

Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій
Кафедра комп'ютерних наук та інженерії програмного
забезпечення

Кваліфікаційна робота бакалавра

на тему «Розробка веб-застосунку для допомоги волонтерам та внутрішньо
переміщеним особам»

Виконав: студент групи ПЗ19-2

Спеціальність 121 Інженерія програмного
забезпечення

Пирогов Владислав Ігорович
(прізвище та ініціали)

Керівник к.т.н., доц. Ульяновська Ю.В
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент в.о. завідувача кафедри
кібербезпеки
(місце роботи)
та інформаційних технологій

к.т.н., доц. Прокопович-Ткаченко

(науковий ступінь, вчене звання, прізвище та ініціали)

Д.І.

Дніпро – 2023

АНОТАЦІЯ

Пирогов В.І. «Розробка веб-застосунку для допомоги волонтерам та внутрішньо переміщеним особам»

Кваліфікаційна робота на здобуття освітнього ступеня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення». – Університет митної справи та фінансів, Дніпро, 2023.

Кваліфікаційну роботу бакалавра присвячено розробці Розробка веб-застосунку для допомоги волонтерам та внутрішньо переміщеним особам «UANHelper».

В першому розділі було проведено аналіз предметної області. Досліджено актуальність та проаналізовано аналоги WEB-застосунків, проведено детальний аналіз та порівняння переваг використання WEB – застосунків.

В другому розділі було проведено визначення програмних засобів, технологій та методів необхідних для розробки програмного продукту та їх обґрунтування.

В третьому розділі опис та результати розробки веб-застосунку. Проведення незалежного тестування веб-застосунку на різних пристроях та тестування за допомогою зовнішніх програмних комплексів для розробників.

Кваліфікаційна робота містить 64 сторінок, 26 рисунків, список літератури із 23 найменувань, 2 додатки.

Ключові слова

WEB-ДОДАТОК, PHPSTORM, VISUAL STUDIO CODE, ASP.NET, WEB API, FIGMA, HTML, C#, CSS, TYPESCRIPT, ANGULAR, MICROSOFT AZURE, POSTGRESQL, БД.

Список публікацій здобувача

1. Наукова(конкурсна) робота «Помічник старости», керівник Мала Юлія Анатоліївна(2021 р.) – 3 місце (2 тур) в м. Дніпро
2. Мала Ю.А., Пирогов В.І., Лапець О.В. Використання ІТ-технологій в навчальному процесі. Міжнародна наукова інтернет-конференція "Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення (випуск 59)" / Збірник тез доповідей: випуск 59 (м. Тернопіль, 8 червня 2021 р.). С.37-39.

ЗМІСТ

АНОТАЦІЯ.....	1
ЗМІСТ.....	3
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ.....	5
ВСТУП.....	6
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ.....	10
1.1. Аналіз предметної області. Поняття «Web-застосунок».....	10
1.2. Переваги використання Web-застосунків.....	11
1.3. Аналіз аналогічних Web-застосунків.....	13
1.4. Висновки до першого розділу	14
РОЗДІЛ 2. ВИЗНАЧЕННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ЗАСТОСУНКУ ТА ЇХ ОБГРУНТУВАННЯ.....	16
2.1. Середовище розробки PHPStorm.....	16
2.2. Середовище розробки Visual Studio Code.....	17
2.3. Особливості платформи ASP.NET WEB API.....	20
2.4. Графічний редактор Figma.....	23
2.5. Мова програмування та розмітки гіпертексту HTML.....	25
2.6. Мова програмування C#.....	28
2.7. Каскадні таблиці стилей CSS.....	27
2.8. Фреймворк Angular.....	31
2.9. Хмарні бази даних Microsoft Azure.....	32
2.10. Хмарні бази даних PostgreSQL.....	35
2.11. Висновки до другого розділу.....	37
РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ	38
3.1. Розробка плану проекту	38
3.2. Структура проекту.....	40
3.3. Створення дизайнерського макету.....	40

3.4. Реалізація системи авторизації.....	43
3.5 Реалізація взаємодії волонтера та переселенця.....	48
3.6. Тестування веб-застосунку.....	49
3.7. Рекомендації щодо подальшого вдосконалення.....	55
3.8. Висновки до третього розділу	57
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТКИ.....	63

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА ТЕРМІНІВ

ПЗ – програмне забезпечення.

Фреймворк — це готовий до використання комплекс програмних рішень, включаючи дизайн, логіку та базову функціональність системи або підсистеми.

Фронтенд – клієнтська сторона WEB-додатку.

RHPStorm - це спеціалізований інструмент для веб-розробки.

Visual Studio Code - середовище розробки.

ASP.NET WEB API – технологія створення веб-застосунків і веб-сервісів

Figma – графічний редактор.

HTML – Hyper Text Markup Language, стандартизована мова розмітки документів в мережі.

C# - мова програмування.

CSS – Cascading Style Sheets, каскадні таблиці стилів.

WEB – застосунок – застосунок, в якому клієнтом є оглядач Інтернету.

Cookie – фрагмент даних.

Angular – фреймворк для розробки сайтів.

Microsoft Azure - хмарна обчислювальна платформа.

PostgreSQL - відкрита об'єктно-реляційна база даних.

Авторизація – керування рівнями та засобами доступу до ресурсу залежно від ідентифікатора і пароля користувача.

Автентифікація – процедура встановлення належності користувачеві певної інформації за допомогою наданого ним ідентифікатора.

HTTP – Hyper Text Transfer Protocol, прикладний протокол передачі даних у комп'ютерних мережах

БД – База даних

ВСТУП

Внаслідок повномасштабного вторгнення Росії мільйони українців були змушені залишити свої домівки. Частина стали біженцями, частина – внутрішніми переселенцями. Статус останніх в Україні мають майже п'ять мільйонів осіб, хоча реальна кількість ВПО вища, зазначають у Міністерстві реінтеграції територій.

Ці люди опинилися в дуже скрутній ситуації, шукають будь-яку швидку та якісну допомогу у волонтерів. Швидкість знаходження волонтерів залежить від кількості людей, які відгукнуться на прохання про допомогу.

Волонтерський рух набуває все більшої популярності у світі, що є поширеною практикою надання безоплатної допомоги. Усюди по країні діють різні групи волонтерів, які надають допомогу людям або країні. Всім подобається допомагати по-своєму, але цей рух об'єднує всіх людей.

Незважаючи на наявність великої кількості волонтерів, які створюють гуманітарне забезпечення, допомагають у пошуку нового житла для родини у різних містах або областях, на жаль, немає централізованого web-додатку, який зміг би об'єднати всіх бажаючих допомогти людей для досягнення найшвидшої та найефективнішої волонтерської допомоги.

Важливо активно просувати і рекламувати такий рух, щоб залучати до нього нових людей і спонукати їх до допомоги. Багато волонтерських організацій мають свої сторінки в соціальних мережах, але часто вони не мають спеціалізованих веб-сайтів, де організації можуть знайти волонтерів, а волонтери - місії для допомоги.

Пропоную спростити цей процес за допомогою web-платформи, яка допоможе забезпечити ефективно поєднання потреб переселенців з бажанням волонтерів допомогти.

Отримати волонтерську допомогу можуть переселенці з окупованих територій, з територій, на яких можливі чи ведуться бойові дії, українці, чий оселі зруйновані або непридатні для життя через війну. За допомогою такого web-

застосунку, збільшується шанси швидкості створення та обробки заяви знаходження внутрішньо переміщеної особи, що безпосередньо впливає на надію та ймовірність отримати будь-яку конкретну допомогу.

Крім того, такий web-застосунок допоміг би об'єднати всіх українських волонтерів по території нашої країни, організувати гуманітарне забезпечення, пошук житла, допомогу військовим, тощо. Такий спеціалізований web-застосунок стане центральною точкою для обміну інформацією та координації діяльності волонтерів.

Додаток повинен відповідати наступним критеріям: має бути легко доступним; використання повинно бути простим, а інтерфейс повинен бути зрозумілим для всіх користувачів.

Метою написання кваліфікованої роботи бакалавра є автоматизація допомоги переселенцям шляхом створення системи об'єднання та забезпечення волонтерської допомоги, тобто web-застосунку для організації online взаємодії волонтерів та внутрішньо переміщених осіб.

Для досягнення поставленої мети в дипломній роботі ставились та вирішувались наступні завдання дослідження:

- Дослідити попит на реалізацію даного веб-застосунку
- Проаналізувати методи та засоби для реалізації веб-застосунку
- Розробити веб-застосунок і розвернути його на сервері для користування

Завдання розробити адаптивний web-застосунок для забезпечення волонтерської допомоги, створити зручну систему подачі заявок, сформувати базу даних з заявками на допомогу внутрішньо переміщеним особам та переселенцям.

Об'єктом дослідження є процес створення online - заявки для організації волонтерської допомоги.

Предметом дослідження є створення веб-застосунку для допомоги волонтерам та внутрішньо переміщеним особам.

Методи вирішення завдань роботи: для створення серверної частини web - застосунку online взаємодії волонтерів та внутрішньо переміщених осіб було використано фреймворк Angular, який написано мовою програмування TypeScript, та технологію ASP.NET Core WEBAPI, яка пишеться високорівневою мовою C#. Для клієнтської частини використовувались: мова розмітки HTML, мова опису зовнішнього виду документа, CSS, а також мова програмування JavaScript, зокрема бібліотека Angular та jQuery. Для роботи з базою даних додаток буде використовувати систему керування базами даних Microsoft Azure та PostgreSQL. Web-застосунок складатиметься з адмін-панелі в якій можна зареєструватися волонтеру, ознайомитися з видами волонтерської допомоги, створити online заявку, без реєстрації та дізнатися про благодійність та партнерство веб – застосунку, зорієнтуватися на карті де знаходяться найближчі пункти волонтерської допомоги.

Практичне значення полягає в розробці web-застосунку та вдосконаленні функціонального блоку з online-заявки отримання будь-якої волонтерської допомоги. Такий додаток можна застосовувати у багатьох країнах світу, розширювати функціонал та його корисність.

Структура роботи зумовлена її метою та поставленими задачами і складається з переліку умовних скорочень, вступу, трьох розділів, загальних висновків по роботі та списку використаних джерел (23 – найменувань). Повний обсяг роботи становить 64 сторінки, з них основного тексту – 59 сторінок. У роботі налічується 26 рисунків.

Перший розділ складається з обґрунтування актуальності теми для дослідження, аналізу наявних аналогів та аналізу їх переваг та недоліків. На основі проведеної роботи була сформульована мета та завдання програмного продукту, перелік необхідного функціональності для коректної роботи веб-застосунку, а також обрано програмне забезпечення для створення веб-застосунку. У другому розділі розглядаються теоретичні питання розробки веб-застосунків на мові C# із використанням технологій .NET Core. Також обґрунтовується вибір програмного

забезпечення. Третій розділ описує технічне завдання, вимоги до даних, структуру проекту, безпосередньо реалізацію проекту, розповідаються деталі розробки (написання класів та функцій), показуються скріншоти реалізації програми та тестування застосунку.

РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Аналіз предметної області. Поняття «WEB-додаток»

Головна мета проекту полягає у створенні простого в використанні додатку для внутрішньо переміщених осіб у пошуку волонтерської допомоги.

З інтернетом змінилося наше ставлення до інформації. Майже у всіх нас є смартфон, і ми можемо швидко знайти в мережі те, що нас цікавить в будь-який момент. Також змінився наш підхід до інформації - більше немає потреби зберігати її на особистих пристроях, але набагато важливіше знати, де можна знайти актуальну інформацію в Інтернеті.

Саме тому оптимальним рішенням є розробка веб-додатку, що дозволить залучити найбільшу аудиторію користувачів. Веб-додаток представляє собою розподілений застосунок, в якому браузер виступає клієнтом, а вебсервер - сервером [1].

Web-застосунки - це програми, розроблені для використання в мережі Інтернет, які дозволяють користувачам взаємодіяти з даними шляхом введення, отримання та маніпулювання інформацією. Ці програми зазвичай мають активний зв'язок з сервером, відправляючи йому багато запитів. Вони можуть бути інтегровані в веб-сторінки або виступати самостійними додатками. Наприклад, такими веб-застосунками є Facebook, Gmail, YouTube, eBay, Twitter, Amazon.

Web-додатки використовують ідентифікаційні дані, такі як ім'я користувача і пароль, для аутентифікації користувачів. Вони надають можливість відвідувачам обмінюватися миттєвими повідомленнями через чат-платформи, соціальні мережі або блоги, створювати контент, спеціально налаштований для вподобань користувача, і забезпечують необмежений доступ до цього контенту. Також вони можуть використовувати міні-вбудовані програми для розваг. Багато веб-додатків

можуть не мати значимого інформаційного наповнення, але використовуються для виконання додаткових завдань, таких як інтернет-перекладачі, конвертери файлів або месенджери. Оскільки використання веб-додатків часто пов'язане з введенням особистих даних та платіжних реквізитів, розробники несуть відповідальність за захист конфіденційної інформації.

1.2. Переваги використання Web-застосунків

Останнім часом кількість сервісів і ресурсів в Інтернеті швидко зростає. Інтернет перетворився з простих статичних сторінок на потужний засіб для взаємодії та спілкування з користувачами. У зв'язку з цим веб-додатки здобули велику популярність, оскільки вони пропонують багато важливих переваг, які відсутні в традиційних додатках. Індустрія програмного забезпечення відзначає значний прогрес в розвитку веб-додатків як окремого сегмента цього напрямку.

Звичайні додатки не розвиваються так швидко, як веб-технології. Багато компаній переходять з традиційних додатків на веб-додатки, оскільки бачать в них майбутнє і готові використовувати їх замість звичайних додатків. Це призводить до значних інвестицій, що стимулює розвиток цих технологій. У зв'язку з цим хочу наголосити на їх основних перевагах: [2].

- Встановлення веб-додатків є більш доступним і простим. Використання web-додатків дозволяє підприємствам знизити витрати на ІТ-відділи, які відповідають за встановлення та підтримку програмного забезпечення. У такому випадку користувачам потрібен лише комп'ютер з веб-браузером та доступом до Інтернету або корпоративної мережі.

- Оновлення web-додатків є більш ефективними і простими. Вартість обслуговування програмного забезпечення завжди має велике значення. Оновлення веб-додатків схоже на їх встановлення, тому переваги, які були згадані раніше,

застосовуються і в цьому випадку. Для оновлення веб-додатку потрібно змінити лише версію на сервері, і всі користувачі відразу отримають доступ до нової версії.

- Web-застосунки є більш універсальними та зручними для кінцевого користувача. Достатньо встановити веб-додаток на сервер, який підтримує будь-яку сучасну операційну систему, і ви зможете користуватися ним через Інтернет на будь-якому комп'ютері або мобільному пристрої, що працює під управлінням Mac OS, Windows, Linux або будь-якої іншої.

Web- застосунки, якщо вони розроблені якісно, працюватимуть однаково добре у будь-якому веб-браузері, такому як Mozilla Firefox, Opera, Google Chrome, Internet Explorer або Safari.

Web -додатки спрощують організацію зберігання даних. Якщо потрібно мати доступ до одних і тих самих даних з різних місць, значно легше організувати їх зберігання в одному місці, замість розкидання їх по різних базах даних. Це усуває потребу у синхронізації і підвищує рівень безпеки даних.

Такі причини, на мою думку, призвели до великої популярності веб-додатків і сприяли їхньому значному розвитку. Тому ми все частіше користуємося Інтернетом.

1.3. Аналіз аналогічних застосунків

З метою аналізу аналогів можна виділити наступні види які мені вдалося знайти в інтернеті:

- безліч груп в месенджерах або соціальних мережах наприклад: Facebook, Instagram, Viber, Telegram, LinkedIn, тощо;
- веб web-застосунки Google Play: Волонтер[3]; YesHelp[4].
- сайти з пошуку волонтерської допомоги.

Перед початком розгляду переваг і недоліків представлених вище аналогів, варто відзначити що всі групи створені в соціальних мережах і месенджерах володіють такими загальними недоліками:

- залежність безпосередньо від розробників соціальної мережі і можливими проблемами при додаванні оголошень про допомогу в такі групи;
- незручністю пошуку будь-якої волонтерської допомоги;
- відносно невелика кількість учасників у кожній окремій групі;
- відсутність автоматизації процесу подачі заявки на допомогу.

При обговоренні переваг та недоліків груп у Viber можна зазначити, що цей мобільний месенджер має широке поширення в Україні.

Однак, серед недоліків можна виділити наступне:

- незареєстровані або ті, хто не використовує Viber постійно, не можуть отримати доступ до цих груп.
- відсутня можливість отримати швидко відповідь;
- додавання та подача будь-якої волонтерської допомоги в такі групи є не зручним, оскільки потрібно залежати від адміністраторів групи для подачі та видалення заявок про допомогу;
- ефективність таких груп є низькою.

Раніше було також багато груп у соціальній мережі Vkontakte. Проте, через блокування даної соціальної мережі на території України, ці групи втратили актуальність.

Розглянемо веб-додаток для українських біженців «Волонтер» [31], це також повноцінний зручний веб-додаток.

Переваги веб-додатку:

- не вимагає жодних фінансових вкладень;
- є можливість зв'язку між користувачами за допомогою номера телефону;

- доступна можливість публікації заявки про потребу у волонтерах або про волонтерські пропозиції, а також можливість розміщення оголошень про пошук або знаходження волонтерів.

Недоліки:

- на веб-додатку може міститись значна кількість реклами, що у свою чергу заважає пошуку необхідних оголошень та відштовхує користувачів;
- немає можливості надіслати добровольцеві оголошення безпосередньо, відсутня система чатів для швидкого зв'язку між учасниками;
- має застарілий дизайн.

Мою увагу, заслуговую спеціальний додаток YesHelp [4] для українських біженців, який розробила канадська компанія OneIQ Corp. Також, компанія співпрацює з такими відомими брендами як Lenovo, Nutanix та VMware.

Переваги веб-додатку:

- користувачі можуть знайти легко собі волонтера, житло та розібратися в багатьох інших важливих речах.

За результатами аналізу було встановлено, що веб-додаток YesHelp є найкращим серед порівнюваних аналогів. Він має новітній дизайн, адаптивний інтерфейс та доступний для завантаження на платформах IOS та Android в понад 60 країнах. І в першу чергу його можуть завантажити біженці, які знаходяться на території Європейського Союзу.

1.4 Висновки до першого розділу. Постановка задач дослідження

У першому розділі докладно проаналізовано предметну область, а саме поняття «Web-додаток», а також проведено детальний аналіз та порівняння переваг використання web-застосунків. Аналіз останнього дав чітке рішення стосовно проектування web-застосунків, які спрощують організацію зберігання даних.

Також було виконано аналіз аналогічних застосунків, розглянуто принципи роботи аналогів у месенджерах або соціальних мережах. Віддільно було розглянуто кожний аналог: facebook, vk, instagram, viber, telegram; віділені переваги і недоліки.

Детально розглянуто веб-додаток для українських біженців «Волонтер» це повноцінний зручний web -додаток.

Заслуговує уваги новий сучасний web-додаток YesHelp, який розробила канадська компанія OneIQ Corp. Він має новітній дизайн, адаптивний інтерфейс та доступний для завантаження на платформах IOS та Android в понад 60 країнах.

Враховуючи всі недоліки, що були перелічені вище, виникла ідея створення web-застосунку «UAHelper» [5] для забезпечення волонтерської допомоги переселенцям, створити зручну систему подачі заявок, сформувати базу даних внутрішньо переміщених осіб.

Програмне забезпечення, яке розроблено в кваліфікаційній роботі повинно відповідати наступним вимогам:

- мати зручний та зрозумілий інтерфейс для користувачів;
- мати ефективну систему фільтрації заявок;
- мати зручну та зрозумілу для користувачів систему додавання, редагування та видалення заявок;
- забезпечувати швидку обробку запитів до сервера;
- мати багатофункціональну адміністративну панель для управління веб-додатком.
- мати різні рівні доступів при авторизації та аунтефікації.
- мати адаптивність під будь-які розміри екранів та пристроїв.
- мати власний домен з посиланням на веб-застосунок.
- має бути розроблений на сучасних мовах програмування, використовувати швидкі та потужні бази даних та методи для реалізації.

РОЗДІЛ 2. ВИЗНАЧЕННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ПРОГРАМНОГО ПРОДУКТУ ТА ЇХ ОБГРУНТУВАННЯ

2.1. Середовище розробки PhpStorm

У цій даній роботі розглянутий редактор вихідного коду PhpStorm [6] - це спеціалізований інструмент для веб-розробки, який фокусується на web-додатках та різних програмах, створених з допомогою PHP, HTML, JavaScript і CSS. Він безпосередньо синхронізується з розгортанням та керує проектами за допомогою протоколу FTP.

Основні функції PhpStorm включають автоматичне завершення коду PHP, перевірку коду, різні методи рефакторінгу та швидку навігацію по коду. PhpStorm також має велику кількість плагінів для співпраці з провідними бекендами та фронтенд-фреймворками, а також забезпечує можливість статичного аналізу коду.

Головний плюс в програмному середовищі PhpStorm - це його висока продуктивність та розширені можливості, спеціально розроблені для роботи з мовою програмування PHP. Ось кілька основних переваг PhpStorm:

- Підтримка PHP: PhpStorm надає потужні інструменти для розробки PHP-проектів. Він має розуміння синтаксису PHP, автодоповнення коду, перевірку помилок та підказки, що допомагають розробникам писати чистий та ефективний код.
- Розширені можливості редагування: PhpStorm має багато корисних функцій редагування коду, таких як переформатування коду, швидке переходи до визначень функцій та класів, пошук та заміна, вбудований термінал та багато іншого. Всі ці функції роблять редагування коду швидшим та зручнішим.

- Вбудована система контролю версій: PHPStorm має інтегровану підтримку популярних систем контролю версій, таких як Git, Subversion і Mercurial. Це дозволяє розробникам легко взаємодіяти зі своїми репозиторіями, виконувати коміти, відгалуження, злиття та інші операції безпосередньо з інтерфейсу PHPStorm. На малюнку нижче (рис. 2.1.) зображено інтерфейс PHPStorm.

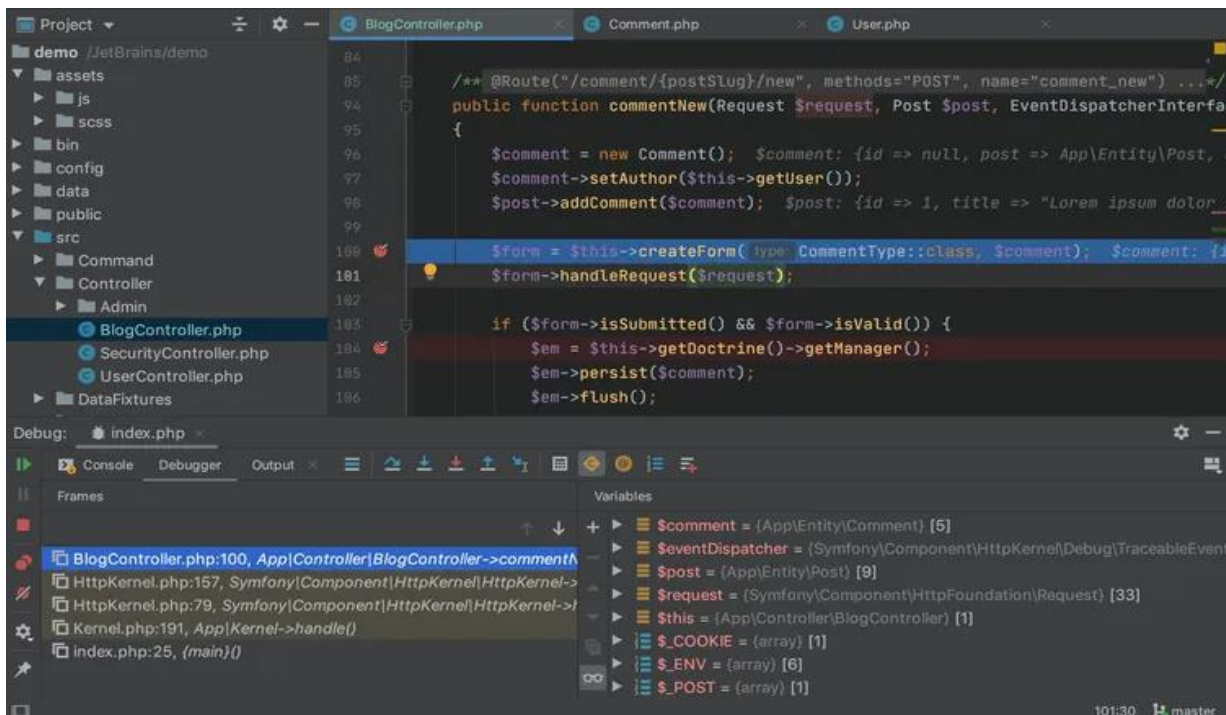


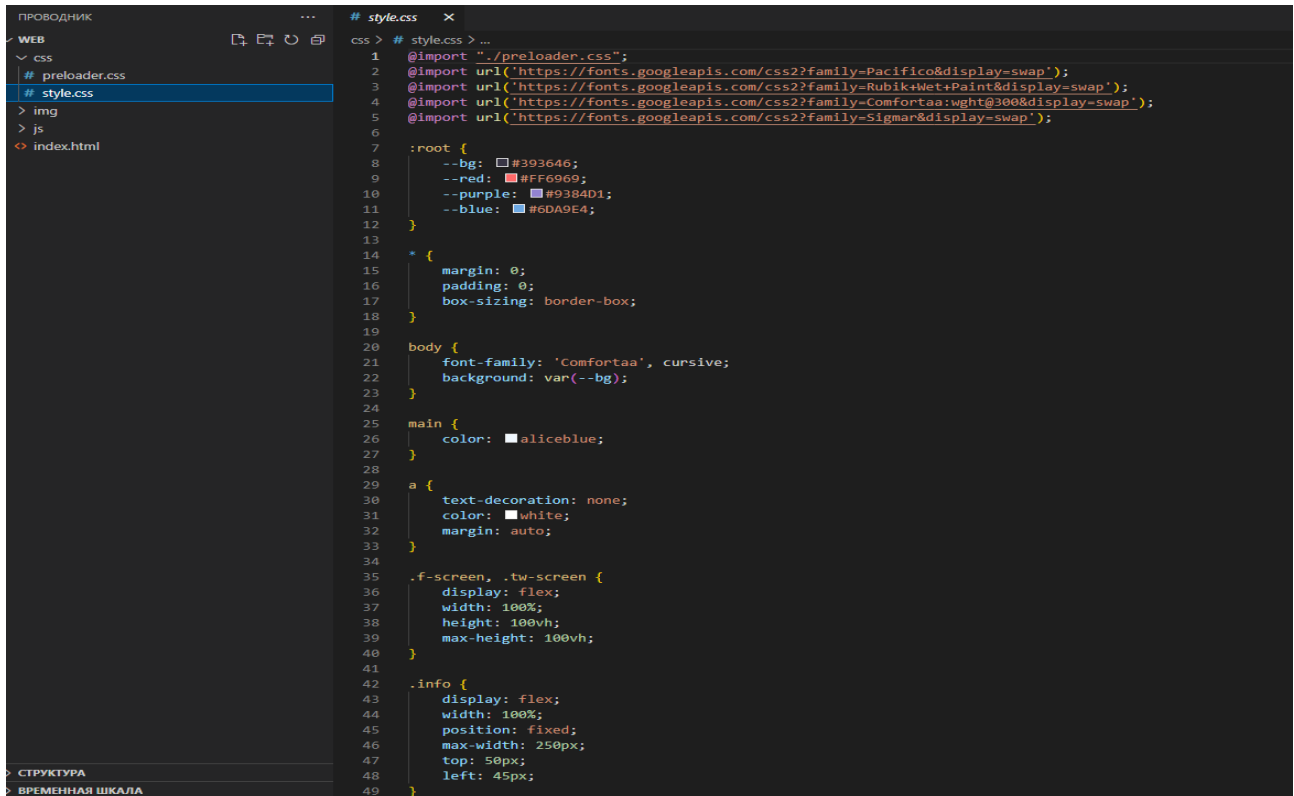
Рис. 2.1 — Інтерфейс PHPStorm

2.2. Середовище розробки Visual Studio Code

Visual Studio Code [7] - це редактор коду, спеціально розроблений для кроссплатформенної розробки web і хмарних додатків, який має репутацію "легкого" редактора. Він дозволяє розробляти як консольні, так і графічні додатки, включаючи ті, що використовують технологію Windows Forms, а також web- сайти, web -додатки та web -служби, як в рідному, так і керованому кодах, на всіх платформах. Редактор має вбудований відладчик, інструменти для роботи з Git, а

також засоби рефакторинга, навігації по коду, автодоповнення типових конструкцій та контекстні підказки.

На рисунку 2.2 зображено інтерфейс Visual Studio Code



```

1 @import url("../preloader.css");
2 @import url('https://fonts.googleapis.com/css2?family=Pacifico&display=swap');
3 @import url('https://fonts.googleapis.com/css2?family=Rubik+K&Paint&display=swap');
4 @import url('https://fonts.googleapis.com/css2?family=Comfontaa:wght@300&display=swap');
5 @import url('https://fonts.googleapis.com/css2?family=Sigmar&display=swap');
6
7 :root {
8   --bg: #393646;
9   --red: #FF6969;
10  --purple: #9384D1;
11  --blue: #6DA9E4;
12 }
13
14 * {
15   margin: 0;
16   padding: 0;
17   box-sizing: border-box;
18 }
19
20 body {
21   font-family: 'Comfontaa', cursive;
22   background: var(--bg);
23 }
24
25 main {
26   color: #aliceblue;
27 }
28
29 a {
30   text-decoration: none;
31   color: #white;
32   margin: auto;
33 }
34
35 .f-screen, .tw-screen {
36   display: flex;
37   width: 100%;
38   height: 100vh;
39   max-height: 100vh;
40 }
41
42 .info {
43   display: flex;
44   width: 100%;
45   position: fixed;
46   max-width: 250px;
47   top: 50px;
48   left: 45px;
49 }

```

Рисунок 2.2. - зразок інтерфейсу сайту середовища розробки Visual Studio Code

Цей продукт забезпечує розробку для платформ ASP.NET і Node.js, та є легковажним рішенням, яке дозволяє уникнути використання повного інтегрованого середовища розробки. Він має великий плюс у підтримці багатьох мов програмування, таких як C++, C#, Python, PHP, JavaScript та інших.

Сервіс має наступні можливості:

- вбудовані інструменти для інтеграції з GitHub, GIT та Visual Studio Team Services для швидкого тестування, складання, упаковки та розгортання різних типів додатків;
- зручний інтерфейс роботи з Unity-проектами;

- можливість роботи з Mono та Node.js за допомогою вбудованого відладчика; підтримка TypeScript і JavaScript;
- можливість публікації створених додатків в Microsoft Azure через сервіс Visual Studio Team Services; можливість написання коду для конкретного завдання з подальшою інтеграцією в проект з надбудовою або безпосередньо;
- наявність великої бібліотеки шаблонів, готових фрагментів коду і сніпетів з можливістю додавання своїх елементів;
- можливість одночасної роботи з декількома проектами в декількох вікнах;
- інтерфейс можна розділити на дві панелі для порівняння коду;
- функція налагодження.

Переваги цього сервісу:

- можливість налаштування багатьох параметрів (так і інтерфейсу);
- наявність розширеної бібліотеки додатків та готових рішень;
- багатофункціональність (редактор підтримує майже всі мови, що використовуються для створення додатків);
- простота використання та гнучкість.

2.3. Особливості платформи ASP.NET WEB API

Веб-застосунок був розроблений з використанням платформи .NET Core та фреймворку ASP.NET Core, що підтримується компанією Microsoft та мови програмування C#. .NET Core є модульною платформою з відкритим кодом, спрямованою на розробку програмного забезпечення. [9].

Фреймворк ASP.NET Core, у свою чергу, є розширенням платформи .NET Core та призначений для створення сучасних веб-застосунків різного розміру та складності - від невеликих блогів до великих інтернет-магазинів. [10].

Один з головних плюсів використання ASP.NET Core полягає в тому, що він може працювати на різних операційних системах, включаючи Windows, Mac і Linux. Фреймворк був створений з метою забезпечення оптимізованого середовища розробки для застосунків, які можуть бути розгорнуті в хмарі або запущені локально.

Основні переваги ASP.NET Core включають:

- Відкритий вихідний код;
- Підтримка національних алфавітів завдяки використанню Unicode;
- Висока продуктивність завдяки відкомпільованому коду;
- Можливість запуску веб-застосунків на різних пристроях, таких як смартфони, планшети, ноутбуки та ПК;
- Кросплатформенність - можливість роботи веб-застосунку на різних операційних системах, таких як Windows, Linux та macOS;
- Інтеграція з сервісами Microsoft;
- Функціональність Razor Pages;
- Можливість використання типових шаблонів авторизації та аутентифікації користувача;
- Підтримка різних сервісів, які дозволяють реалізувати типову логіку клієнт-серверної архітектури з використанням паттерну MVC (Model-View-Controller).

Web API представляє спосіб побудови додатка ASP.NET, який спеціально заточений для роботи в стилі REST (Representation State Transfer або "передача стану подання"). REST-архітектура передбачає застосування таких методів або типів запитів HTTP для взаємодії із сервером [11]:

- GET
- POST
- PUT

- DELETE

Найчастіше REST-стиль особливо зручний при створенні всіякого роду Single Page Application, які нерідко використовують спеціальні javascript-фреймворки типу Angular, React або Vue.js. По суті Web API являє собою веб-службу, до якої можуть звертатися інші додатки. Причому ці додатки можуть представляти будь-яку технологію і платформу - це можуть бути веб-додатки, мобільні або десктопні клієнти.

На малюнку нижче (рис. 2.4) зображено шаблонів API

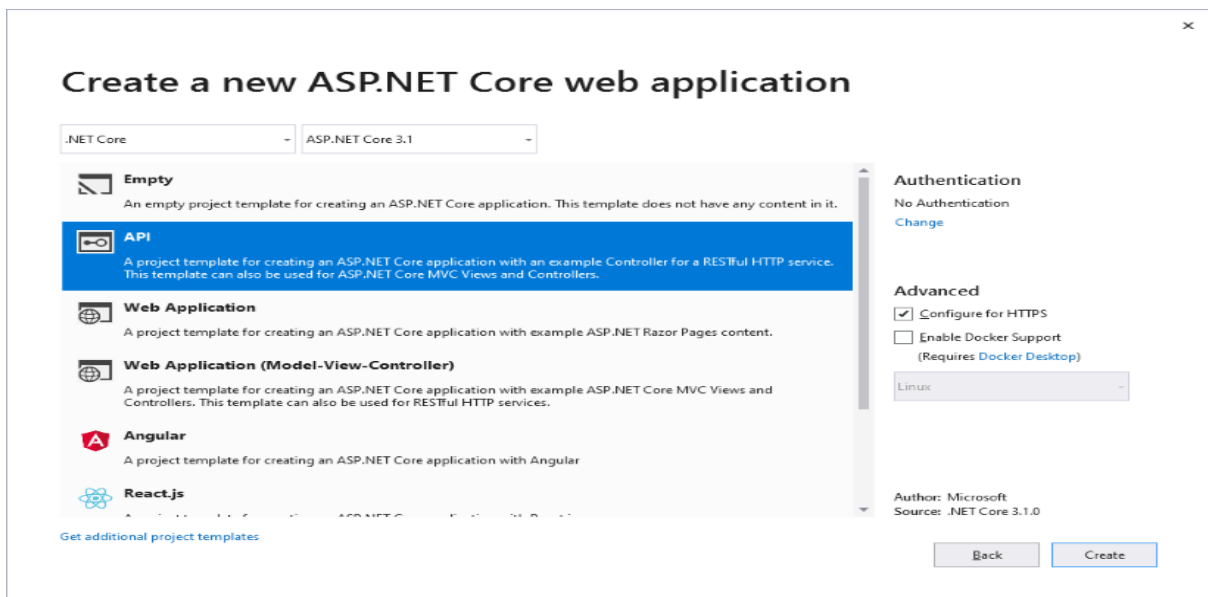


Рис. 2.4. - Проєкт ASP.NET Core серед шаблонів API

Web API - це інтерфейс прикладного програмування (API), який використовується для забезпечення зв'язку або взаємодії програмних компонентів один з одним.

Web API - це розширена форма веб-додатку для надання послуг на різних пристроях, таких як ноутбук, мобільний телефон та інші.

Компонент Web API додатку діє як інтерфейс на основі HTTP, який приймає HTTP-запити від сторонніх клієнтських додатків і виконує операції з даними.

Наступна діаграма пояснює можливу архітектуру цього гіпотетичного додатку з використанням Web API [12]:

На малюнку нижче (рис. 2.5) зображено архітектуру Web API

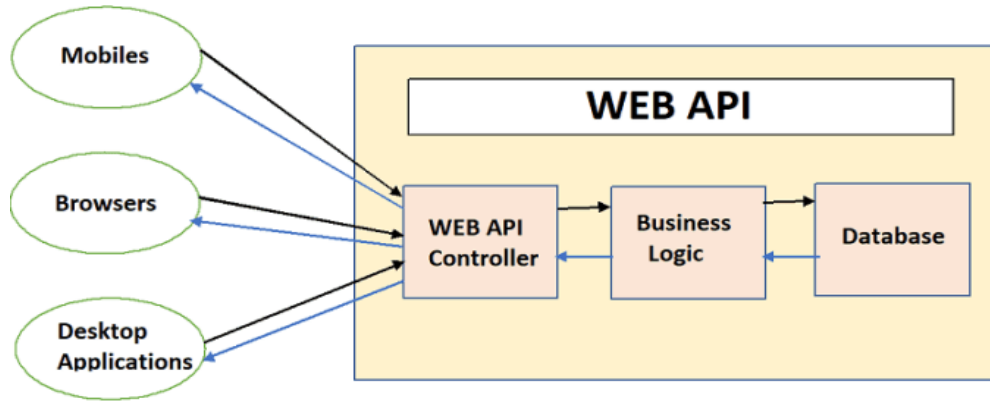


Рис. 2.5. - Архітектура додатку з використанням WEB API.

Для створення веб-застосунків з використанням технології ASP.NET Core найчастіше використовують мову програмування C#, оскільки вона має такі переваги:

- Підтримка об'єктно-орієнтованого підходу: C# дозволяє використовувати принципи об'єктно-орієнтованого програмування, що сприяє структурованості та модульності коду.
- Мова компільованого типу: C# використовується для компіляції коду в машинний код, що забезпечує високу швидкодію та оптимізацію програм.
- Наявність великої кількості бібліотек та шаблонів, написаних мовою C#: Існує багато готових бібліотек та шаблонів, що спрощують розробку веб-застосунків із використанням ASP.NET Core.
- Наявність зрозумілої документації: Мова C# та платформа .NET мають детальну та доступну документацію, що допомагає розробникам зрозуміти та використовувати їх ефективно.

- Сувора типізація: C# використовує статичну типізацію, що дозволяє виявляти помилки на етапі компіляції і покращує стабільність та безпеку програм.

В якості середовища програмування для розробки веб-застосунків на платформі ASP.NET Core використовується Visual Studio 2019 від компанії Microsoft. Це функціональне інтегроване середовище розробки (IDE), яке надає зручну навігацію по коду, пошук помилок, підказки щодо виправлення та інші корисні функції для зручної розробки на платформі .NET.

2.4. Графічний редактор Figma

Фігма - відносно нова безкоштовна програма для веб дизайнера, який знає ціну своєму часу і йде в ногу з тим самим часом.

Figma є відносно новим сервісом для створення дизайн-рішень, який в основному використовується для розробки інтерфейсів мобільних, десктопних та веб-додатків. Одна з основних відмінностей Figma. [13] від програм, таких як Photoshop і Paint, полягає в тому, що цей графічний редактор є мультиплатформним і працює у веб-середовищі, не прив'язуючись до конкретного комп'ютера. Всі дані зберігаються в хмарі, що дозволяє всій команді одразу мати доступ до перегляду та роботи з проектом.

Figma має добре продуманий функціонал, який, хоч і спрямований на конкретні завдання дизайнерів, повністю задовольняє їх потреби. Крім того, програма включає інші інструменти, не пов'язані безпосередньо з графікою, наприклад, формування характеристик об'єктів для подальшої розробки та симуляція взаємодії елементів на веб-сайті.

Хоча ми розглядаємо Figma як сервіс для роботи з растровою графікою, в ньому також є можливість створювати векторні зображення та ілюстрації.

До інших переваг Figma входять:

- Зручний інтерфейс;

- Багатопарова структура роботи, подібна до Photoshop;
- Крім основної веб-версії, доступна десктопна версія для MacOS та Windows;
- Можливість без дискомфорту працювати з безкоштовним тарифом.
- Просто робите роботу, як дизайнер, закриваєте браузер, а все вже і так збережено, і нічого не пропаде ніколи. Це значно прискорює роботу, особливо для великих команд.

На малюнку нижче (рис. 2.6) зображено інтерфейс графічного додатка Figma

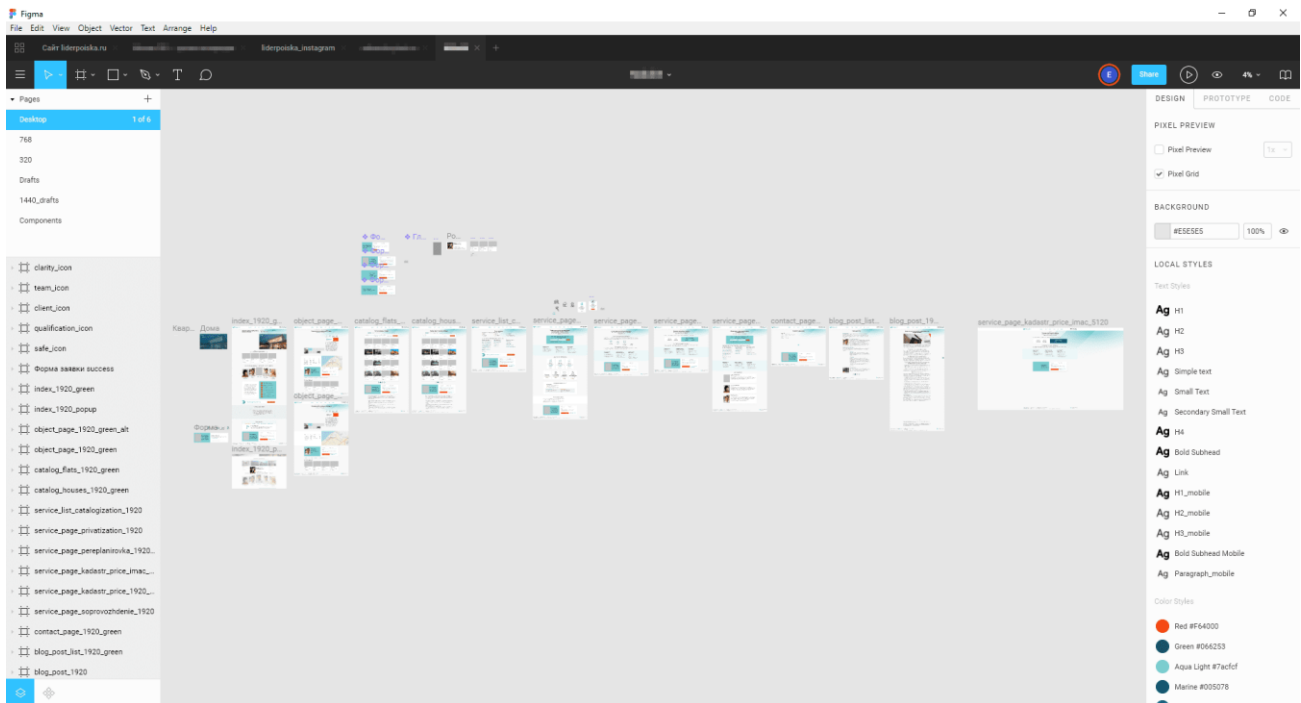


Рис. 2.6 - Інтерфейс графічного додатка Figma

Однак, Figma має деякі недоліки:

- Відсутність підтримки СМΥК (хоча програма не призначена для поліграфії);
- Обмежена кількість інструментів для обробки зображень.
- Платну підписку для відкриття всіх програмних можливостей.

2.5. Мова програмування та розмітки гіпертексту HTML

HTML є мовою розмітки гіпертексту, яка використовується для створення веб-сторінок та додатків, доступних через Інтернет. Розроблена на початку 1990-х, HTML дозволяє розробникам веб-сайтів описувати елементи, такі як зображення, текст, форми та інтерактивні функції, щоб інтернет-браузери могли їх відображати. Для створення веб-сайтів та додатків, які можуть бути доступні через Інтернет-браузерні програми, розробники часто використовують HTML разом з каскадними таблицями стилів (CSS) та JavaScript. Історично, стандарти для HTML, CSS та JavaScript були розроблені Всесвітнім консорціумом із веб-сторінок (W3C). Тім Бернерс-Лі був творцем HTML та створив його прототип на початку 1980-х, а до кінця десятиліття розробив мову, Інтернет-браузер та серверне програмне забезпечення. Згодом, у 1995 році, була випущена друга версія HTML, а пізніше створено кілька додаткових оновлень. В даний час, Всесвітній консорціум із веб-сторінок продовжує розробку стандартів для HTML та інших веб-технологій. [14]

HTML розроблено на основі правил SGML - стандартної узагальненої мови розмітки, а XHTML - на основі правил XML, суворішої підмножини правил SGML. Тому, для обробки XHTML документів можна використовувати стандартні інструменти обробки XML документів, що робить їх обробку швидшою та простішою порівняно з HTML. XHTML можна розглядати як перетин HTML та XML, що робить його більш гнучким і потужним стандартом. Рекомендації W3C для XHTML були введені в 2000 та 2001 роках для версій 1.0 та 1.1 відповідно.

HTML є текстовим форматом представлення веб-документів, який інтерпретується браузером та відображається у вигляді документа, зручного для людини. Цей формат вибраний через його простоту, можливість безпосередньої інтерпретації в будь-якій операційній системі, мінімальний розмір файлу, зручність редагування і інтерпретації. HTML є додатком до стандарту SGML та відповідає міжнародному стандарту ISO 8879.

Мова розмітки гіпертекстових документів HTML дозволяє використовувати різні типи елементів, які забезпечують функціональність документа, такі як текстові фрагменти з заданими параметрами форматування, списки, таблиці, зображення, гіперпосилання та інші. Для оголошення цих елементів використовуються команди розмітки, які називаються тегами. Усі 25 HTML-тегів, які зустрічаються в тексті документа, інтерпретуються браузером при відображенні документа. При створенні веб-сторінки автор використовує декілька ключових компонентів та атрибутів, які зможе прочитати будь-який Інтернет-браузер, що отримує доступ до веб-сайту. Багато з цих атрибутів будуть в поєднанні зі "початковим тегом" та "кінцевим тегом". Наприклад, автор може використовувати тег "p" з набором дужок для позначення початку абзацу та тег "/p" у другому наборі дужок для позначення кінця абзацу. Перший тег відкриває абзац, а другий тег закриває його.

HTML 5 є поточним стандартом мови розмітки, який використовується для створення більшості веб-сторінок в Інтернеті. Цей стандарт дозволяє підтримувати мультимедіа, такі як відео та аудіо. HTML 5 є головним будівельним блоком веб-розробки, оскільки він забезпечує статичний вигляд веб-сторінок. Всі інші стилі та функції можуть бути додані за допомогою CSS, PHP, Javascript. Тому, перед вивченням інших мов програмування, варто знати HTML. На сьогоднішній день, HTML 5 є основою для багатьох веб-сайтів та сервісів створення веб-сайтів, таких як WordPress, Wix, Weebly.

На малюнку нижче (рис. 2.7) зображено структура HTML і контейнери

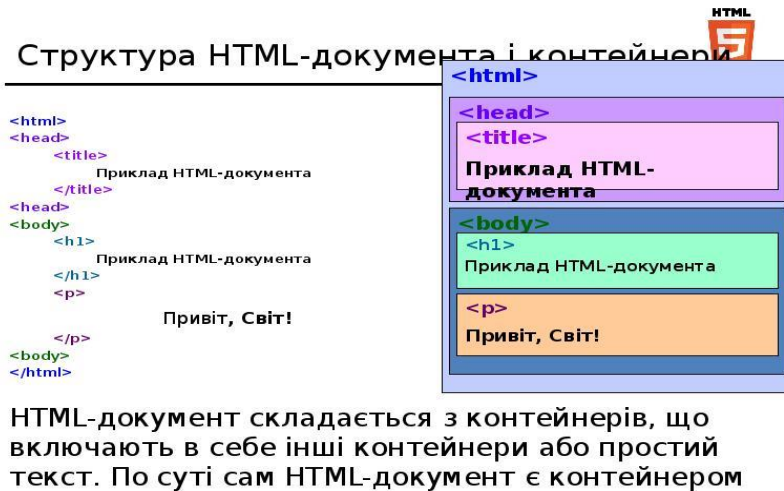


Рис. 2.7 - Структура HTML і контейнери

HTML-елемент відрізняється від іншого тексту в документі завдяки "тегам", які складаються з імені елемента, оточеного "<" і ">". Назва елемента всередині тегу може бути написана в будь-якому регістрі, тому <title> може бути записано як <Title>, <TITLE>, тощо.

Програмісти веб-сайтів можуть використовувати атрибути та елементи HTML, щоб визначити практично будь-який аспект сторінки та її відображення користувачам. Вони також можуть використовувати CSS та JavaScript, а також інші мови, такі як PHP, jQuery та XML, для створення веб-сайту. [15]

Багато новачків у програмуванні починають з вивчення HTML як своєї першої мови, а потім продовжують навчання більш складними мовами, такими як JavaScript та CSS.

Вивчення основ HTML може бути корисним не тільки для майбутніх програмістів, але й для тих, хто хоче зрозуміти, як створюються веб-сторінки, які вони щодня переглядають на комп'ютері або смартфоні.

2.6. Мова програмування C#

При створенні web- застосунку було обрано мову програмування C# для розробки веб-застосунку, проте цього недостатньо для створення повноцінного проекту. Для цього необхідно вибрати платформу та фреймворк для розробки проекту. Сучаснішою платформою для розробки мовою C# є .Net Core, яка є модульною платформою з відкритим вихідним кодом і є сумісною з операційними системами Windows, Linux і macOS. [16].

Натомість, для створення веб-додатків було вибрано ASP.Net Core - вільно-розповсюджуваний крос-платформний фреймворк з відкритим вихідним кодом, розроблений Microsoft спільно зі спільнотою, який має високу продуктивність порівняно з ASP.NET. ASP.NET Core може працювати поверх крос-платформного середовища .NET Core, що може бути розгорнуто на операційних системах Windows, Mac OS та Linux. За допомогою ASP.NET Core можна створювати крос-платформні додатки і запускати їх не лише на ОС Windows, але і на Linux і Mac OS. Для розгортання веб-додатків можна використовувати традиційний IIS або крос-платформний веб-сервер Kestrel.

Завдяки модульності фреймворка, компоненти веб-додатків можуть завантажуватися як окремі модулі через пакетний менеджер NuGet. .NET Core має декілька визначальних особливостей, зокрема гнучке розгортання, яке дозволяє встановлювати платформу окремо від застосунку. Вбудоване введення залежностей та модульний конвеєр HTTP-запитів також є важливими компонентами .NET Core для сучасної веб-розробки.

Додатково, .NET Core має крос-платформну модульну інфраструктуру, доступну через NuGet, яка дозволяє розробникам створювати легкі додатки з безпечнішим та продуктивнішим пакетним функціоналом. Крім того, є доступ до командного рядка (CLI) на будь-якій підтримуваний платформі, що дозволяє виконувати певні команди для запуску та розгортання проектів.

Аналізуючи існуючі технології для веб-застосунків, було встановлено, що оптимальними для реалізації поставленої задачі є веб-сервер IIS, сервер застосунків C# на платформі .Net Core, сервер БД MS SQL та Microsoft Sql Management Studio для адміністрування баз даних.

2.7. Каскадні таблиці стилей CSS

CSS, або "Каскадні таблиці стилів", є способом описування різних елементів HTML та розширення їх властивостей. Він вбудований в HTML-код і не вимагає використання спеціальних редакторів чи компіляторів, оскільки інтерпретується звичайним браузером. Цей механізм став частиною розробки специфікації HTML 3.0, але його підтримка з'явилася тільки в 1997 році, коли з'явилися браузери, такі як Netscape Navigator 4.0 та Internet Explorer 4.0. Сьогодні CSS використовується творцями веб-сторінок для встановлення кольорів, шрифтів, макетів та інших аспектів презентації документів.

CSS є мовою каскадних таблиць стилів (англ. cascading style sheets), яка використовується для опису стилів, що можуть бути застосовані до багатьох елементів розмітки документа. Ця мова була розроблена компанією Hakon Wium Lie в 1994 році, а специфікація була створена W3C у грудні 1996 року. CSS дозволяє веб-розробникам змінювати зовнішній вигляд веб-сторінок, такий як шрифти, кольори, розміри та розташування елементів. Один CSS-файл може бути застосований до декількох сторінок, що дозволяє змінювати зовнішній вигляд всіх сторінок одночасно. Використання CSS не обов'язково вимагає створення нових селекторів класу або id. Наприклад, CSS можна використовувати для зміни стилю гіперпосилань, включаючи їх колір, коли курсор миші наводиться на них. Наведений нижче приклад демонструє використання CSS для зміни стилю двох тегів HTML без створення нових селекторів класу або id.

Основною метою CSS було розділення вмісту, написаного мовою розмітки (наприклад, HTML), та його презентації, написаної за допомогою CSS. Це дозволяє збільшити доступність документа, надає більше гнучкості та контролю над його виглядом, та зменшує складність та повторюваність структурованого вмісту. Крім того, використання CSS дає можливість відображати один і той же документ у різних стилях та форматах, таких як екран, друк, читання голосом з використанням голосового браузера або програми зчитування, та виведення на пристрої з використанням шрифту Брайля. [17].

Стандарт CSS встановлює порядок застосування правил стилю та пріоритет випадків, коли кілька правил мають властивості, що підходять для одного й того ж елемента (або коли в одному правилі маються властивості з однаковими назвами). Цей порядок називається "каскадом" і включає в себе призначення ваг властивостям, щоб забезпечити передбачувані результати.

Основні переваги використання CSS:

- Економія часу - CSS дозволяє визначити стиль для кожного HTML елемента та повторно використовувати цей стиль на багатьох веб-сторінках.
- Швидше завантаження сторінок - застосовуючи CSS, не потрібно кожен раз записувати атрибути для HTML тегів, що зменшує код та прискорює завантаження сторінки.
- Простота обслуговування - одна зміна у CSS може оновити стиль на всіх сторінках, що полегшує роботу розробників.
- Покращення стилів для HTML - CSS має більший набір атрибутів, що дозволяє давати кращий вигляд веб-сторінкам порівняно з атрибутами HTML.
- Сумісність з різними пристроями - CSS дозволяє оптимізувати вміст для різних типів пристроїв, таких як КПК та мобільні телефони, що дозволяє представляти веб-сторінки на різних пристроях.

2.8. Фреймворк Angular

Angular є альтернативою AngularJS, яка базується на Typescript для розробки динамічних веб-додатків. AngularJS, який був створений у 2009 році і написаний на Javascript, може використовуватися як термін для всіх версій Angular v1.x. Оновлена версія Angular має кращий модульний дизайн та є простішою у розробці, а також є швидшою в порівнянні з AngularJS. Одна з ключових відмінностей між ними полягає у мові, на якій вони базуються: Angular використовує Typescript (супернабір ES6), тоді як AngularJS базується на Javascript. Це призводить до різниці у складових фреймворків.

Одна з головних відмінностей між Angular та AngularJS полягає в підходах до використання компонентів та директив. Angular використовує ієрархічну структуру компонентів для організації роботи, тоді як AngularJS має набір директив, які дозволяють повторно використовувати код та писати окремі модулі. У Angular також є стандартні директиви, але реалізація їх функцій відрізняється від AngularJS. [18].

Фреймворк Angular базується на використанні компонентів, які включають в себе структурні та атрибутивні директиви для модифікації макета та поведінки DOM. Структурні директиви змінюють елементи, щоб досягнути своєї мети, тоді як директиви атрибутів змінюють зовнішній вигляд елементів. У підході Angular функціональні та логічні компоненти відокремлені для кращої служби додатку. [19].

У свою чергу, AngularJS використовує фреймворк модель-представлення-контролер (MVC), де контролер відіграє центральну роль у керуванні даними, правилами та обробці вхідних даних. Модель забезпечує збереження та керування отриманими даними, а View відображає дані, які містяться в моделі. MVC простий, цілісний та допомагає прискорити розробку, а зв'язування даних дозволяє зменшити кількість необхідного коду.

- Angular використовує ієрархічну систему ін'єкції залежностей для підвищення продуктивності. На відміну від AngularJS, який використовує директиви замість DI.

- У відмінності від AngularJS, який використовує `@routeProvider.when` для визначення маршрутів, Angular використовує URL-адреси, щоб імітувати директиви, і конфігурація `@Route` використовується для визначення маршрутів. Це дозволяє Angular працювати ефективніше, що робить його перевагою над AngularJS. [20].

- Angular має кращу структуру, що допомагає підвищити швидкість і продуктивність операцій, ніж у AngularJS. Незважаючи на те, що AngularJS має функцію двостороннього зв'язування, що зменшує час та зусилля, Angular здатний працювати швидше і ефективніше завдяки своїй оптимізованій структурі.

AngularJS використовує сторонні інструменти, такі як IDE та Webstorm, для створення програм та виявлення помилок. З іншого боку, Angular надає інтерфейс командного рядка (CLI) для створення проектів, що забезпечує швидкість та доступність під час тестування.

Керування проектами Angular більш структуроване, що робить його більш придатним для розробки великих програм, в порівнянні з AngularJS.

2.9. Хмарні бази даних Microsoft Azure

Microsoft Azure - це потужна платформа, яка використовується 95% компаній зі списку Fortune 500. Завдяки понад двом сотням продуктів, які включає Azure, існує рішення для більшості поширених бізнес-проблем. Вона надає широкий спектр послуг, починаючи від віртуальних машин і закінчуючи послугами квантових обчислень наступного покоління.

Azure Active Directory або AAD - це сервіс ідентифікації, який забезпечує можливість одноразового входу (SSO), багатофакторну аутентифікацію (MFA) та умовний доступ. AAD є частиною найновішого роду продуктів Microsoft Entra, який також включає управління дозволами та перевірену ідентифікацію[21].

AAD дозволяє підвищити рівень безпеки завдяки функції MFA, яка може знизити ризик компрометації облікового запису на 99,9%, а також впроваджувати політики з урахуванням ризику завдяки умовному доступу.

Один з найбільш цінних аспектів AAD - це можливість підвищення продуктивності завдяки можливості одноразового входу, який може інтегруватися зі сотнями популярних програм SaaS.

Безкоштовна версія AAD, яка входить до кожної підписки на комерційні послуги Azure, може додати значну цінність для бізнесу будь-якого розміру та будь-якої галузі.

Azure Logic Apps - це платформа, яка допомагає компаніям створювати автоматизовані робочі процеси з мінімальним кодом або без нього. Це спрощує з'єднання між старими та новими системами за допомогою хмари. Одним з ключових факторів, які роблять Azure Logic Apps потужними, є їх зв'язок з іншими хмарними службами, такими як Google Cloud, Microsoft 365, Twitter та Hubspot. Для компаній, які хочуть інтегрувати дані, програми та пристрої, Azure Logic Apps є дуже корисним. Якщо є достатній досвід, цей інструмент можна поєднувати з Azure для підвищення гнучкості.

У 2022 році Postman повідомив, що 51% компаній витрачають більше половини свого часу на розробку API. Цей час може пришвидшити розвиток, дозволяючи швидше створювати додатки. Microsoft намагається спростити процес за допомогою Azure API Management, яка допомагає оптимізувати роботу в гібридних та багатохмарних середовищах за допомогою єдиного місця керування всіма API.

Багато компаній використовують Azure API Management через покращену безпеку, оскільки всі API зберігаються за єдиною статичною IP-адресою з ключами, маркерами та IT-фільтрацією для підвищення захисту. Azure API Management може навіть перетворити існуючі застарілі веб-сервіси на API на основі REST через фасади ваших серверних служб.

Однією з ключових тенденцій у технологіях останніх 5 років є широке впровадження функцій штучного інтелекту (ШІ) в програмах. Azure Cognitive Services надає такі можливості ШІ розробникам у вигляді готових функцій, які можна використовувати за допомогою єдиного виклику API програми. Ці функції дозволяють програмам бачити, чути, говорити, шукати, розуміти та приймати рішення за допомогою штучного інтелекту.

Розробники цінують Azure Cognitive Services, тому що використовувані моделі попередньо навчені з високоякісними дослідженнями від Microsoft AI. Це також дешевше, ніж створення власної моделі з нуля або використання аналогічних послуг.

Microsoft Azure Active Directory (AD) [22] є ефективним рішенням для керування ідентифікацією та доступом (IAM), яке використовується багатьма організаціями для безпечного та зручного керування доступом. IT-адміністратори використовують Azure AD для контролю доступу користувачів до програм та ресурсів та надання ролей та дозволів.

Azure AD дає користувачам можливість отримати доступ до внутрішніх та зовнішніх ресурсів, включаючи хмарні програми, які розроблені спеціально для вашої організації, такі як Microsoft 365 та портал Azure. За допомогою Azure AD та SSO користувачі можуть без зусиль отримати доступ до різноманітних програм SaaS, не вводячи паролі багато разів.

Azure AD може реалізувати багатофакторну автентифікацію (MFA) для критично важливих ресурсів компанії. Маркери, згенеровані Azure AD, зберігаються локально на пристроях співробітників.

Особливості:

- Azure AD дозволяє захистити доступ до ресурсів за допомогою кількох факторів автентифікації та політики доступу.
- Єдиний вхід, доступ без пароля та спеціальний користувацький портал забезпечують зручну роботу користувачів.
- Централізована система керування ідентифікацією дозволяє підключити до будь-якої служби або пристрою всіх внутрішніх та зовнішніх користувачів.

Недоліки:

- Занадто складне управління та недостатня автоматизація можуть зробити рутинні завдання важкими та втомлюючими.
- В порівнянні з іншими рішеннями, використання Azure AD може бути менш економічно вигідним.

2.10. Хмарні бази даних PostgreSQL

PostgreSQL є потужною системою управління об'єктно-реляційною базою даних (ORDBMS) з відкритим вихідним кодом, яка дотримується стандартів та акцентує на розширюваності. Вона використовує мову SQL у поєднанні з багатьма функціями, що забезпечують безпечне зберігання та масштабування складних робочих навантажень даних. [23]

PostgreSQL є транзакційною системою, сумісною з ACID, яка зберігає дані у табличному форматі та використовує обмеження, тригери, ролі, збережені процедури та представлення як основні компоненти.

PostgreSQL забезпечує паралельні обчислення за допомогою MVCC - багатоверсійного контролю паралельності, що забезпечує кожній транзакції унікальний ідентифікатор версії. Це дозволяє уникнути блокування процесу читання та гарантує підтримку принципів ACID. Крім того, PostgreSQL дозволяє

ізоляцію транзакцій на трьох рівнях: repeatable read, serializable та read committed. Система включає вбудовану синхронну реплікацію, що забезпечує гарантовану збереження даних у випадку відмови вузла.

Довговічність транзакцій у PostgreSQL може бути визначена на рівні бази даних, користувача, за сеанс або за транзакцію. Це корисно для робочих навантажень, які не вимагають таких транзакцій, але може не бути потрібним для усіх даних, оскільки це може сповільнити продуктивність через вимогу підтвердження цих транзакцій у синхронному режимі очікування.

Сервери, які зарезервовані, можуть бути синхронними або асинхронними. Синхронні сервери вказуються в конфігурації і визначають, які саме сервери є кандидатами для синхронної реплікації. Поточним синхронним сервером буде використовуватися перший у списку, який активно транлюється. Коли стається збій, система автоматично змінює сервер на наступний у черзі.

На малюнку нижче (рис. 2.8) зображено інтерфейс хмарної бази даних PostgreSQL

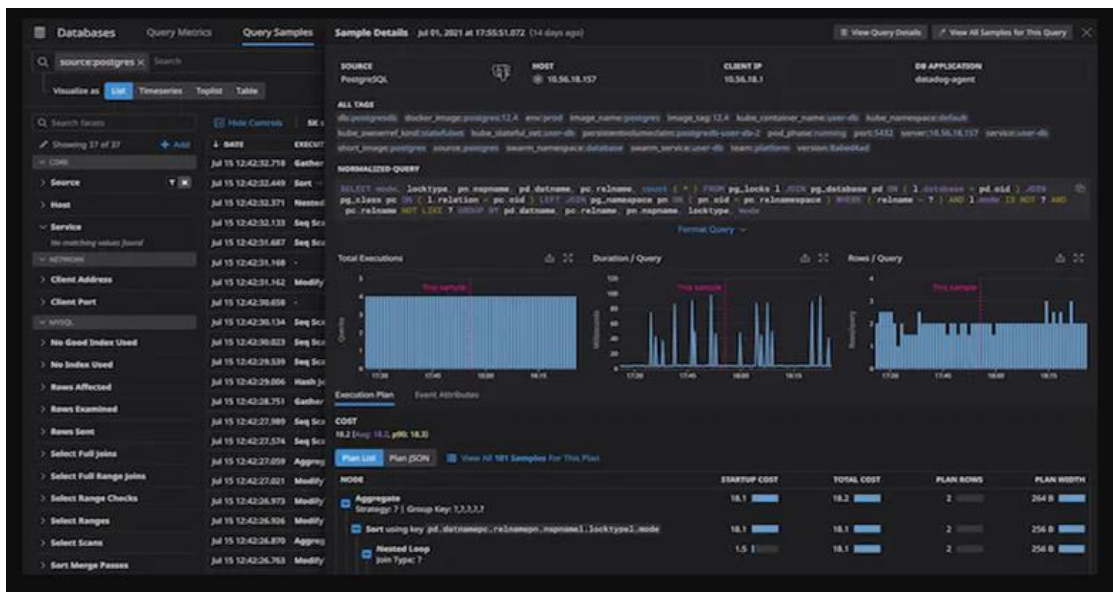


Рис. 2.8 - Інтерфейс хмарної бази даних PostgreSQL

PostgreSQL містить вбудовані інтерфейси для lib pg та ECPG, які дозволяють підключатися до програм. Існують також бібліотеки третіх сторін для прямого підключення до PostgreSQL з багатьох мов програмування, таких як Java, C++, Python, Ruby, NodeJs тощо.

Основні особливості PostgreSQL, які роблять його популярним використовувати:

- Безкоштовний продукт з відкритим кодом;
- Доступний на декількох мовах програмування;
- Високо масштабований;
- Захищає цілісність даних;
- Створює надійне оточення.

2.11. Висновки до другого розділу.

У другому розділі були проаналізовані програмні засоби, які будуть використовуватися при розробці Web-застосунку "UAHelper". Було розглянуто середовище розробки Visual Studio Code, мови розмітки HTML, CSS, графічний редактор Figma. Також було проаналізовано мову програмування C#, та її основні фреймворки. Після опису порівняння фреймворків було описано обраний фреймворк для створення додатку, його переваги та недоліки. Також були описані обрані інструменти для розробки, такі як PHPStorm, та серверна платформа Microsoft Azure. Ці інструменти завжди залишатимуться у моєму наборі для розробки, оскільки вони дуже зручні та мають необхідний функціонал для успішної та швидкої розробки. Головне - вибрати правильні технології відразу, щоб у подальшому уникнути складнощів з розширенням проєкту.

РОЗДІЛ 3. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ЗАСТОСУНКУ

3.1. Розробка плану проекту

Розробка плану-проекту є важливим кроком у процесі управління проектом і має наступні цілі:

1. Орієнтація: План-проект допомагає визначити напрямок розробки проекту і його цілі. Він допомагає всім учасникам проекту, включаючи команду розробників, замовника та зацікавлені сторони, зрозуміти, яким чином проект буде виконуватися та досягати своїх мет.

2. Планування ресурсів: План-проект допомагає визначити потрібні ресурси, такі як людські ресурси, фінансові кошти, матеріали, обладнання тощо. Це дозволяє оптимізувати використання ресурсів і забезпечити їх належне розподілення.

3. Управління термінами: План-проект містить розклад робіт та визначає критичні етапи та мілістоуни, що допомагає управляти термінами проекту. Він дозволяє визначити дати поставки, встановити пріоритети та керувати часовими обмеженнями.

4. Керування ризиками: План-проект допомагає ідентифікувати потенційні ризики та прийняти заходи для їх запобігання або зменшення впливу. Він дозволяє управляти ризиками, передбачати можливі проблеми та розробляти плани реагування.

Починаємо розробляти власний план-проект для розробки веб-застосунку:

1. Визначення вимог;

- Встановлення мети та цілей проекту.
- Аналіз потреб користувачів.

- Визначення функціональних та нефункціональних вимог до застосунку.

2. Проектування;

- Створення структури та архітектури системи.
- Розробка схем бази даних та моделей даних.
- Розробка інтерфейсу користувача (UI/UX).

3. Розробка;

- Налаштування робочого середовища та інструментів розробки.
- Реалізація функціональності застосунку на основі визначених вимог.
- Розробка серверної логіки, включаючи обробку запитів, взаємодію з базою даних та зовнішніми сервісами.

4. Тестування;

- Виконання тестування на рівні користувача (UI тестування).

5. Впровадження;

- Розгортання застосунку на сервері або хостингу.
- Налаштування потрібного середовища для роботи застосунку.
- Забезпечення безпеки та захисту даних.

6. Супровід та підтримка;

- Вирішення помилок, які виявлені після випуску.
- Внесення змін та додаткового функціоналу на основі фідбеку користувачів.
- Регулярне оновлення системи, включаючи захист від вразливостей та удосконалення процесів.

7. Керування проектом;

- Планування та відстеження роботи над проектом.
- Керування ресурсами та термінами.

8. Методологія розробки - Agile (гнучкий) - це методологія розробки програмного забезпечення, яка акцентує на ітераційному та інкрементальному підході до

розробки, співпраці між розробниками та замовником проекту, а також на відповідності змінним вимогам та здатності швидко адаптуватися до змін.

3.2. Структура проекту

Структура проекту в галузі ІТ визначає організацію та ієрархію комунікаційних, функціональних та ресурсних зв'язків всередині проектної команди або організації. Вона визначає, які ролі та відповідальності призначені учасникам проекту, як вони спілкуються, як розподіляються завдання і ресурси. Оскільки проект реалізовується в рамках написання дипломної бакалаврської роботи, то ієрархія відсутня, адже проект написаний з нуля виконавцем дипломної роботи.

3.3. Створення дизайнерського макету

Під час створення дизайнерського макету, був зроблений упор на те, що користувачами веб-застосунку можуть бути як і діти, так і люди похилого віку. Тобто, додаток має бути дуже гнучким та максимально зрозумілим для користувачів всіх вікових категорій. Також, додаток має бути адаптивний під ПК, Ноутбуки, Планшети та смартфони навіть старих моделей з маленьким екраном.

Спираючись на вищевказані показники був створений макет(Рис. 3.1).

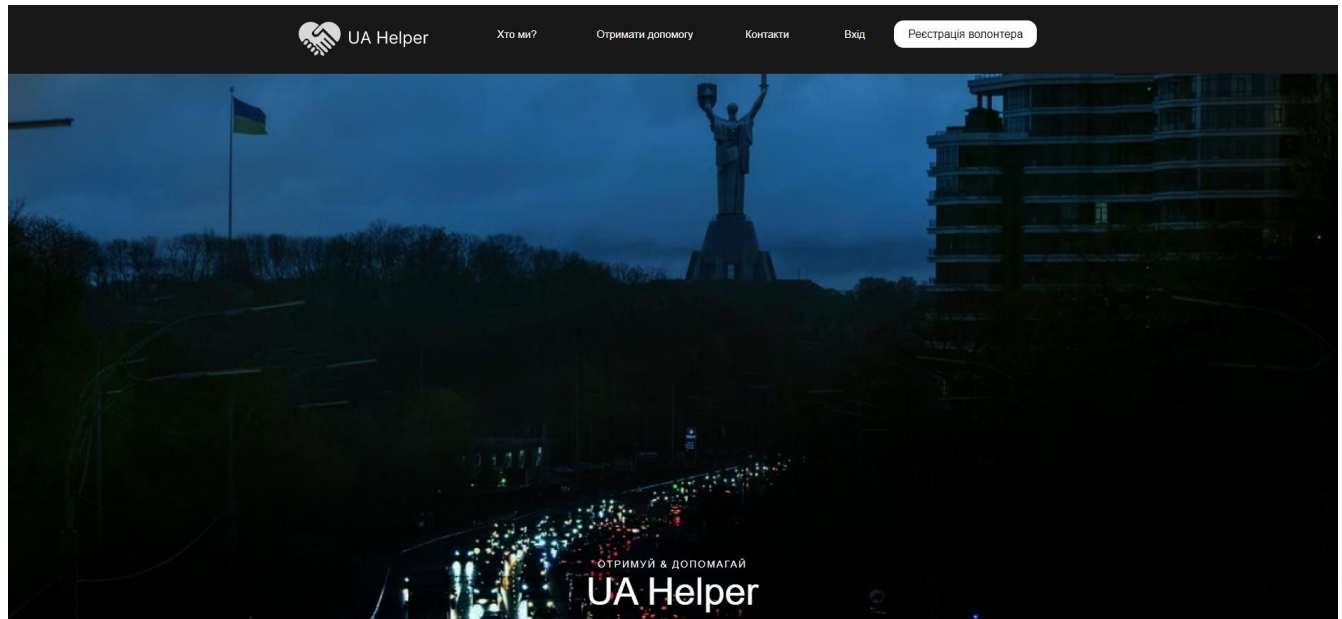


Рис 3.1 – Макет веб-застосунку

Макет по задумці мав містити стрічку з новинами світових передач(Рис.3.2)

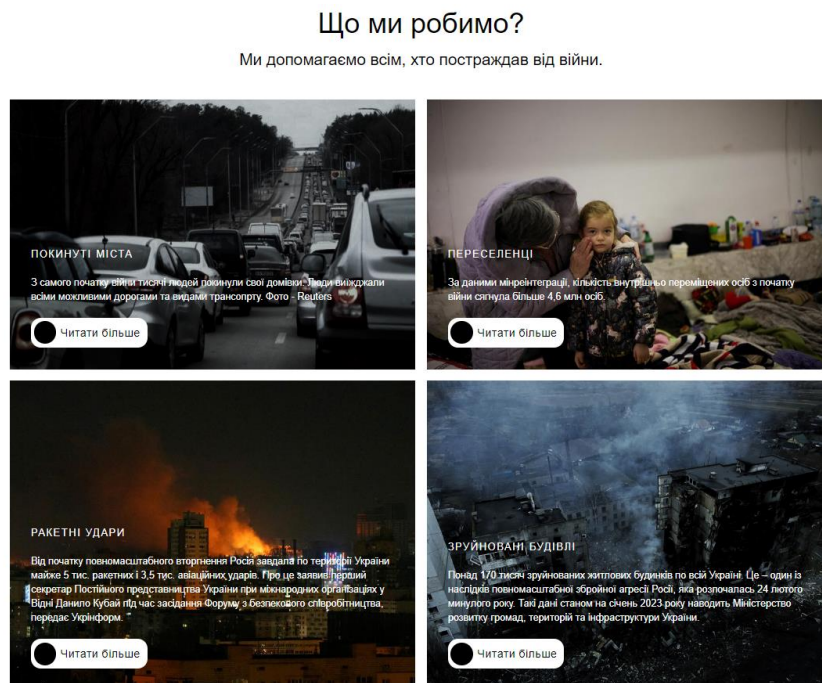


Рис. 3.2 – макет стрічки з новинами

Також, має міститись загальна інформація, про допомогу переселенцям, та людям, які були змушені покинути свої домівки через повномасштабне вторгнення(Рис. 3.3)



Рис. 3.4 – Загальна інформація про допомогу для переселенців

Інформація про благодійні організації та партнерів(Рис. 3.5)

Наші партнери

Благодійність



UNITED24

КОЛО

Рис. 3.5 – партнери та благодійні організації

Дизайн сторінки переходу на форму заповнення для отримання волонтерської допомоги(Рис.3.6)

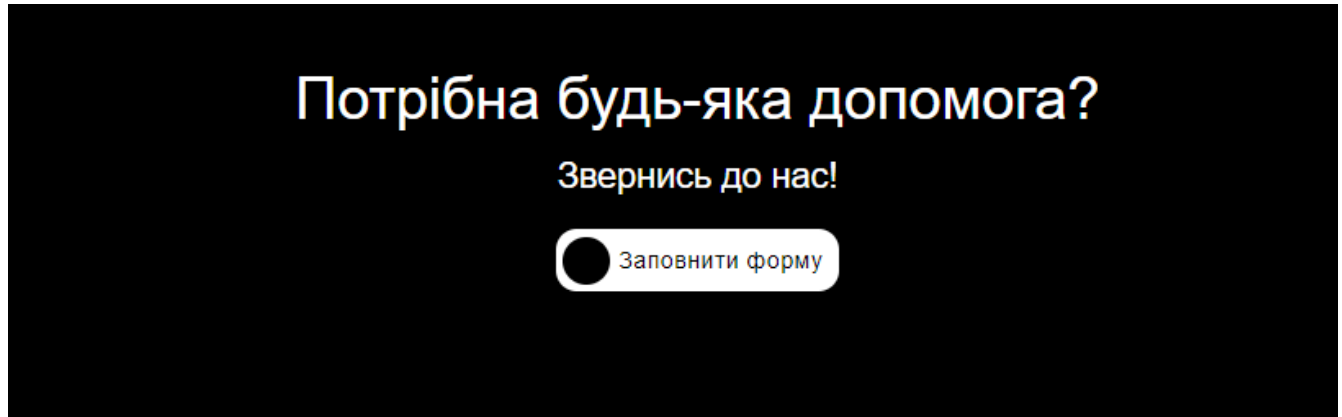


Рис. 3.6. – Перехід на заповнення форми для отримання допомоги

Та мапа, яка працюватиме на будь-яких пристроях з відмітками волонтерських штабів, де саме можна звернутись, для отримання допомоги.(Рис.3.7)

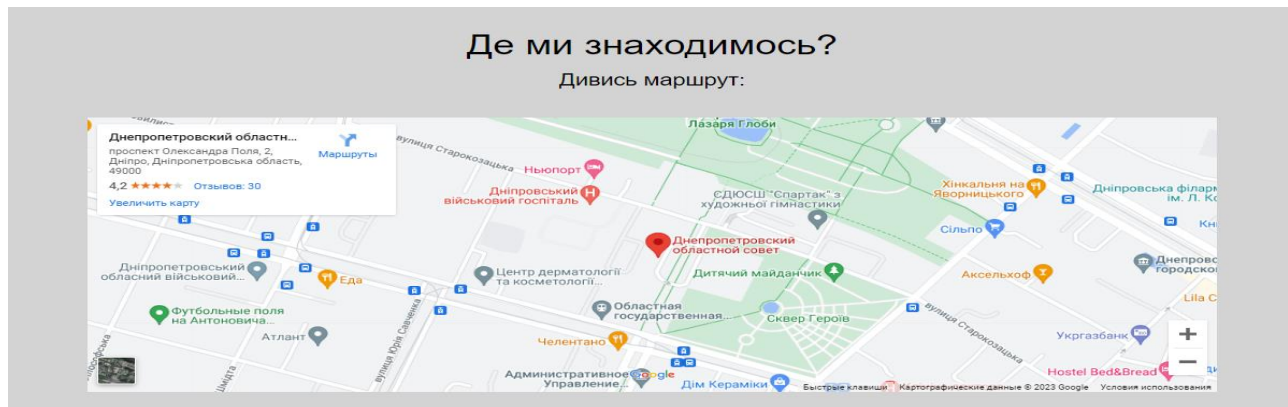


Рис. 3.7 – мапа волонтерських штабів

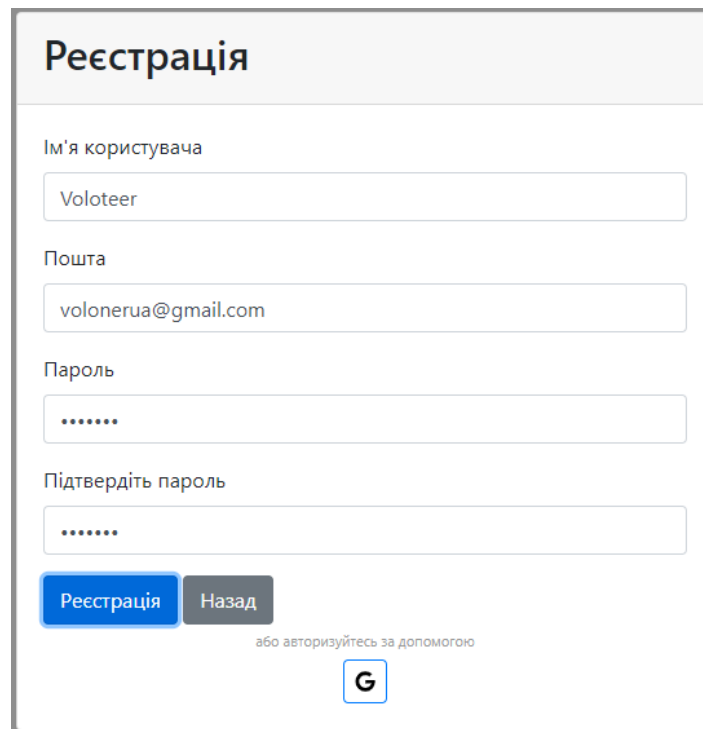
3.4. Реалізація системи авторизації

Авторизація - це процес перевірки та підтвердження ідентифікації користувача або суб'єкта з метою надання доступу до певних ресурсів, функцій

або інформації. Це процес, який визначає права та привілеї користувача та переконується, що він має дозвіл на використання обраних ресурсів.

Авторизація зазвичай використовується для захисту конфіденційної інформації, контролю доступу до систем, додатків або веб-сторінок, забезпечення цілісності даних та ідентифікації користувачів. Вона забезпечує контроль над тим, хто може отримувати доступ до ресурсів, та обмежує доступ для неповноважених користувачів.

Логіка роботи веб-застосунку така, що авторизуватись може лише той, хто має статус волонтера(Рис.3.8). Але спочатку, він має пройти реєстрацію, після чого дані будуть відправлені Адміністратору сайту, який в свою чергу має підтвердити дані та надати доступ на перегляд всіх заявок від людей, які очікують на допомогу.



The image shows a web form for volunteer registration. The title is 'Реєстрація'. It contains four input fields: 'Ім'я користувача' (Username) with the value 'Voloteer', 'Пошта' (Email) with the value 'volonerua@gmail.com', 'Пароль' (Password) with masked characters '.....', and 'Підтвердіть пароль' (Confirm password) also with masked characters '.....'. Below the fields are two buttons: a blue 'Реєстрація' button and a grey 'Назад' button. At the bottom, there is a link 'або авторизуйтесь за допомогою' followed by a Google logo.

Рис. 3.8 - Реєстрація волонтера

Після заповнення реєстраційної форми волонтера, виходить таке повідомлення, в якому вказується, що волонтеру присвоюється унікальний ідентифікатор користувача, та його сторінка буде підтверджена адміністратором протягом доби.(Рис.3.9)

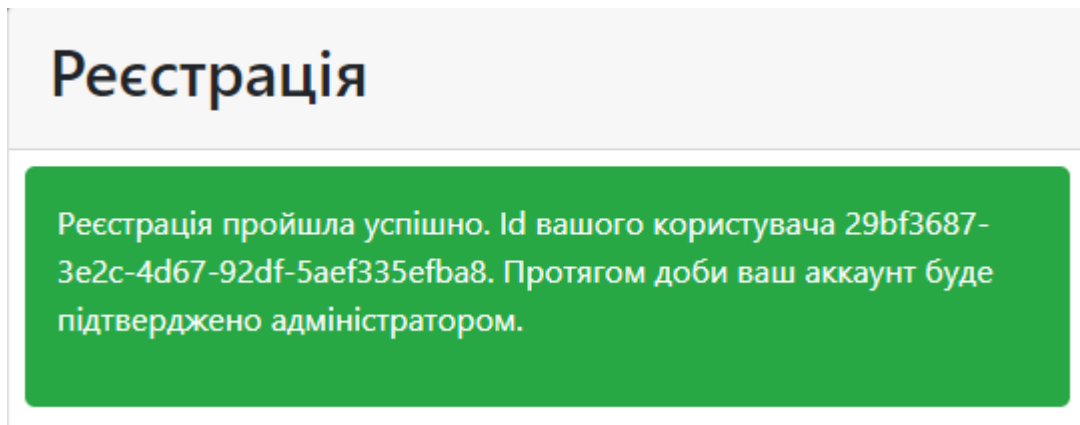


Рис. 3.9 – повідомлення після заповнення форми волонтера

Після аутенфікації особи Адміністратором сторінки, волонтер отримує доступ до списку всіх заявок від людей, які потребують допомоги(Рис. 3.10). На сторінці він може взяти в роботу, щоб інші волонтери бачили, що питанням по заявці вже займається людина. Може редагувати дані, якщо, наприклад, переселенець вказав не всі нюанси.

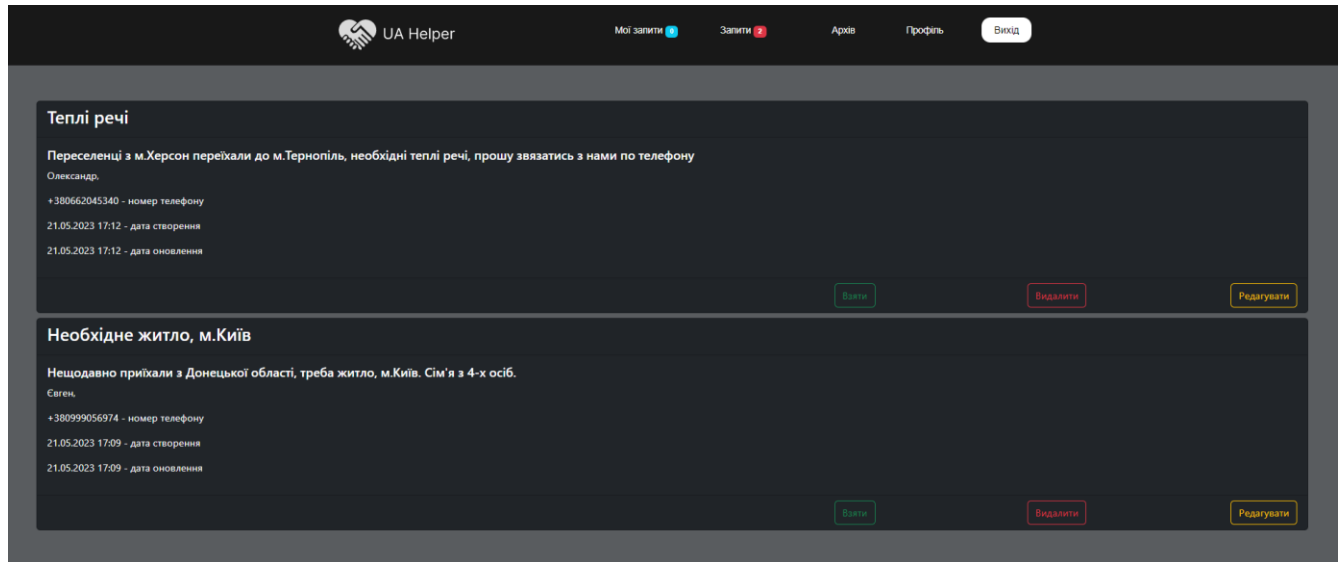


Рис. 3.10 – робоча сторінка волонтера з заявками на допомогу

Також, реалізована можливість перегляду вже відпрацьованих заявок в архіві, та зміна профілю волонтера(Рис. 3.11).

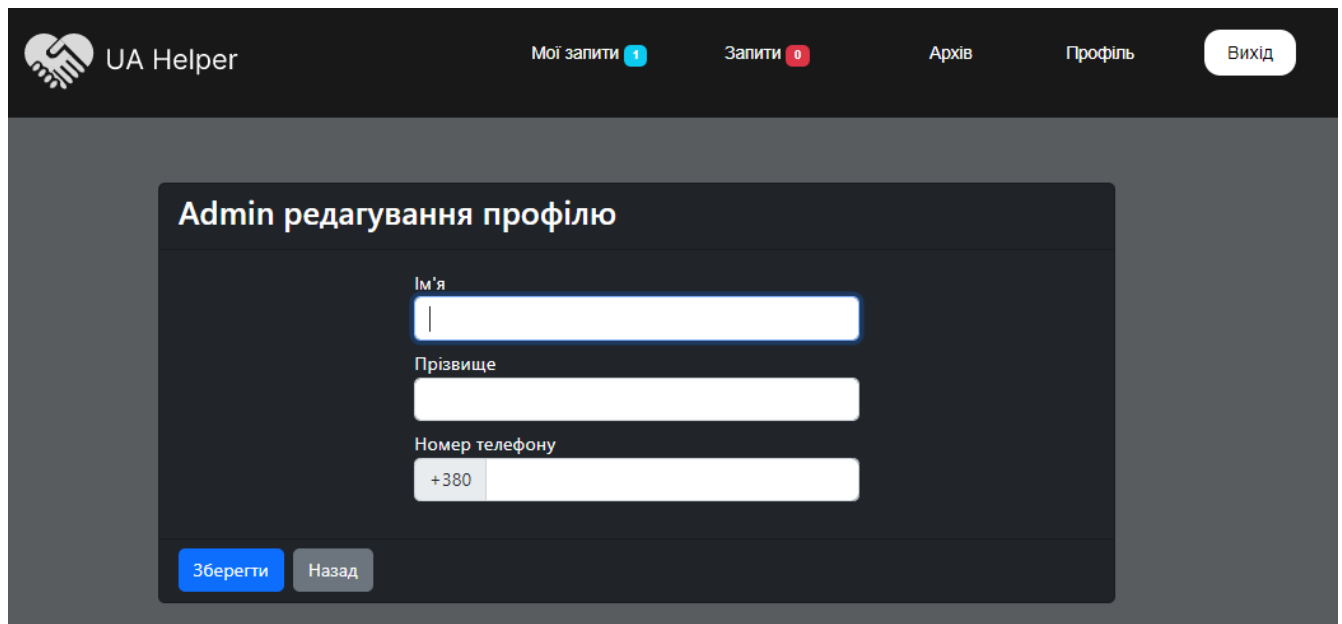


Рис. 3.11 – форма редагування профілю волонтера

Код реалізації авторизації бек-енд частини мовою С# зображена на Рис.3.12

```
public async Task<ApplicationUser> AutoProvisionUser(string provider, string userId,
List<Claim> claims)
{
    // create a list of claims that we want to transfer into our store
    var filtered = new List<Claim>();
    foreach (var claim in claims)
        // if the external system sends a display name - translate that to the
standard OIDC name claim
        if (claim.Type == ClaimTypes.Name)
            filtered.Add(new Claim(JwtClaimTypes.Name, claim.Value));
        // if the JWT handler has an outbound mapping to an OIDC claim use that
// copy the claim as-is
        else
            filtered.Add(claim);

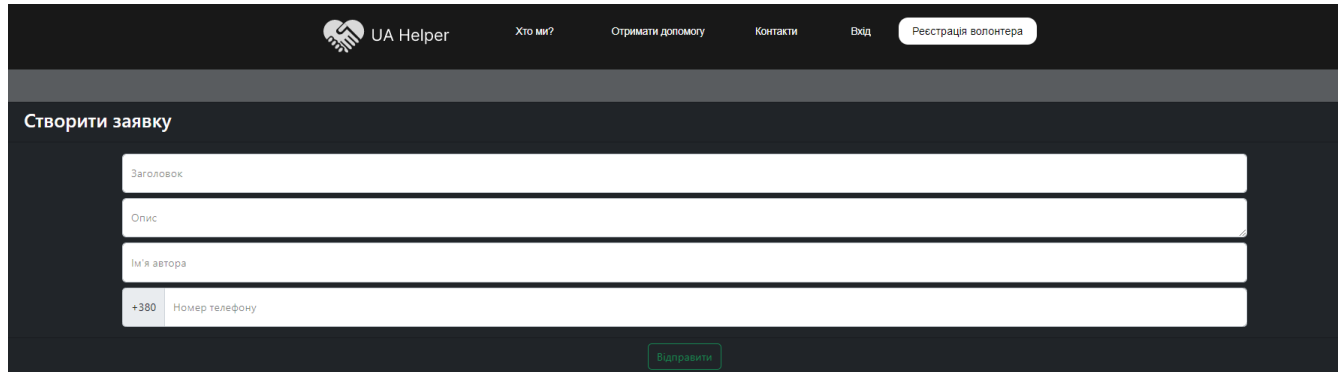
    // if no display name was provided, try to construct by first and/or last name
    if (filtered.All(x => x.Type != JwtClaimTypes.Name))
    {
        var first = filtered.FirstOrDefault(x => x.Type ==
JwtClaimTypes.GivenName)?.Value;
        var last = filtered.FirstOrDefault(x => x.Type ==
JwtClaimTypes.FamilyName)?.Value;
        if (first != null && last != null)
            filtered.Add(new Claim(JwtClaimTypes.Name, first + " " + last));
        else if (first != null)
            filtered.Add(new Claim(JwtClaimTypes.Name, first));
        else if (last != null) filtered.Add(new Claim(JwtClaimTypes.Name, last));
    }

    // check if a display name is available, otherwise fallback to subject id
    var name = filtered.FirstOrDefault(c => c.Type == ClaimTypes.GivenName)?.Value
?? userId;
    var email = filtered.FirstOrDefault(c => c.Type == ClaimTypes.Email)?.Value;
    var lastName = filtered.FirstOrDefault(c => c.Type ==
ClaimTypes.Surname)?.Value;
```

Рис. 3.12 – Код реалізації авторизації бек-енд частини мовою С#

3.5. Реалізація взаємодії волонтера та переселенця

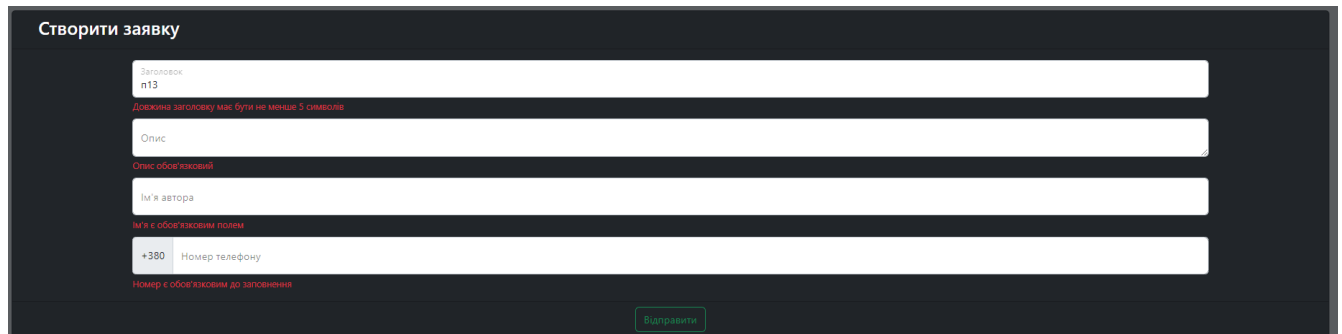
Для реалізації взаємодії було зроблено форму, яку можна заповнити якнайшвидше. Тобто, без реєстрації та підтвердження користувача через пошту або номер телефону. Форма має містити Заголовок, короткий опис, Ім'я людини та номер телефону(Рис. 3.12).



The screenshot shows the 'UA Helper' website interface. At the top, there is a navigation bar with the logo and links for 'Хто ми?', 'Отримати допомогу', 'Контакти', 'Вхід', and a highlighted 'Реєстрація волонтера' button. Below the navigation bar, the main content area is titled 'Створити заявку'. It contains four input fields: 'Заголовок', 'Опис', 'Ім'я автора', and '+380 Номер телефону'. A 'Відправити' button is located at the bottom right of the form.

Рис.3.12 - Форма заповнення заявки для переселенця

Задля зменшення незрозумілих заявок та невалідних повідомлень, була розроблена валідація форми, де користувачу форма підказує, правильність заповнення(Рис.3.13)



The screenshot shows the same form as in Figure 3.12, but with validation error messages. The 'Заголовок' field has a red error message: 'Довжина заголовку має бути не менше 5 символів'. The 'Опис' field has a red error message: 'Опис обов'язковий'. The 'Ім'я автора' field has a red error message: 'Ім'я є обов'язковим полем'. The '+380 Номер телефону' field has a red error message: 'Номер є обов'язковим до заповнення'. The 'Відправити' button is still visible at the bottom right.

Рис. 3.13 – валідація полів в формі створення заявки переселенця

3.6 Тестування веб-застосунку

Одним з найголовніших пунктів в розробці веб-застосунку була адаптивність сайту. Адаптивність веб-застосунку - це здатність сайту адекватно відобразитися та працювати на різних пристроях та розмірах екранів, таких як комп'ютери, планшети, смартфони та інші пристрої з доступом до Інтернету. Головною метою адаптивності є забезпечення оптимального користувацького досвіду на будь-якому пристрої, з урахуванням його особливостей та обмежень.

Основні причини, чому адаптивність веб-сайту є важливою:

- Зручність для користувачів: Адаптивний сайт пристосовується до різних розмірів екранів, що дозволяє користувачам легко переглядати та навігувати по вмісту незалежно від пристрою, яким вони користуються. Це робить веб-сайт зручним і доступним для широкого кола користувачів.

- Покращення пошукової оптимізації: Адаптивність веб-сайту впливає на його рейтинг у пошукових системах. Враховуючи те, що пошукові системи віддають перевагу адаптивним сайтам, ваш сайт має більші шанси з'явитися у результатах пошуку для користувачів різних пристроїв.

- Економія часу та зусиль: Замість розробки окремих версій сайту для кожного пристрою, адаптивний дизайн дозволяє створити один універсальний шаблон, який автоматично адаптується до різних пристроїв, тощо.

- Розширення аудиторії: Адаптивний веб-сайт дозволяє досягти більшої аудиторії, оскільки люди можуть легко взаємодіяти з вашим сайтом незалежно від пристрою, яким вони користуються.

Саме тому, почнемо тестувати застосунок на пристроях з різними розмірами екранів. Відкривши сторінку з Apple Iphone 12 PRO MAX(Рис.3.14) бачимо, що шапка сайту стала адаптивна на мобільній версії.

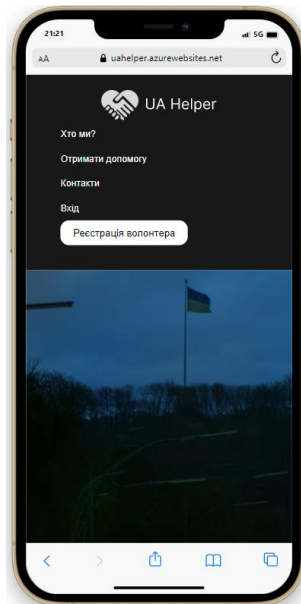


Рис. 3.14 – тестування застосунку на мобільній версії використовуючи Apple Iphone 12 PRO MAX

Оскільки сайт не «зламався», всі блоки, шрифти, картинки та стилі відображені правильно, то можемо продовжувати тестування. Протестуємо форму створення заявки переселенцем на допомогу, використовуючи смартфон ХІАОМІ 12(Рис.3.15)

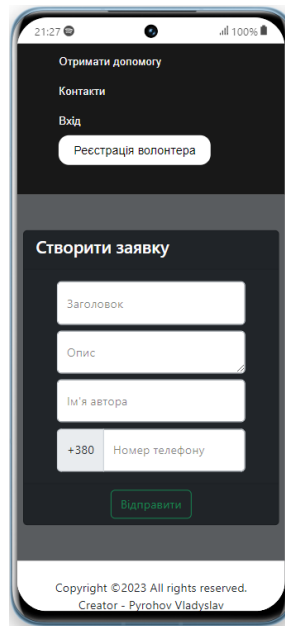


Рис.3.15 – тестування створення заявки переселенцем на допомогу використовуючи смартфон ХІАОМІ 12

Продовжуємо тестувати наш застосунок, спробуємо провести авторизацію волонтера використовуючи пристрій Apple Ipad Mini(Рис.3.16)

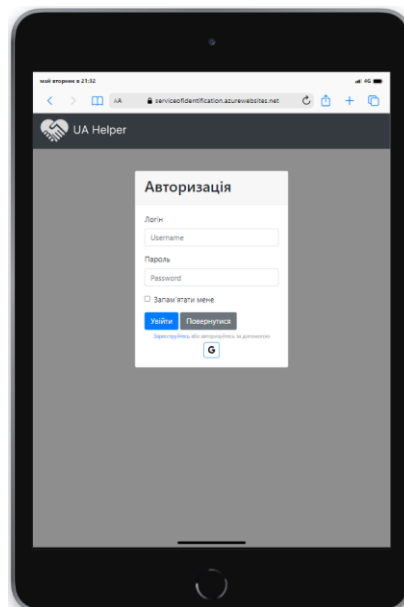


Рис.3.16 – тестування авторизації волонтера використовуючи планшет Apple Ipad Mini

Все працює правильно, тому спробуємо протестувати веб-застосунок на більш старих моделях смартфонів, в яких маленький екран, наприклад, Apple iPhone SE 2016(Рис.3.17)

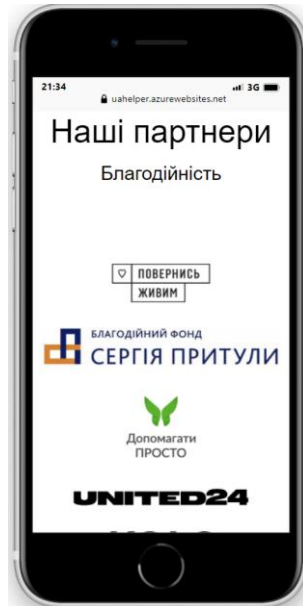


Рис. 3.17 – тестування веб-застосунку використовуючи Apple iPhone SE 2016

Тестування адаптивності веб-застосунку успішне. А зараз спробуємо протестувати, як Веб-браузери спроможні читати наш код і відображати його користувачу. Skorистаємось додатком для тестування сторінок(Рис.3.18), яким користуються розробники - <https://pagespeed.web.dev/>. [23]

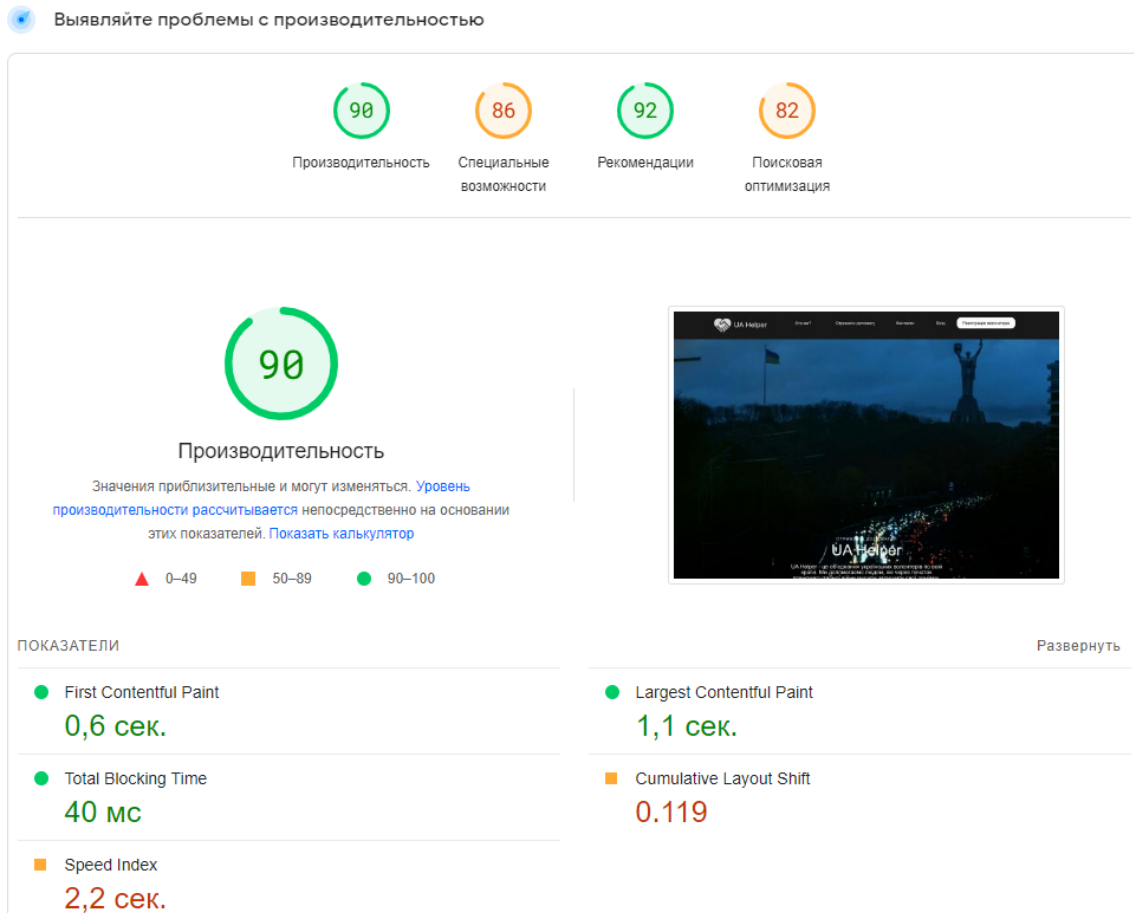


Рис.3.18 – результати тестування веб-застосунку і відображення на версії комп'ютера чи ноутбука.

Результати в цілому позитивні, адже показник вище 90, тобто браузер дуже гарно сприймає веб-застосунок на комп'ютерах та ноутбуках. Протестуємо на мобільних версіях(Рис.3.19)

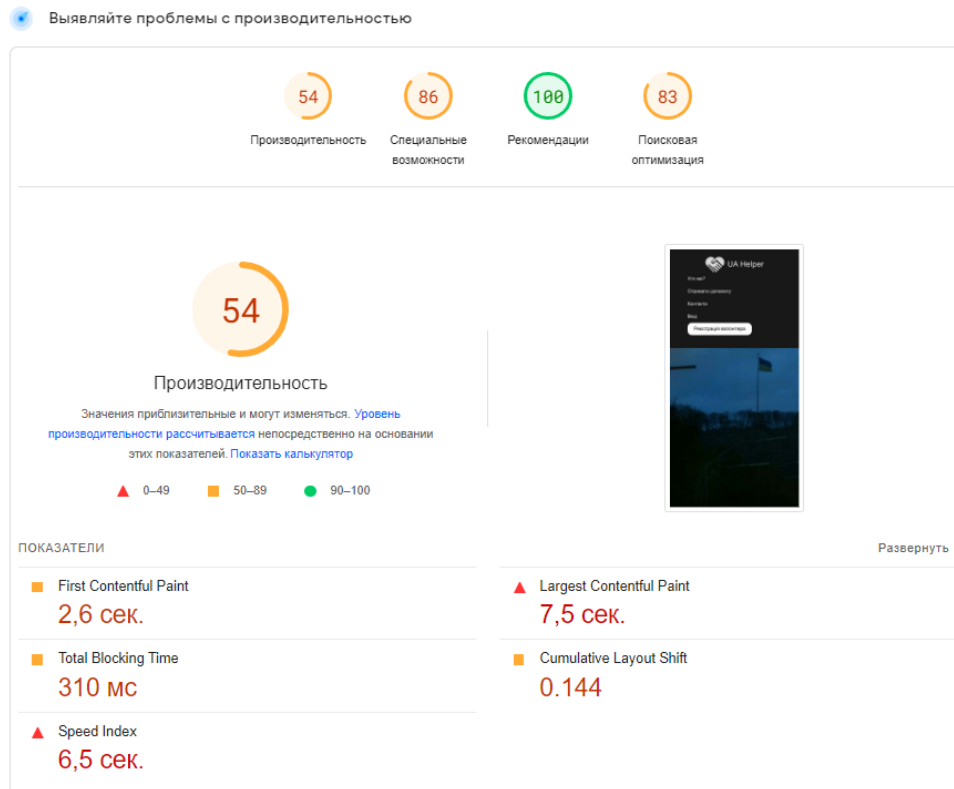


Рис.3.19 - результати тестування веб-застосунку і відображення на версії смартфонів

Результати гірші, ніж на комп'ютері, але слід пам'ятати про те, що наш сайт було опубліковано на сервері Microsoft Azure, який в свою чергу працює на безкоштовній версії не на всю. Також, через те, що сайт має багато фотографій та картинок великих розмірів та мають велику вагу, браузеру трішки важко це відкривати. Але ж, найголовніше – щоб користувачу було все доступно та зрозуміло, тому змінювати картинки та зменшувати їх розмір ми не будемо.

3.7. Рекомендації щодо подальшого вдосконалення

На основі аналізу роботи веб-застосунку і його функціональності, ми можемо запропонувати декілька рекомендацій щодо подальшого вдосконалення:

- Розширення функціональності: Визначити можливості для розширення функціоналу веб-застосунку. Проведення аналізу потреб користувачів та дослідити ринок, щоб виявити нові функції, які можуть покращити досвід користувачів та зробити застосунок більш привабливим для цільової аудиторії.

- Оптимізація продуктивності: Провести додатковий аналіз продуктивності веб-застосунку і виявіть можливості для оптимізації. Звернути увагу на завантаження сторінок, час відгуку сервера і загальну продуктивність. Застосувати техніки кешування, оптимізацію запитів до бази даних та інші методи, щоб покращити швидкість і продуктивність застосунку.

- Покращення користувацького досвіду: Здійснити аналіз користувацького досвіду та зібрані повідомлення від користувачів. Виявити можливі проблеми, затримки або незручності, з якими зустрічаються користувачі. Запропонувати покращення і вдосконалення інтерфейсу користувача, навігації та узагальнення функцій, щоб зробити веб-застосунок більш зручним і легким у використанні.

- Забезпечення безпеки: Перевірити поточні заходи безпеки веб-застосунку та забезпечте відповідність найновішим стандартам безпеки. Розглянути використання шифрування, вдосконалення механізмів аутентифікації та авторизації, виявлення й відповідь на потенційні загрози безпеки.

- Моніторинг та аналітика: Встановити систему моніторингу та аналітики для збору даних про використання та продуктивність веб-застосунку. Аналізувати ці дані, щоб виявити слабкі місця, незвичайну активність та покращити процеси. Це допоможе нам зрозуміти, як користувачі взаємодіють з вашим застосунком та виявити можливості для подальшого вдосконалення.

- Тестування: Здійснювати систематичне тестування нашого веб-застосунку. Використовуйте автоматичні тести для перевірки функціональності, та модульне Unit-тестування для виявлення помилок та валідації коду.

- Розробка: розробити мобільні додатки для ОС Android та iOS, з метою комфортного користування програмним забезпеченням для будь-якого віку користувачів.

- Сервер публікації: перейти на незалежний та більш швидкий сервер для оптимізації та швидкості роботи веб-застосунку.

Саме ці рекомендації допоможуть нашому веб-застосунку розвиватися, і найголовніше – допомагати людям.

3.8. Висновки до третього розділу

Під час проектування та реалізації веб-застосунку в рамках кваліфікаційної роботи було зроблено декілька ключових висновків. Основні висновки з проектування та реалізації веб-застосунку в дипломній роботі такі:

- Ефективне проектування архітектури: У результаті аналізу вимог та вивчення технологій була розроблена ефективна архітектура веб-застосунку. Ця архітектура дозволяє розширювати та підтримувати функціональність застосунку з мінімальними зусиллями.

- Використання сучасних технологій: У процесі реалізації веб-застосунку були використані сучасні технології та інструменти інженерії програмного забезпечення, що дозволяють забезпечити ефективну та стабільну роботу застосунку. Це включає в себе використання популярних фреймворків, бібліотек та стандартів розробки.

- Користувацький досвід: При проектуванні і реалізації веб-застосунку було приділено увагу користувацькому досвіду. Застосунок був розроблений з орієнтацією на зручність використання, інтуїтивність та приємний вигляд. Це сприяє задоволенню користувачів та покращує їх взаємодію з системою.

- Функціональність та надійність: Реалізований веб-застосунок успішно виконує всі визначені функції та завдання. Виявлені помилки та недоліки були виправлені, що сприяє надійній роботі застосунку.

- Тестування та валідація: Веб-застосунок був підданий тестуванню та валідації для перевірки його функціональності та якості. Це дозволило виявити та виправити помилки, забезпечивши якісну та стабільну роботу застосунку.

- Адаптація під всі можливі розміри екрану, від комп'ютера до старих моделей смартфонів, що дає можливість використання застосунку для всіх користувачів з будь-якого віку.

У цілому, проектування та реалізація веб-застосунку в кваліфікаційній роботі були успішно виконані з урахуванням вимог та вибраних технологій. Результатом є функціональний, ефективний та користувачем-орієнтований веб-застосунок, який задовольнятиме потреби користувачів і найголовніше – буде приносити користь у використанні його.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була проаналізована предметна область для програмного забезпечення, яке є результатом роботи. Показана актуальність розробки веб-застосунку для допомоги волонтерам та внутрішньо переміщеним особам. На даний момент в Україні не існує повністю цифрового додатку для пошуку місій в якому можна зареєструватися як волонтер. Спираючись на результати дослідження предметної області були описані вимоги до програмного забезпечення на основі яких проводились аналізи технологій які потрібно використовувати для його реалізації. Для охоплення найбільшої кількості аудиторії ПЗ представляє собою веб-додаток з клієнтською та серверною частинами. Для клієнтської частини була обрана мова програмування JavaScript, так як це найзручніший інструмент для створення клієнтської частини. Після вибору мови програмування був проведений аналіз інструментів, що прискорюють та поліпшують розробку ПЗ, на основі якого був обраний фреймворк Angular, через його переваги, та те, що він найбільш підходить до виконання цілей клієнтської частини додатку.

На основі аналізу двох найпопулярніших IDE, через більшу кількість переваг була обрана IDE WebStorm. Для написання серверної частини додатку була обрана технологія програмування ASP.NET Core Web API як найновіший інструмент для написання серверних частин на мові C#. Проаналізувавши технології баз даних серед найпопулярніших PostgreSQL та MySQL була обрана БД PostgreSQL через те, що вона більш продуктивніша ніж MySQL та не має її недоліків. Після завершення аналізу усіх частин програмного забезпечення були сформовані макети сторінок веб-додатку, схема роботи додатку та схема БД.

За макетами була створена клієнтська частина веб-додатку зі сторінками, вікнами реєстрації та функціоналом, а також серверна частина з таблицями та формами користувача для функціонування додатку. Останнім етапом розробки

було тестування системи та виправлення недоліків. Після їх виправлення та проведення тестів додатку, його можна вважати завершеним. Таким чином, програмне забезпечення, яке було розроблено в кваліфікаційній роботі, є завершеним програмним продуктом і відповідає всім поставленим вимогам та дає можливість переселенцям та людям, які постраждали від війни, швидко та якісно отримати допомогу від перевірених волонтерів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-додаток URL: <https://uk.wikipedia.org/wiki/Вебзастосунок> (дата звернення: 15.04.2023).
2. Переваги використання Web –застосунків URL: <http://sanchiz.net/blog/advantage-of-webapplications>. (дата звернення 19.03.2023).
3. Веб-додаток URL: <https://play.google.com/store/apps/details?id=ru.taxi.id9904>(дата звернення 19.03.2023).
4. Веб-додаток URL: https://zakordon.24tv.ua/noviy-dodatok-yeshelp-dlya-ukrayintsiv-dopomozhe-znayti-volonteriv_n2182349 (дата звернення 19.03.2023).
5. Веб-додаток URL: <https://uahelper.azurewebsites.net/> (дата звернення 20.02.2023).
6. Веб-додаток URL: PHPStorm [Електронний ресурс] — Режим доступу: <https://itpro.ua/product/jetbrains> (дата звернення 20.03.2023).
7. Visual Studio Code описание, подбор аналогов. URL: <https://bizzapps.ru/p/visualstudio-code>. (дата звернення 27.03.2023). 20
8. Руководство по ASP.NET MVC 5 URL: <https://metanit.com/sharp/mvc5>. (дата звернення 18.04.2023)
9. Документація .NET [Електронний ресурс] — Режим доступу: <https://docs.microsoft.com/en-us/dotnet/fundamentals/> (дата звернення 18.04.2023)
10. Документація ASP.NET Core [Електронний ресурс] — Режим доступу: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0> (дата звернення 18.04.2023)
11. Документація ASP.NET Core [Електронний ресурс] — Режим доступу: <https://metanit.com/sharp/aspnet5/23.1.php> (дата звернення 18.04.2023)
12. Веб-додаток URL: <https://dev.to/renukapatil/create-web-api-with-aspnet-core-60-4614> - (дата звернення 18.04.2023)
13. Веб-додаток URL: <https://beonmax.com/blog/figma/> - (дата звернення 18.04.2023)
14. Веб-додаток URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics - (дата звернення 18.04.2023)

- 15 Веб-додаток URL: <https://www.hostinger.com/tutorials/what-is-html> - (дата звернення 18.04.2023)
16. Веб-додаток URL: <https://learn.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/> - (дата звернення 18.04.2023)
17. Большая книга CSS3 [Электронный ресурс]. — Режим доступа: ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/Дэвид Сойер Макфарланд — Большая книга CSS3/(дата звернення 15.04.2023) – Назва з екрану
18. TypeScript [Электронный ресурс]. Режим доступа: <https://www.typescriptlang.org/> (дата звернення 02.04.2023) – Назва з екрана.
19. All of Angular [Электронный ресурс]. Режим доступа: <https://angular.io/> (дата звернення 02.04.2023) – Назва з екрана
20. AngularJs [Электронный ресурс]. Режим доступа <https://angularjs.org/> (дата звернення 02.04.2023) – Назва з екрана.
21. Веб-додаток URL: <https://carecomputers.co.uk/top-9-microsoft-azure-services-for-businesses-in-2022/> (дата звернення 02.04.2023) – Назва з екрана.
22. Веб-додаток URL: <https://www.zluri.com/blog/microsoft-azure-active-directory-alternatives/> <https://www.datadoghq.com/dg/monitor/postgresql-23>. Мониторинг Datadog | Датадог ([datadoghq.com](https://www.datadoghq.com/enefits/?utm_source=advertisement&utm_medium=search&utm_campaign=dg-google-dbm-emea))enefits/?utm_source=advertisement&utm_medium=search&utm_campaign=dg-google-dbm-emea-(дата звернення 02.04.2023) – Назва з екрана.
23. Веб-додаток URL: <https://pagespeed.web.dev/> - (дата звернення 03.04.2023) – Назва з екрана.

ДОДАТКИ

1. Посилання на веб-додаток, розроблений згідно бакалаврської дипломної роботи. Веб-додаток URL: <https://uahelper.azurewebsites.net/>
2. Посилання на проект та повний код веб-застосунку в GitHub. Веб-додаток URL: <https://github.com/stanlocktop/BachelorProject>