАЦІЯ ДОДАТКА

4.1 Опис призначеного для користувача інтерфейсу

В цьому розділі буде представлено інтерфейс веб-сайту. На мою думку, спочатку слід показати головну сторінку. На ній розміщені рекомендовані новини та пропозиції.

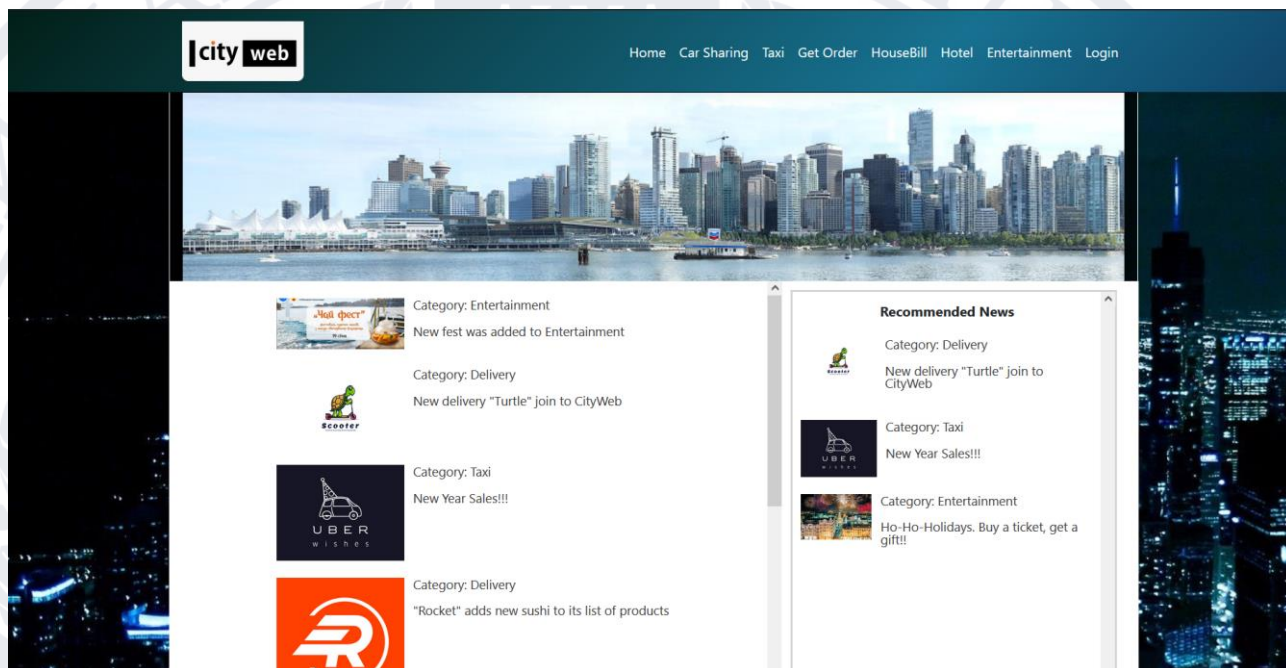


Рисунок 4.1 – Головна сторінка



Рисунок 3.2 – Логотип веб-сайту

Також на головній сторінці, так само як і на кожній, міститься шапка, в якій знаходяться гіперпосилання на всі доступні сервіси, які входять до веб-додатку, кнопку авторизації і логотип. При натисненні на логотип користувача буде переадресовано на головну сторінку.

Home Car Sharing Taxi Get Order HouseBill Hotel Entertainment Login

Рисунок 4.3 – Шапка сайту

4.2 Аунтефікація та особистий кабінет користувача

При натисненні на кнопку «Login» відкривається модальне вікно аунтефікації, з можливістю зареєструватися або увійти у вже створений акаунт. Для реєстрації потрібно ввести ім'я користувача, пошту, пароль і номер телефону. Для входу лише пошту і пароль.

Рисунок 4.4-5 – Форма реєстрації та входу

Після чого в шапці буде відображено ім'я користувача та кнопка для виходу з облікового запису :

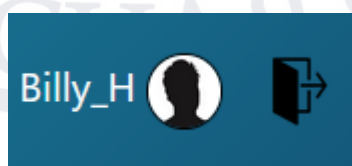


Рисунок 4.6 – Кнопка виходу і міні аватар профілю

Натиснувши на ім'я користувача, ми потрапляємо в його особистий кабінет, де поки що немає ніякої інформації про його операції на сайті, але є його особисті дані. Також є можливість зміни фото профілю.

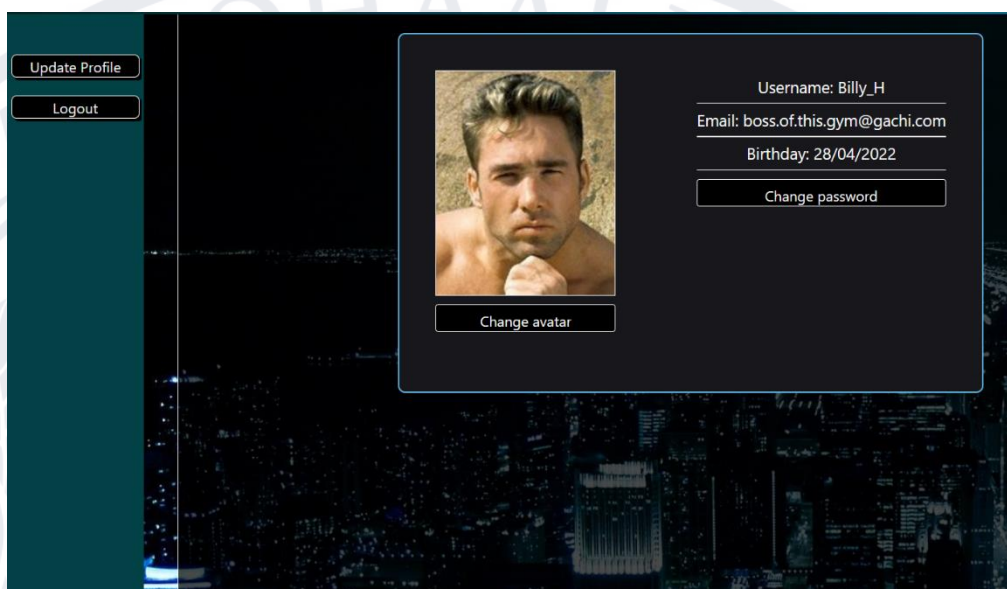


Рисунок 4.7 – Особистий кабінет користувача

Зліва присутнє меню в якому можна вийти з акаунту або відредагувати його.

Рисунок 4.8 – Форма редагування акаунта

Так виглядав особистий кабінет звичайного користувача, але також є кабінет адміністратора, в якому доступне додаткове поле зі списком всіх користувачів. Адміністратор може переглянути інформацію кожного користувача, видалити його або ж змінити йому роль назначивши його іншим адміністратором або менеджером контенту.

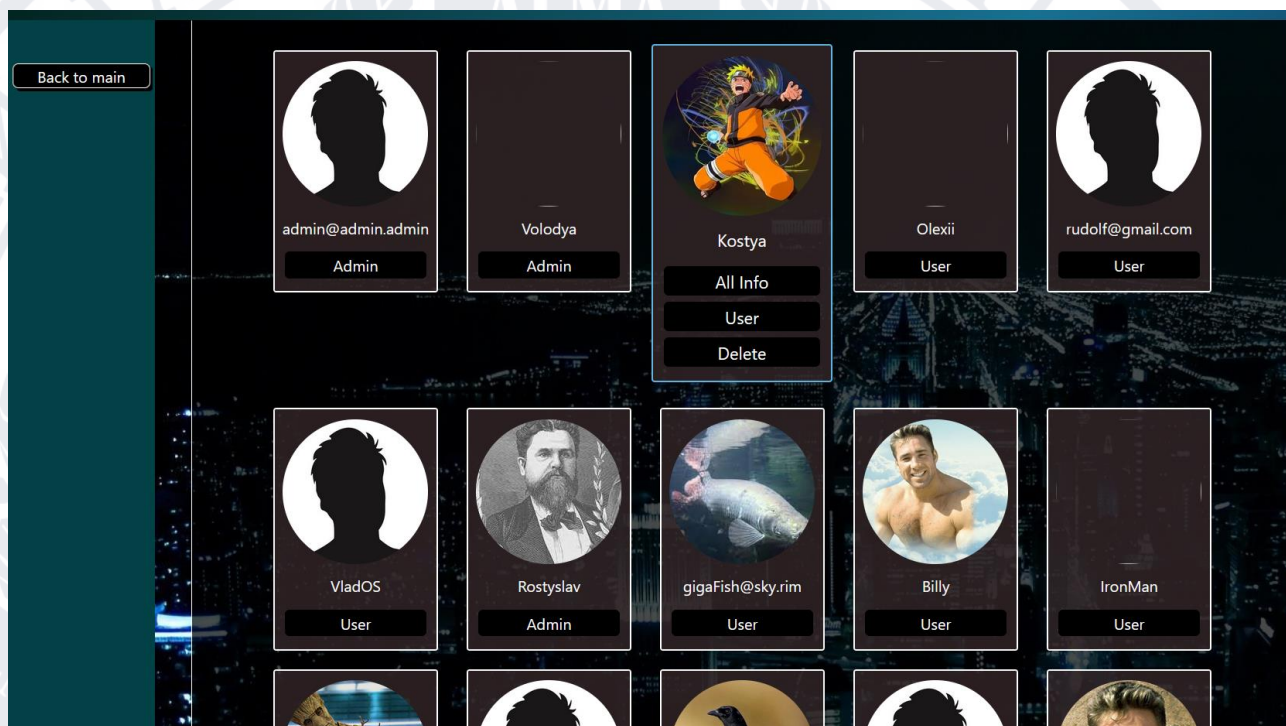


Рисунок 4.9 – Список користувачів і можливі дії над ними

4.3 Опис структури та функціоналу сервісів

На цьому можна завершити огляд особистого кабінету і перейти до першого сервісу, а саме сервісу резервування готелів. Для цього треба перейти в шапці на вкладку Hotel. На ній ми бачимо всі готелі в яких можна зарезервувати номер. Готелі відсортовано за їхнім рейтингом, який відображено на картці готелю знизу.

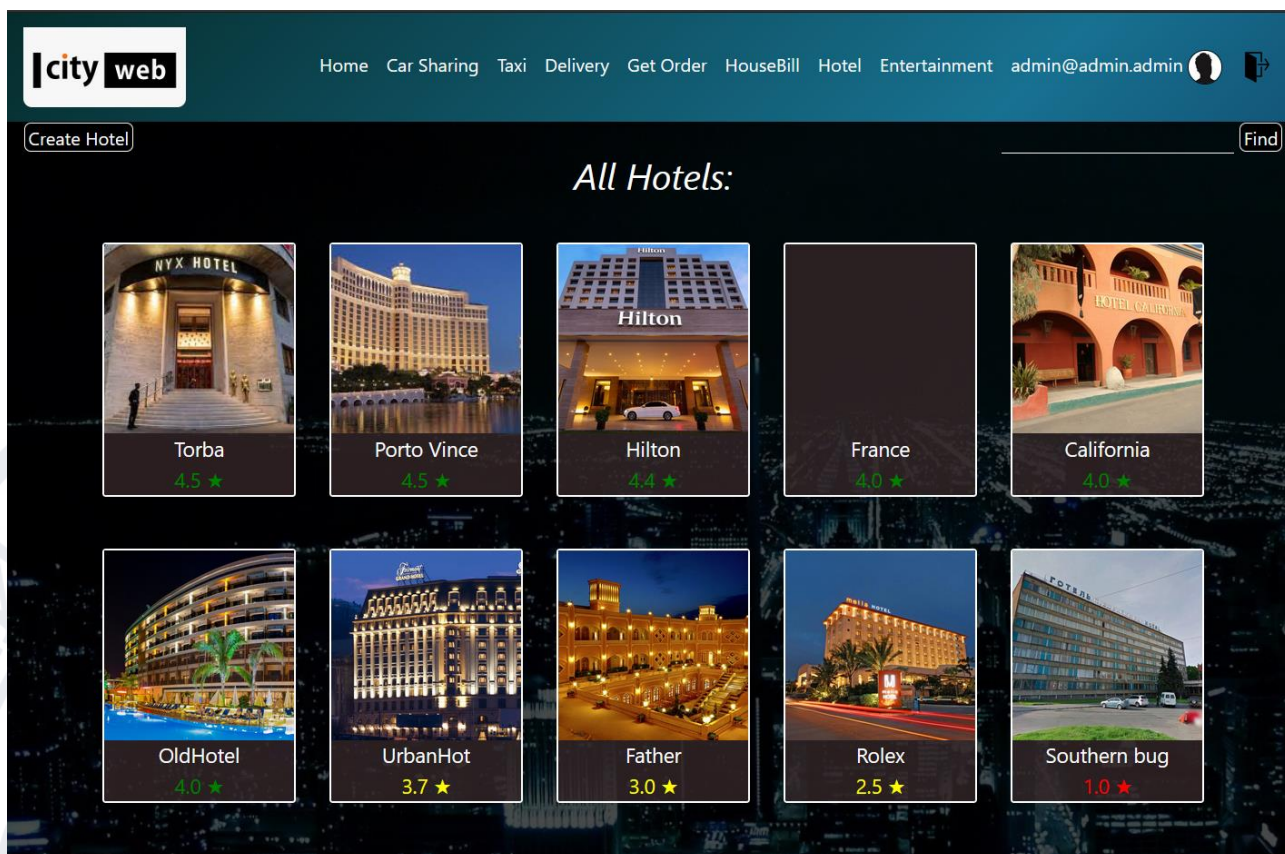


Рисунок 4.10 – Список усіх готелів

Якщо користувач володіє правами адміністратора в нього буде відображатися кнопка «Create hotel», за допомогою якої він може додати новий готель до бази даних. Адміністратор також має змогу редагувати та видаляти готель з бази.

Enter data to add new hotel

Title:

Hotel 777

Description:

777

Hotel image:

/Jf3zb67/A/jn6w7Z/RpUAAAAAEFTkSuQmCC

Street name:

Пирогова

House number:

21

Enter hotel data

Back

Рисунок 4.11 – Форма додавання нового готелю

Після натиснення на комірку готелю відкривається сторінка з більш детальною інформацією про нього. На сторінці готелю показані такі поля :

- ❖ Назва(Title);
- ❖ Опис (Description);
- ❖ Назва вулиці (Street name);
- ❖ Номер вулиці (Street number).

Також на сторінці є такі кнопки :

- ❖ Назад до списку готелю – повертає користувача на сторінку зі списком всіх готелів;
- ❖ Редагувати дані готелю – перехід на сторінку зміни всіх ключових даних про готель;
- ❖ Видалити готель – відкриває спливаюче вікно для підтвердження видалення готелю з бази даних або відміни операції;
- ❖ До кімнат – перехід на сторінку зі всіма кімнатами готелю.

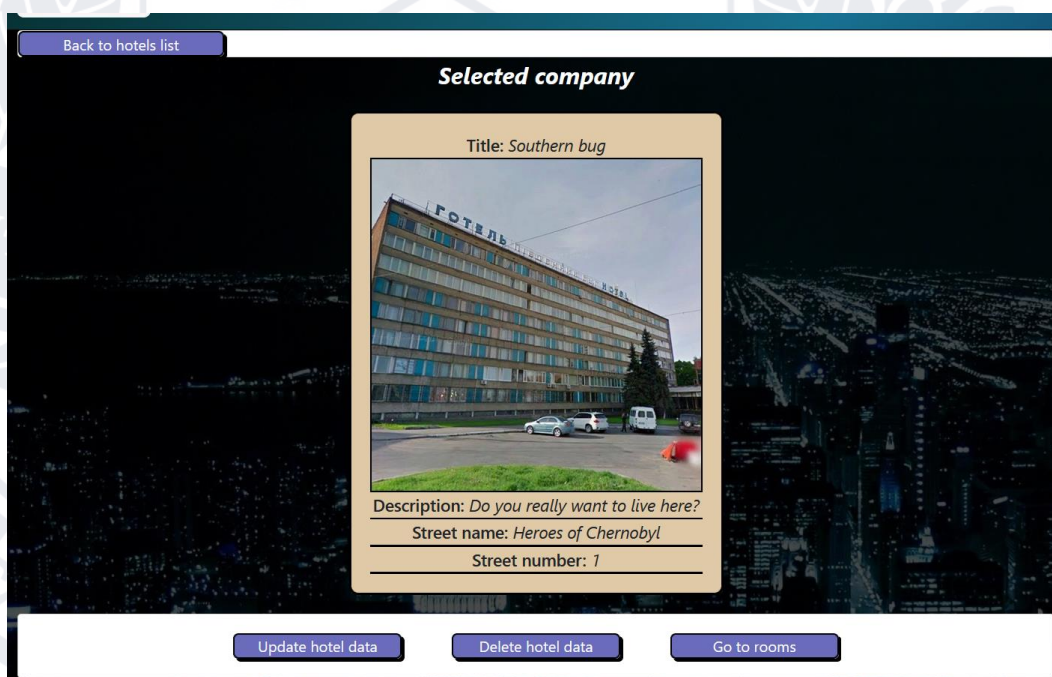


Рисунок 4.12 – Сторінка готелю

Сторінка кімнат схожа на сторінку списку готелів, але на ній не має відображення рейтингу кімнати, так як ця функція доступна лише для компаній, в даному випадку готелів. Також на сторінці присутні такі кнопки:

- ❖ Назад до готелю – повертає на сторінку готелю в якому знаходяться дані кімнати;
- ❖ Додати кімнату – додає нову кімнату до бази даних;

Крім цього на сторінці присутній пошуковий рядок в якому можна знайти кімнату за її номером або класифікацією.



Рисунок 4.13 – Сторінка списку доступних кімнат

За аналогією до готелів, також можна перейти на сторінку кожної кімнати. На сторінці кімнати можна побачити фото кімнати та інформацію про неї, а саме:

- ❖ Номер кімнати;
- ❖ Тип кімнати;
- ❖ Ціну за день проживання;

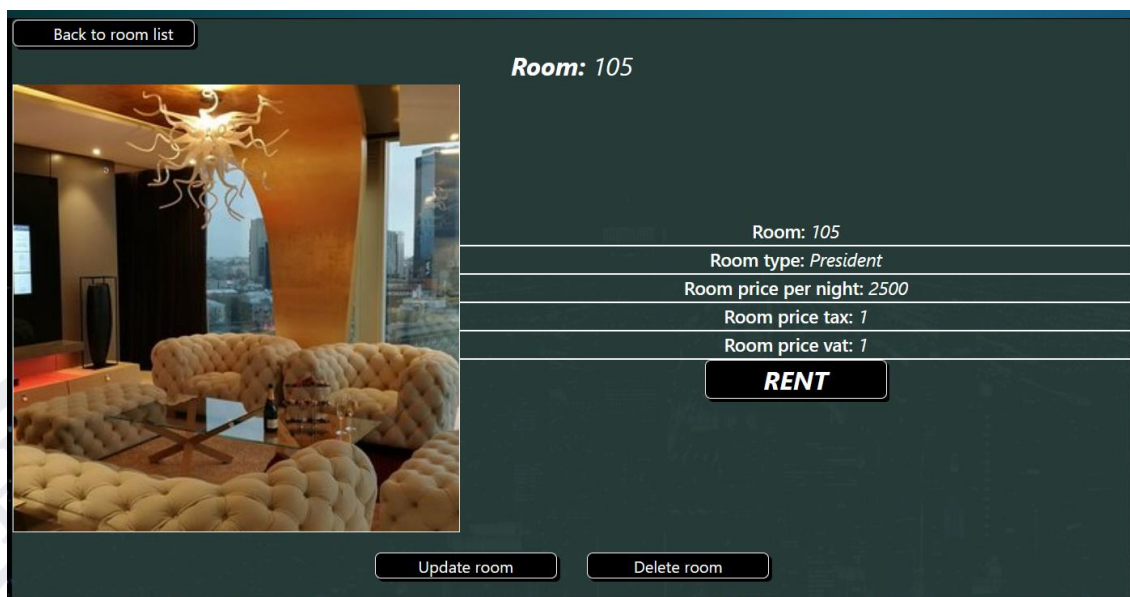


Рисунок 4.14 – Сторінка кімнати

Також на сторінці знаходиться такий набір кнопок:

- ❖ До списку кімнат – посилання на сторінку зі всіма кімнатами готелю;
- ❖ Редагувати кімнату – перехід на сторінку, де можна змінити поля наведені вище;
- ❖ Видалити кімнату - відкриває спливаюче вікно для підтвердження видалення кімнати з бази даних або відміни операції;

- ❖ Зарезервувати(Rent) – відправляє користувача на сторінку оплати, де він може обрати термін на який він бронює цей номер та безпосередньо здійснити оплату, яка автоматично вираховується в залежності від вартості кімнати та терміну оренди.

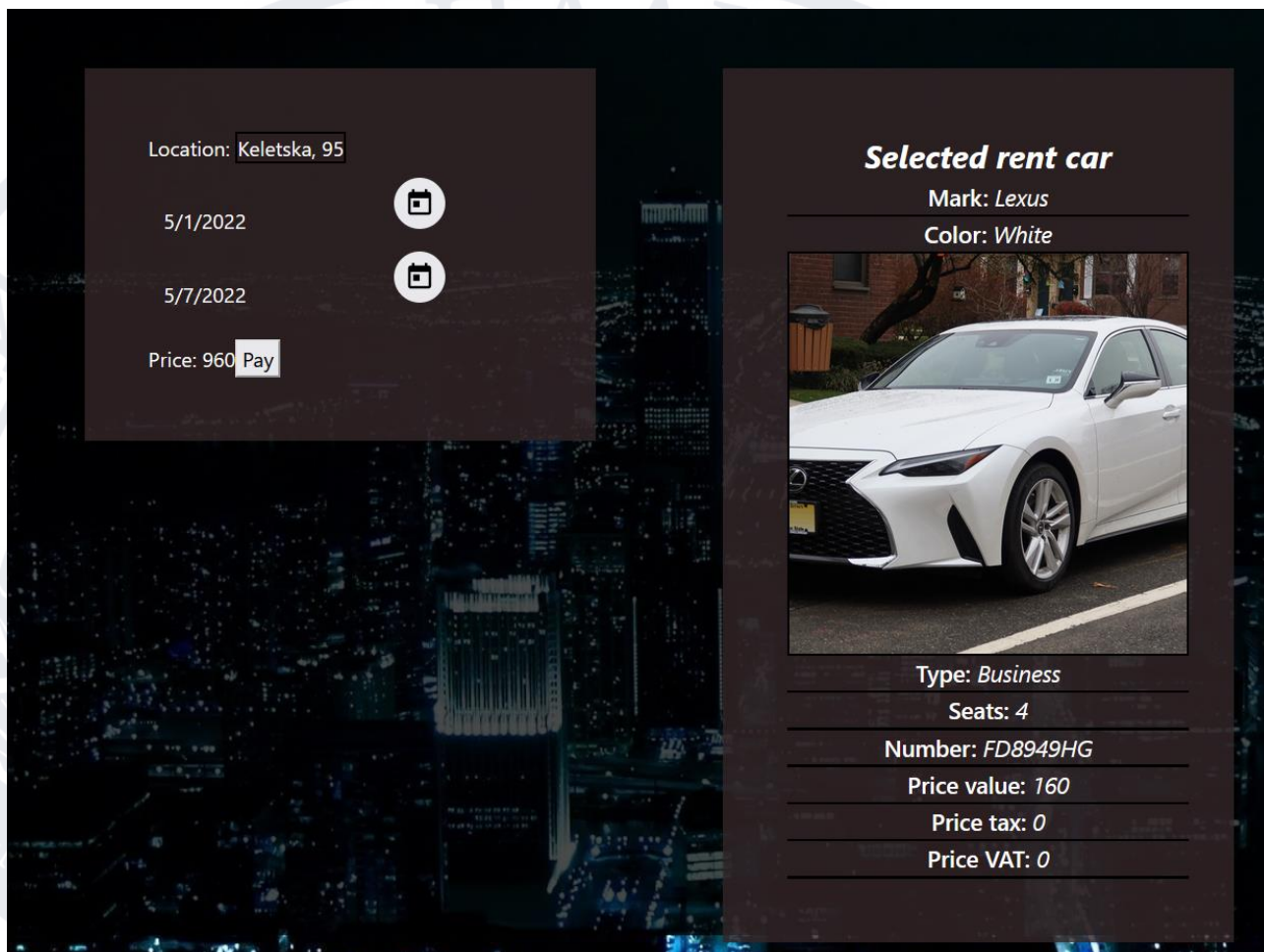


Рисунок 4.15 – Сторінка оплати замовлення

Таку ж саму структуру та послідовність і перелік операцій мають сервіси орендування машин, бронювання білетів та сервіс оплати комунальних послуг, тому розповідати про них більш детально не є дуже доречно.

4.4 Опис спеціальних функцій сайту

Але крім них також ще є два сервіси які містять у собі певні унікальні функції, які попередні сервіси не мали. Ці функції присутні в сервісі для замовлення товарів і подальшої доставки їх та в сервісі замовлення таксі.

Що ж розпочнімо з таксі. Перш за все відмінність полягає у вмісті сторінки яка відповідає за цей сервіс. На даній сторінці розміщена карта від Google, яка автоматично визначає ваше місце знаходження за вашою згодою і позначає його на карті синім маркером. Потім ви обираєте на карті точку в яку вам потрібно дістатись, після чого вираховується відстань у кілометрах яку вам потрібно подолати.

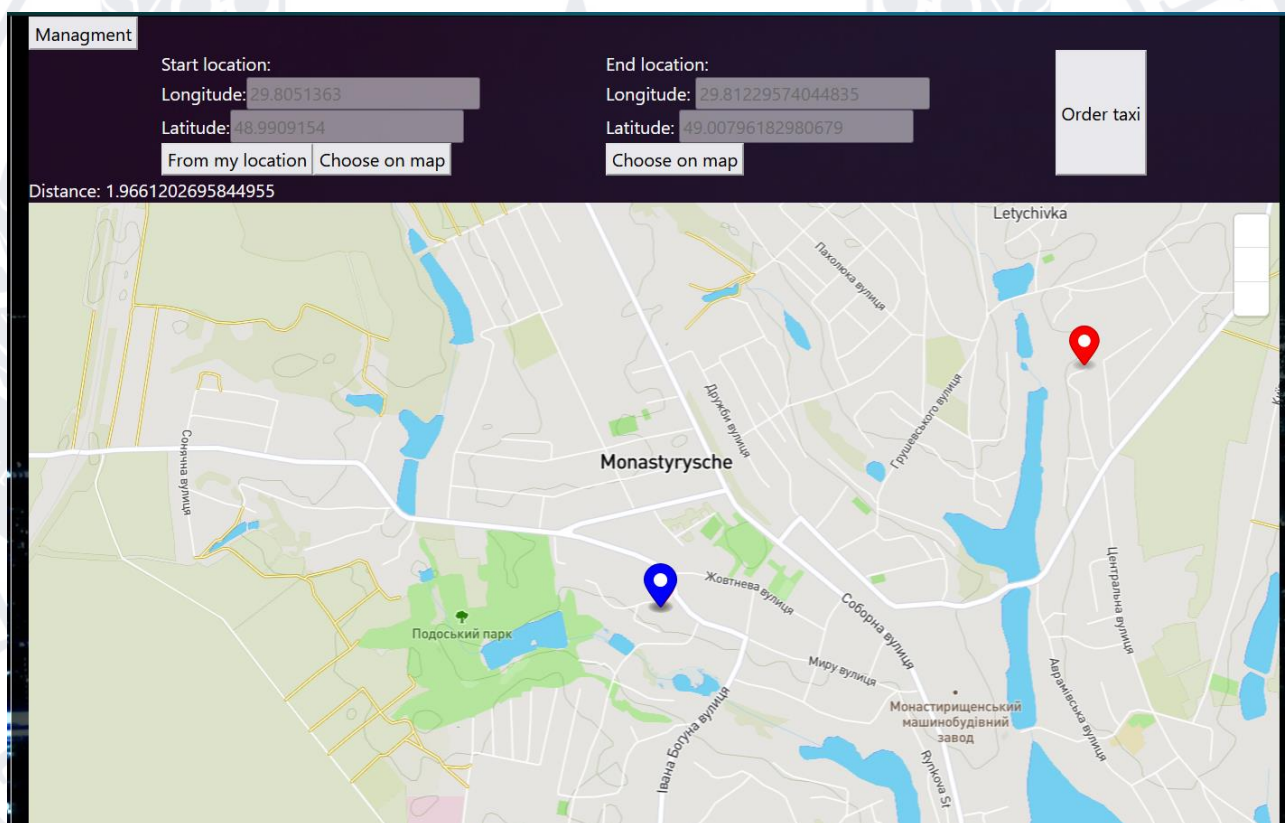


Рисунок 4.16 – Сторінка сервісу таксі

Щоб продовжити замовлення, потрібно натиснути кнопку «Замовити таксі» (Order taxi), після чого користувача буде перенаправлено на сторінку вибору доступних таксі компаній у даному регіоні. На цій сторінці користувач може обрати клас таксі. Від вибору класу буде залежати ціна за кілометр і різноманітність автомобілів.

На даний момент доступні такі види класів:

- ❖ Економ(Econom);
- ❖ Стандарт(Standart);
- ❖ Бізнес(Business);

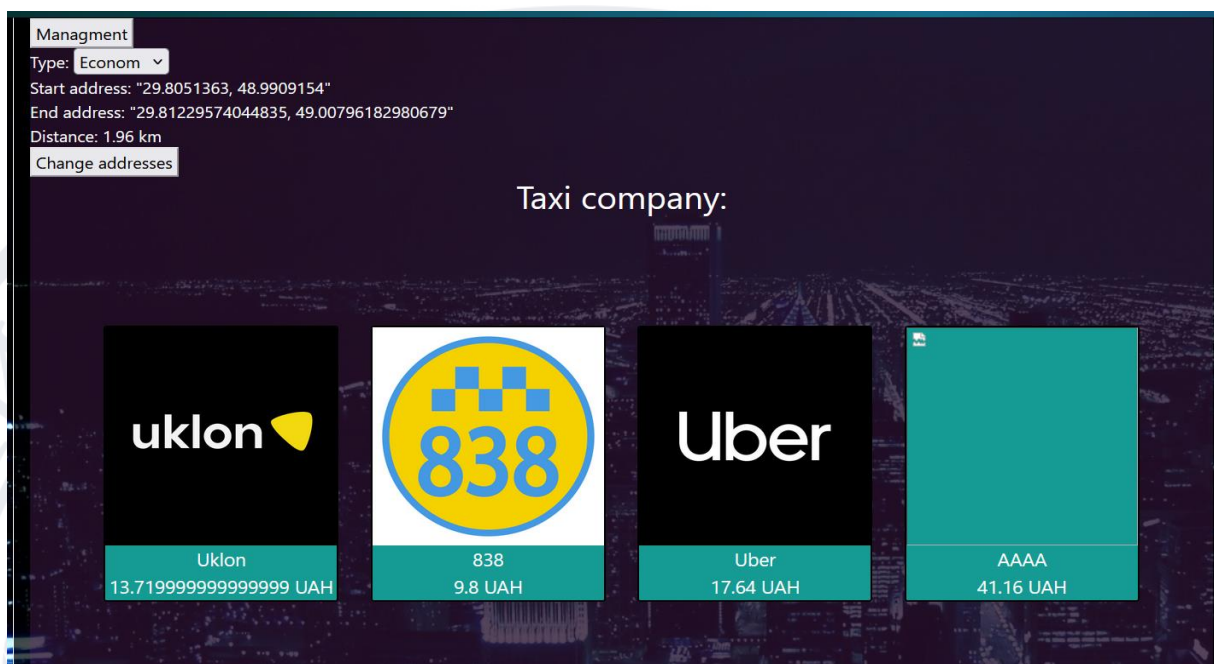


Рисунок 4.17 – Сторінка таксі компаній

Останній та від цього не менш важливий сервіс веб-платформи – це сервіс замовлення та доставки товарів. На головну сторінку сервісу можна потрапити натиснувши кнопку «Get order» в шапці сайту. На цій сторінці знаходиться список всіх служб доставки, якими можливо скористатися на веб-сайті. Список представляє собою перелік компаній в прямокутній області, де знаходиться логотип організації, її назва, рейтинговий бал та кількість користувачів, які цей бал виставили.

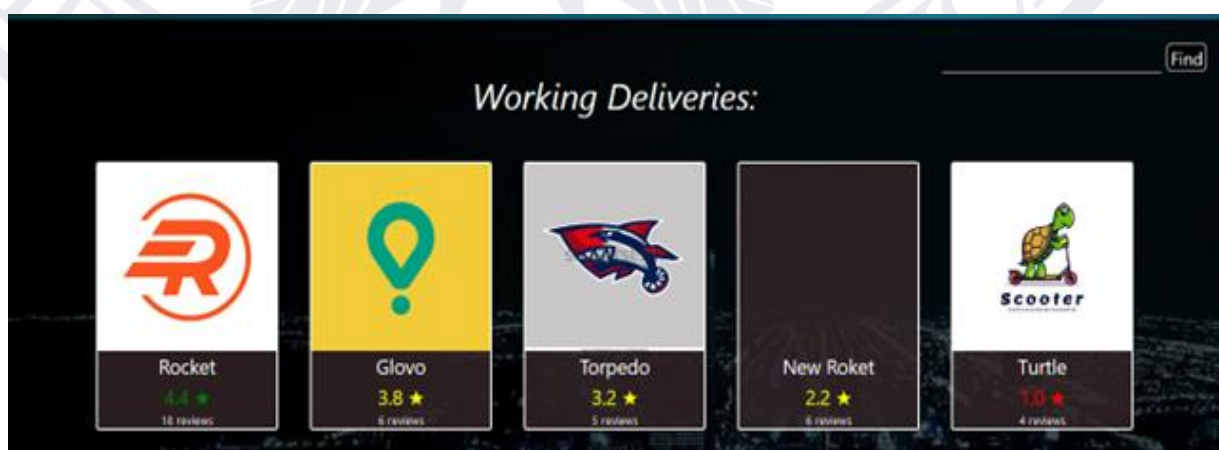


Рисунок 4.18 – Сторінка списку служб доставки

Після кліку на іконку компанії, сайт переадресовує користувача на сторінку доступних продуктів, які ця компанія доставляє. В залежності від товарів які доставляє ця фірма змінюється перелік доступних видів продуктів, який знаходиться над списком продуктів. На даний момент доступні такі види продуктів:

- ❖ Суші;
- ❖ Салати;
- ❖ Бургери;
- ❖ Піци;
- ❖ Напої;
- ❖ Десерти;
- ❖ Фастфуд;

Кожен продукт має своє фото, під яким вказано його назву, ціну у гривнях та кнопку «Додати до кошика»(To basket). Після натиснення цієї кнопки продукт додається до кошика. В кошик можна додати кілька товарів. Після додавання хоча б однієї страви до кошику іконка кошика зверху справа стає активною і доступною для переходу на сторінку замовлення.

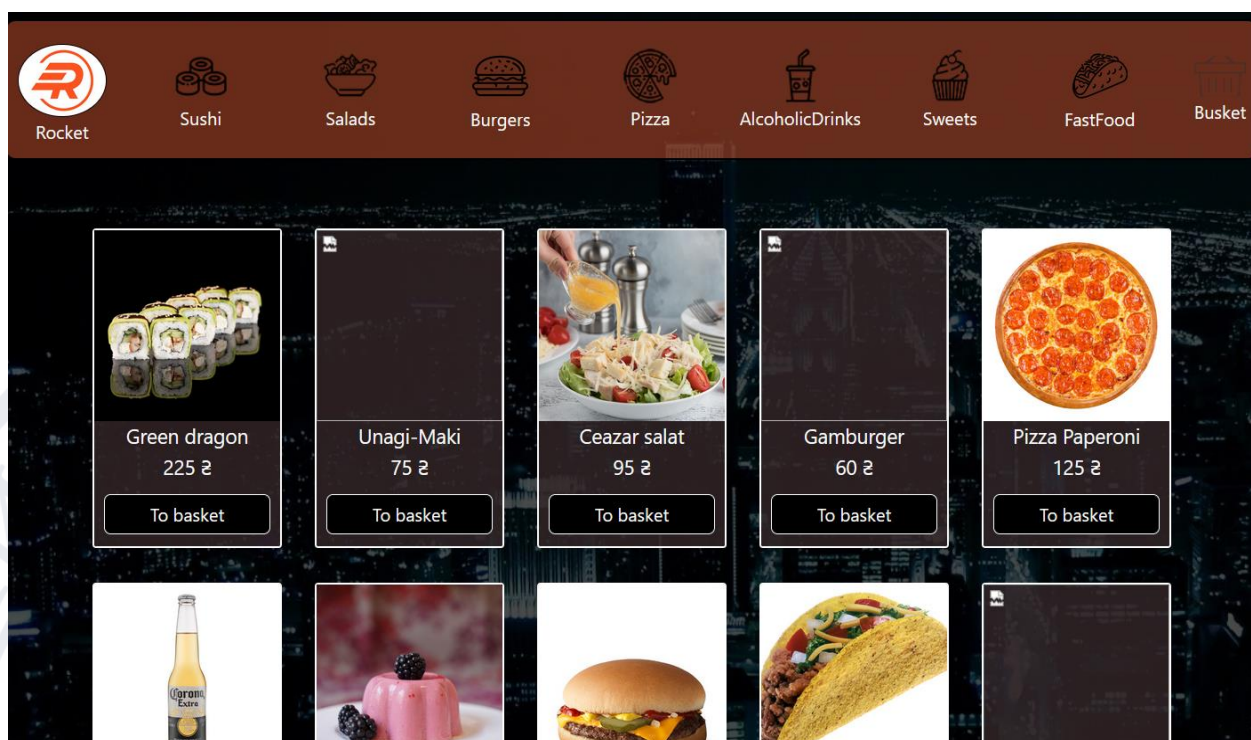


Рисунок 4.19 – Сторінка списку продуктів компанії

На сторінці кошику зліва розміщені всі додані до цього товари в кількості однієї одиниці. За потреби можна збільшити кількість будь-якого товару натиснувши знак «+» під ним, або забрати його з кошика натиснувши знак «-». Також можна очистити корзину натиснувши кнопку «Очистити кошик» (Clear the basket). При потребі можна повернутися на сторінку товарів і додати ще якісь продукти до кошику, для цього потрібно натиснути кнопку «Назад до списку продуктів» (Back to product list). При чому обрані раніше продукти нікуди не зникнуть з корзини.

Після введення адреси, номеру телефону та ім'я отримувача, стає доступною кнопка «Підтвердити замовлення» (Confirm order). Ціна буде автоматично порахована залежно від продуктів і їх кількості так вказана біля цієї кнопки.

Back to product list

Clean the basket

Make your order

	Pizza Paperoni 375 ₴	+ 3 -
	Green dragon 225 ₴	+ 1 -
	Panna cotta 80 ₴	+ 1 -
	Corona 180 ₴	+ 4 -
	Cheeseburger 225 ₴	+ 5 -

Enter your name
Влад

Enter your phone number
55555

Enter street name
Soborna st.

Enter house number
1

Enter apartment number
66

Enter some coment
Хороша піцца

Your total price: 1110 ₴

Confirm order

Рисунок 4.20 – Сторінка оформлення замовлення

Після оформлення замовлення на екрані з'явиться модальне вікно в якому клієнт може написати відгук ти поставити оцінку цій компанії.

ВИСНОВКИ

У межах цієї бакалаврської роботи була розглянута та досліджена сфера розробки веб-сайтів. Актуальність переходу певних видів бізнесу в онлайн формат, переваги та недоліки даного рішення, вплив на прибуток. Також були знайдені та порівняні аналоги даного проекту.

Крім цього ми проаналізували сучасні ефективні і комфортні інструменти , програми та фреймворки для розробки як клієнтської так і серверної частини сайту.

В результаті даної роботи було створено добре функціонуючий веб-сайт для оплати міських сервісів, таких як: бронювання кімнат в готелі, квитків, замовлення та доставки їжі, виклик таксі, оренда авто і оплата комунальних послуг. Цей сайт-платформа буде актуальним не лише під час пандемії, а ще довгий час після її завершення.

СПИСОК ДЖЕРЕЛ

1. Вплив коронавірусу на бізнес URL: <https://www.bdo.ua/uk-ua/about-2/business-impact-of-covid-19/insights>
2. Rozetka URL: <https://rozetka.com.ua/>
3. Розетка (інтернет-магазин) URL: [https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%B5%D1%82%D0%BA%D0%B0_\(%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD\)](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%B5%D1%82%D0%BA%D0%B0_(%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82-%D0%BC%D0%B0%D0%B3%D0%B0%D0%B7%D0%B8%D0%BD))
4. Starwood (інтернет-магазин) URL: <https://starwood.vn.ua/ua/>
5. Microsoft Visual Studio URL: <https://visualstudio.microsoft.com/vs/>
6. Найпопулярніші IDE для мови C# URL: <http://chvv.com.ua/top-populyarnyh-ide-dlya-programuvannya-na-s/>
7. IIS Express URL: <https://www.microsoft.com/en-us/download/details.aspx?id=48264>
8. Visual Studio Code URL: <https://code.visualstudio.com/>
9. TypeScript URL: <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>
10. Angular URL: <https://angular.io/>
11. Model–view–controller (MVC) URL: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
12. Node.js URL: <https://nodejs.org/uk/>
13. Angular CLI URL: <https://angular.io/cli>
14. AngularJS URL: <https://angularjs.org/>
15. Angular component URL: <https://angular.io/api/core/Component>
16. .NET Framework URL: <https://support.microsoft.com/uk-ua/topic/microsoft-net-framework-4-8-%D0%B0%D0%B2%D1%82%D0%BE%D0%BD%D0%BE%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE-%D1%96%D0%BD%D1%81%D1%82%D0%B0%D0%BB%D1%8F%D1%82%D0%BE%D1%80%D0%B0-%D0%B4%D0%BB%D1%8F-windows-9d23f658-3b97-68ab-d013-aa3c3e7495e0>
17. Мова програмування C# URL: <https://docs.microsoft.com/uk-ua/dotnet/csharp/>
18. ASP.NET MVC Core URL: <https://docs.microsoft.com/uk-ua/aspnet/core/mvc/overview?view=aspnetcore-6.0>

19.10 кращих веб-ресурсів, створених на ASP .NET URL:

<https://www.danylkoweb.com/Blog/top-10-websites-written-using-aspnet-mvc-JK>

20. Термін База даних та різновиди СУБД URL:

https://uk.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85

21. Документація з Entity Framework URL: <https://docs.microsoft.com/ru-ru/ef/>

22. Microsoft SQL Server URL: <https://www.microsoft.com/en-us/sql-server/sql-server-2019?rtc=1>

23. Рихтер Дж. P55CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. — СПб.: Питер, 2013. — 896 с.: ил. — (Серия «Мастер-класс»)

24. [Книга] Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages 9th ed. Edition – by Adam Freeman

25. [Книга] C# 10 in a Nutshell: The Definitive Reference 1st Edition – by Joseph Albahari

26. ASP.NET Core in Action, Second Edition – by Andrew Lock

27. Programming Entity Framework 1st Edition – by Julia Lerman

28. Programming Entity Framework: DbContext Released February 2012
Publisher(s): O'Reilly Media, Inc. ISBN: 9781449312961 - by Julia Lerman, Rowan Miller

29. SQL Cookbook: Query Solutions and Techniques for Database Developers (Cookbooks (O'Reilly)) 1st Edition – by Anthony Molinaro

30. SQL Tutorial URL: <https://www.w3schools.com/sql/>

31. Bootstrap 5 Tutorial URL: <https://www.w3schools.com/bootstrap5/index.php>

32. HTML Tutorial URL: <https://www.w3schools.com/html/default.asp>

33. CSS Tutorial URL: <https://www.w3schools.com/css/default.asp>

34. C# Tutorial URL: <https://www.w3schools.com/cs/index.php>

35. Язык программирования C# и .NET URL: <https://metanit.com/sharp/tutorial/>

36. Evil by Design: Interaction Design to Lead Us into Temptation Paperback – Illustrated, June 17, 2013 – by Chris Nodder

37. Книга HTML in Easy Steps от In Easy Steps

38. ASP.NET Core Crash Course - C# App in One Hour [Відеоресурс] URL: <https://youtu.be/BfEjDD8mWYg>

39. Angular Tutorial for Beginners: Learn Angular & TypeScript [Відеоресурс] URL: <https://youtu.be/k5E2AVpwsko>

- 40.C# Programming All-in-One Tutorial Series (6 HOURS!) [Відеореcypc] URL:
<https://youtu.be/qOruiBrXlAw>
- 41.ng-book: The Complete Guide to Angular 4 – Видавництво: CreateSpace
Independent Publishing Platform. Автори: Nathan Murray, Ari Lerner, Felipe
Coury, Carlos Taborda
- 42.ASP.NET Core 2 and Angular 5. Full-Stack Web Development with .NET
Core and Angular - Видавництво: Packt Publishing. Автор: Valerio De
Sanctis
- 43.Angular Development with TypeScript - Автори: Антон Мойсеев, Яков
Файн

