

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРИКЛАДНОЇ ІНФОРМАТИКИ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ В.П. Гамаюн

«_____» _____ 2021 р.

ДИПЛОМНИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСНИКА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР

Тема: «Онлайн моніторинг громадського транспорту м. Ірпінь»

Виконавець: Руденко Орест Ігорович

Керівник: Кірхар Наталія Володимирівна

Нормоконтролер: Боровик Володимир Миколайович

Київ 2021

ВСТУП

Розвиток цифрових технологій, який ми спостерігаємо останні декілька десятиліть років, спрямований не тільки на загальний прогрес людства, а й на підвищення рівня комфорту кожної людини. «Розумні» пристрої, типу сучасних електронних годинників, холодильників, кондиціонерів та інших побутових девайсів непомітно стали звичними для більшості населення розвинених країн світу. Літаючі дрони, які моніторять стан сільськогосподарських угідь, камери відеоспостереження на кожному кутку в крупних містах, світлофори, які адаптуються під інтенсивність трафіку – ці всі винаходи є прикладами такої технології, як інтернет речей (англ. internet of things, IoT).

Інтернет речей активно розвивається в країнах першого світу, але Україна теж не відстає, тому ми можемо своїми очима спостерігати за локальним використанням таких технологій на території нашої країни. В першу чергу, інтернет речей активно розвивається в аграрному секторі, логістиці та в рамках проєктів Smart City. Хорошим конкретним прикладом може слугувати сервіс EasyWay [1], який був створений в Україні в 2011 році для моніторингу громадського транспорту у Львові та Києві. Сервіс чудово показав себе під час Євро 2012, пізніше став партнером міжнародних корпорацій і продовжує полегшувати життя жителів та гостей Львова та Києва сьогодні.

Невеликі міста, особливо міста-супутники Києва, теж потребують такої системи моніторингу, адже вони вже мають розвинену систему громадського транспорту, навігація для якої існує поки тільки у вигляді дорожніх знаків та статичних маршрутів на онлайн картах.

Тому завданням дипломної роботи є створення веб-сайту для онлайн моніторингу громадського транспорту в місті Ірпінь.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Поняття інтернету речей та суміжних технологій

Інтернет речей (англ. Internet of Things, IoT) — концепція мережі, що складається із взаємозв'язаних фізичних пристроїв, які мають вбудовані передатчики та програмне забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі, за допомогою використання стандартних протоколів зв'язку. Окрім передатчиків, мережа може мати виконавчі пристрої, вбудовані у фізичні об'єкти і пов'язані між собою через дротові чи бездротові мережі. Ці взаємопов'язані пристрої мають можливість зчитування та приведення в дію, функцію програмування та ідентифікації, а також дозволяють виключити необхідність участі людини, за рахунок використання інтелектуальних інтерфейсів.

Набуває поширення також термін «інтернет всього» (англ. Internet of Everything, IoE) — всеохопний, або всеосяжний інтернет. Це явище спричинило занепокоєння в конфіденційності інформації й сприяло появі такого терміну, як «безпека інтернету речей» [2].

Слід уточнити, що термін «інтернет речей» не є інтернетом, яким ми користуємося в повсякденному житті, в звичному розумінні. Інтернет, в даному випадку, слід розуміти як мережу, яка об'єднує фізичні речі та програмне забезпечення, яке ці речі моніторить та надає необхідну інформацію про їхній стан користувачу.

1.2. Засоби ідентифікації

Для того, щоб використовувати предмети фізичного світу в мережі інтернету речей, потрібні відповідні засоби ідентифікації, тобто визначення певного предмету, як складової частини мережі.

До таких засобів відносяться GPS-трекери, якщо мова йде про об'єкти, які напряму не підключені до глобальної мережі інтернет, МАК-адреси та IP-адреси для об'єктів, які мають підключення до інтернету. Також засобами ідентифікації можуть слугувати спеціальні знаки, які можна розпізнати оптично за допомогою програмного забезпечення: штрих-коди, QR-коди, спеціальні набори символів (рис.1.1.).



Рис.1.1. Приклад QR-коду

1.3. Засоби виміру

Ключову роль в технології інтернету речей відіграють засоби виміру, які перетворюють аналогову інформації з фізичного, зовнішнього світу в набір цифрових даних, зрозумілий для всієї системи.

Використовується широкий набір засобів виміру:

- датчики температури, рівня освітлення, тиску, швидкості тощо (рис.1.2.);
- прилади обліку споживання (наприклад, інтелектуальні лічильники);
- складні інтегровані вимірювальні системи.



Рис.1.2. Приклад датчика температури та вологості повітря

Принципово важливо об'єднувати засоби виміру в системи якомога більшого розміру, для підвищення точності виміру, надійності, полегшення технічного обслуговування та зменшення загального рівня енергоспоживання.

Вирішення проблеми ефективного автономного живлення для засобів виміру, як і всіх автономних електроприладів мережі, це шлях до підвищення загальної продуктивності системи, адже система не буде хоча б мінімально робочою, якщо її автономні елементи зможуть працювати без підзарядки всього декілька годин.

1.4. Засоби передачі даних

Передача даних всередині мережі інтернету речей може відбуватися бездротовими та дротовими методами та всіма можливими комбінаціями цих технологій.

Для бездротової передачі даних особливо важливу роль в побудові інтернету речей відіграють такі характеристики, як ефективність, відмовостійкість, адаптивність, можливість самоорганізації. Великий інтерес для реалізації вище описаних характеристик представляє стандарт IEEE 802.15.4, що управляє доступом для організації енергоефективних персональних мереж, і є основою для таких протоколів, як ZigBee та 6LoWPAN.

ZigBee — це комунікаційна технологія, заснована на протоколі IEEE 802.15.4 для реалізації низькошвидкісних бездротових приватних мереж. Наразі ZigBee використовується в основному для комунікації між елементами мережі, які знаходяться на невеликій відстані та не потребують високої швидкості передачі даних між собою.

Для передачі даних також використовується Wi-Fi (рис.1.3.). Wi-Fi — це локальна бездротова технологія, яка використовує 2,4 ГГц надвисокої частоти або 5 ГГц супер-високочастотної радіохвилі. Wi-Fi підходить для передачі великих об'ємів інформації, але має високе рівень споживання електроенергії.



Рис.1.3. Приклад вай-фай роутера для «розумного» будинку

Технологія Bluetooth використовується для передачі даних в персональних мережах. Під персональними мережами слід розуміти домашні мережі (наприклад, електричний дверний замок, який управляється через смартфон), особисті мережі (наприклад, мережа смартфон-годинник), невеликі робочі мережі (наприклад, аудіо-система в кафе, яка управляється зі смартфона чи ноутбука).

Важливу нішу займають технології, які використовують лінії електропередач для передачі даних, так як багато фізичних об'єктів мають стаціонарне підключення до електромережі: банкомати, торгові автомати, лічильники тощо).

1.5. Засоби обробки даних

Засоби обробки даних представляють собою програмне забезпечення, яке аналізує, систематизує та обробляє інформацію, отриману від всіх датчиків, лічильників та приладів системи.

Обробка даних може відбуватися як на стороні серверу, коли кінцевий користувач отримує лиш частину даних, так і на стороні клієнта, коли на девайс користувача приходять максимальна кількість даних, які потенційно можуть знадобитися. Використання підходу обробки інформації на стороні клієнта може дуже знизити навантаження на центральний сервер у випадку, коли користувачів багато, а потужність серверу не висока.

До категорії засобів обробки даних також відносяться засоби зберігання та систематизації даних, яких в процесі роботи може накопичуватися дуже багато. Тут постає питання, чи потрібно взагалі витратити ресурси на зберігання даних, які вже були використані і не будуть знову задіяні в тих самих цілях. Відповідь на це питання дає наука Data Science та технологія обробки даних Big Data.

Data Science (рис.1.4.) - це наука про методи аналізу даних і вилучення з них цінної інформації, знань. Вона тісно перетинається з машинним навчанням (Machine Learning), наукою про мислення (Cognitive Science) і, звичайно ж, технологіями для роботи з великими даними (Big Data).

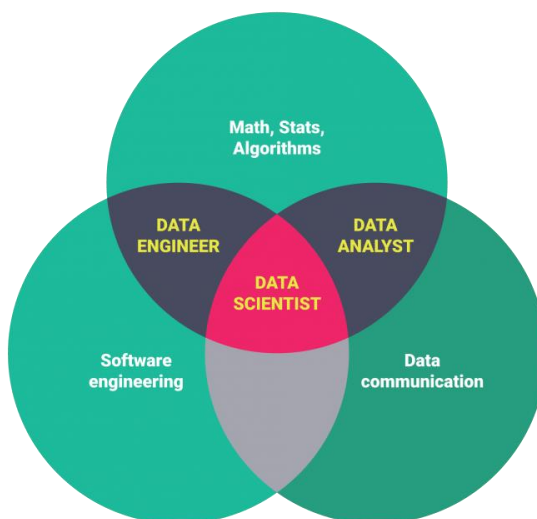


Рис.1.4. Діаграма, яка пояснює, з чого складається Data Science

Будь-яка система, яка має достатню кількість користувачів та щоденно обробляє великі масиви даних, має змогу збирати величезні об'єми персоналізованої та загальної інформації, яка після аналізу та спеціалізованої обробки методами Big Data та Data Science набуває комерційної цінності. Великі корпорації, як то Google, Facebook, Microsoft, роками збирають подібні дані, аналізують їх та отримують нову, абсолютно унікальну інформацію щодо звичок, вподобань, переміщень, покупок та навіть режиму сну своїх користувачів.

Збір та обробка великих даних необхідна для підвищення комерційної ефективності компаній, покупки чи продажу максимально влучної реклами для окремих категорій користувачів та конкретної людини. Аналіз великих даних дає змогу не тільки провести ретроспективу, а й зробити досить точні прогнози на майбутнє, зрозуміти шаблони поведінки різних людей, розбити їх на категорії та прослідкувати зміну цих шаблонів з часом.

Тому питання зберігання чи знищення інформації в наш час стає риторичним. Компанії рідко прямим текстом говорять про те, що вони збирають особисту інформацію користувачів, але за законом цей пункт має бути обов'язково вказаний в договорі користувача.

1.6. Безпека

Інтернет речей може викликати величезні зміни у повсякденному житті, надавши звичайним користувачам абсолютно новий рівень комфорту. Але якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання за допомогою надійного криптографічного алгоритму, то замість користі вони принесуть шкоду, надавши кіберзлочинцям можливість для підриву інформаційної безпеки. Оскільки речі із вбудованими комп'ютерами зберігають дуже багато інформації про свого власника, зокрема можуть знати його точне місцезнаходження, доступ до такої інформації відчутно нашкодити користувачу.

Питання безпеки хоч і відноситься до всіх сфер використання інтернету речей, але для деяких все ж не є критичним. Наприклад, збій в роботі онлайн моніторингу громадського транспорту не призведе до фатальних наслідків, а просто зробить день певної кількості користувачів менш комфортним.

Інше діло, коли мова йде про системи типу «розумного» будинку. Датчик диму, електричний дверний замок, дистанційне регулювання температури та освітлення – несанкціоноване втручання в будь-який з цих модулів може нашкодити здоров'ю чи навіть життю користувача. Навіть за допомогою злому доступу до електричного чайника можна спровокувати пожежу в квартирі чи будинку.

Відсутність на даний час стандартів для захисту таких автономних мереж дещо сповільнює впровадження інтернету речей у повсякденне життя.

1.7. Інтерфейс користувача

Є декілька варіантів реалізації інтерфейсу користувача: веб-сайт, мобільний додаток та десктоп додаток.

Розробка мобільних додатків (рис.1.5.) коштує дорого і має сенс тільки у випадку, коли додаток має потенціал на велику аудиторію користувачів на часте, в ідеалі щоденне, використання. Також ціна розробка збільшується через те, що потрібно розробляти та підтримувати додаток відразу для двох популярних мобільних операційних систем: iOS та Android.

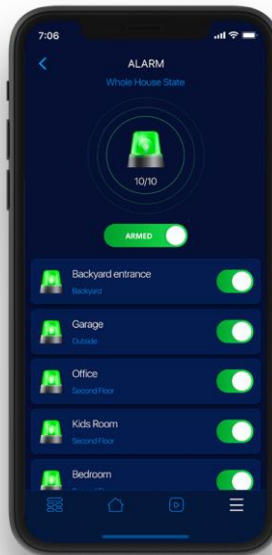


Рис.1.5. Графічний інтерфейс мобільного додатку для керуванням сигналізацією

Десктоп додатки розробляються рідко і потрібні для вузькоспеціалізованих задач. Наприклад, інтерфейс моніторингу стану сільськогосподарських угідь за допомогою зручно надавати саме у вигляді комп'ютерної програми.

Найбільш універсальним варіантом інтерфейсу користувача є веб-сайт. Веб-сайти такого типу відносно не дорогі в розробці. Це може бути всього одна веб-сторінка, на якій реалізовано весь необхідний функціонал. Хорошим прикладом такої реалізації є веб-сайт eway.in.ua, де, не виходячи за межі головної сторінки, можна зручно відслідкувати потрібний автобус.

Великою перевагою веб-сайтів є їхня крос-платформність, що дає змогу користувачам всіх популярних операційних систем отримати доступ до інтерфейсу, навіть користувачам Windows, Linux та macOS.

Тому інтерфейс користувача у вигляді веб-сайту – оптимальний варіант для системи онлайн моніторингу громадського транспорту в місті Ірпінь.

1.8. Огляд аналогів

Найкращим прикладом успішної реалізації моніторингу громадського транспорту є сервіс EasyWay (рис.1.6.), який був коротко описаний вище.

По-перше, сервіс представлений у вигляді веб-сайту. По-друге, це український сервіс, який враховує реалії нашої держави. По-третє, веб-сайт успішно працює у Львові та Києві протягом вже майже десяти років.

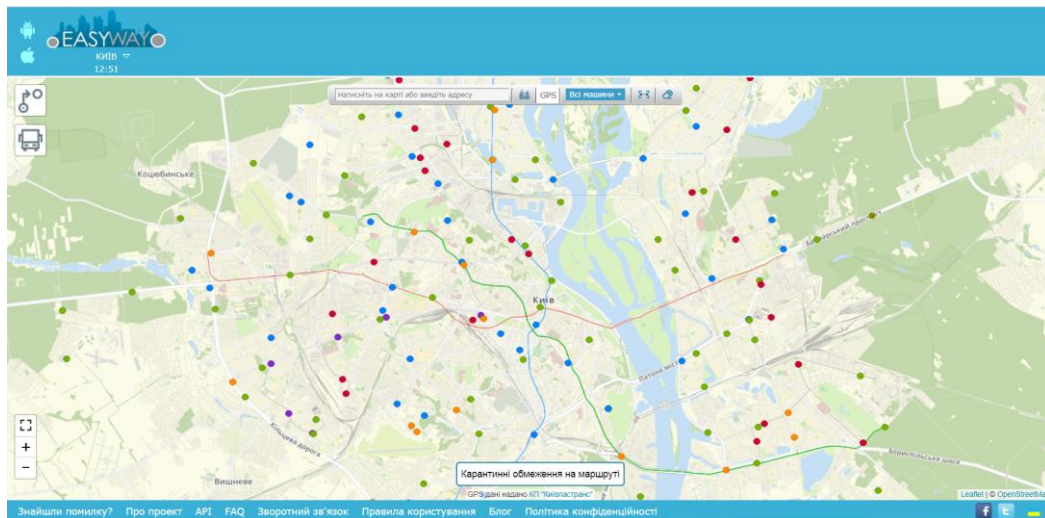


Рис.1.6. Головна сторінка веб-сайту eway.in.ua

EasyWay відслідковує положення автобусів за допомогою технології GPS, а точніше за допомогою GPS-трекерів, які установлені в кожному автобусі. Ключовою перевагою GPS-трекерів є їхнє низьке електроспоживання. Хоча точність визначення місцезнаходження не дуже висока (похибка може складати декілька метрів), але в даному випадку це не є проблемою, так як користувача цікавить якраз приблизне розташування автобуса і приблизний час прибуття.

Отже, система онлайн моніторингу громадського транспорту міста Ірпінь буде реалізована за аналогією до сервісу EasyWay, з поправкою на обмежені ресурси розробника-студента та обмежений об'єм дипломного проекту бакалавра.

1.9. Вимоги до проектування програмного забезпечення

Перед початком проектування і розробки програмного забезпечення для онлайн моніторингу громадського транспорту потрібно визначити декілька ключових моментів.

По-перше, дана система буде розроблятися у вигляді веб-сайту, тому що, як описано вище, такий формат оптимальний для більшості популярних операційних систем та не вимагає занадто багато ресурсів для розробки.

По-друге, веб-сайт буде розроблятися на мові програмування JavaScript з використанням фреймворку React JavaScript (рис.1.7.). Даний фреймворк дозволяє розробляти адаптивні веб-сайти, що відповідає вже описаним вимогам.

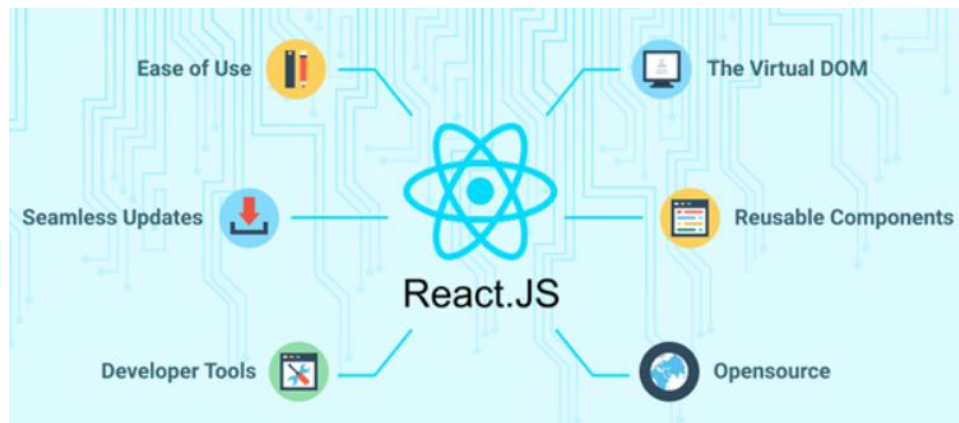


Рис.1.7. Структура React JavaScript

По-третє, система онлайн моніторингу громадського транспорту буде представляти собою прототип, який буде демонструвати основні можливості такого сервісу. В першу чергу, це інтерфейс користувача, який має бути зручним, зрозумілим та стабільно працюючим.

Також важливо визначити об'єм роботи та кількість ресурсів які вже були та які ще будуть витрачені, що дуже важливо для проектування. Тому прототип системи онлайн моніторингу не буде надлишковим, буде реалізовано тільки необхідний функціонал.

Після проектування, на ранніх етапах розробки, командою розробників (в даному випадку одним розробником) приймається рішення щодо вигляду демонстраційної версії продукту. Існує, як мінімум, два варіанти реалізації такої версії: прототип та MVP.

Прототип – це суто технічний продукт, який потрібен розробнику чи розробникам для того, щоб зрозуміти правильність вибраного вектору розробки, протестувати деякі гіпотези та, можливо, показати його стороннім

розробникам. Це хороший варіант, якщо необхідно показати свій продукт на ранніх етапах розробки іншим технічним спеціалістам.

MVP (англ. minimum viable product) – мінімально життєздатний продукт, який має дуже обмежений, але достатній для перших користувачів, функціонал (рис.1.8.). MVP зазвичай розробляється з ціллю тестування нових технологій, залучення аудиторії та перевірки гіпотез. Тому MVP вважається маркетинговою версією прототипу, що дуже актуально для тих сфер інформаційних технологій, які потребують великих інвестицій: індустрія відеоігор, мобільних додатків, робототехніка тощо.

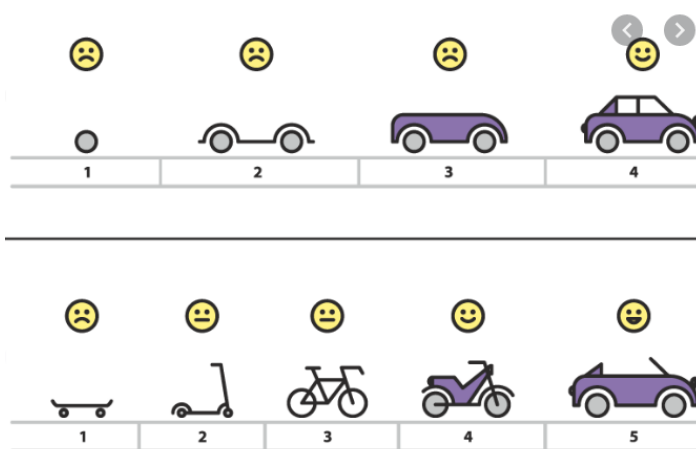


Рис.1.8. Схематичне зображення поняття MVP

В даному випадку вибір однозначно падає формат прототипу, тому що фактичні результати дипломного проекту будуть презентуватися серед технічних спеціалістів.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

2.1. Проектування та вимоги до розробки веб-сайту

Розробка будь-якого програмного продукту починається з розробки концепту та проектування. Хоча дані процеси на перший погляд дуже схожі, але все ж мають низку серйозних відмінностей.

Концепт – це зазвичай інноваційна ідея чи набір ідей, які спочатку формуються в осмислений текстовий чи мультимедійний документ для демонстрації людям, які добре розбираються в даній предметній області та тим чи іншим чином будуть задіяні в подальшій розробці.

У випадку з системою моніторингу громадського транспорту міста Ірпінь, концепт майбутнього продукту представляє собою низку ідей:

- розробка веб-сайту, яким буде зручно користуватися;
- моніторинг не тільки маршрутних таксі, а й служб таксі і навіть приватних перевізників, які користуються сервісом BlaBlaCar;
- можливість переглядати детальну аналітику перевізників.

Наступним кроком є безпосереднє проектування веб-сайту. Найбільш наглядний спосіб проектування на ранніх етапах – це створення UML діаграм, в даному випадку діаграми варіантів використання або Use Case Diagram (рис.2.1.).

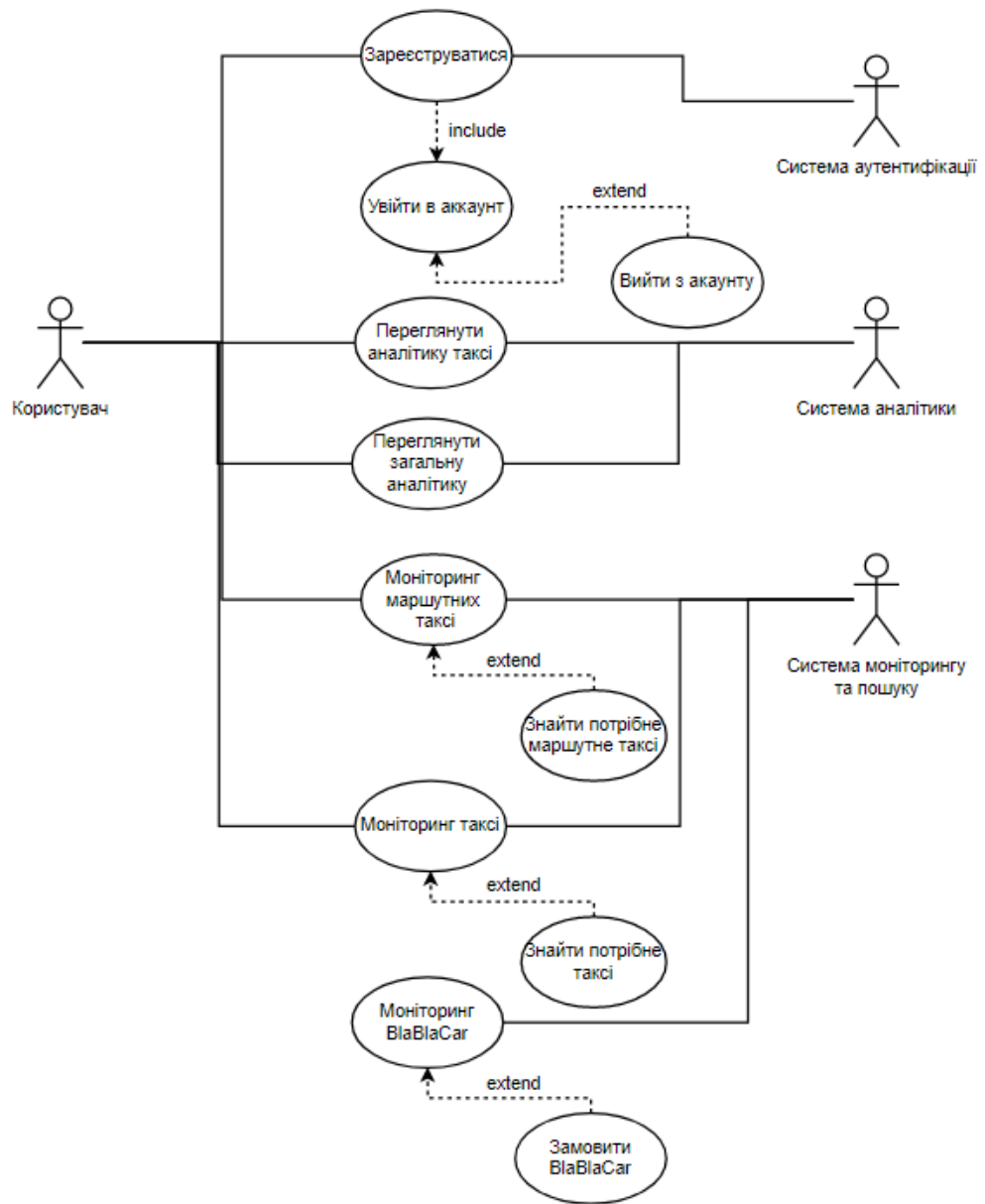


Рис.2.1. Діаграма варіантів використання

Як видно на діаграмі, веб-сайт складається з трьох підсистем: системи аутентифікації, системи аналітики та системи моніторингу й пошуку. Кожна підсистема виконує свої унікальні функції.

Система аутентифікації відповідає за безпосередньо за процес реєстрації користувача, входу в обліковий запис та виходу з нього. Також ця система перевіряє введені логін, пароль та інші данні на відповідність правилам вводу в конкретне поле аутентифікації.

Система аналітики відповідає за збір даних, аналіз та графічне представлення різноманітних статистичних даних, які тим чи іншим чином пов'язані з транспортом. Сюди відноситься загальна аналітика та аналітика

маршрутних таксі. Невелика аналітика приватних перевізників, які користуються супутнім сервісом BlaBlaCar, відображається на сторінці з цими перевізниками.

Відповідно система моніторингу та пошуку відповідає за збір даних щодо транспорту, включаючи в себе найменування транспорту, ціну перевезення, час роботи, маршрут (якщо мова йде про маршрутні таксі) та рейтинг (якщо мова йде про служби таксі). Також дана система дозволяє проводити пошук серед представлених варіантів шляхом використання рядка пошуку.

2.2. Вибір технологій, плагінів, утиліт для розробки

Так як мова йде про розробку веб-програми, це означає, що потрібно підібрати сучасні технології, можливості яких задовольняють вище описаний функціонал та вимоги. Також необхідно обрати платформи, утиліти, плагіни та технології, які спростять та прискорять процес розробки.

JavaScript – мова програмування, яка має декілька основних парадигм. JavaScript підтримує об'єктно-орієнтований, імперативний та функціональний стилі (рис.2.2.).

JavaScript зазвичай використовується як інтегрована мова для програмного доступу до об'єктів веб-додатків. Найчастіше знаходить застосування в браузері як мова сценаріїв для створення інтерактивних веб-сторінок.

Дана мова програмування має динамічну та слабку типізацію, автоматичне керування пам'яттю та реалізацію функцій як об'єктів першого класу.



Рис.2.2. Логотип JavaScript

Webpack – це пакет модулів JavaScript з відкритим кодом. В більшості випадків використовується для JavaScript, але також здатний працювати з зовнішніми ресурсами, такими як HTML та CSS. Webpack приймає на вхід модулі із залежностями та генерує статичні активи, які містять ці модулі (рис.2.3.).

Використовувати Webpack можна за допомогою консолі (командного рядка) або зробити налаштування за допомогою конфігураційного файлу, який має назву `webpack.config.js`. Даний файл визначає правила, список плагінів та інших деталей проекту.

Для роботи з Webpack необхідне використання Node.js.

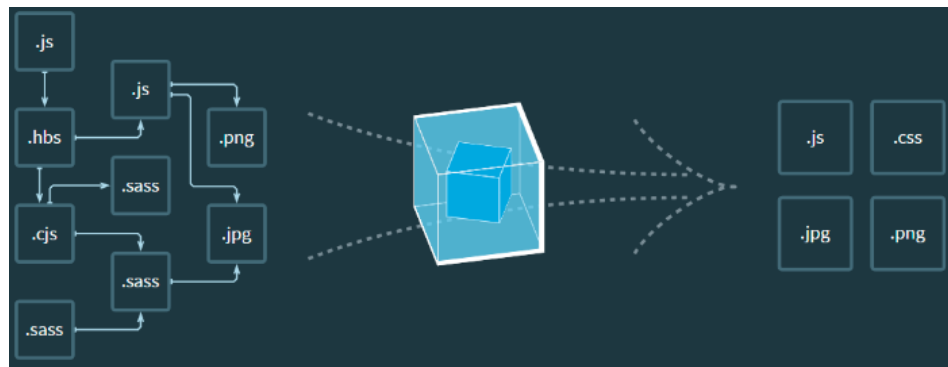


Рис.2.3. Схема роботи Webpack

React – це бібліотека JavaScript, яка створена для розробки інтерфейсів користувача у вигляді дерева невеликих фрагментів, які називаються компонентами. Компонент - це поєднання HTML і JavaScript, який містить в собі всю логіку, яка потрібна для коректного відображення невеликої

частини інтерфейсу користувача. Після створення окремих компонентів, масив компонентів послідовно збирається в цілісну картину.

Redux – це бібліотека JavaScript з відкритим кодом для моніторингу та керування станом програми. В більшості випадків використовується з бібліотеками React та Angular для створення інтерфейсів користувача. Redux має схожий функціонал з React, але використовується для більших проєктів, так як має властивість легко масштабуватися.

Redux – це невелика бібліотека з простим та лаконічним API, який, в тому числі, призначений для контейнерів стану програми (веб-сайту). Redux працює подібно до функції зменшення, що використовується в функціональному програмуванні (рис.2.4.).

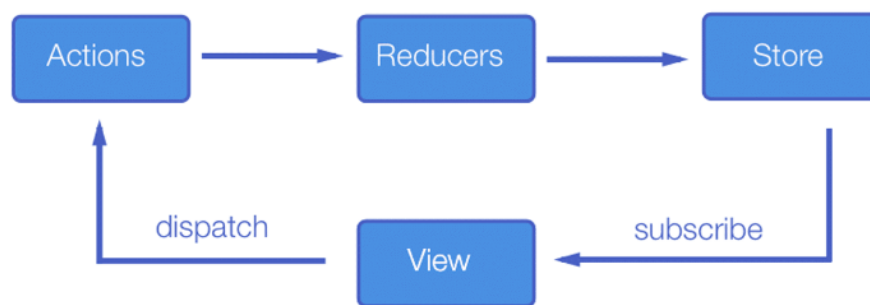


Рис.2.4. Схема роботи Redux

Jest – універсальна платформа для тестування програмного коду будь-якої бібліотеки або фреймворку JavaScript. Jest підтримується Facebook, Inc., розроблена програмістом Крістофом Наказавою з акцентом на простоту та зручність тестування великих веб-програми. Платформа підтримує проєкти, які написані на Babel, TypeScript, Node.js, React, Angular, Vue.js та Svelte. Jest не потребує довгого чи складного налаштування конфігурації, що зручно для людей, які вперше користуються платформою.

Enzyme – це утиліта для тестування конкретно JavaScript React, яка дає змогу просто та швидко протестувати роботу компонентів React. Утиліта дає змогу проводити симуляції, маніпуляції, переходи та різними способами імітувати час роботи програми, враховуючи всі необхідні результати.

API Enzyme інтуїтивно зрозумілий та зручний, схожий на API jQuery, який часто використовується для маніпулювання з DOM.

Immer – це невеликий пакет (бібліотека), який дає можливість працювати зі статичними та динамічними станами коду. Принцип роботи Immer базується на механізмі копіювання тимчасових фрагментів коду для подальшого використання (рис.2.5.).

Основна ідея полягає в тому, що програміст вносить зміни в тимчасовий стан `draftState`, який є проксі-сервером поточного стану `State`. Коли всі зміни й перетворення будуть закінчені, Immer створює наступний стан, базуючись на змінах стану чернетки. Тобто, програміст може взаємодіяти з даними шляхом простих модифікацій, при цьому зберігаючи всі переваги незмінних даних.

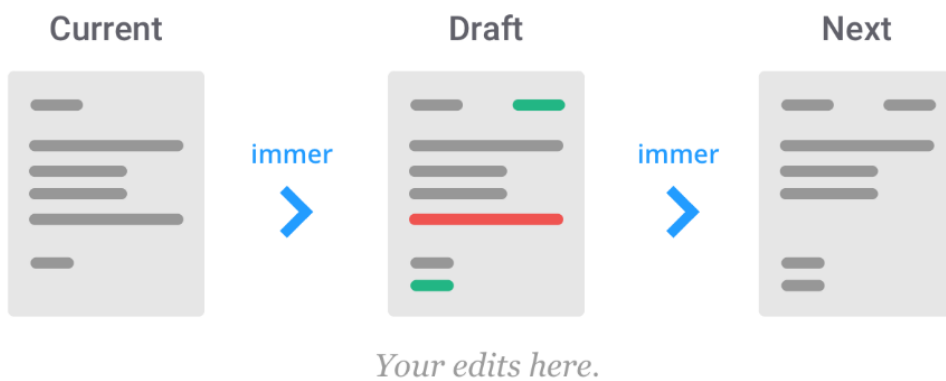


Рис.2.5. Схема роботи Immer

Redux Saga – це бібліотека проміжного програмного забезпечення Redux, основною задачею якої є спрощення та покращення виконання побічних ефектів, таких як асинхронні операції, завантаження даних, робота з доступом до кешу браузера тощо (рис.2.6.). Це досягається за допомогою використання функції ES6, яка ще називається функцією-генератором, що дозволяє писати асинхронний код, який буде перетворений в синхронний і таким чином буде легким для розуміння та тестування. Тобто Redux Saga – це зручний спосіб обробляти довготривалі транзакції та запити з побічними ефектами або потенційними збоями.

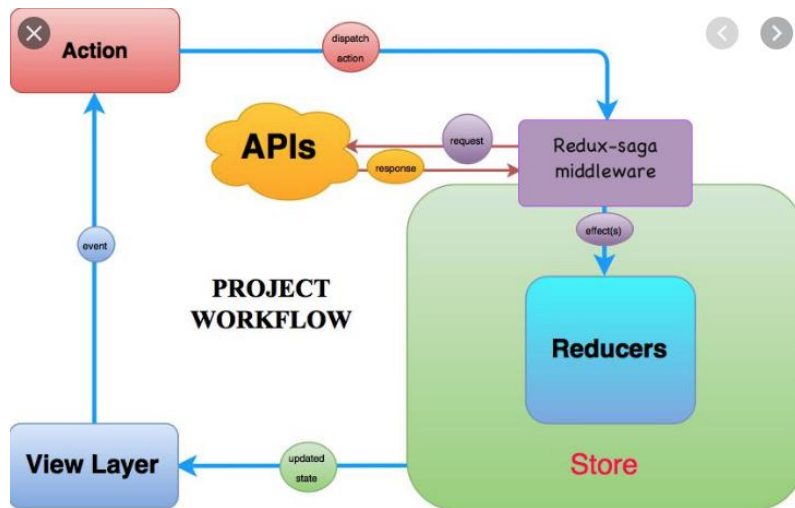


Рис.2.6. Схема роботи Redux Saga

Redux Thunk – це бібліотека проміжного програмного забезпечення, яка дозволяє повертати функції, а не просто дії. Найбільш поширеним випадком використання цієї бібліотеки є обробка дій, які не можуть бути синхронними. Наприклад, використання `axios` для надсилання запиту GET. Наприклад, Redux Thunk дає змогу генератору дій інвертувати керування за рахунок диспетчерських функцій. Генератор отримує відправлення як аргумент і може виконувати цю дію асинхронно (рис.2.7.).

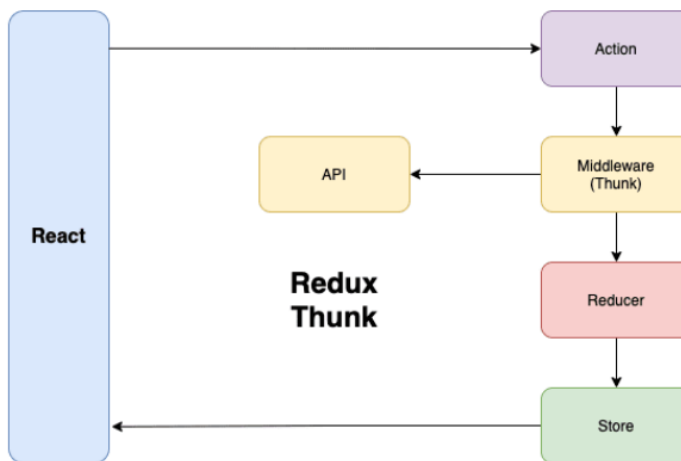


Рис.2.7. Схема роботи Redux Thunk

2.3. Вибір програмного забезпечення

Оптимальним інструментом розробки даного веб-сайту є редактор коду Visual Studio Code (рис.2.8.) і на це є ряд причин:

- безліч налаштувань (як всього редактору, так і графічного інтерфейсу);
- бібліотека доповнень, яка постійно розширюється й оновлюється;

- безліч актуальних готових рішень, які регулярно оновлюються;
- багатфункціональність (редактор підтримує майже всі актуальні мови програмування);
- простота користування та гнучкість налаштування;
- редактор потребує небагато, в порівнянні з аналогами, обчислювальної потужності комп'ютера.

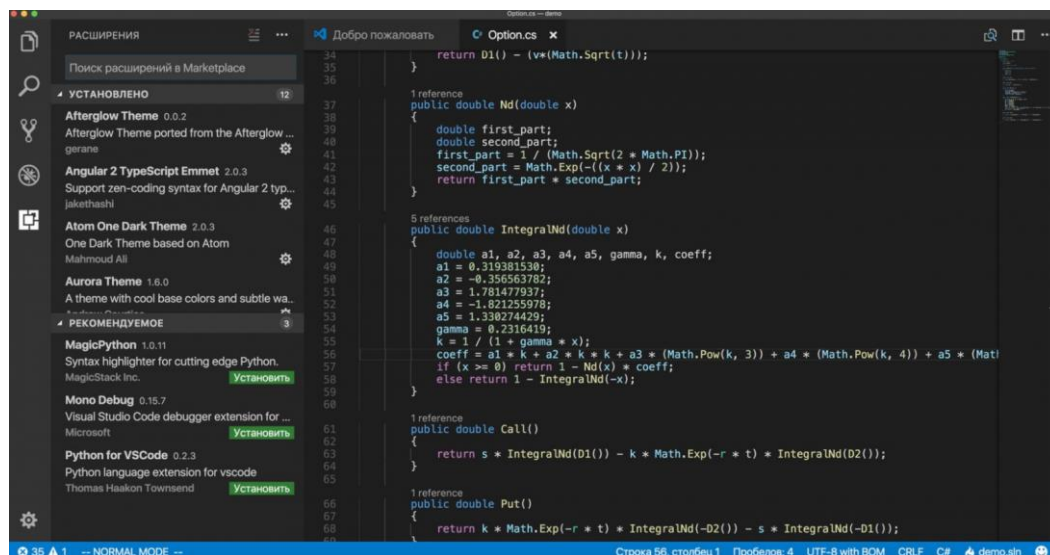


Рис.2.8. Інтерфейс редактора коду Visual Studio Code

Visual Studio Code - це безкоштовний редактор вихідного коду, створений компанією Microsoft для операційних систем Windows, Linux та macOS. Основними особливостями є підтримка індивідуального налаштування редактору, комплексне підсвічування синтаксису, інтелектуальне заповнення коду та фрагментів коду, рефакторинг коду та вбудований Git. Користувачі можуть змінювати тему, комбінації клавіш, налаштування та встановлювати розширення, що підвищує рівень функціональності та комфорту при роботі з редактором.

2.4. Вибір сервісів для тестування

На всіх етапах розробки продукт повинен проходити тестування за різними критеріями і для цього потрібно визначити декілька сервісів для тестування веб-сайтів, які допоможуть виявляти проблеми системи на ранніх етапах та допоможуть проводити контроль якості окремих модулів.

WAVE – це безкоштовний онлайн інструмент оцінки доступності та зручності використання сайту від WebAIM (рис.2.9.). Даний онлайн інструмент видає результати наочно – на сторінці сайту, який перевіряється, з'являються зауваження-блоки різних кольорів (червоний, жовтий, зелений) з вказівками на те, що потрібно виправити. Якщо перейти на вкладку меню «Structure / Order», оформиться схема та структура сайту. Крім того, можна перевірити як сайт виглядає в текстовому режимі, що дуже корисно в деяких окремих випадках.



The image shows the user interface of the WAVE (Web Accessibility Evaluation Tool) website. At the top, there is a dark header with the WAVE logo on the left, which consists of a green circle containing a white wave symbol. To the right of the logo, the text 'WAVE' is written in a large, bold, green font, with 'web accessibility evaluation tool' in a smaller, white font below it. Further right, there is a search bar with the placeholder text 'Web page address...' and a button labeled 'WAVE' to its right. Below the search bar, there is a link that says 'or upload a file'. The main content area has a white background. It starts with the heading 'Welcome to WAVE' in bold. Below this, there is a paragraph of text explaining that WAVE is a free tool provided by WebAIM, used to aid humans in the web accessibility evaluation process. It states that instead of a complex technical report, WAVE shows the original web page with embedded icons and indicators. Below the text, there are three main sections: 1. 'Enter a web site address' with a sub-heading 'Enter the URL of the web site you want to evaluate:' and a text input field followed by a 'WAVE this page!' button. 2. 'Upload a file' with a sub-heading 'If you have files that are not publicly available on the internet, you can upload the files for WAVE evaluation. Simply browse to the file using the form below.' and a 'Choose File' button followed by 'No file chosen' and a 'WAVE this file!' button. 3. 'Check HTML code' with a sub-heading 'Paste HTML code into the text area below.' and a large text input field.

Рис.2.9. Інтерфейс сервісу Wave

W3C CSS Validation Service – досить відомий серед веб-програмістів онлайн сервіс перевірки валідності CSS (рис.2.10.). Можна вводити як посилання на сайт, так і просто завантажувати CSS-код в форму перевірки. В результаті з'явиться звіт з помилками, зауваженнями і поясненнями, якщо тільки CSS-код не є ідеальним на момент перевірки.



Рис.2.10. Інтерфейс сервісу CSS Validation Service

Pingdom Tools – корисний онлайн-сервіс для перевірки швидкості завантаження сайту і його елементів (рис.2.11.). В результаті роботи даний сервіс видає дуже наочний звіт, де видно які елементи сайту перевантажені або створюють проблеми при завантаженні, а сторінка «Page Analsis» дає розгорнуті звіти щодо швидкості відгуку сервера, помилок, відмов тощо.



Рис.2.11. Інтерфейс сервісу Pingdom Tools

РОЗДІЛ 3. РОЗРОБКА ВЕБ-САЙТУ

3.1. Глобальний дизайн інтерфейсу веб-сайту та навігація

Починаючи розробку веб-сайту, потрібно в першу чергу визначатися з глобальним дизайном інтерфейсу та навігацією. Тобто створити приємний, зручний та інтуїтивно зрозумілий графічний інтерфейс, враховуючи актуальні тенденції в сфері дизайну інтерфейсу користувача (англ. UI Design – User Interface Design) та дизайну досвіду користувача (англ. UX Design – User Experience Design).

Слід зазначити, що дана система моніторингу громадського транспорту представлена у вигляді прототипу, який демонструє можливості системи, які потенційно можуть бути реалізовані у випадку залучення більшої кількості ресурсів.

Опираючись на вище спроектовану UML діаграму варіантів використання, можна сформулювати список необхідних сторінок веб-сайту:

- сторінка реєстрації;
- сторінка входу в обліковий запис;
- сторінка моніторингу маршрутних таксі;
- сторінка моніторингу маршрутів;
- сторінка моніторингу таксі;
- сторінка моніторингу перевізників BlaBlaCar;
- сторінка загальної аналітики;
- сторінка аналітики таксі.

Навігація веб-сайту представлена у вигляді невеликого блоку у верхньому лівому куті кожної сторінки і включає в себе гіперпосилання на всі основні сторінки (рис.3.1.).

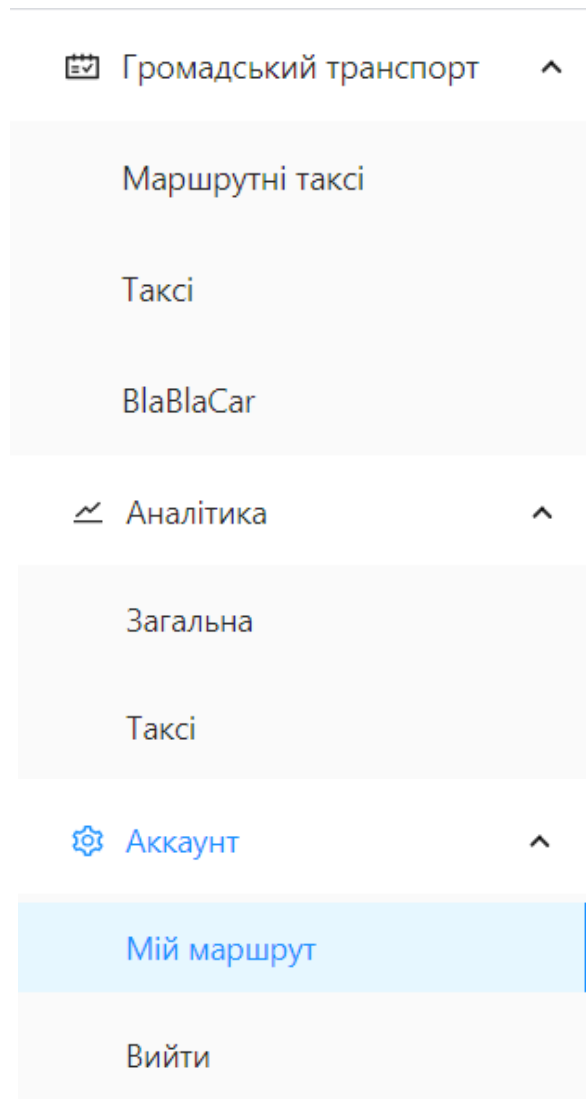


Рис.3.1. Блок навігації

3.2. Розробка інтерфейсу сторінок «Реєстрація» та «Вхід»

Сторінка реєстрації представляє собою блок з чотирьох полів (рис.3.2.):

- Ім'я;
- Імейл (електронна пошта);
- Пароль;
- Повторити пароль.

Реалізована функція приховування паролю на випадок, якщо користувач не хоче, щоб його пароль відображався на екрані, так як це конфіденційна інформація і користувачі часто використовують одні й ті ж самі паролі для різних сервісів.

Після введення всіх даних користувач натискає кнопку «Зареєструватися», реєстрація завершується й відбувається перехід безпосередньо на сторінку моніторингу транспорту.

Реєстрація

* Імя:

* Імейл:

* Пароль:

* Повторити пароль:

Рис.3.2. Сторінка реєстрації

У випадку, якщо користувач залишає одне чи декілька полів незаповненими або вводить невідповідні символи, з'являється попередження (рис.3.3.).

Реєстрація

* Імя:
Імя обов'язкове поле для заповнення

* Імейл:
Імейл обов'язкове поле для заповнення

* Пароль:
Пароль обов'язкове поле для заповнення

* Повторити пароль:
Повторити пароль обов'язкове поле для заповнення

Рис.3.3. Незаповнені поля реєстрації

Схожим чином реалізована сторінка входу (рис.3.4.), яка має всього два поля: «Логін» та «Пароль». Для зручності й покращення досвіду користувача, реалізована функція «Запам'ятати мене», яка дозволяє не вводити логін та пароль при кожному запуску веб-сайту.

* Логін:

* Пароль:

Ще не зареєструвались ?

Запам'ятати мене

Рис.3.4. Сторінка входу

3.3. Розробка інтерфейсу сторінки «Маршрутні таксі»

Інтерфейс сторінки «Маршрутні таксі» представляє собою таблицю з інформацією про маршрутні таксі міста Ірпінь та околиць (рис.3.5).

Таблиця містить наступні стовпці:

- номер маршрутного таксі;
- напрямок;
- ціна;
- робочі дні;
- час роботи.

Цієї базової інформації достатньо для того, щоб користувач міг оцінити різні варіанти того, як дістатися того чи іншого населеного пункту.

Планувалося також додати інформацію про періодичність, з якою кожне маршрутне таксі курсує між зупинками, але для цього потрібні додаткові дані, які не зрозуміло де можна дістати.

Номер	Напрямок	Ціна	Робочі дні	Час роботи
1-Д	Військове містечко – Ірпінь	16 грн	ПН СР ПТ	8:00 - 17:00
2	м. Буча – с. Мироцьке	24 грн	ПН ВТ СР ЧТ ПТ	8:00 - 19:00
3	Ірпінь – Буча	16 грн	ПН ВТ СР ЧТ ПТ НД	5:30 - 23:00
6	ст. Ірпінь – Машторф	8 грн	ПН ВТ СР ЧТ ПТ СБ НД	9:00 - 20:00

Рис.3.5. Сторінка з таблицею маршрутних таксі

3.4. Розробка інтерфейсу сторінки «Таксі»

Враховуючи недосконалість інфраструктури громадського транспорту в місті Ірпінь та недоступність більшості транспорту в нічний час, створений ще один модуль для моніторингу сервісів таксі та, відповідно, окрема сторінка для цього.

Інтерфейс сторінки «Таксі» представляє собою таблицю з інформацією про сервіси таксі міста Ірпінь та околиць (рис.3.6.).

Таблиця містить наступні стовпці:

- назва таксі;
- рейтинг серед користувачів;
- ціна виклику;
- ціна за км шляху;
- контактні дані (номер телефону).

Назва	Рейтинг	Ціна виклику	Ціна за км	Номер телефону
Таксі десятка	9.2	40грн	9грн	+380676566663
Євролайн	6.7	40грн	9грн	+380933388190
Авангард	9.9	30грн	6грн	+380632331848
Deluxe Таксі	3.8	50грн	8грн	+380935003500
Лінкор таксі	6.1	40грн	7грн	+537
Ірпінь таксі	9.1	40грн	7грн	+380634097751
Аврора	5.3	30грн	6грн	+380970412520

Рис.3.6. Сторінка з таблицею сервісів таксі

Сервіси таксі стали дуже популярні у великих містах та містах-супутниках, але не завжди є можливість оперативно дізнатися детальну інформацію про всіх перевізників в конкретному місті. Даний модуль веб-сайту надає таку можливість.

3.5. Розробка інтерфейсу сторінки «BlaBlaCar»

BlaBlaCar – це французький сервіс пошуку автомобільних попутчиків. Тобто водій, який прямує з точки А в точку Б і має вільні місця в своєму автомобілі, може знайти попутчиків онлайн.

Система онлайн моніторингу має інтегрований модуль сервісу BlaBlaCar, який дає можливість швидко переглянути всіх водіїв, які вирушають з Ірпеня в інші міста України. В контексті всього веб-сайту, цей зручна функція, адже можна спланувати свою поїздку не тільки по місту, а й за його межами.

Інтерфейс сторінки складається з блоків, які формуються за стандартизованим шаблоном, що надає веб-сторінці приємний, акуратний вигляд.

Кожен блок містить наступну інформацію про водія (рис.3.7.):

- фото;
- маршрут (в форматі «місто-місто»);
- ім'я;
- марка автомобіля;
- ціна.

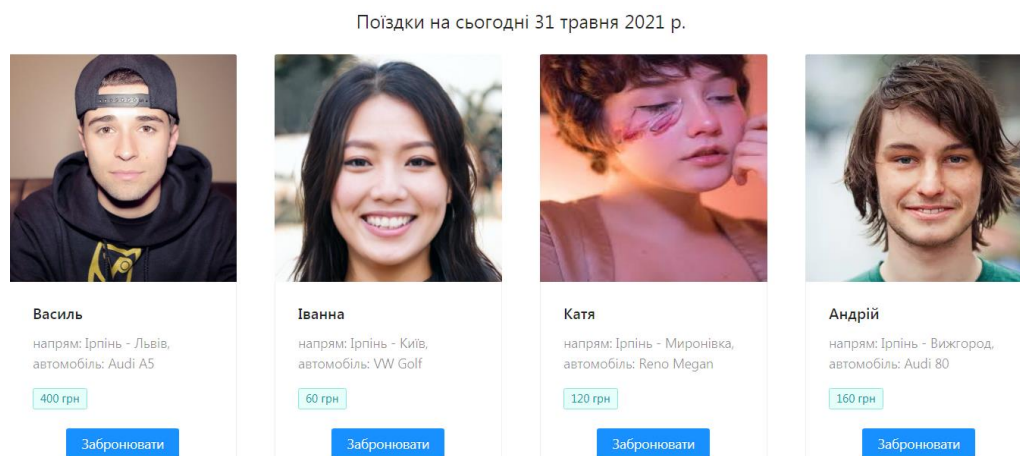


Рис.3.7. Сторінка з водіями BlaBlaCar

3.6. Розробка інтерфейсу сторінки «Мій маршрут»

Зараз сторінка «Мій маршрут» представляє собою карту формату Google Maps, на якій відображається маршрут, побудований від точки А в точку Б (рис.3.8.). Стартова та фінальна точки вказуються у верхній частині сторінки, над картою.

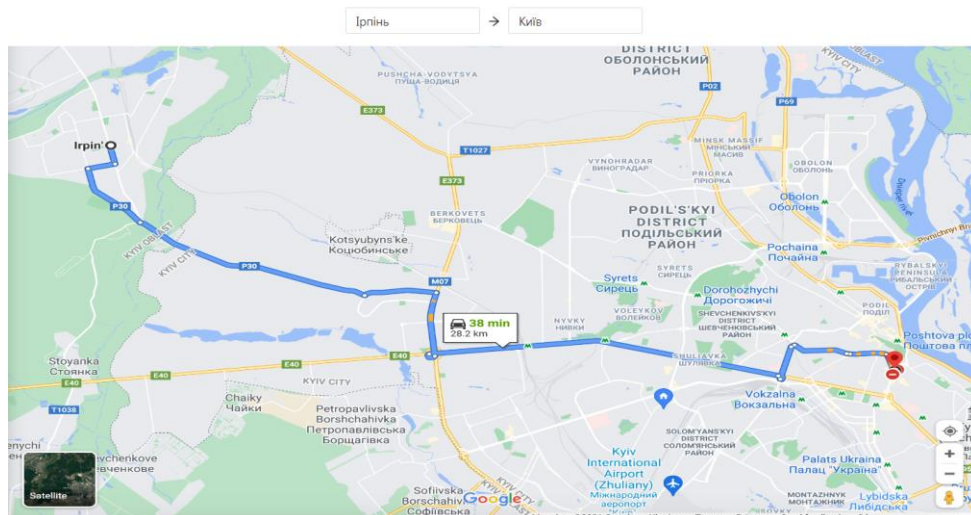


Рис.3.8. Карта з маршрутом

В подальшому, при умові використання спеціального апаратного забезпечення, можлива реалізація унікальної карти, на якій транспортні засоби будуть відображатися в режимі реального часу і можна буде відслідковувати потрібне маршрутне таксі з точністю до декількох метрів.

3.7. Розробка інтерфейсу сторінки «Загальна аналітика»

Інтерфейс сторінки «Загальна аналітика» представляє собою дві діаграми, які відображають потенційно цікаву для користувача інформацію щодо транспорту.

Перша діаграма показує співвідношення використання транспорту між маршрутними таксі, сервісами таксі та приватними перевізниками BlaBlaCar (рис.3.9.).

Співвідношення використання транспорту

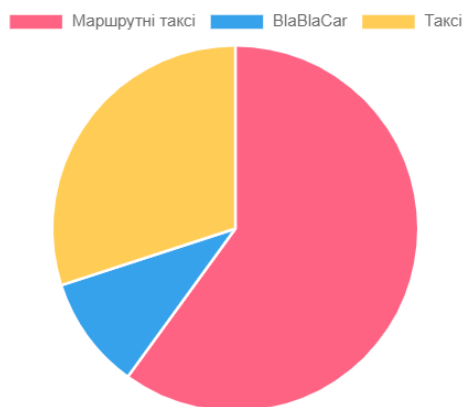


Рис.3.9. Діаграма співвідношення використання видів транспорту

Друга діаграма демонструє порівняння швидкості пересування різними видами транспорту, включаючи персональний альтернативний транспорт, типу велосипеда та електросамоката (рис.3.10.).

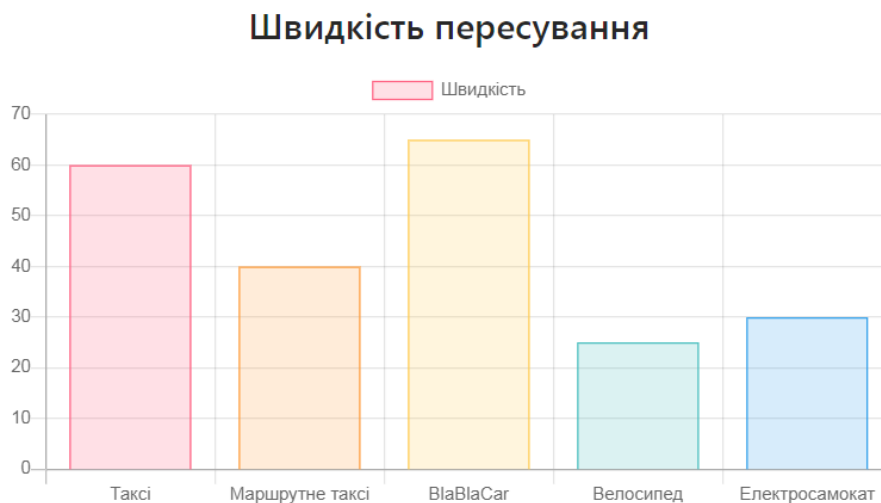


Рис.3.10. Діаграма співвідношення швидкості пересування різними видами транспорту

3.8. Розробка інтерфейсу сторінки «Аналітика таксі»

Інтерфейс сторінки «Аналітика таксі» представляє собою дві діаграми, які відображають потенційно цікаву для користувача інформацію щодо доступних в регіоні сервісів таксі.

Перша діаграма відображає порівняльну характеристику рейтингів кожного сервісу таксі в період з 2015 по 2021 роки (рис.3.11.). Така ретроспективна діаграма дозволяє зрозуміти який сервіс має найкращі відгуки серед користувачів на даний момент, детальніше проаналізувати середню оцінку протягом всього проміжку часу та, в результаті, обрати найбільш оптимальний варіант.

Порівняльна характеристика лідерів серед таксі

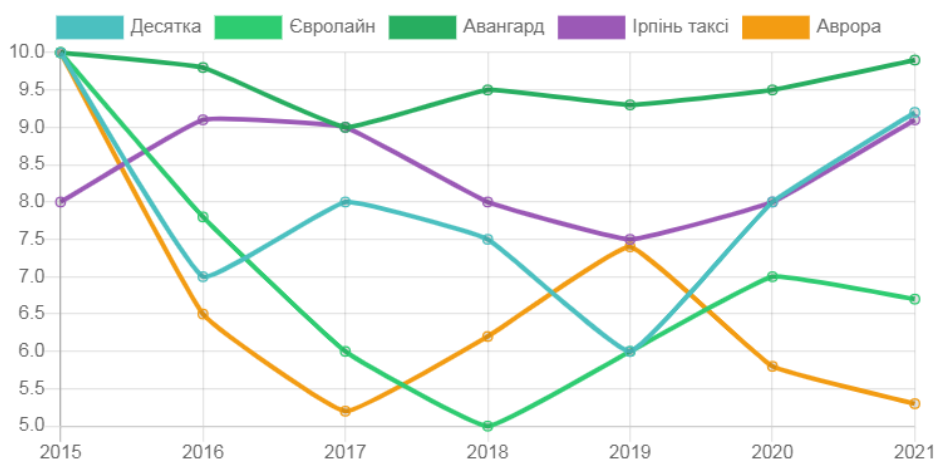


Рис.3.11. Порівняльна характеристика сервісів таксі за рейтингом

Друга діаграма відображає порівняльну характеристику кількості відгуків серед п'яти найкращих сервісів (рис.3.12.). Ця п'ятірка формується на основі результатів першої діаграми. Дуже часто саме кількість відгуків формує враження потенційного користувача про сервіс: чим більше відгуків, тим більша ймовірність якісного обслуговування.

Кількість відгуків

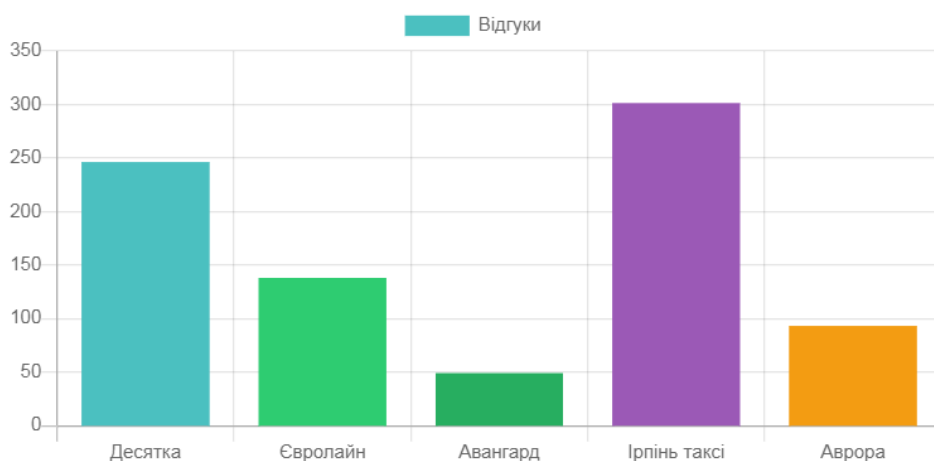


Рис.3.12. Порівняльна характеристика кількості відгуків сервісів таксі

Комбінація даних двох діаграм дає можливість отримати потенційному користувачу найбільш повну та вичерпну інформацію щодо сервісів таксі і обрати оптимальний для себе варіант.

3.9. Розробка функціоналу сторінок «Реєстрація» та «Вхід»

Сторінка реєстрації – обов’язкова частина більшості веб-сайтів і вона має свої певні стандарти. Звичайно, ці стандарти реалізовані в даному веб-сайті (рис.3.13.). По-перше, це перевірка валідності введених даних в поля форми реєстрації. Тобто поле в будь-якому випадку не повинно бути пустим та повинно містити більше заданою в коді кількості символів. Також введена електронна пошта обов’язково має містити символ «@». Поля, обов’язкові для заповнення, помічені спеціальними позначками.

```
import styles from './Registration.module.sass';

const validateMessages = {
  required: "${label} обов'язкове поле для заповнення",
  types: {
    email: 'Не валідний імейл',
  },
  string: {
    range: '${label} повинен бути мінімум ${min} символів',
  },
};
```

Рис.3.13. Програмування сторінки реєстрації

Сторінка входу в обліковий запис містить також функцію «Запам’ятати мене» та зручне гіперпосилання на сторінку реєстрації – для випадків, коли користувач потрапив на сторінку входу до того, як зареєстрував свій обліковий запис. Після успішного входу користувач потрапляє на сторінку моніторингу маршрутних таксі, так як це основна й прогнозовано найбільш популярна сторінка веб-сайту.

У випадку неуспішної автентифікації користувача сторінка перезавантажується та вказує на те, що логін чи пароль введені невірно (рис.3.14.).

```

const Login = (props) => {
  const history = useHistory();

  const onFinish = (values) => {
    console.log('Success:', values);
    history.push('/traffic/buses');
  };

  const onFinishFailed = (errorInfo) => {
    console.log('Failed:', errorInfo);
  };

  const onNavigateToRegistration = () => {
    history.push('/registration');
  };
};

```

Рис.3.14. Програмування сторінки входу

Після успішного входу в обліковий запис, користувач має можливість вийти за допомогою кнопки, яка знаходиться в меню навігації, щоб зайти в інший обліковий запис, якщо це необхідно.

3.10. Розробка функціоналу сторінки «Маршрутні таксі»

Крім вище описаних можливостей інтерфейсу, сторінка «Маршрутні таксі» також пропонує зручний пошук, який реалізований за принципом пошукових систем (рис.3.15.). Тобто користувач може ввести відповідне ключове слово, наприклад, назву населеного пункту і побачити відповідні маршрутні таксі, якщо такі існують.

```

return (
  <div className={styles.container}>
    <div className={styles.search_container}>
      <Search
        placeholder="Введіть ваш пункт призначення"
        allowClear
        size="large"
        enterButton="Шукати"
        onSearch={onSearch}
      />
    </div>
    <Table columns={columns} dataSource={busesDataSet.filter((bus) => bus.isShown)} />
  </div>
);
};
export default Buses;

```

Рис.3.15. Програмування пошуку маршрутних таксі

У випадку, якщо ключове слово не було знайдене в масиві інформації, якою володіє система, буде показано відповідний напис замість рядків таблиці (рис.3.16.).

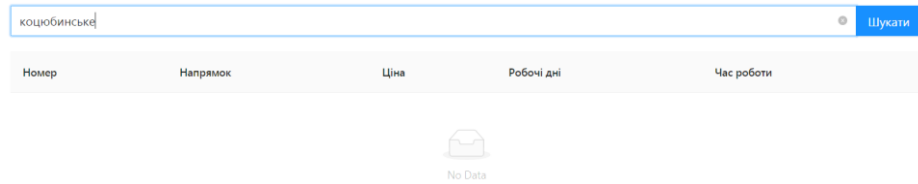


Рис.3.16. Блок пошуку маршрутних таксі

3.11. Розробка функціоналу сторінки «Таксі»

Аналогічно зі сторінкою «Маршрутні таксі» реалізовано пошук сервісів таксі. Тільки в даному випадку пошук за ключовими словами відбувається виключно серед назв сервісів.

Для зручності користувача, плашки, на яких вказаний рейтинг того чи іншого сервісу, мають градацію за кольором (рис.3.17.). Сервіс з рейтингом менше чотирьох балів має плашку червоного кольору; якщо рейтинг від чотирьох до семи, то компанія отримує плашку жовтого кольору, і якщо більше семи – зеленого.

```
{tags.map((tag) => {  
  let color = 'green';  
  
  if (tag > 0 && tag < 4) {  
    color = 'red';  
  } else if (tag > 4 && tag < 7) {  
    color = 'yellow';  
  } else {  
    color = 'green';  
  }  
})
```

Рис.3.17. Програмування відображення рейтингу сервісів таксі

3.12. Розробка функціоналу сторінки «VlaVlaCar»

Як описано вище, сторінка «VlaVlaCar» відображає список доступних водіїв, які користуються цим сервісом. Поїзду з будь-яким водієм можна забронювати, після чого кнопка «Забронювати» зміниться на кнопку «Заброньовано» і стане неактивною. Також після бронювання з'явиться плашка «Лист із відправленням відправлено на вашу пошту», яка зникне через 5 секунд (рис.3.18.).

```

const handleBookDriver = (id) => {
  const updatedDrives = drivers.map((driver) => {
    if (id === driver.id) {
      message.success('Лист із підтвердженням відправлено на вашу пошту');
      return {
        ...driver,
        isBooked: true,
      };
    }
    return driver;
  });
  setDrivers(updatedDrives);
};

```

Рис.3.18. Програмування функції бронювання водіїв BlaBlaCar

3.13. Розробка функціоналу сторінки «Мій маршрут»

Сторінка «Мій маршрут» зараз працює в демонстраційному режимі (рис.3.19.), так як для нормальної роботи такого складного модуля потрібно, по-перше, створити складну карту, на якій будуть вказані всі можливі маршрути, будуть враховані дорожні знаки й правила дорожнього руху. По-друге, при подальшій розробці планується реалізація функції відслідковування потрібного транспорту в режимі реального часу, для чого необхідно відповідне апаратне й програмне забезпечення та залучення додаткових ресурсів.

```

function Map(props) {
  return (
    <div className={styles.container}>
      <div className={styles.input_container}>
        <Input value='Ірпінь' />
        <div className={styles.icon_container}>
          <ArrowRightOutlined />
        </div>
        <Input value='Київ' />
      </div>
      <img
        className={styles.map_container}
        src={map}
      />
    </div>
  );
}

```

Рис.3.19. Програмування сторінки «Мій маршрут»

Тому зараз дана сторінка візуально показує, яким чином даний модуль буде працювати при виконанні всіх вище перерахованих умов.

3.14. Розробка функціоналу сторінки «Загальна аналітика»

Сторінка загальної аналітики має дві діаграми. Обидві діаграми мають унікальну анімацію появи, що відповідає сучасним тенденціям розробки веб-сайтів. Компоненти для діаграм та анімації появи реалізовані за допомогою спеціальної бібліотеки Turography, яка значно економить ресурси під час розробки діаграм такого типу (рис.3.20.).

```
import React, { useRef, useEffect } from 'react';
import { Typography } from 'antd';
import Chart from 'chart.js';
import styles from './OverallAnalytics.module.sass';

const { Title } = Typography;

const OverallAnalytics = (props) => {
  const pieChartRef = useRef();
  const barChartRef = useRef();
```

Рис.3.20. Програмування сторінки загальної аналітики

На даний момент сторінка загальної аналітики працює із завчасно заданою інформацією, але легко може бути переформатована для парсингу інформації з інших джерел, наприклад, сторонніх веб-сайтів, таблиць Microsoft Excel тощо.

3.15. Розробка функціоналу сторінки «Аналітика таксі»

Сторінка аналітики сервісів таксі має дві діаграми. Обидві діаграми мають унікальну анімацію появи, що відповідає сучасним тенденціям розробки веб-сайтів. Компоненти для діаграм та анімації появи реалізовані за допомогою спеціальної бібліотеки Turography, яка значно економить ресурси під час розробки діаграм такого типу (рис.3.21.).

```
useEffect(() => {
  const customPieChartRef = pieChartRef.current.getContext('2d');
  const customBarChartRef = barChartRef.current.getContext('2d');

  new Chart(customPieChartRef, {
    type: 'line',
    data: {
      labels: ['2015', '2016', '2017', '2018', '2019', '2020', '2021'],
```

Рис.3.21. Розробка сторінки аналітики сервісів таксі

На даний момент сторінка аналітики сервісів таксі працює із завчасно заданою інформацією, але легко може бути переформатована для парсингу

інформації з інших джерел, наприклад, сторонніх веб-сайтів, таблиць Microsoft Excel тощо.

3.16. Тестування системи

Тестування сервісом W3C markup validation service показало дуже хороші результати, оцінка продуктивності 94, час завантаження всього 500 мс (рис.3.22.)

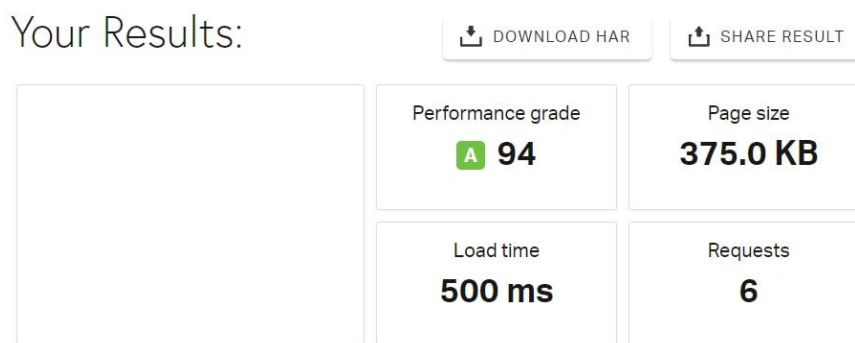


Рис.3.22. Загальні результати тестування сервісом W3C markup validation service

Проблеми виявилися тільки в заголовках терміну дії, тут оцінка всього 56 (рис.3.23.). Заголовки терміну дії повідомляють браузеру, чи потрібно робити запит щодо певного ресурсу в джерела веб-сайту, чи його можна отримати даний ресурс з кешу браузера.

Improve page performance

GRADE	SUGGESTION
E 56	Add Expires headers
A 100	Avoid empty src or href
A 100	Put JavaScript at bottom
A 100	Reduce the number of DOM elements
A 100	Make favicon small and cacheable
A 100	Avoid HTTP 404 (Not Found) error
A 100	Avoid URL redirects

Рис.3.23. Детальні результати тестування сервісом W3C markup validation service

Тестування сервісом WAVE не показало ніяких проблем за вказаними характеристиками тестування (рис.3.24.)

Showing results for https://maximys963.github.io/PTC_Demo/

Checker Input

Show source outline image report

Check by

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was utf-8.
Total execution time 57 milliseconds.

Рис.3.24. Результати тестування сервісом WAVE

Так як сервіс Pingdom проводить дещо більш комплексне тестування, ніж попередні сервіси, було виявлено ряд некритичних проблем (рис.3.25.). Серед них проблеми дуже низького контрасту в таблицях, недосконалої системи заголовків, наявності надлишкових компонентів та неструктурованого коду.

The screenshot displays the results of an accessibility audit. On the left, a summary panel shows: 0 Errors, 1 Contrast Error, 5 Alerts, 3 Features, 0 Structural Elements, and 5 ARIA issues. The main content area lists the following findings:

- 1 Contrast Error:** 1 X Very low contrast.
- 5 Alerts:**
 - 1 X No heading structure
 - 1 X No page regions
 - 1 X Noscript element
 - 2 X Redundant title text
- 5 ARIA:**
 - 2 X Form label
 - 1 X Language
 - 1 X ARIA
 - 1 X ARIA label
 - 1 X ARIA tabindex
 - 1 X ARIA button
 - 1 X ARIA hidden

Рис.3.25. Результати тестування сервісом Pingdom

Всі вище перераховані знайдені в процесі тестування баги не є критичними, тому можна вважати, що веб-сайт успішно пройшов тестування і готовий до роботи.

ВИСНОВКИ

Результатом виконання дипломного проекту є демонстраційний веб-сайт системи онлайн моніторингу громадського транспорту міста Ірпінь.

У першому розділі досліджено предметну область та сфери використання технології інтернету речей, а саме сфери логістики, транспорту, аграрної промисловості, урбаністики.

У другому розділі описано безпосередній процес проектування системи онлайн моніторингу громадського транспорту, вимоги до розробки, вибір технологій, плагінів, утилів, сервісів тестування та вибір програмного забезпечення.

У третьому розділі відображені етапи розробки даного веб-сайту, починаючи зі глобального дизайну інтерфейсу користувача і закінчуючи веб-сторінками аналітики доступного транспорту.

Завдяки оптимальному проектуванню, правильному формуванню вимог до розробки та менеджменту ресурсів, необхідних для отримання якісного та цільного продукту, спроектовано та розроблено демонстраційну версію веб-сайту системи онлайн моніторингу громадського транспорту, що повністю відповідає сучасним вимогам, описаним характеристикам та має відповідний функціонал. Даний веб-сайт стабільно працює та надає корисну інформацію для користувачів.

Система онлайн моніторингу громадського транспорту міста Ірпінь є перспективним, амбіційним проектом, який має велике майбутнє у випадку залучення достатніх ресурсів для подальшої розробки та розвитку.