

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти бакалавр

на тему: **«Проектування вебдодатку для сервісу пошуку квитків і обробки
замовлень користувачів»**

Виконав: здобувач вищої освіти
за освітньою програмою
Інформаційні управляючі системи
спеціальності 126 Інформаційні
системи та технології
ступеня вищої освіти бакалавр
групи 126ІСТ_бд_2022
Єщенко Єгор Анатолійович
Керівник: Копішинська Олена
Петрівна
Рецензент: Яхін Сергій Валерійович

Полтава – 2026 року

ПОЛТАВСЬКИЙ ДЕРЖАВНИЙ АГРАРНИЙ УНІВЕРСИТЕТ
Навчально-науковий інститут економіки, управління, права та
інформаційних технологій
Кафедра інформаційних систем та технологій

Освітня програма Інформаційні управляючі системи
Спеціальність 126 Інформаційні системи та технології
Рівень вищої освіти перший (бакалаврський)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Юрій УТКІН

«10» червня 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Єщенка Єгора Анатолійовича

1. Тема кваліфікаційної роботи:
«Проектування вебдодатку для сервісу пошуку квитків і обробки замовлень користувачів»,
Керівник роботи: к. ф.-м. н., доцент, професор кафедри інформаційних систем та технологій Копішинська Олена Петрівна.
Затверджено наказом закладу вищої освіти від «10» квітня 2026 року № 383-ст
2. Строк подання здобувачем вищої освіти роботи «08» червня 2026 року.
3. Вихідні дані до роботи: інформаційні ресурси бібліотек JavaScript React <https://react.dev/>, редактор вихідного коду <https://code.visualstudio.com/>, <https://www.figma.com/design/>, статистичні дані <https://mate.academy/>, <https://dou.ua/>
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
Розділ 1. Аналіз предметної області та сучасних вебтехнологій для організації онлайн-бронювання квитків
Розділ 2. Проектування та розробка вебдодатку для пошуку квитків і обробки замовлень
Розділ 3. Практичні результати програмної реалізації функцій розробленого вебдодатку
5. Перелік графічного матеріалу: схеми, рисунки, діаграми за темою та об'єктом дослідження

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання отримав
Оцінювання економічної ефективності результатів дослідження	Дивнич О. Д., к. е. н., доцент кафедри економіки та публічного управління	01.04.2026 р.	29.05.2026 р.

7. Дата видачі завдання «10» червня 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Вибір і затвердження теми роботи	02.06.2025 р. – 04.06.2025	
2	Складання і затвердження розгорнутого плану та завдання на кваліфікаційну роботу	05.06.2025 р. – 10.06.2025 р.	
3	Опрацювання джерел інформації	11.06.2025 р. – 15.06.2025 р.	
4	Збір, вивчення і обробка інформації, необхідної для виконання роботи	16.06.2025 р. – 20.06.2025 р.	
5	Виконання теоретико-методологічного розділу роботи	08.09.2025 р. – 01.11.2025 р.	
6	Виконання дослідницько-аналітичного розділу роботи	19.01.2026 р. – 14.02.2026 р.	
7	Виконання проєктно-рекомендаційного розділу роботи	16.02.2026 р. – 31.03.2026 р.	
8	Оцінювання економічної ефективності результатів дослідження	01.04.2026 р. – 29.05.2026 р.	
9	Оформлення тексту роботи	01.06.2026 р. – 06.06.2026р.	
10	Попередній захист роботи на кафедрі	08.06.2026 р.	
11	Доопрацювання роботи з урахуванням зауважень і пропозицій	09.06.2026 р. – 10.06.2026 р.	
12	Нормоконтроль	11.06.2026 р. – 15.06.2026 р.	
13	Захист кваліфікаційної роботи	16.06.2026 р. - 19.06.2026 р.	

Здобувач вищої освіти

Єгор ЄЩЕНКО

Керівник роботи

Олена КОПШИНСЬКА

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАК	6
ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ ВЕБТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ОНЛАЙН-БРОНЮВАННЯ КВИТКІВ	10
1.1 Огляд сучасних технологій і засобів розробки вебдодатків.....	10
1.2 Особливості організації пасажирських перевезень та онлайн-продажу квитків.....	15
1.3 Аналіз існуючих сервісів пошуку та бронювання квитків	18
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ПОШУКУ КВИТКІВ І ОБРОБКИ ЗАМОВЛЕНЬ	23
2.1 Проєктування структури та функціональних модулів вебдодатку.....	23
2.2 Розробка дизайну та адаптивного користувацького інтерфейсу	26
2.3 Обґрунтування технологій програмної реалізації вебдодатку засобами HTML5, CSS3 та JavaScript.....	30
РОЗДІЛ 3 ПРАКТИЧНІ РЕЗУЛЬТАТИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ФУНКЦІЙ РОЗРОБЛЕНОГО ВЕБДОДАТКУ	38
3.1 Перевірка працездатності та тестування функціональних можливостей системи.....	38
3.2 Архітектурні паттерни та модулі клієнтської частини на JavaScript.....	42
3.3 Перевірка та опис економічної ефективності розробки.....	47
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАК

SEO (Search Engine Optimization) – оптимізація сайту під запити до пошукових систем

IDE (Integrated development environment) – Інтегроване середовище розробки

JS – JavaScript, мова програмування

CSS (Cascading Style Sheets) – Каскадні таблиця стилів

HTML (HyperText Markup Language) – мова гіпертекстової розмітки

UX (User Experience) – враження користувача від використання продукту

ВСТУП

Актуальність теми кваліфікаційної роботи. В умовах цифровізації транспортної галузі та стрімкого розвитку електронних сервісів особливої актуальності набувають веборієнтовані системи пошуку, бронювання та придбання квитків на пасажирські перевезення. Зростання мобільності населення, розширення міжнародних транспортних маршрутів та потреба в оперативному доступі до інформації про рейси зумовлюють необхідність створення зручних і функціональних вебдодатків, які дозволяють користувачам швидко знаходити необхідні маршрути, порівнювати доступні варіанти перевезень та оформлювати замовлення в режимі онлайн.

Особливого значення набуває автоматизація процесів обробки замовлень у туристичних агентствах та транспортних компаніях, оскільки використання сучасних вебтехнологій сприяє зменшенню навантаження на персонал, підвищенню якості обслуговування клієнтів та скороченню часу виконання операцій. Водночас значна кількість існуючих рішень характеризується надмірною складністю, недостатньою адаптивністю або обмеженими можливостями інтеграції з бізнес-процесами підприємств. Створюються нові транспортні компанії, які віддають перевагу власним брендам і намагаються використовувати власний сервіс.

У зв'язку з цим проєктування вебдодатку для сервісу пошуку квитків і обробки замовлень користувачів є актуальним завданням, спрямованим на підвищення ефективності роботи транспортно-туристичних компаній та покращення якості надання послуг пасажирам.

Метою кваліфікаційної роботи є проєктування та розробка вебдодатку для пошуку автобусних маршрутів, бронювання квитків та автоматизованої обробки замовлень користувачів із використанням сучасних вебтехнологій.

Завдання кваліфікаційної роботи:

- провести аналіз предметної області та особливостей діяльності туристичних агентств, що здійснюють пасажирські перевезення;

- дослідити існуючі онлайн-сервіси пошуку та бронювання транспортних квитків, визначити їх переваги та недоліки;
- розробити вимоги до функціональності вебдодатку та сформувати його структуру;
- спроектувати інтерфейс користувача з урахуванням принципів зручності та адаптивності;
- розробити програмну реалізацію вебдодатку засобами HTML, CSS та JavaScript;
- реалізувати механізми пошуку маршрутів, бронювання квитків та оформлення замовлень;
- провести тестування створеного вебдодатку та оцінити його працездатність;
- виконати економічне обґрунтування доцільності розробки програмного продукту.

Об'єктом дослідження є процеси пошуку, бронювання та оформлення замовлень на пасажирські перевезення через веборієнтовані інформаційні системи.

Предметом дослідження є методи, технології та програмні засоби проєктування і реалізації вебдодатків для автоматизації пошуку квитків та обробки замовлень користувачів.

Методи дослідження: під час виконання роботи використовувалися методи системного аналізу предметної області, методи вебпроєктування та моделювання інформаційних систем, методи структурного проєктування інтерфейсів користувача, технології клієнтської веброзробки HTML5, CSS3 та JavaScript, а також методи функціонального та користувацького тестування програмного забезпечення..

Практична значущість полягає у створенні працездатного вебдодатку для туристичного агентства, який забезпечує автоматизацію процесів пошуку автобусних рейсів, бронювання квитків та оформлення замовлень користувачів через мережу Інтернет. Розроблене рішення може бути

використане транспортними та туристичними компаніями для розширення каналів надання послуг, підвищення ефективності взаємодії з клієнтами та зменшення витрат часу на обробку заявок. Запропоновані технічні рішення можуть бути адаптовані для впровадження в інших інформаційних системах сфери пасажирських перевезень.

Структура і обсяг кваліфікаційної роботи: пояснювальна записка складається із вступу, трьох розділів основної частини, висновків, списку використаних джерел та додатків. Вона викладена на 50 сторінках формату А4, містить 22 рисунки, 4 таблиці та додатки. У роботі використано 32 науково-технічних інформаційних джерела.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СУЧАСНИХ ВЕБТЕХНОЛОГІЙ ДЛЯ ОРГАНІЗАЦІЇ ОНЛАЙН-БРОНЮВАННЯ КВИТКІВ

1. 1 Огляд сучасних технологій і засобів розробки вебдодатків

Сучасний розвиток інформаційних технологій сприяв стрімкому поширенню вебдодатків у різних сферах діяльності людини. Вебдодатки використовуються для автоматизації бізнес-процесів, електронної комерції, дистанційного навчання, туристичних послуг, транспортних перевезень та багатьох інших напрямів. Їх популярність пояснюється доступністю через мережу Інтернет, незалежністю від операційної системи користувача та можливістю централізованого оновлення функціоналу без необхідності встановлення додаткового програмного забезпечення.

Задаючись питанням, що таке вебсайт, можна знайти таке визначення: «... вебсайт – це сукупність програмних, інформаційних, а також медійних засобів, логічно пов'язаних між собою. По суті ж вебсайт – це віддзеркалення успішності фірми, її обличчя» [1].

Вебдодаток являє собою програмну систему, доступ до якої здійснюється за допомогою веббраузера. На відміну від традиційних настільних програм, вебдодатки не потребують інсталяції на пристрої користувача та можуть працювати на різних платформах за умови наявності сучасного браузера. Основними перевагами вебдодатків є універсальність, масштабованість, простота підтримки та можливість швидкого розгортання нових функцій.

Основою сучасної веброзробки є технології HTML, CSS та JavaScript, які утворюють так званий клієнтський рівень вебдодатку. Мова розмітки HTML (HyperText Markup Language) використовується для створення структури вебсторінок. За допомогою HTML визначаються текстові блоки, заголовки, форми введення даних, таблиці, навігаційні елементи та інші компоненти інтерфейсу [2]. HTML-елемент відокремлюється від іншого тексту в документі

за допомогою «тегів», які складаються з назви елемента, оточеної символами `<teg>`. Назва елемента всередині тегу не враховує регістр. Тег `<html>` представляє корінь (елемент верхнього рівня) HTML-документа, тому його також називають кореневим елементом. Усі інші елементи повинні бути нащадками цього елемента, як показано на рис. 1.1.

```
<!doctype html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width" />
    <title>My test page</title>
  </head>
  <body>
    <img src="" alt="My test image" />
  </body>
</html>
```

Рисунок 1.1 – Запис структури HTML-документу за допомогою базових тегів

Остання версія стандарту HTML5 значно розширила можливості веброзробників, забезпечивши підтримку мультимедійного контенту, семантичних елементів і сучасних вебінтерфейсів [3].

Архітектура більшості сучасних вебдодатків базується на розподілі функціональності між двома основними складовими: клієнтською (Front-end) та серверною (Back-end) частинами. Front-end (клієнтська частина) відповідає за взаємодію користувача з інформаційною системою та забезпечує відображення даних у веббраузері. До її основних завдань належать формування інтерфейсу користувача, навігація між сторінками, валідація введених даних, виконання пошуку, фільтрація інформації та забезпечення адаптивності вебресурсу.

Базовими технологіями Front-end розробки є HTML, CSS та JavaScript. Для оформлення та візуального представлення вебсторінок застосовується технологія CSS (Cascading Style Sheets). Каскадні таблиці стилів дозволяють відокремити логіку відображення від структури документа, що спрощує супровід

програмного коду та забезпечує єдність дизайну на всіх сторінках вебдодатку. Використання сучасних можливостей CSS3 дає змогу створювати адаптивні інтерфейси, які коректно відображаються на персональних комп'ютерах, планшетах і мобільних пристроях [4].

Важливу роль у створенні інтерактивних вебдодатків відіграє мова програмування JavaScript. Вона забезпечує динамічну зміну вмісту сторінок без їх перезавантаження, обробку дій користувача, перевірку введених даних та взаємодію з віддаленими серверами. Саме JavaScript дозволяє реалізувати функції пошуку, фільтрації інформації, бронювання послуг, формування замовлень та інші механізми, необхідні для сучасних вебсервісів [5].

Останніми роками значного поширення набули спеціалізовані бібліотеки та фреймворки для веброзробки, серед яких найбільш відомими є React, Angular та Vue.js. Вони спрощують створення складних інтерфейсів користувача та забезпечують повторне використання програмних компонентів. Разом із тим використання таких засобів не завжди є виправданим для невеликих або середніх за складністю проєктів [6]. У випадку розробки вебдодатку для пошуку та бронювання квитків достатньо ефективним може бути використання класичного стеку HTML, CSS та JavaScript, який забезпечує необхідну функціональність за менших витрат часу на розробку та підтримку.

Для створення адаптивних інтерфейсів часто застосовуються CSS-фреймворки, зокрема Bootstrap та Tailwind CSS. Вони містять готові стилі, сіткові системи та компоненти інтерфейсу, що дозволяє прискорити процес розробки. Однак у багатьох випадках використання власних стилів забезпечує більшу гнучкість дизайну та дозволяє уникнути надмірного програмного коду.

Back-end (серверна частина) відповідає за обробку бізнес-логіки, взаємодію з базами даних, керування користувацькими запитами та забезпечення функціонування всієї інформаційної системи [6]. Основними функціями серверної частини є:

- обробка запитів користувачів;
- управління даними про маршрути та квитки;

- формування та обробка замовлень;
- зберігання інформації про клієнтів;
- адміністрування системи;
- забезпечення безпеки та захисту інформації.

Для реалізації Back-end можуть використовуватися різні мови програмування, серед яких найбільш поширеними є PHP, Python, Java, C#, JavaScript (Node.js) та Ruby.

Одним із найпоширеніших рішень для невеликих і середніх вебпроектів залишається PHP, який характеризується простотою інтеграції з вебсерверами та широкою підтримкою систем керування базами даних. У випадку масштабних систем часто застосовують Node.js, Python або Java, які забезпечують високу продуктивність та можливість обробки великої кількості одночасних запитів. Рейтинг популярності мов програмування серед розробників за версією аналітиків DOU [7] показано на рис. 1.2.

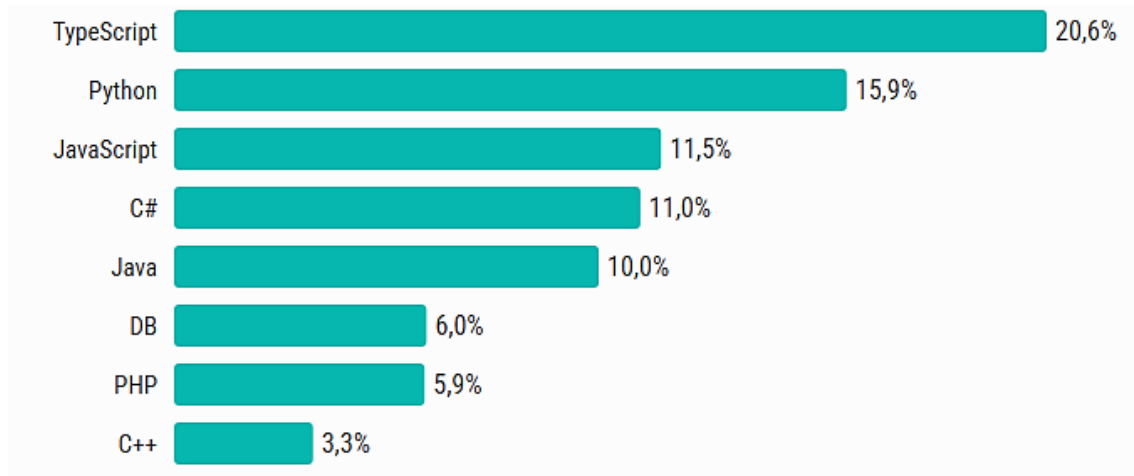


Рисунок 1.2 – Рейтинг мов програмування у 2026 р. [7]

Статистичні дані показали (див. рис. 1.2), що за підсумками 2025 р. п'ятірка лідерів залишається незмінною: TypeScript підвищив позиції, JavaScript залишається класикою фронтенд-розробки, Python обирають новачки. Серед 6782 опитаних більшість – це працівники Software Engineering (67%), далі – QA (12%). Помітно зросли в рейтингу і мови баз даних.

Важливим компонентом серверної архітектури є системи керування базами даних (СКБД), призначені для зберігання та обробки інформації. Найбільш поширеними є MySQL, PostgreSQL, MariaDB та SQLite. У контексті сервісу пошуку квитків база даних може містити інформацію про маршрути перевезень, пункти відправлення та прибуття, розклади руху, вартість квитків, замовлення користувачів та персональні дані клієнтів.

Обмін інформацією між клієнтською та серверною частинами здійснюється за допомогою технологій HTTP, HTTPS та API (Application Programming Interface). Найчастіше передавання даних реалізується у форматі JSON, який забезпечує компактне та структуроване представлення інформації. Останнім трендом є перехід до fullstack-розробки.

Важливим аспектом сучасної веброзробки є адаптивний дизайн. Згідно зі статистичними даними, значна частина користувачів здійснює доступ до вебресурсів за допомогою мобільних пристроїв. Тому вебдодатки повинні коректно працювати на екранах різного розміру та підтримувати принципи Responsive Web Design. Для цього використовуються медіазапити CSS, гнучкі сітки та адаптивні елементи інтерфейсу [8].

Окремої уваги заслуговують засоби розробки та тестування вебдодатків. Найбільш популярними редакторами програмного коду є Visual Studio Code, Sublime Text та WebStorm. Вони надають інструменти автоматичного доповнення коду, налагодження програм та контролю версій. Для тестування функціональності вебдодатків використовуються вбудовані засоби браузерів, які дозволяють аналізувати структуру сторінок, перевіряти коректність виконання сценаріїв JavaScript та контролювати мережеві запити.

При проектуванні вебдодатку для сервісу пошуку квитків особливого значення набувають зручність користувацького інтерфейсу, швидкість пошуку інформації та надійність роботи системи. Використання сучасних технологій HTML5, CSS3 та JavaScript дозволяє реалізувати всі необхідні функціональні можливості, забезпечити адаптивність вебдодатку та створити комфортні умови для взаємодії користувачів із сервісом бронювання квитків.

Таким чином, сучасні вебтехнології надають широкий спектр інструментів для створення ефективних інформаційних систем. Для розробки вебдодатку з пошуку та бронювання квитків доцільним є використання класичного стеку HTML, CSS та JavaScript, який забезпечує достатній рівень функціональності, простоту реалізації та зручність подальшого супроводу програмного продукту.

Адаптивність сайту – це можливість правильного відображення на пристроях з різними технічними характеристиками. Тобто це здатність вебсайту підлаштовуватись під пристрій, з якого його переглядають і залежить вона від ширини екрану користувача.

1.2 Особливості організації пасажирських перевезень та онлайн-продажу квитків

Пасажирські перевезення є однією з найважливіших складових транспортної системи будь-якої держави, оскільки забезпечують мобільність населення, розвиток міжрегіональних та міжнародних зв'язків, а також сприяють соціально-економічному розвитку територій. В умовах глобалізації та цифрової трансформації економіки зростає попит на швидкі, доступні та зручні способи організації поїздок, що стимулює впровадження сучасних інформаційних технологій у сферу транспортних послуг.

В Україні автобусні перевезення залишаються одним із найбільш затребуваних видів транспорту як для внутрішнього, так і для міжнародного сполучення. Це пояснюється широким охопленням населених пунктів, відносно невисокою вартістю поїздок, гнучкістю маршрутної мережі та можливістю оперативного коригування графіків руху залежно від потреб пасажирів. Особливої актуальності міжнародні автобусні перевезення набули в останні роки у зв'язку зі збільшенням кількості поїздок до країн Європи з туристичною, освітньою, трудовою та гуманітарною метою. Традиційно процес придбання квитків здійснювався через каси автовокзалів, представництва транспортних

компаній або туристичних агентств. Такий підхід має низку недоліків, серед яких необхідність особистого відвідування пункту продажу, обмежений графік роботи, значні часові витрати та недостатня оперативність оновлення інформації про доступні рейси.

Розвиток цифрових технологій сприяв переходу до онлайн-моделі обслуговування клієнтів, яка дозволяє автоматизувати більшість процесів взаємодії між перевізником та пасажиром. Онлайн-продаж квитків являє собою комплекс інформаційних сервісів, що забезпечують пошук маршрутів, перегляд розкладу руху, ознайомлення з вартістю проїзду, бронювання місць та оформлення замовлення через мережу Інтернет [9].

Сучасні системи онлайн-бронювання дозволяють користувачам самостійно виконувати необхідні операції незалежно від часу доби та місця перебування. Використання вебтехнологій значно спрощує процес придбання квитків та підвищує якість обслуговування клієнтів. Крім того, автоматизація процесів сприяє зменшенню навантаження на працівників туристичних агентств і транспортних компаній.

Організація онлайн-продажу квитків базується на взаємодії декількох функціональних компонентів, які забезпечують повний цикл обслуговування користувача. До основних функціональних процесів належать:

- пошук необхідного маршруту за пунктами відправлення та прибуття;
- вибір дати та часу поїздки;
- відображення доступних рейсів;
- перегляд інформації про перевізника та вартість квитка;
- оформлення бронювання;
- обробка замовлення користувача;
- підтвердження успішного оформлення заявки.

Процес взаємодії користувача із системою онлайн-бронювання можна представити у вигляді послідовності дій. Спочатку користувач вводить параметри поїздки, після чого система виконує пошук доступних маршрутів та відображає результати. Далі користувач обирає потрібний рейс, заповнює

необхідні контактні дані та оформлює замовлення. Після обробки заявки система формує підтвердження бронювання.

Однією з ключових вимог до сучасних сервісів продажу квитків є забезпечення високого рівня зручності використання. Інтерфейс повинен бути інтуїтивно зрозумілим, адаптованим до різних пристроїв та містити мінімальну кількість дій для виконання основних операцій. Важливими характеристиками таких систем також є швидкість роботи, надійність, актуальність інформації та захист персональних даних користувачів [10].

Суттєву роль відіграє адаптивність вебресурсу, оскільки значна частина користувачів здійснює пошук та бронювання квитків за допомогою смартфонів і планшетів. У зв'язку з цим вебдодатки повинні підтримувати коректне відображення інформації незалежно від роздільної здатності екрана та типу пристрою.

Окрім користувацьких переваг, впровадження онлайн-систем приносить значні організаційні переваги підприємствам сфери перевезень. Серед них можна виділити автоматизацію рутинних операцій, зменшення витрат на обслуговування клієнтів, скорочення часу обробки заявок, підвищення точності ведення обліку замовлень та розширення географії надання послуг.

Для туристичних агентств, які організовують автобусні перевезення по Україні та країнах Європи, використання вебдодатків стає одним із ключових інструментів конкурентоспроможності. Наявність власного онлайн-сервісу дозволяє забезпечити постійний доступ користувачів до актуальної інформації про маршрути, оперативно обробляти замовлення та підвищувати рівень довіри клієнтів до компанії.

Таким чином, сучасні тенденції розвитку сфери пасажирських перевезень свідчать про необхідність активного впровадження цифрових технологій та автоматизації процесів продажу квитків. Створення вебдодатку для пошуку маршрутів і обробки замовлень користувачів є актуальним напрямом розвитку туристично-транспортних послуг, що сприяє підвищенню ефективності роботи підприємств та покращенню якості обслуговування пасажирів.

1.3 Аналіз існуючих сервісів пошуку та бронювання квитків

Розвиток цифрових технологій сприяв появі великої кількості онлайн-сервісів, призначених для пошуку, бронювання та придбання квитків на пасажирські перевезення. Сучасні інформаційні системи дозволяють користувачам отримувати актуальну інформацію про маршрути, порівнювати пропозиції різних перевізників, здійснювати бронювання та оформлювати замовлення в режимі реального часу.

На ринку транспортних послуг функціонують як міжнародні агрегатори перевезень, так і спеціалізовані платформи, орієнтовані на окремі країни або регіони. Найбільш популярними сервісами, що підтримують автобусні перевезення по Україні та країнах Європи, є Busfor, INFOBUS, FlixBus та BlaBlaCar Bus.

Одним із найбільш відомих сервісів є Busfor, який спеціалізується на продажу автобусних квитків та співпрацює з великою кількістю перевізників [11]. Платформа дозволяє здійснювати пошук маршрутів, переглядати розклад руху, порівнювати ціни та оформлювати електронні квитки. До переваг сервісу належать широке охоплення маршрутів та простий механізм бронювання. Водночас користувачі інколи стикаються з недостатньою деталізацією інформації про перевізників та обмеженими можливостями персоналізації пошуку. Сервіс орієнтований суто на автобусні маршрути кількох об'єднаних перевізників. Сторінка сайту компанії наведена на рис. 1.3.

Сервіс INFOBUS орієнтований на міжнародні перевезення між Україною та країнами Європи [12]. Платформа надає користувачам інформацію про доступні рейси, тривалість поїздки, пересадки, вартість квитків та особливості перевезення багажу. До переваг сервісу належать значна кількість міжнародних маршрутів та підтримка багатомовного інтерфейсу.

Недоліком можна вважати перевантаженість окремих сторінок великою кількістю інформації, що ускладнює навігацію для нових користувачів. Доступні функції показані на рис. 1.4



Рисунок 1.3 – Сайт для оформлення квитків Busfor

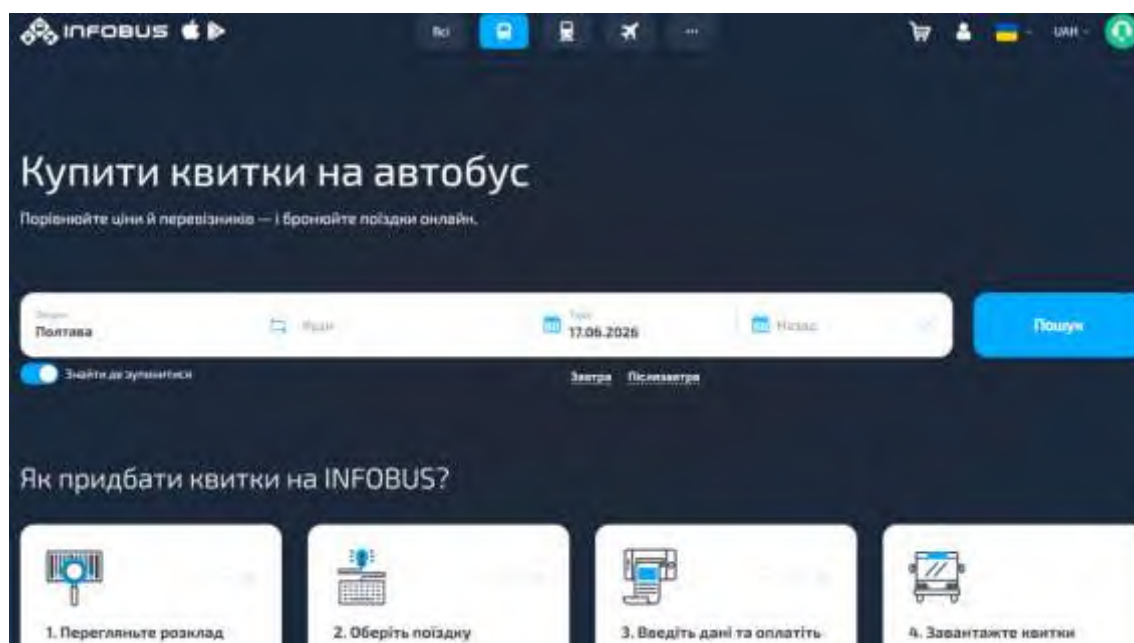


Рисунок 1.4 – Вебсайт бронювання квитків на транспорт INFOBUS

Одним із найбільших європейських операторів автобусних перевезень є FlixBus [13]. Компанія пропонує сучасний вебсервіс із високим рівнем інтеграції функціональних можливостей. Система забезпечує швидкий пошук маршрутів, онлайн-оплату, керування бронюваннями та відстеження поїздок. Перевагами сервісу є мінімалістичний дизайн, висока продуктивність та адаптивність

інтерфейсу. Проте основна діяльність компанії орієнтована на власну транспортну мережу, що обмежує кількість альтернативних перевізників.

Популярним міжнародним рішенням також є BlaBlaCar Bus, який поєднує функціональність автобусних перевезень із сервісами спільних подорожей. Користувачам доступний пошук маршрутів, перегляд інформації про перевізників та оформлення електронних квитків. До переваг належать зручний інтерфейс та інтеграція з іншими транспортними сервісами. Водночас для деяких напрямків кількість доступних рейсів може бути обмеженою.

Порівняльний аналіз існуючих сервісів дозволив виділити найбільш поширені функціональні можливості, які повинні бути реалізовані в сучасному вебдодатку:

- пошук маршрутів за пунктами відправлення та прибуття;
- вибір дати та часу поїздки;
- перегляд детальної інформації про рейс;
- сортування та фільтрація результатів пошуку;
- оформлення бронювання;
- формування замовлення користувача;
- адаптивний дизайн для мобільних пристроїв;
- можливість подальшого розширення функціоналу.

Разом із тим аналіз показав наявність певних недоліків, характерних для більшості існуючих платформ:

- надмірна складність інтерфейсу;
- перевантаженість інформацією;
- велика кількість рекламних елементів;
- обмежені можливості персоналізації;
- багаторівнева процедура оформлення замовлення;
- залежність від сторонніх сервісів та інтеграцій.

На основі детального аналізу популярних сервісів проведено оцінювання якості послуг методом ієрархій за основними критеріями [14]. Оцінювання

проведене за двобальною шкалою, де «2» – функція повністю реалізована; «1» – функція реалізована частково або має обмеження; «0» – функція відсутня або недоступна (табл. 1.1).

Таблиця 1.1 – Порівняльний аналіз існуючих сервісів онлайн-пошуку та бронювання автобусних квитків

Критерій	Busfor	INFOBUS	FlixBus	BlaBlaCar Bus	Пропонований вебдодаток
Пошук маршрутів	2	2	2	2	2
Бронювання квитків онлайн	2	2	2	2	2
Перевезення по Україні	2	2	1	0	2
Міжнародні перевезення до країн Європи	2	2	2	2	2
Простий та інтуїтивний інтерфейс	1	1	2	2	2
Адаптивність для мобільних пристроїв	2	2	2	2	2
Мінімальна кількість дій для оформлення замовлення	1	0	2	2	2
Можливість персоналізації під потреби туристичного агентства	0	0	0	0	2
Спрощена система обробки замовлень	1	1	2	1	2
Можливість подальшого розширення функціоналу	0	0	1	1	2
Сумарний показник якості	13	12	16	14	20

Аналіз даних таблиці 1.1 свідчить, що існуючі сервіси успішно реалізують базові механізми пошуку та бронювання квитків, однак більшість із них є універсальними платформами, орієнтованими на широке коло перевізників. Це ускладнює адаптацію їх функціональних можливостей до потреб конкретного туристичного агентства. Крім того, окремі системи характеризуються перевантаженістю інтерфейсу та багатоетапною процедурою оформлення замовлення.

Розроблення власного вебдодатку дозволяє усунути зазначені недоліки шляхом створення спрощеної системи пошуку маршрутів, оптимізації процесу бронювання квитків та забезпечення ефективної обробки замовлень користувачів.

З огляду на проведений аналіз, під час проєктування власного вебдодатку доцільно орієнтуватися на принципи простоти, швидкодії та зручності користування. Особлива увага повинна приділятися мінімізації кількості дій, необхідних для пошуку та бронювання квитків, а також забезпеченню інтуїтивно зрозумілої навігації.

Для туристичного агентства, яке здійснює автобусні перевезення по Україні та країнах Європи, доцільним є створення спеціалізованого вебдодатку, що об'єднуватиме функції пошуку маршрутів, перегляду інформації про рейси та обробки замовлень користувачів в єдиному середовищі. На відміну від універсальних платформ, такий підхід дозволить адаптувати функціональні можливості системи під конкретні потреби підприємства та спростити взаємодію клієнтів із сервісом.

Таким чином, аналіз існуючих сервісів показав, що сучасні платформи успішно реалізують базові механізми онлайн-бронювання квитків, проте мають низку недоліків, пов'язаних зі складністю інтерфейсу та надлишковою функціональністю. Це підтверджує доцільність розробки власного вебдодатку, орієнтованого на швидкий пошук автобусних маршрутів та ефективну обробку замовлень користувачів.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ВЕБДОДАТКУ ДЛЯ ПОШУКУ КВИТКІВ І ОБРОБКИ ЗАМОВЛЕНЬ

2.1 Проєктування структури та функціональних модулів вебдодатку

Проєктування структури вебдодатку є одним із ключових етапів розробки програмного продукту, оскільки саме на цьому етапі визначаються функціональні можливості системи, взаємозв'язки між окремими компонентами та принципи взаємодії користувача з інформаційним середовищем. Грамотно спроектована структура забезпечує зручність використання, швидкодію, адаптивність та можливість подальшого розширення функціоналу.

Метою проєктування було створення вебдодатку для туристичного агентства, яке забезпечує пошук автобусних маршрутів, бронювання квитків та обробку замовлень користувачів через єдиний вебінтерфейс. Особлива увага приділялася простоті використання, мінімізації кількості дій користувача та адаптації сервісу до різних типів пристроїв.

При розробці структури вебдодатку було застосовано модульний підхід, відповідно до якого функціональність системи поділяється на окремі взаємопов'язані компоненти. Такий підхід спрощує процес розробки, тестування та подальшого супроводу програмного продукту.

Архітектура вебдодатку реалізована за принципом клієнтської взаємодії, де всі основні процеси виконуються безпосередньо у веббраузері користувача із застосуванням технологій HTML5, CSS3 та JavaScript. Вебдодаток має односторінкову структуру (Single Page Application), що дозволяє забезпечити швидке переміщення між функціональними блоками без перезавантаження сторінки [15].

На етапі проєктування було виділено декілька основних функціональних модулів. Опишемо назву і призначення кожного з них.

1. Модуль головної сторінки.

Головна сторінка виконує роль центрального інформаційного середовища, з якого користувач отримує доступ до всіх функціональних можливостей системи. На ній розміщено основні елементи навігації, інформаційні блоки та інтерактивні компоненти, що забезпечують швидкий доступ до пошуку маршрутів і бронювання квитків. Особлива увага приділялася візуальному оформленню, яке формує позитивне враження та спрощує взаємодію із сервісом.

2. Модуль навігації.

Для забезпечення швидкого доступу до функціональних розділів було реалізовано адаптивне навігаційне меню. Навігаційна система забезпечує переміщення між окремими інформаційними секціями та автоматично адаптується до різних розмірів екрана. Використання меню типу «гамбургер» дозволило оптимізувати відображення вебдодатку на мобільних пристроях та підвищити ергономічність інтерфейсу.

3. Модуль пошуку маршрутів.

Одним із центральних елементів вебдодатку є модуль пошуку автобусних маршрутів. Його призначення полягає в забезпеченні швидкого доступу користувачів до необхідної інформації про доступні перевезення. Користувач отримує можливість вводити параметри поїздки, після чого система здійснює обробку введених даних та відображає результати пошуку. Реалізація модуля спрямована на мінімізацію часових витрат під час вибору необхідного маршруту.

4. Модуль бронювання квитків.

Модуль бронювання забезпечує взаємодію користувача із системою оформлення замовлення. Його функціональність передбачає введення контактних даних, формування заявки та підтвердження виконання операції. Для підвищення зручності передбачено перевірку правильності введення інформації та відображення повідомлень про результати виконання дій користувача.

5. Модуль обробки замовлень.

Обробка замовлень реалізована за допомогою JavaScript-сценаріїв, які забезпечують взаємодію між формами введення даних та механізмами їх опрацювання. На поточному етапі розробки обробка заявок виконується у

демонстраційному режимі без використання серверної частини та систем керування базами даних. Такий підхід дозволяє продемонструвати логіку функціонування майбутньої інформаційної системи та може бути основою для її подальшої модернізації.

6. Модуль мультимедійної взаємодії.

Для підвищення привабливості вебдодатку були реалізовані додаткові інтерактивні компоненти, серед яких анімовані переходи, звукові сповіщення, графічні елементи та плавне прокручування сторінки. Використання мультимедійних елементів сприяє покращенню користувацького досвіду та формує сучасний вигляд вебресурсу. Під час проектування також були враховані принципи адаптивного дизайну. Інтерфейс автоматично змінює своє відображення залежно від роздільної здатності екрана, що забезпечує коректну роботу як на персональних комп'ютерах, так і на планшетах та смартфонах.

Узагальнену структуру вебдодатку можна представити у вигляді взаємодії функціональних модулів (рис. 2.1).

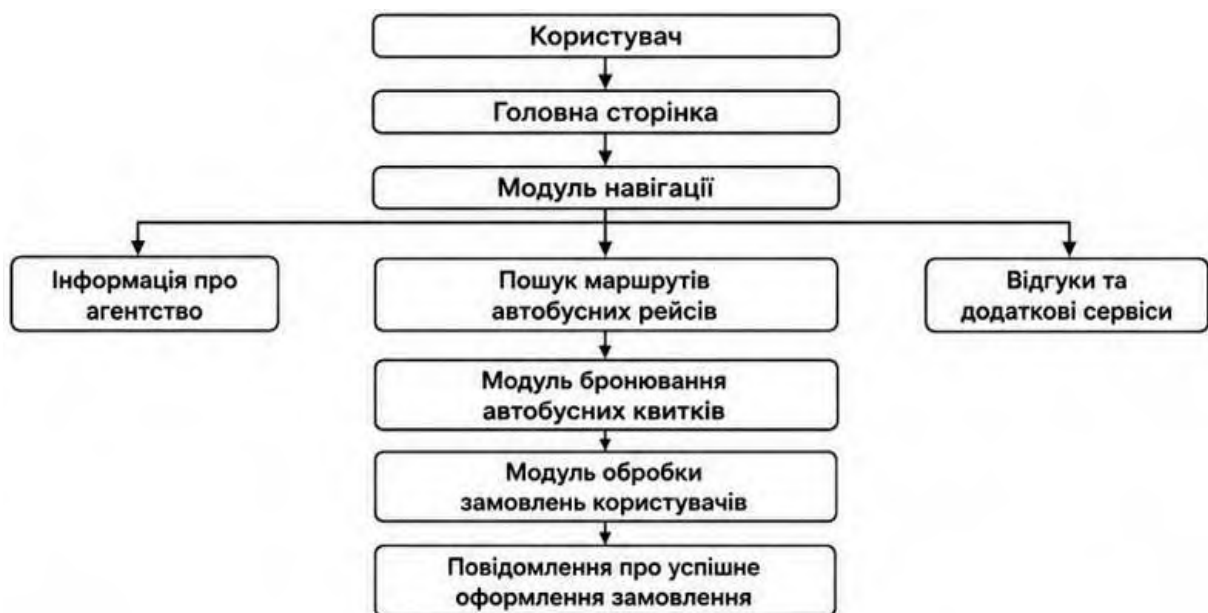


Рисунок 2.1 – Структурна схема вебдодатку

На рис. 2.1 представлено структурну схему розробленого вебдодатку для пошуку автобусних квитків та обробки замовлень користувачів. Центральним

елементом системи є головна сторінка, через яку здійснюється доступ до всіх функціональних модулів. Навігаційний модуль забезпечує перехід до інформаційних розділів, пошуку маршрутів та додаткових сервісів. Основний сценарій взаємодії користувача передбачає послідовне виконання пошуку автобусного рейсу, бронювання квитка, обробки замовлення та отримання повідомлення про успішне завершення операції. Така модульна організація спрощує структуру вебдодатку та забезпечує можливість його подальшого розширення.

2.2 Розробка дизайну та адаптивного користувацького інтерфейсу

Розробка дизайну та користувацького інтерфейсу є одним із найважливіших етапів створення сучасних вебдодатків, оскільки саме зовнішній вигляд системи формує перше враження користувача та значною мірою впливає на ефективність взаємодії з програмним продуктом. Грамотно спроектований інтерфейс дозволяє мінімізувати часові витрати на пошук необхідної інформації, спростити виконання операцій та підвищити загальний рівень задоволеності користувачів.

При розробці вебдодатку для туристичного агентства основною метою було створення сучасного, інтуїтивно зрозумілого та адаптивного середовища, яке забезпечує швидкий пошук автобусних маршрутів, бронювання квитків та обробку замовлень незалежно від типу пристрою.

Процес проєктування інтерфейсу базувався на принципах User Interface (UI) та User Experience (UX). Концепція UI спрямована на створення візуальної складової вебдодатку, тоді як UX орієнтована на забезпечення комфортної взаємодії користувача із системою.

Під час розробки були враховані такі принципи проєктування інтерфейсу:

- простота та зрозумілість навігації;
- мінімізація кількості дій користувача;

- логічне групування функціональних елементів;
- візуальна узгодженість усіх компонентів;
- адаптивність до різних типів пристроїв;
- швидкий доступ до основних функцій системи.

При створенні дизайну було використано односторінковий підхід (Single Page Design), який передбачає розміщення основних функціональних блоків на одній вебсторінці з плавною навігацією між ними. Такий підхід дозволяє скоротити час переходу між розділами та спрощує взаємодію із сервісом.

Важливим елементом інтерфейсу стала головна сторінка, яка виконує роль центрального інформаційного вузла системи. На ній розміщено ключові функціональні компоненти, необхідні для роботи користувача із сервісом.

Для забезпечення сучасного вигляду вебдодатку використано принцип візуальної ієрархії, який передбачає виділення найбільш важливих елементів за допомогою розмірів, контрастності та просторового розташування. Центральне місце відведено пошуковому модулю та кнопкам переходу до основних функцій системи. Особлива увага приділялася навігаційній системі. Для цього було реалізовано адаптивне меню типу «гамбургер», яке автоматично змінює спосіб відображення залежно від розміру екрана. Такий підхід дозволяє ефективно використовувати робочий простір як на персональних комп'ютерах, так і на мобільних пристроях.

Під час створення інтерфейсу використовувався принцип мінімізації когнітивного навантаження. Це означає, що користувач повинен приймати якомога менше рішень для досягнення необхідного результату. Зокрема, процес пошуку та бронювання автобусних квитків організований як послідовність взаємопов'язаних кроків із чіткими підказками.

Важливим елементом сучасного дизайну є використання інтерактивних компонентів. У вебдодатку реалізовано:

- плавні анімаційні переходи;
- ефекти наведення курсора;
- динамічне відображення елементів;

- звукові сповіщення;
- автоматичне оновлення окремих блоків інтерфейсу.

Використання таких рішень дозволяє зробити взаємодію з вебдодатком більш комфортною та природною.

Для оформлення інтерфейсу було застосовано каскадні таблиці стилів CSS3. Використання єдиної системи стилів забезпечило візуальну цілісність усіх елементів вебдодатку та спростило подальшу модифікацію дизайну.

Окрему увагу приділено адаптивності вебдодатку. Адаптивний дизайн (Responsive Web Design) передбачає автоматичне пристосування інтерфейсу до параметрів пристрою користувача [16]. Реалізація адаптивності здійснювалася за допомогою медіазапитів CSS, гнучких контейнерів та відносних одиниць вимірювання. При зміні розмірів екрана автоматично відбувається:

- перебудова навігаційного меню;
- зміна розташування інформаційних блоків;
- масштабування графічних елементів;
- оптимізація відображення тексту;
- коригування відступів між компонентами.

Важливим аспектом сучасного вебпроектування є також забезпечення високої швидкодії інтерфейсу. Для цього було мінімізовано кількість зовнішніх залежностей, а реалізація основного функціоналу виконана за допомогою власних JavaScript-сценаріїв без використання сторонніх фреймворків. Такий підхід дозволяє зменшити обсяг завантажуваних ресурсів та прискорити відображення вебсторінки.

Сучасні дослідження у сфері UX-дизайну свідчать, що користувач формує перше враження про вебресурс протягом перших 3–5 секунд взаємодії. Саме тому під час проектування особлива увага приділялася простоті інтерфейсу, швидкому доступу до основних функцій та відсутності перевантаження сторінок надлишковою інформацією.

Крім того, при розробці було враховано правило «трьох кліків», відповідно до якого користувач повинен отримати доступ до необхідної функції не більше

ніж за три послідовні дії. Це дозволяє значно підвищити ергономічність системи та скоротити час виконання операцій. Інтегрована схема принципів дизайну представлена на рис. 2.2.



Рисунок 2.2 – Основні принципи проектування користувацького інтерфейсу

Таким чином, розроблений користувацький інтерфейс поєднує сучасні принципи UI/UX-дизайну, адаптивні технології та інтерактивні компоненти, що забезпечують комфортну взаємодію користувачів із вебдодатком (див. рис. 2.2). Реалізовані дизайнерські рішення сприяють спрощенню процесів пошуку автобусних маршрутів, бронювання квитків та оформлення замовлень, а також створюють передумови для подальшого розвитку функціональних можливостей системи.

2.3 Обґрунтування технологій програмної реалізації вебдодатку засобами HTML5, CSS3 та JavaScript

Програмна реалізація вебдодатку є завершальним етапом його проектування, під час якого створюється працездатний програмний продукт, що забезпечує виконання всіх передбачених функціональних можливостей. Для розробки вебдодатку туристичного агентства, призначеного для пошуку автобусних квитків та обробки замовлень користувачів, було використано класичний стек вебтехнологій, до складу якого входять HTML5, CSS3 та JavaScript.

Обраний підхід дозволяє реалізувати повноцінний вебдодаток без використання складних програмних платформ та сторонніх фреймворків. Крім того, застосування стандартних вебтехнологій забезпечує високу швидкодію, сумісність із сучасними браузерами та спрощує подальший супровід програмного продукту.

Архітектура вебдодатку побудована за принципом клієнтської взаємодії, відповідно до якого всі основні процеси виконуються безпосередньо у веббраузері користувача. Програмна структура реалізована у вигляді односторінкового вебдодатку, де всі функціональні блоки об'єднані в межах єдиного HTML-документа.

Реалізація структури вебдодатку засобами HTML5 [17].

Мова розмітки HTML5 використовується для формування логічної структури вебдодатку та організації взаємозв'язків між його функціональними компонентами.

Під час розробки було реалізовано такі структурні блоки:

- головний екран вебдодатку;
- навігаційне меню;
- інформаційні секції;
- модуль пошуку маршрутів;
- модуль бронювання квитків;

- блок відгуків користувачів;
- форми введення даних;
- інформаційні повідомлення.

Для побудови структури сторінки використовувалися семантичні елементи HTML5, що покращують читабельність програмного коду та спрощують його подальший супровід. Семантичний підхід також позитивно впливає на пошукову оптимізацію вебресурсу та доступність інтерфейсу.

Реалізація візуального оформлення зазвичай описується кодами CSS3 [18]. Зовнішній вигляд вебдодатку реалізовано за допомогою каскадних таблиць стилів CSS3, які забезпечують єдине оформлення всіх компонентів інтерфейсу.

Основними завданнями CSS3 стали:

- формування єдиної стилістичної концепції;
- організація розташування елементів;
- створення адаптивного дизайну;
- реалізація анімацій;
- оформлення інтерактивних компонентів.

Особлива увага приділялася використанню сучасних засобів позиціонування елементів на пристроях різної ширини екрану під потреби користувачів. Для цього застосовувалися гнучкі контейнери, відносні одиниці вимірювання та медіазапити. Завдяки цьому вдалося забезпечити коректне відображення вебдодатку на різних типах пристроїв:

- персональних комп'ютерах;
- ноутбуках;
- планшетах;
- смартфонах.

Для покращення користувацького досвіду були реалізовані плавні анімаційні переходи між станами елементів інтерфейсу, ефекти наведення курсора та динамічна поява інформаційних блоків.

Інтерактивна складова вебдодатку реалізована за допомогою мови програмування JavaScript. Використання JavaScript дозволило реалізувати

механізми взаємодії користувача із системою без необхідності перезавантаження сторінки.

Основними функціональними можливостями, реалізованими засобами JavaScript, є:

- обробка дій користувача;
- керування навігаційним меню;
- організація пошуку маршрутів;
- обробка даних форм бронювання;
- перевірка правильності введеної інформації;
- виведення повідомлень;
- керування мультимедійними компонентами;
- реалізація анімаційних ефектів.

Особливу роль відіграє механізм обробки замовлень користувачів. Алгоритм його роботи передбачає послідовне виконання таких операцій:

- введення користувачем параметрів пошуку;
- вибір відповідного маршруту;
- заповнення форми бронювання;
- перевірка коректності введених даних;
- формування замовлення;
- відображення повідомлення про успішне виконання операції.

У поточній реалізації обробка замовлень здійснюється у демонстраційному режимі без використання серверної частини та систем керування базами даних. Ці функції до завершення клієнтської частини знаходяться в стані прототипування. Проте, запропонована програмна архітектура передбачає можливість подальшої інтеграції із зовнішніми інформаційними системами.

Створений вебдодаток має компактну структуру, що складається з одного основного HTML-файлу та допоміжних мультимедійних ресурсів.

Концептуальна схема використання вебінструментів у розрізі функцій та архітектури вебдодатку показана на рис. 2.3.



Рисунок 2.3 – Структура рпрограмної реалізації вебдодатку

До складу проєкту входять: файл `index.html`, який містить основну структуру вебдодатку; графічні файли, що використовуються для візуального оформлення; звуковий файл для реалізації системи сповіщень.

Такий підхід дозволив зменшити складність програмної структури та спростити процес розгортання вебдодатку. Використання класичного стеку вебтехнологій забезпечило низку переваг:

- простоту реалізації;
- високу швидкодію;
- кросплатформність;
- адаптивність;
- легкість супроводу;
- можливість масштабування;
- сумісність із сучасними браузерами;
- відсутність залежності від сторонніх програмних платформ.

Особливістю розробленого вебдодатку є поєднання простоти програмної архітектури та достатньої функціональності для реалізації основних бізнес-процесів туристичного агентства. Для роботи з кодом було обрано інтегроване середовище розробки зпоміж відомих редакторів коду (WebStorm [19], Sublime

Text та ін.), використання яких є необхідною частиною сучасної культури програмної інженерії. Серед основних переваг сучасних середовищ розробки можна виділити наступні:

- безкоштовність; з малої літери всі пункти;
- автодоповнення коду;
- можливість редагування в режимі реального часу або, так званий, Live Edit;

- зручність;
- функціональна зрозумілість інтерфейсу;
- гнучкість;
- простота освоєння;
- практичність.

Далі заносимо оцінки виділених характеристик редакторів коду в табл. 2.1 і визначаємо бали за шкалою від 1 до 10, де 1 – абсолютно не задовільно, а 10 – повністю відповідає на основі досконалості даних переваг у редактора.

Таблиця 2.1 – Порівняльна характеристика середовищ розробки

Критерії порівняння	Бальна оцінка за критеріями кожного із середовищ розробки				
	VS Code	Sublime Text	WebStorm	Dreamweaver	Vim
Безкоштовність	10	9	5	5	10
Автодоповнення коду	10	9	10	9	5
Live Edit	10	10	10	9	9
Зручність	10	8	10	8	10
Простота інтерфейсу	8	8	7	8	6
Гнучкість	10	8	10	7	8
Простота освоєння	9	9	9	9	4
Практичність	9	8	10	8	10
Середній бал якості	9.5	8.63	8.88	7.88	7.75

Після розрахунку середнього балу якості (див. табл. 2.1) найбільше переваг має редактор коду Visual Studio Code (результати – 9,5). Отже, підтверджено, що VS Code є найоптимальнішим варіантом редактору коду для розробника, через свою зручність, гнучкість та безкоштовність. Саме через ціну і простоту редакторів середній бал WebStorm та Vim значно знизився, хоча без сумніву всі

розглянуті вище редактори є гарними програмними засобами і доцільні для використання, але в індивідуальних умовах. Ключові можливості та характеристики Visual Studio Code [20] розглянемо більш детально по пунктах.

1. Інтерфейс користувача. VS Code має чистий та інтуїтивно зрозумілий користувацький інтерфейс з бічною панеллю для навігації, центральною областю редагування та рядком стану для відображення інформації та інструментів. Він забезпечує вільний робочий простір, дозволяючи розробникам зосередитися на своєму коді (рис. 2.4).

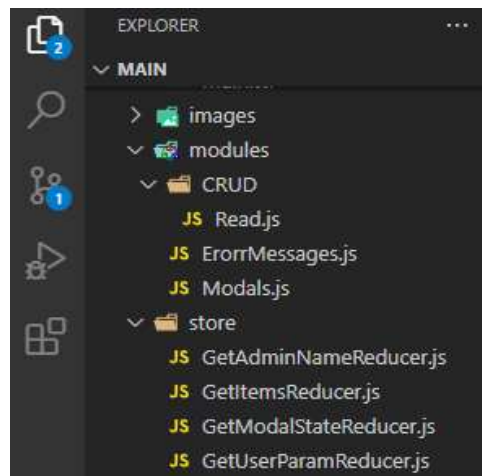


Рисунок 2.4 – Вигляд бічної панелі Visual Studio Code

Бічна панель Visual Studio Code забезпечує швидкий доступ до різних функцій та можливостей редактора коду. Бічна панель містить іконки навігації, такі як Провідник, Розширення та інші, що дозволяють користувачам легко перемикаватися між різними файлами та інструментами в редакторі.

2. Крос-платформна сумісність. Visual Studio Code доступний для Windows, macOS та Linux, що забезпечує сумісність та однакову роботу з різними операційними системами.

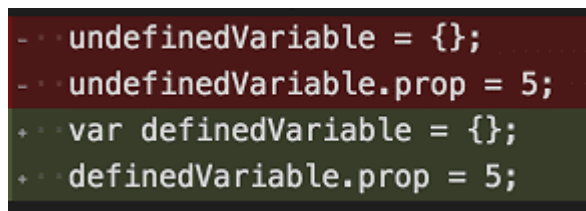
3. Розширення та маркетплейс. Однією з визначних особливостей VS Code є його широка екосистема розширень. Розробники можуть налаштовувати та покращувати свій досвід кодування, встановлюючи широкий спектр розширень для різних мов програмування, фреймворків та інструментів. На

Visual Studio Code Marketplace розміщено величезну колекцію розширень, створених спільнотою.

4. Мовна підтримка. Комплексна мовна підтримка для багатьох мов програмування, включаючи такі популярні, як JavaScript, Python, C++, Java, HTML, CSS та багато інших. Вона забезпечує підсвічування синтаксису, завершення коду, форматування, розбиття на рядки та можливості налагодження, специфічні для кожної мови.

5. Інтегрований термінал. Дозволяє розробникам запускати команди, компілювати код і виконувати різні завдання без перемикання на окремий термінальний додаток. Це полегшує безперебійний робочий процес розробки в самому редакторі.

6. Інтеграція з Git. VS Code має вбудовану інтеграцію з Git, що дозволяє контролювати версії та співпрацювати з колегами по команді (рис. 2.5). Розробники можуть легко переглядати та керувати репозиторіями Git'a, фіксувати зміни, переглядати різниці та виконувати операції з Git'ом, не виходячи з редактора [21].



```
- · undefinedVariable = {};  
- · undefinedVariable.prop = 5;  
+ · var definedVariable = {};  
+ · definedVariable.prop = 5;
```

Рисунок 2.5 – Вигляд різних версій проєкту Visual Studio Code

7. Можливості налагодження. Visual Studio Code забезпечує надійну підтримку налагодження для різних мов і платформ. Вона пропонує покрокове налагодження, перевірку змінних та інші важливі функції налагодження, які допомагають розробникам виявляти та виправляти проблеми в коді.

8. IntelliSense. Інтелектуальна функція завершення коду у VS Code, яка надає контекстно-залежні пропозиції під час введення коду. Вона пропонує автозавершення, підказки щодо підписів функцій та документацію, щоб пришвидшити кодування та зменшити кількість помилок [22].

9. Автоматизація завдань. VS Code підтримує автоматизацію завдань за допомогою інтегрованого бігуна завдань. Розробники можуть визначати власні завдання збірки, запускати скрипти і виконувати різні автоматизовані завдання в редакторі, підвищуючи продуктивність і ефективність робочого процесу.

10. Кастомізація. Visual Studio Code дуже добре налаштовується, що дозволяє розробникам персоналізувати своє середовище кодування. Користувачі можуть змінювати теми, встановлювати розширення, налаштовувати параметри та створювати власні комбінації клавіш, щоб пристосувати редактор до своїх уподобань [23].

Загалом, Visual Studio Code надає потужне та гнучке середовище кодування з багатим набором функцій, розширень та можливостей налаштування, що робить його популярним вибором для розробників усіх рівнів. Його легкість, широка мовна підтримка та динамічна екосистема сприяють його широкому розповсюдженню у спільноті розробників.

Таким чином, використання HTML5, CSS3 та JavaScript і введення в Visual Studio Code дозволило створити ефективний вебдодаток для пошуку автобусних квитків та обробки замовлень користувачів. Запропонована програмна реалізація забезпечує високу швидкодію, зручність використання та можливість подальшої модернізації системи шляхом інтеграції серверних технологій та баз даних.

РОЗДІЛ 3

ПРАКТИЧНІ РЕЗУЛЬТАТИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ФУНКЦІЙ РОЗРОБЛЕНОГО ВЕБДОДАТКУ

3.1 Перевірка працездатності та тестування функціональних можливостей системи

Згідно обраних технологій клієнтську частину (Frontend) вебдодатка розроблено у вигляді SPA (Single Page Application) на базі HTML5, CSS3 та Vanilla JavaScript (чистий JS). Проєкт має адаптивну верстку, складні CSS-анімації та логіку взаємодії з користувачем на стороні клієнта. Важливою частиною створення будь-якого сайту є семантична оптимізація коду. Семантична оптимізація коду фокусується на використанні відповідних тегів HTML5 для відображення структури та ієрархії контенту. В роботі використано семантичну розмітку (`<header>`, `<main>`, `<section>`, `<footer>`) для забезпечення високої SEO-оптимізації й доступності (Accessibility) [24]. Результатом розмітки є вебдодаток, який складається з кількох послідовних секцій. Першим оформлене горизонтальне меню, яке залишається закріпленим на екрані під час скролінгу всієї сторінки (рис. 3.1). Код горизонтального меню показано на рис. 3.2.

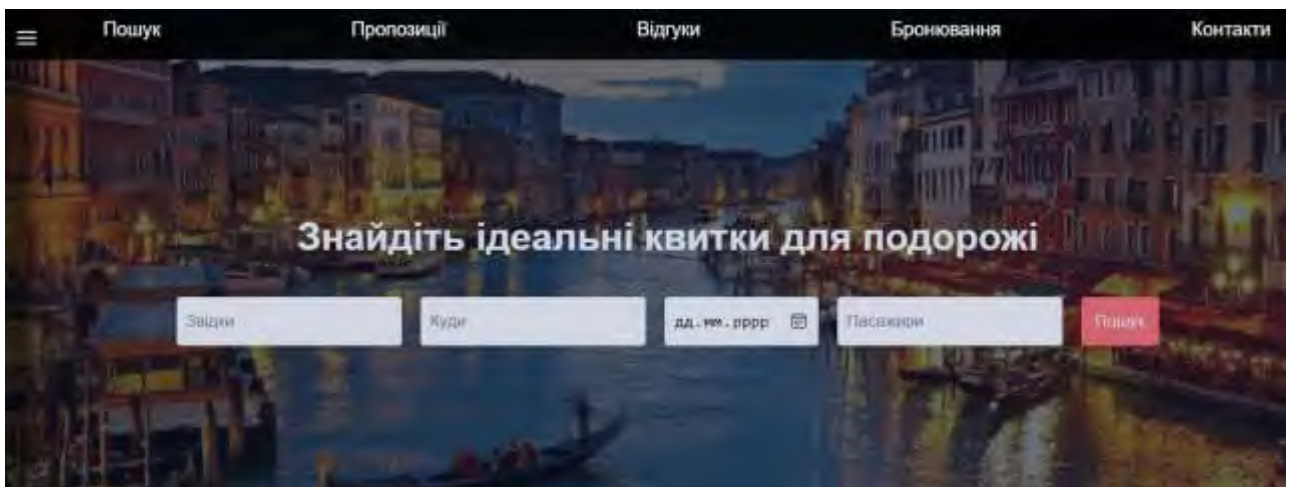


Рисунок 3.1 – Інтерфейс верхньої частини вебсайту

```

652 <body>
653 > <div class="sidebar-toggle" id="sidebarToggle" onclick="toggleSidebar()">...
657 </div>
658
659 <div class="sidebar" id="sidebar">
660 <ul class="sidebar-nav">
661 <li><a href="#features" onclick="smoothScroll('features')">Пропозиції</a></li>
662 <li><a href="#testimonial" onclick="smoothScroll('testimonial')">Відгуки</a></li>
663 <li><a href="#booking" onclick="smoothScroll('booking')">Бронювання</a></li>
664 <li><a href="#contacts" onclick="smoothScroll('contacts')">Контакти</a></li>
665 </ul>
666 </div>
---
```

Рисунок 3.2 – HTML-код верхньої частини і горизонтального меню

Використовуючи описові та змістовні імена для елементів HTML, розробники можуть зробити свій код більш зрозумілим і легким для сприйняття. Це не тільки приносить користь іншим розробникам, які можуть працювати над проектом, але й сприяє загальній легкості обслуговування кодової бази. Повний HTML-код вебсторінки представлено в додатку А.

Ще одним ключовим аспектом семантичної оптимізації коду є розмежування проблем, пов'язаних із контентом, представленням і поведінкою. Такий підхід дозволяє розробникам підтримувати чітке розділення HTML, CSS і JavaScript, що полегшує оновлення та модифікацію певних аспектів вебсайту, не впливаючи на інші.

Зліва вгорі знаходиться компактне гамбургер-меню, написане на HTML і JavaScript (рис. 3.3), яке активується в момент перегляду сайту на малих екранах, відкривається та закривається після натискання відповідної кнопки (рис. 3.4).

```

812 <script>
813   function toggleSidebar() {
814     const sidebar = document.getElementById('sidebar');
815     const toggle = document.getElementById('sidebarToggle');
816
817     sidebar.classList.toggle('open');
818     toggle.classList.toggle('active');
819   }
820
821   document.addEventListener('click', function(event) {
822     const sidebar = document.getElementById('sidebar');
823     const toggle = document.getElementById('sidebarToggle');
824
825     if (!sidebar.contains(event.target) && !toggle.contains(event.target) && sidebar.classList.contains('open')) {
826       toggleSidebar();
827     }
828   });

```

Рисунок 3.3 – Код JavaScript адаптивного гамбургер-меню

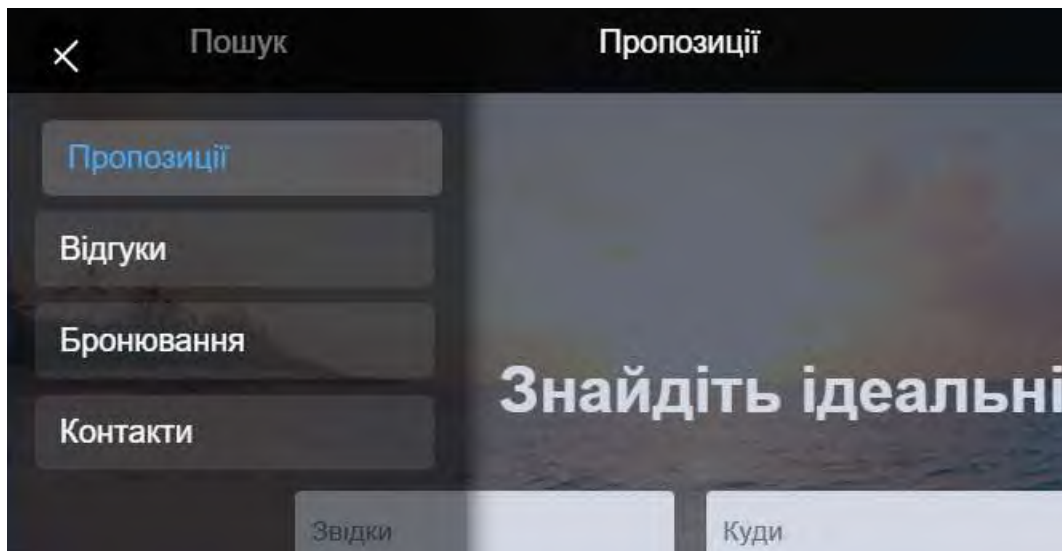


Рисунок 3.4 – Розгорнуте адаптивне бокове меню на екрані

Для побудови адаптивного інтерфейсу використано технологію flexbox та медіазапити CSS3 [25]. При зменшенні ширини екрана автоматично змінюється розташування елементів, що забезпечує коректне відображення вебдодатку на смартфонах і планшетах. Код показано на рис. 3.5.

```

225     #hero {
226         text-align: center;
227         padding: 4rem 2rem;
228         min-height: calc(100vh - 100px);
229         display: flex;
230         flex-direction: column;
231         justify-content: center;
232         position: relative;
233         overflow: hidden;
234         transition: opacity 0.5s ease;
235         opacity: 1;
236         margin-top: 0;
237     }

```

Рисунок 3.5 – Реалізація адаптивного дизайну за допомогою flexbox CSS3

Окремі елементи мають ефект анімації, наприклад, для покращення користувацького досвіду реалізовано плавну появу інформаційних блоків (рис. 3.6). Анімація створює сучасний вигляд інтерфейсу та робить взаємодію користувача із системою більш комфортною.



Рисунок 3.6 – Ефект анімації (2D повертання) при наведенні на елемент вебсату

Для опису властивостей використані стандартні функції трансформації [26] `transform: translateY(20px); transition: all 0.5s ease.` Фрагмент запису відповідного коду показано на рис. 3.7.

```

447     .testimonial-content {
448         max-width: 500px;
449         opacity: 0;
450         transform: translateY(30px);
451         transition: all 1.6s ease;
452         z-index: 2;
453     }
454
455     .testimonial-content.animated {
456         opacity: 1;
457         transform: translateY(0);
458     }

```

Рисунок 3.8 – Опис властивостей анімованих елементів на CSS

Для інтерактивних елементів використано ефект масштабування під час наведення курсора (властивість `.btn:hover`). Такий підхід забезпечує візуальний зворотний зв'язок між системою та користувачем і підвищує зручність використання вебдодатку.

Особливих ефектів при взаємодії користувача досягнуто завдяки поєднанню JavaScript з властивостями CSS або HTML5. Які розглядаються в наступному підрозділі.

3.2 Архітектурні паттерни та модулі клієнтської частини на JavaScript

Перш за все, заслуговує на увагу модуль динамічної ротації контенту (Hero Section Slider), який керує станом фонових зображень за допомогою циклічного таймера `setInterval` та синхронізованих мікроанімацій індикаторів (пагінації). Логіка автоматичної зміни тем реалізована за допомогою чергування класів `background-current` та `background-next` у поєднанні зі скиданням та оновленням інтервалу часу: JavaScript-код на рис. 3.9.

```

993     const backgroundThemes = ['airport', 'beach', 'mountain', 'city'];
994     let currentBackgroundIndex = 0;
995     let backgroundInterval;
996     let isTransitioning = false;
997
998     function updateCircleButtons(oldIndex, newIndex) {
999         const circleButtons = document.querySelectorAll('.circle-button');
1000
1001         circleButtons[oldIndex].classList.add('expanding-peak');
1002
1003         if (oldIndex > 0) {
1004             circleButtons[oldIndex - 1].classList.add('contracting-far');
1005         }
1006         if (oldIndex < circleButtons.length - 1) {
1007             circleButtons[oldIndex + 1].classList.add('expanding-far');
1008         }
1009
1010         setTimeout(() => {
1011             circleButtons[oldIndex].classList.remove('expanding-peak');
1012             circleButtons[oldIndex].classList.add('expanding-final');
1013             circleButtons[oldIndex].classList.remove('active');
1014
1015             if (oldIndex > 0) {
1016                 circleButtons[oldIndex - 1].classList.remove('contracting-far');
1017                 circleButtons[oldIndex - 1].classList.add('contracting');
1018             }
1019             if (oldIndex < circleButtons.length - 1) {
1020                 circleButtons[oldIndex + 1].classList.remove('expanding-far');
1021                 circleButtons[oldIndex + 1].classList.add('expanding');
1022             }
1023         }, 500);
1024     }

```

Рисунок 3.9 – Код каруселі фонових зображень

У фрагменті коду JavaScript (див рис. 3.9) видно, що фонові зображення складають масив, встановлено зміну 4-х варіантів фону. зміна відбувається кожні 5 секунд. Логіка автоматичної зміни тем реалізована за допомогою чергування класів `background-current` та `background-next` у поєднанні зі скиданням та оновленням інтервалу часу. Так досягається ефект каруселі.

Наступний функціональний ефект досягнуто в підсистемі обробки подій взаємодії з формами пошуку та бронювання. Після вибору дати, напрямку перевезень, кількості квитків заповнюється форма бронювання (рис. 3.10). Користувачу надходить повідомлення про підтвердження бронювання, деталі – на пошту. Нижній рисунок демонструє відповідний код на JavaScript (рис. 3.11).

Рисунок 3.10 – Заповнення форм бронювання

```
document.getElementById('booking-form').addEventListener('submit', function(e) {
    e.preventDefault(); // Скасування перезавантаження сторінки
    const formData = new FormData(this);
    console.log('Дані бронювання:', Object.fromEntries(formData));
    alert('Бронювання підтверджено!');
});
```

Рисунок 3.11 – Код на JavaScript для обробки подій форми

Механізм взаємодії з формами пошуку та бронювання.

1. Виклик функції `e.preventDefault()` блокує стандартне надсилання HTTP-запиту, що дозволяє обробити дані асинхронно (концепція AJAX) [27].
2. Об'єкт `FormData(this)` автоматично збирає всі значення полів форми (Name, Email, Class).

3. Конструкція `Object.fromEntries(formData)` трансформує зібрані дані у чистий JavaScript-об'єкт `key-value`, який готовий для відправки на майбутній серверний ендпоінт за допомогою `fetch()` або `axios()`.

Для покращення UX реалізовано алгоритм відстеження положення екрана користувача відносно цільових блоків під час скролінгу (рис. 3.12).

```

875     function smoothScroll(targetId) {
876         const targetElement = document.getElementById(targetId);
877         if (targetElement) {
878             const headerHeight = document.getElementById('main-header').offsetHeight;
879             const targetPosition = targetElement.getBoundingClientRect().top + window.pageYOffset - headerHeight;
880
881             window.scrollTo({
882                 top: targetPosition,
883                 behavior: 'smooth'
884             });
885         }
886     }
887
888     let lastActiveSection = 'features';

```

Рисунок 3.12 – Код на JavaScript, відповідальний за плавний скролінг сторінки

На завершення опису функціональності вебдодатку узагальнено заходи з оптимізації. До них відносять:

1. Графіка та мультимедіа. Вставлені візуальні елементи, які доповнюють тематику сторінки, такі як ілюстрації, фотографії та відео. При цьому використані формати, що підтримуються веббраузерами, для оптимального завантаження та відтворення. Окремо обрані спеціальні шрифти з бібліотеки Google, які надають виразності тексту.

2. Оптимізація контенту та ключові слова, яскравий та змістовний заголовок, який привертає увагу користувачів, а також підзаголовок, який уточнює основне повідомлення.

3. Оптимізація та тестування на різних пристроях. Готовий вебсайт проходив попереднє тестування з метою усунення як недоліків у написанні коду, так і для правильного відображення на різних екранах. У попередньому розділі було зазначено про використання флекс-елементів для досягнення адаптивності різних блоків. Приклад адаптивності всього додатку представлено на рис. 3.13. Як видно, всі елементи вибудовані в одну вертикальну колонку.

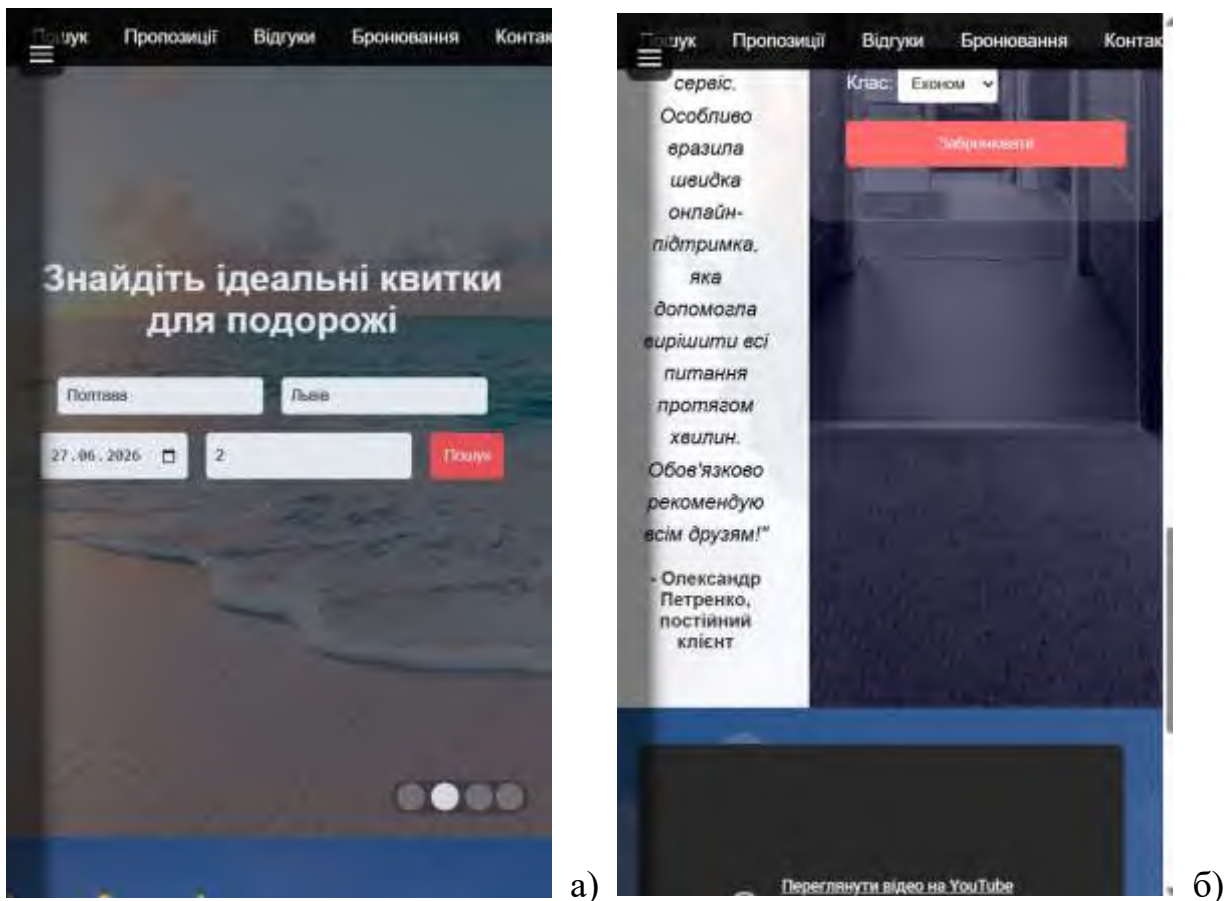


Рисунок 3.13 – Тестування вебдодатку при перегляді на смартфоні смартфоні, меню в згорнутому вигляді (а) та нижня частина (б)

Адаптивний дизайн спрямований на те, щоб вебсайти були візуально привабливими, незалежно від пристроїв, з яких користувачі отримують доступ до них (стаціонарного ПК, ноутбука, планшета або смартфона [28]). Адаптивний дизайн був розроблений для всього сайту (рис. 3.12).

Перед розгортанням вебсайту в мережі, було проведене тестування, щоб переконатися в її правильному відображенні та функціональності на різних пристроях та браузерях за допомогою сервісу [29]. Виявлені проблеми виправлені, а також оптимізоване завантаження сторінки для поліпшення її продуктивності та швидкості завантаження.

Для перетворення цього прототипу на повноцінний сервіс пошуку квитків та обробки замовлень користувачів і зв'язку з реальною базою даних будь-якої транспортної компанії API варто запропонувати клієнт-серверну архітектуру [30], основні компоненти якої представлені в табл. 3.1.

Таблиця 3.1 – Рекомендована архітектура та стек для Backend-частини

Компонент	Технологія	Обґрунтування
Backend API	Node.js (Express) або Python (FastAPI)	Висока швидкість обробки асинхронних запитів від користувачів під час пошуку квитків у реальному часі.
База даних (БД)	PostgreSQL	Реляційна СУБД для збереження транзакційних даних (користувачі, квитки, рейси, статуси замовлень).
Кешування	Redis	Кешування результатів пошуку найпопулярніших рейсів для зниження навантаження на основну БД.

В табл. 3.1 показані засоби і технології, які є основою для розвитку проєкту та створення повноцінної серверної частини, а в поясненні розробки функціоналу буде опис інтеграції фронтенду з бекендом. Зараз у коді стоять "заглушки" (`alert('Пошук виконано!')`), які можна замінити на код:

```

async function sendBookingData(data) {
  try {
    const response = await fetch('https://api.tickets-
service.com/v1/orders', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json'
},
      body: JSON.stringify(data)
    });
    const result = await response.json();
    if (response.ok) {
      alert(`Замовлення успішно створено! ID:
${result.orderId}`);
    }
  } catch (error) {
    console.error("Помилка обробки замовлення:",
error);
  }
}

```

Наведений код – опис заміни заглушки на реальний API-запит.

3.3 Перевірка та опис економічної ефективності розробки

Перевірка та опис економічної ефективності розробки проєкту передбачає детальну оцінку різних аспектів для визначення його життєздатності та ефективності у створенні економічних переваг.

Оцінка проведена методом витрат. Для розглянутого проєкту розробки, що включає створення вебдодатку з використанням HTML5, CSS3, SCSS, JS та деяких прикладних програм, були враховані фактори витрат, описані нижче [31].

1. Інструменти та програмне забезпечення для розробки. Використаний Visual Studio Code як редактор коду, який є безкоштовним і широко використовуваним інструментом у спільноті розробників. Він допоміг ефективно писати та керувати кодом.

2. Хостинг та інфраструктура. Firebase пропонує безкоштовний план хостингу, що дозволить мені розгорнути свій вебдодаток без жодних витрат на хостинг. Ця економічна опція забезпечила зручну та надійну платформу для розміщення додатку. В майбутньому компанія може оплатити власний домен.

3. Ретельно підібрані бібліотеки та пакети з відкритим вихідним кодом для свого проєкту дозволили уникнути накладних витрат, пов'язаних із придбанням або ліцензуванням програмного забезпечення.

4. Оскільки працював над проєктом самотійно, додаткових витрат на персонал не було. Загалом же використовується тариф на оплату праці програміста, вебдизайнера середнього рівня.

5. Витрати на інтеграцію та API. Розробляючи вебдодаток також врахував будь-які потенційні витрати, пов'язані з інтеграцією сторонніх сервісів або використанням зовнішніх API. Залежно від конкретних може виникнути потреба в інтеграції з платіжними шлюзами, постачальниками послуг або іншими зовнішніми системами.

Інтеграція з цими сервісами часто пов'язана з додатковими витратами, такими як комісія за транзакції, підписка або плата за використання. Важливо оцінити моделі ціноутворення та умови цих інтеграцій, щоб переконатися, що

вони відповідають бюджету та вимогам проєкту. З вище описаних витрат, могли бути витрати тільки за оплату за замовлення та комісію.

6. Обслуговування та підтримка. При розрахунках економічної вартості потрібно врахувати потенційні майбутні витрати на обслуговування та підтримку вебдодатку. Це включає такі фактори, як поточне обслуговування сервера (за наявності), виправлення помилок і розширення функцій. Якщо використовувати проєкт для багатьох користувачів, то потрібно врахувати витрати на базу вдалося ефективно керувати витратами. Ця економічна ефективність дозволила зосередитися на успішному виконанні проєкту без фінансових обмежень. Розрахунок типових витрат наведено в табл. 3.2.

Таблиця 3.2 – Перелік витрат при розробці комерційного вебсайту, станом на початок 2026 р.

№ п/п	Найменування послуги (зміст)	Вартість, грн
1	Планування та дизайн. Збір вимог, структурування та UX дизайну	9217,00
2	Frontend розробка. HTML/CSS кодування, реалізація UI	18536,00
3	Backend розробка. Налаштування бази даних, логіка на стороні сервера	18536,00
4	Інтеграція API (платіжний шлюз, тощо)	3000,00
5	Тестування та QA. Забезпечення якості, виправлення помилок	9217,00
6	Впровадження та запуск. Налаштування сервера, реєстрація домену використовуючи Firebase	370,00
7	Поточне обслуговування. Оновлення, виправлення помилок, технічна підтримка (терміном дії на місяць)	7414,00
8	Сумарні витрати на розробку, встановлення та обслуговування комерційного вебсайту	66290,00

Етап розробки завершений успішно, але в майбутньому можуть виникнути непередбачувані технічні проблеми. Ці проблеми можуть включати проблеми сумісності між різними версіями бібліотек або потенційні вразливості безпеки, які можуть з'явитися. Щоб зменшити ці ризики, необхідно залишатися в курсі останніх версій та проводити регулярний моніторинг. Поєднання використання економічно ефективних ресурсів, проведення ретельної оцінки ризиків та впровадження відповідних заходів сприяло успішному завершенню проєкту.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на тему «Проектування вебдодатку для сервісу пошуку квитків і обробки замовлень користувачів» досягнуто таких результатів.

1. Проведено аналіз сучасного стану сфери пасажирських перевезень та встановлено, що цифровізація транспортних послуг є одним із пріоритетних напрямів розвитку туристичної галузі, оскільки сприяє підвищенню доступності, оперативності та якості обслуговування користувачів.

2. Досліджено сучасні технології та засоби розробки вебдодатків, зокрема особливості застосування клієнтських технологій HTML5, CSS3 та JavaScript, а також принципи побудови адаптивних вебінтерфейсів.

3. Виконано аналіз існуючих сервісів пошуку та бронювання автобусних квитків, що дозволило виявити їхні переваги та недоліки, а також визначити функціональні вимоги до розроблюваного програмного продукту.

4. Спроектовано структуру вебдодатку на основі модульного підходу, який передбачає логічне розділення функціональних компонентів системи та забезпечує можливість її подальшого розширення.

5. Розроблено вебдодаток для туристичного агентства, який реалізує функції пошуку автобусних маршрутів, бронювання квитків та обробки замовлень користувачів у єдиному інформаційному середовищі.

6. Створено адаптивний користувацький інтерфейс із використанням сучасних принципів UI/UX-дизайну, що забезпечує комфортну взаємодію користувачів із системою на персональних комп'ютерах, планшетах та мобільних пристроях.

7. Реалізовано інтерактивні механізми навігації, анімаційні ефекти, мультимедійні компоненти та засоби візуального супроводу, які покращують користувацький досвід та підвищують привабливість вебдодатку.

Особливістю розробленого програмного продукту стало використання класичного стеку вебтехнологій без залучення складних програмних платформ і

сторонніх фреймворків, що дозволило забезпечити високу швидкодію, простоту супроводу та мінімізувати обчислювальні витрати.

Практична цінність роботи полягає у створенні готового до використання вебдодатку, який може бути впроваджений у діяльність туристичних агентств та підприємств, що організовують автобусні пасажирські перевезення по Україні та країнах Європи. Запропоноване рішення може слугувати основою для подальшого розвитку повноцінної інформаційної системи з розширеними функціональними можливостями.

Перспективами подальшого розвитку вебдодатку є інтеграція серверної частини та системи керування базами даних, автоматизація формування електронних квитків, реалізація особистого кабінету користувача, впровадження онлайн-оплати, підключення геолокаційних сервісів, інтеграція із зовнішніми системами перевізників та застосування механізмів аналітики користувацьких даних.

Отримані результати підтверджують, що використання сучасних вебтехнологій дозволяє створювати ефективні програмні продукти для автоматизації транспортно-туристичних послуг, підвищувати якість обслуговування клієнтів та сприяти цифровій трансформації сфери пасажирських перевезень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-сайт: визначення й застосування. URL: <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site> (дата звернення 20.04.2026).
2. Пелешишин А. М., Тимовчак-Максимець О. Ю. Інтернет-технології та ресурси : навчальний посібник. Львів : Видавництво Львівської політехніки, 2021. 244 с.
3. HTML basics. URL: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics (дата звернення 20.04.2026).
4. Duckett J. HTML and CSS: Design and Build Websites. Indianapolis : John Wiley & Sons, 2019. 490 p.
5. Usage statistics and market shares of JavaScript libraries. *W3Techs*: вебсайт. URL: https://w3techs.com/technologies/overview/javascript_library (дата звернення: 27.04.2026).
6. Vue vs React vs... Svelte?! *Medium*. URL: <https://medium.com/@faulknerproject/vue-vs-react-vs-svelte-5f93d70d2618> (дата звернення: 27.04.2026).
7. Рейтинг мов програмування 2026. <https://dou.ua/lenta/articles/language-rating-2026/> (дата звернення 20.04.2026).
8. Desktop vs Mobile vs Tablet Market Share Worldwide. Apr 2023 - Apr 2024. Statcounter: вебсайт. URL: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet> (дата звернення 20.04.2026).
9. Галишич Р. Я., Богуславський М. В. Принципи комунікативного дизайну інтерактивної мультимедійної вебінфографіки в сучасних інформаційних системах. *Український мистецтвознавчий дискурс*. 2024. № 4. С. 44–52. DOI: 10.32782/uad.2024.4.6.
10. Ратушняк Т. В., Омельчук А. В., Гладченко О. В. та ін. Інклюзивний вебдизайн: стандарти та цифрові інструменти тестування доступності вебсайту.

Applied Questions of Mathematical Modeling. 2024. Т. 7, № 2. С. 189–206. DOI: 10.32782/mathematical-modelling/2024-7-2-17.

11. Busfor. Busline: вебсайт. URL: <https://busline.info/uk/busfor> (дата звернення 20.04.2026).

12. Купуйте квитки на автобус, поїзд і літак. Infobus. URL: <https://infobus.eu/ua/bus> (дата звернення 20.04.2026).

13. FlixBus. URL: <https://www.flixbus.ua/> (дата звернення 20.04.2026).

14. Метод аналізу ієрархій. URL: <https://dss.knsa.chdtu.edu.ua/ahp-help> (дата звернення 20.04.2026).

15. Чаус І. В., Марусенкова Т. В. Front-end Framework for Building Applications With Adaptive User Interfaces Using Machine Learning Methods. *Computer Systems and Networks*. 2024. Vol. 6, No. 2. P. 252–267. DOI: 10.23939/csn2024.02.252.

16. Близнюк С. В., Онофрійчук О. В. Адаптивний веб-дизайн і його роль у мобільному маркетингу. *Економіка та суспільство*. 2025. № 82. С.1-6.

17. Що таке HTML5: вебсайт. URL: <https://www.hostinger.com.ua/rukovodstva/chto-takoe-html-i-ih-razlichiya> (дата звернення: 27.04.2026).

18. CSS: Cascading Style Sheets. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення 20.04.2026).

19. IDE in Software: Definition & Examples. URL: <https://study.com/academy/lesson/ide-in-software-definition-examples.html> (дата звернення: 27.04.2026).

20. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення 20.04.2026).

21. Using Git source control in VS Code. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення 20.04.2026).

22. IntelliSense. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення 20.04.2026).

23. Personalize Visual Studio Code. *Visual Studio Code*: вебсайт. URL: <https://code.visualstudio.com/> (дата звернення 20.04.2026).

24. Що таке аудит сайту?.: вебсайт. URL: <https://polyarix.com/blog/chto-takoe-audit-sajta/> (дата звернення: 27.04.2026).

25. Михайлов А. В., Йохна В. В. Еволюція дизайну сайтів як результат адаптації маркетингових інструментів до змін психологічного сприйняття інформації цільовими аудиторіями. *Вісник Хмельницького національного університету. Економічні науки*. 2024. № 336. С. 61–68. DOI: 10.31891/2307-5740-2024-336-9.

26. MDN. URL: <https://developer.mozilla.org/en-US/docs/Web/CSS/Reference/Properties/transform> (дата звернення 20.04.2026).

27. Що таке JavaScript?. URL: https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript#a_high-level_definition] (дата звернення 20.04.2026).

28. Del Castillo R. Top Café Website Designs That Drove Conversions, Bookings & Revenue. DesignRush. 2025. URL: <https://www.designrush.com/best-designs/websites/trends/cafe-website-design> (дата звернення: 04.06.2026).

29. DefineProperties. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/defineProperties (дата звернення 20.04.2026).

30. Копішинська О. П., Уткін Ю. В., Скриль В.К. Особливості технологій і архітектури вебсервісів. *Інтеграція інформаційних систем і інтелектуальних технологій в умовах трансформації інформаційного суспільства*: Матер. IV Міжнар. науково-практичної конференції, Полтава, 21-22 жовтня 2021 року. Полтава: ПДАУ, 2021 С.29-33.

31. Оцінка економічної ефективності Інтернет-ресурсів. : вебсайт. URL: http://dipplus.com.ua/metodicheskiye-ukazaniya-i-informatsiya/article_post/otsinka-yekonomichnoi-yefektivnosti-internet-resursiv (дата звернення: 04.05.2026).