

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

**Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки**

«До захисту в ЕК»
Директор інституту(декан факультету)
_____ Андрій ФОРСЮК
(підпис) (ім'я та прізвище)

«02» червня 2025 р.

«До захисту допущено»
Завідувач кафедри
_____ Сергій ГРИБКОВ
(підпис) (ім'я та прізвище)

«02» червня 2025 р.

**КВАЛІФІКАЦІЙНА РОБОТАХ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності _____ 122 «Комп'ютерні науки»
(код і назва спеціальності)
освітньо-професійної програми «Комп'ютерні науки»
на тему: Розроблення веб-орієнтованої системи управління замовленнями
пасажирських перевезень ФОП Салавор Р.Т.

Виконав: здобувач 4 курсу, групи КН-4-3

_____ Салавор Максим Романович
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник _____ Мошенський Андрій Олександрович
(підпис)

Консультанти _____
(ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

Рецензент _____
(ім'я та прізвище) (підпис)

Я як здобувач Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав і не одержував недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ – 2025р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних системКафедра Інформаційних технологій, штучного інтелекту і кібербезпекиОсвітній ступінь БакалаврСпеціальність 122 Комп'ютерні науки

(код і назва)

Освітньо-професійна програма «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних
технологій, штучного інтелекту і
кібербезпеки Сергій ГРИБКОВ

«28» квітня 2025 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Салавора Максима Романовича

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення веб-орієнтованої системи управління
замовленнями пасажирських перевезень ФОП Салавор Р.Т.керівник роботи Мошенський Андрій Олександрович, к.т.н., доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)затвержені наказом закладу вищої освіти від «28» квітня 2025 р. № 254-кв2. Строк подання здобувачем роботи: 30.05.20253. Вихідні дані до роботи: загальні відомості про ринок пасажирських
транспортних перевезень.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

1. Дослідження предметної області та постановка задачі
2. Технічне завдання на проектування
3. Проектування, створення та апробація інформаційної системи

5. Перелік графічного матеріалу:

1. Організаційно-функціональна схема підприємства.2. Скріншоти інтерфейсу користувача системи.

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Мошенський А. О. ,доцент, кандидат технічних наук	28.04.2025	10.05.2025
2	Мошенський А. О. ,доцент, кандидат технічних наук	28.04.2025	17.05.2025
3	Мошенський А. О. ,доцент, кандидат технічних наук	28.04.2025	23.05.2025

Дата видачі завдання: 28.04.2025

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області та постановка задачі	29.04.2025-01.05.2025	Виконано
2	Проектування бази даних	29.04.2025-05.05.2025	Виконано
3	Розробка back end частини програми	05.05.2025-20.05.2025	Виконано
4	Розробка front end частини програми	05.05.2025-25.05.2025	Виконано
5	Оформлення пояснювальної записки та створення презентації	15.05.2025-28.05.2025	Виконано

Здобувач

_____ (підпис)

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

_____ (прізвище та ініціали)

АНОТАЦІЯ

Тема диплому: “Розроблення веб-орієнтованої системи управління замовленнями пасажирських перевезень ФОП Салавор Р. Т.”

У цій кваліфікаційній роботі розглядається процес розробки та впровадження веборієнтованої інформаційної системи для автоматизації пасажирських перевезень на замовлення. Основною метою роботи є вдосконалення процесу обробки замовлень, бронювання транспорту, взаємодії з клієнтами та підвищення ефективності роботи компанії, що надає послуги з пасажирських перевезень.

У ході дослідження проведено аналіз предметної області, сформульовано технічне завдання, розроблено структуру бази даних, інтерфейс користувача, а також реалізовано функціонал реєстрації, авторизації, бронювання, зворотного зв'язку та перегляду історії замовлень. Система створена з використанням технологій React, Node.js, Express та MongoDB.

Робота складається з 74 сторінок, містить 46 рисунків, 8 таблиць, 21 використаних джерел.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, ПАСАЖИРСЬКІ ПЕРЕВЕЗЕННЯ, АВТОМАТИЗАЦІЯ, БРОНЮВАННЯ, ВЕБЗАСТОСУНОК, ІНТЕРФЕЙС КОРИСТУВАЧА, MONGODB, NODE.JS, REACT, КОРИСТУВАЧ.

ANNOTATION

Diploma topic: "Development of a web-oriented system for managing passenger transportation orders for sole proprietor Salavor R. T."

This qualification work examines the process of developing and implementing a web-based information system for automating on-demand passenger transportation services. The main goal of the work is to improve the process of handling orders, booking transport, interacting with clients, and increasing the efficiency of the company providing passenger transportation services.

During the study, an analysis of the subject area was conducted, a technical specification was formulated, a database structure and user interface were designed, and the functionality for registration, authorization, booking, feedback, and order history viewing was implemented. The system was developed using React, Node.js, Express, and MongoDB technologies.

The work consists of 74 pages, includes 46 figures, 8 tables, and 21 references.

Keywords: INFORMATION SYSTEM, PASSENGER TRANSPORTATION, AUTOMATION, BOOKING, WEB APPLICATION, USER INTERFACE, MONGODB, NODE.JS, REACT, USER.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	8
1.1 Загальна характеристика підприємства	8
1.2 Організаційна структура підприємства.....	11
1.3. Розробка функціональної моделі та аналіз існуючих бізнес-процесів	16
1.4. Аналіз нинішнього стану комп'ютеризації	17
1.4.1.Задачі автоматизації.....	18
1.5. Огляд існуючих рішень для розв'язання виявлених проблем.....	20
1.6. Обґрунтування доцільності проектування й розроблення веб-сайту для надання послуг з пасажирських перевезень на замовлення	25
1.7. Розрахунок економічного ефекту від впровадження системи	26
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ.....	30
2.1. Загальні положення.....	30
2.2. Призначення і цілі створення системи.....	30
2.3. Характеристика об'єкта автоматизації.	31
2.4. Вимоги до системи.....	31
2.5. Склад і зміст робіт по створенню системи	36
2.6. Порядок контролю і приймання системи	36
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію	36
2.8. Вимоги до документації	37
2.9. Джерела розробки	37
РОЗДІЛ 3. ПРОЄКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	38
3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення інформаційної системи	38
3.2. Проєктування та створення бази даних.....	40
3.3. Реалізація функцій системи.....	44
3.4. Інструкція користувача	60
3.4. Тестування програмного продукту	69
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	73
ДОДАТКИ.....	76

ВСТУП

У сучасному світі транспортні послуги відіграють важливу роль у забезпеченні мобільності населення, зокрема, коли йдеться про замовлення пасажирських перевезень. Компанії, що працюють у цьому сегменті, повинні забезпечувати високий рівень обслуговування клієнтів, оперативно реагувати на запити та ефективно управляти своїми ресурсами. Разом з тим, використання застарілих підходів до організації роботи, таких як ручна обробка замовлень та навіть відсутність веб-сайту, призводять до зниження якості послуг і втрати клієнтів.

Автоматизація бізнес-процесів у сфері пасажирських перевезень стає необхідною умовою для забезпечення конкурентоспроможності компанії та її подальшого розвитку. Впровадження сучасної інформаційної системи дозволяє представити послуги компанії на більший ринок, створити зручні умови для онлайн-бронювання транспорту, забезпечувати оперативний контроль за виконанням замовлень і створити зручний інтерфейс для клієнтів.

Метою кваліфікаційної роботи є розроблення вебзастосунку для управління замовленнями пасажирських перевезень. Система спрямована на автоматизацію процесів обробки замовлень, бронювання транспорту, надсилання повідомлень та представлення послуг компанії.

Дослідження теми є актуальним, оскільки в умовах стрімкого розвитку технологій і підвищення вимог споживачів до якості сервісу, лише інноваційні підходи дозволяють транспортним компаніям залишатися конкурентоспроможними та задовольняти потреби клієнтів.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальна характеристика підприємства

Підприємство ФОП «Салавор Р.Т.» спеціалізується на наданні послуг пасажирських перевезень на замовлення. Його історія почалась більше 10 років тому з одного мікроавтобуса, який використовувався для перевезення невеликих груп пасажирів. Завдяки якісному обслуговуванню, надійності та індивідуальному підходу до кожного клієнта, компанія поступово розширювалася, формуючи сталу клієнтську базу, яка повертається за послугами знову і знову.

Сьогодні підприємство має невеликий, але сучасний автопарк, який включає мікроавтобуси та автобуси різної місткості. Це дозволяє компанії задовольняти потреби клієнтів у різних типах перевезень, від невеликих груп до великих колективів.

Основні види послуг:

1. Організація перевезень для корпоративних клієнтів:

- Розвезення персоналу: Компанія надає послуги ранкового та вечірнього розвезення співробітників до місць роботи та назад. Це особливо актуально для підприємств, які розташовані далеко від громадського транспорту або мають ненормований графік роботи.

- Транспортування на корпоративні заходи: Компанія організовує перевезення співробітників на тренінги, конференції, семінари та інші корпоративні заходи. Замовники можуть розраховувати на точність, пунктуальність та комфорт під час поїздки.

- Щоденні маршрути: Для постійних клієнтів компанія пропонує розробку індивідуальних маршрутів, які враховують потреби підприємства та його співробітників.

2. Перевезення для приватних клієнтів:

- Трансфери на святкові заходи: Компанія організовує перевезення гостей на весілля, ювілеї, дні народження та інші урочисті події. Клієнти можуть

замовити транспорт для себе або для своїх гостей, що дозволяє забезпечити комфортну та безпроблемну організацію заходу.

- Екскурсійні поїздки: Компанія пропонує організацію екскурсійних поїздок до інших міст, туристичних об'єктів та пам'яток природи. Це ідеальний варіант для груп, які хочуть насолодитися подорожжю без необхідності турбуватися про транспорт.

- Перевезення на масові заходи: Компанія забезпечує транспортування груп пасажирів на спортивні змагання, концерти, фестивалі та інші масові заходи. Це дозволяє клієнтам дістатися до місця проведення заходу вчасно та без зайвих зусиль.

3. Міжміські перевезення:

- Комфортні поїздки між містами: Компанія надає послуги перевезення груп пасажирів між містами України. Це може бути як одноразова поїздка, так і регулярні рейси для постійних клієнтів.

- Туристичні трансфери: Компанія організовує перевезення туристичних груп до місць відпочинку, готелів, аеропортів та інших об'єктів. Це дозволяє туристам зосередитися на відпочинку, не турбуючись про транспортні питання.

Особливості роботи підприємства:

1. Індивідуальний підхід до кожного клієнта:

- Компанія працює за принципом врахування потреб кожного клієнта. Незалежно від того, чи це корпоративний клієнт, який потребує регулярного транспортування співробітників, чи приватна особа, яка організовує весілля, компанія пропонує гнучкі умови щодо маршруту, часу та типу транспорту.

- Кожен замовник може розраховувати на консультацію щодо вибору оптимального маршруту, типу транспортного засобу та інших деталей, що забезпечують комфортну подорож.

2. Надійний та сучасний автопарк:

- У розпорядженні компанії є мікроавтобуси та автобуси різної місткості, що дозволяє обслуговувати як невеликі групи (8-21 осіб), так і великі колективи (до 50 осіб).

- Всі транспортні засоби обладнані кондиціонерами, зручними сидіннями, багажними відділеннями та іншими зручностями, що забезпечують комфорт під час поїздки.

- Автопарк регулярно проходить технічне обслуговування, що гарантує безпеку та надійність перевезень.

3. Кваліфікований персонал:

- Водії компанії мають великий досвід роботи, регулярно проходять навчання та інструктажі з техніки безпеки. Це дозволяє забезпечити високий рівень сервісу та мінімізувати ризики під час перевезень.

- Кожен водій компанії знає місцеві дороги, що дозволяє уникнути затримок та забезпечити пунктуальність.

4. Доступні ціни та гнучкі умови співпраці:

- Компанія пропонує конкурентні ціни на свої послуги, що робить її доступною для широкого кола клієнтів.

- Для постійних клієнтів передбачені знижки та спеціальні умови співпраці, що робить послуги компанії ще більш привабливими.

5. Екологічна відповідальність:

- Компанія дбає про екологічну безпеку, використовуючи транспортні засоби, які відповідають сучасним екологічним стандартам. Це дозволяє мінімізувати вплив на навколишнє середовище.

Клієнтська база:

До клієнтів компанії належать як приватні особи, так і корпоративні замовники. Серед останніх — невеликі та середні підприємства, які потребують транспортування своїх співробітників, а також туристичні агенції, що організовують екскурсійні поїздки. Завдяки якісному обслуговуванню та

доступним цінам, компанія формує сталу клієнтську базу, яка повертається за послугами повторно.

Перспективи розвитку:

Підприємство, незважаючи на свої скромні масштаби, активно прагне до розвитку. Серед основних планів:

- Розширення автопарку: Закупівля нових транспортних засобів для обслуговування більшої кількості клієнтів.
- Впровадження нових послуг: Організація додаткових маршрутів, включаючи міжнародні перевезення.
- Партнерські відносини: Розвиток співпраці з туристичними компаніями, корпоративними клієнтами та організаторами масових заходів.
- Підвищення якості сервісу: Впровадження нових стандартів обслуговування, щоб забезпечити максимальний комфорт для пасажирів.

1.2 Організаційна структура підприємства

Центральне місце в структурі підприємства займає Директор підприємства. Він визначає стратегічний напрямок розвитку компанії та відповідає за загальне керівництво. Директору підпорядковані такі ключові посадові особи:

1. Заступник директора з перевезень відповідає за організацію та контроль перевезень, управління автопарком та водіями. Його завдання включають планування маршрутів, контроль за виконанням замовлень та забезпечення ефективного використання транспортних засобів.

2. Заступник директора з обслуговування клієнтів забезпечує якісне обслуговування клієнтів, взаємодію з клієнтською базою та розвиток сервісу. Він відповідає за обробку запитів, надання інформації про послуги та забезпечення високого рівня задоволеності клієнтів.

3. Головний бухгалтер відповідає за фінансовий облік, звітність та бюджетування компанії. Його завдання включають контроль за фінансовими потоками, складання звітів та взаємодію з податковими органами.

4. Голова юридичного відділу забезпечує правову підтримку компанії, укладання договорів та дотримання законодавчих норм. Він також відповідає за ліцензії та дозволи, необхідні для виконання перевезень.

5. Головний диспетчер координує роботу диспетчерської служби, яка відповідає за планування маршрутів, розподіл замовлень та контроль за виконанням рейсів. Він забезпечує оперативну взаємодію між клієнтами, водіями та іншими підрозділами компанії.

Основні підрозділи компанії:

1. Відділ перевезень відповідає за організацію та виконання пасажирських перевезень. Його основні завдання включають планування маршрутів з урахуванням потреб клієнтів, контроль за виконанням замовлень на перевезення, забезпечення своєчасного та безпечного перевезення пасажирів, організацію технічного обслуговування та ремонту транспортних засобів, а також координацію роботи водіїв та механіків для забезпечення ефективної роботи автопарку.

2. Відділ обслуговування пасажирів забезпечує комунікацію з клієнтами, надає інформацію про послуги, обробляє запити та замовлення. Відділ також відповідає за збір зворотного зв'язку від клієнтів для покращення якості сервісу.

3. Бухгалтерія відповідає за фінансовий облік, звітність та розрахунки з клієнтами та постачальниками. Відділ забезпечує точність фінансових операцій та дотримання податкового законодавства.

4. Юридичний відділ забезпечує правову підтримку компанії, укладання договорів та дотримання законодавчих норм. Відділ також відповідає за ліцензії та дозволи, необхідні для виконання перевезень [21]

5. Диспетчерська служба координує замовлення, планує маршрути та розподіляє ресурси для виконання перевезень. Відділ забезпечує оперативну взаємодію між клієнтами, водіями та іншими підрозділами компанії.

На рисунку 1.1 зображена організаційно-функціональна схема підприємства ФОП «Салавор Р. Т».

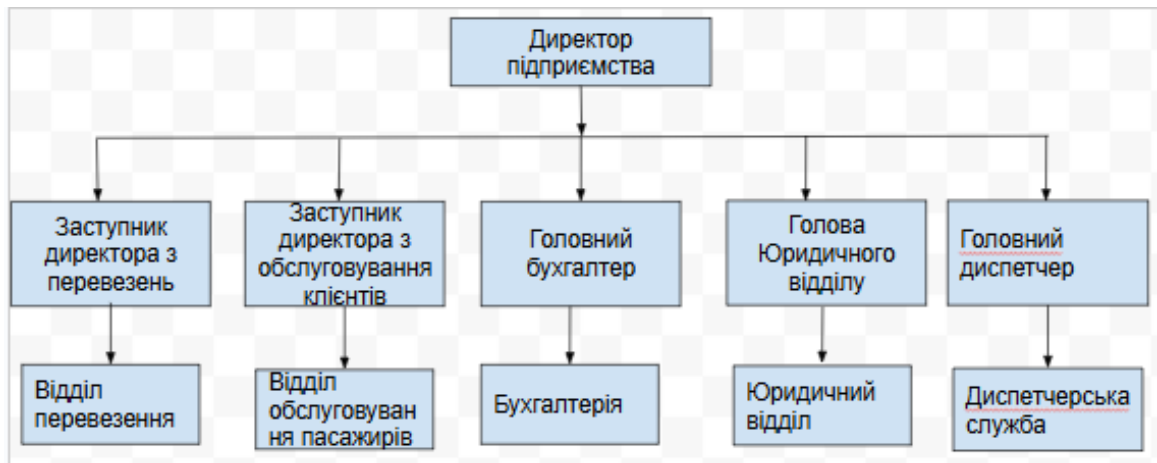


Рисунок 1.1 – Організаційно-функціональна схема підприємства з пасажирських перевезень

Організаційна структура відділу перевезень:

- Керівник відділу перевезень

Відповідає за загальне управління відділом, планування маршрутів та контроль за виконанням замовлень. Він також координує роботу водіїв та механіків.

- Водії

Виконують перевезення пасажирів, дотримуючись встановлених маршрутів та графіків. Водії відповідають за безпеку пасажирів та дотримання правил дорожнього руху.

- Механіки

Забезпечують технічний стан транспортних засобів, проводять технічне обслуговування та ремонт. Вони відповідають за те, щоб транспортні засоби були готові до виконання рейсів.

На рисунку 1.2 зображена організаційна структура відділу перевезення.



Рисунок 1.2. – Структурна схема відділу перевезень

Основним завданням компанії з пасажирських перевезень на замовлення є забезпечення якісного обслуговування клієнтів під час організації перевезень, зручної взаємодії між клієнтами, диспетчерами та водіями, а також оптимізація процесів онлайн замовлень і виконання поїздок. Важливими аспектами є індивідуальний підхід до кожного замовлення, створення пропозицій відповідно до потреб клієнтів, оперативна обробка запитів, надання повної інформаційної підтримки на всіх етапах обслуговування, включно з вибором транспорту, маршруту та розрахунком вартості, а також збір і аналіз зворотного зв'язку для підвищення якості сервісу [17].

Основні функції, що виконує відділ перевезення наведені в таблиці 1.1.

Таблиця 1.1. – Завдання і функції відділу перевезення

№1	Задачі	Функції
1	Підготовка транспорту до перевезень	<ul style="list-style-type: none"> - Перевірка технічного стану транспортних засобів перед рейсом. - Організація технічного обслуговування та ремонту транспорту. - Забезпечення відповідності транспорту вимогам безпеки та комфорту.
2	Призначення водіїв	<ul style="list-style-type: none"> - Планування розкладу роботи водіїв відповідно до маршрутів і графіків. - Призначення водіїв на конкретні рейси. - Контроль виконання норм робочого часу водіїв

№1	Задачі	Функції
		<ul style="list-style-type: none"> - Інструктаж перед рейсом: ознайомлення водіїв із деталями маршруту, умовами перевезення та специфічними вимогами клієнтів. - Моніторинг стану здоров'я водіїв: перевірка самопочуття водіїв перед рейсами, забезпечення їх дотримання режиму відпочинку.
3	Організація перевезень	<ul style="list-style-type: none"> - Формування маршрутів із врахуванням оптимального використання транспорту. - Координація водіїв під час виконання рейсів. - Виконання перевезення: забезпечення своєчасного прибуття транспорту до місця відправлення та прибуття до пункту призначення. - Забезпечення відповідності перевезень нормативним вимогам та стандартам безпеки.

Таблиця 1.2. Взаємодія відділу перевезень з іншими відділами компанії

№	Відділ	Одержання	Надання
1	Диспетчерська служба	<ul style="list-style-type: none"> - Інформація про замовлення (маршрут, час, кількість пасажирів). - Дані про доступність транспорту. 	<ul style="list-style-type: none"> - Затверджені графіки рейсів для виконання замовлень. - Списки призначених водіїв і транспорту.
2	Відділ обслуговування пасажирів	<ul style="list-style-type: none"> - Дані про вимоги клієнтів до перевезень (додаткові послуги, спеціальні запити). 	<ul style="list-style-type: none"> - Затверджені маршрути і транспорт для інформування клієнтів.
3	Юридичний відділ	<ul style="list-style-type: none"> - Договори на виконання перевезень для клієнтів. - Погодження ліцензій та дозволів для виконання замовлень. 	<ul style="list-style-type: none"> - Документи на транспортні послуги для укладання договорів із клієнтами.
4	Бухгалтерія	<ul style="list-style-type: none"> - Дані про укладені замовлення та надані послуги. 	<ul style="list-style-type: none"> - Рахунки на оплату за виконані замовлення. - Звіти про витрати на виконання рейсів.

1.3. Розробка функціональної моделі та аналіз існуючих бізнес-процесів

Для ефективного управління процесами пасажирських перевезень була розроблена функціональна модель, яка охоплює всі ключові етапи від замовлення клієнта до завершення поїздки. Ця модель дозволяє оптимізувати роботу компанії, забезпечуючи високий рівень сервісу та оперативне виконання замовлень. Основні етапи бізнес-процесу включають:

1. Замовлення клієнта: процес починається зі звернення клієнта до компанії для замовлення перевезення. Відділ обслуговування пасажирів обробляє запит, уточнюючи деталі, такі як місце призначення, час виїзду, кількість пасажирів та інші побажання. Цей етап є важливим для забезпечення індивідуального підходу до кожного клієнта.

2. Планування маршруту: після отримання замовлення диспетчерська служба розробляє оптимальний маршрут, враховуючи побажання клієнта та доступність транспортних засобів. На цьому етапі визначається тип транспортного засобу (автобус, мікроавтобус тощо), який найкраще підходить для конкретного замовлення.

3. Підготовка транспорту: відділ перевезень забезпечує готовність транспортного засобу до рейсу. Це включає перевірку технічного стану автомобіля, призначення водія та проведення інструктажу щодо маршруту та умов перевезення. Важливо, щоб транспорт був у належному стані, а водій був добре підготовлений до виконання завдання.

4. Виконання перевезення: водій вирушає за маршрутом, дотримуючись графіку та правил безпеки. Диспетчерська служба здійснює моніторинг руху транспорту, що дозволяє оперативно реагувати на можливі зміни, такі як затори або інші непередбачувані обставини.

5. Прибуття до місця призначення: пасажири благополучно прибувають до пункту призначення, і клієнт отримує підтвердження завершення поїздки. Цей етап є завершальним у процесі перевезення, і його успішне виконання забезпечує високий рівень задоволеності клієнтів.

6. Збирання та обробка відгуків: після завершення поїздки відділ обслуговування пасажирів збирає зворотний зв'язок від клієнтів. Відгуки аналізуються для покращення якості сервісу та виявлення можливих проблем. Цей етап є важливим для постійного вдосконалення роботи компанії.

На рисунку 1.3 представлено бізнес-процеси, від замовлення клієнта до завершення замовлення.

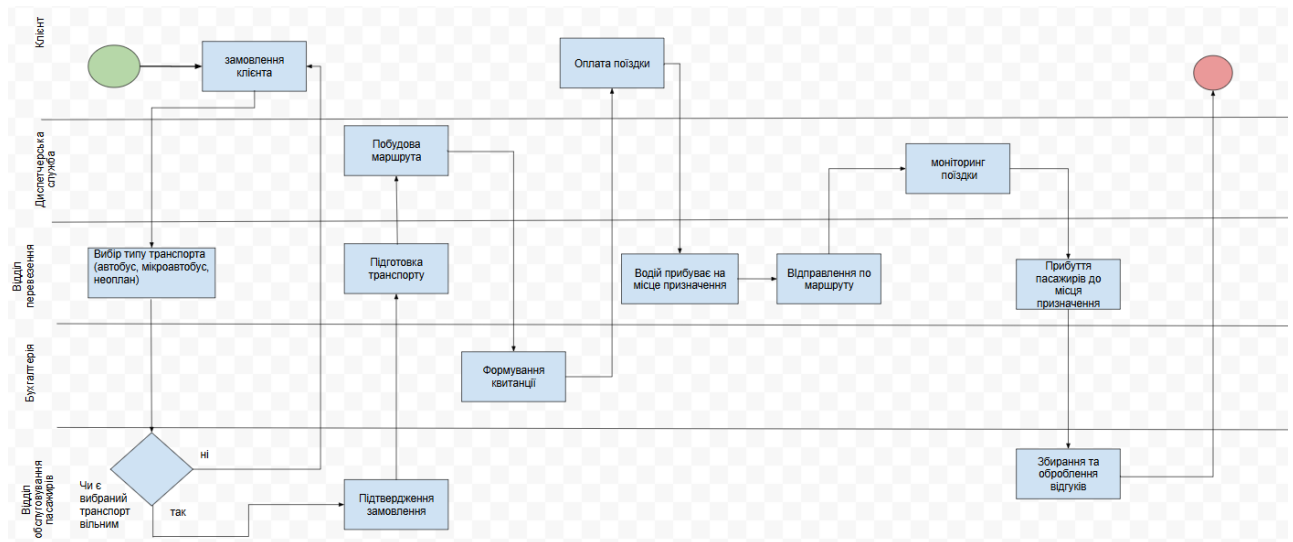


Рисунок 1.3 – Бізнес-процеси від замовлення клієнта до завершення замовлення

1.4. Аналіз нинішнього стану комп'ютеризації

Найбільш очевидною проблемою є відсутність веб-сайту для замовлення послуг та системи онлайн-бронювання. Це обмежує доступ клієнтів до послуг компанії, ускладнює процес замовлення перевезень та призводить до втрати потенційних клієнтів. У сучасному світі, де більшість клієнтів обирають зручність та швидкість онлайн-замовлень, відсутність можливості ознайомлення з послугами через Інтернет може відлякати потенційних клієнтів. Конкуренти, які мають веб-присутність, автоматично стають більш привабливими для клієнтів через більшу доступність та зручність.

Відсутність веб-сайту для замовлення послуг ускладнює процес взаємодії з клієнтами. Клієнти не мають можливості зручно замовляти перевезення онлайн, переглядати доступні маршрути чи розраховувати вартість послуг. Це обмежує

доступ до послуг компанії та ускладнює процес замовлення. Замість автоматизованої системи, яка б дозволяла клієнтам бронювати перевезення в будь-який час, компанія використовує ручну обробку замовлень, що потребує більше часу та людських ресурсів.

Відсутність онлайн-інструментів обмежує можливості просування послуг компанії. На сьогоднішній день соціальні мережі, контекстна реклама та інші онлайн-канали є найефективнішими способами залучення нових клієнтів. Відсутність цих інструментів призводить до втрати потенційних замовлень. Досі клієнти зверталися до компанії завдяки рекомендаціям інших людей, фізичній рекламі та через постійних клієнтів. Проте створення веб-сайту та його SEO-просування дозволило б значно розширити аудиторію, залучаючи клієнтів через пошукові системи. Оптимізація сайту за ключовими запитами, налаштування правильного контенту та покращення технічних характеристик дозволили б займати вищі позиції в пошукових результатах, що сприяло б збільшенню кількості замовлень.

Наслідки відсутності комп'ютеризації є значними. Без онлайн-платформи компанія втрачає клієнтів, які віддають перевагу зручним та швидким способам замовлення послуг. Конкуренти, які вже мають веб-сайти та автоматизовані системи, можуть бути більш привабливими для клієнтів через більшу доступність та зручність. Відсутність автоматизації призводить до ручної обробки замовлень, що потребує більше часу, ніж автоматизована система, і створює додаткове навантаження на персонал. Це негативно впливає на ефективність роботи компанії та її конкурентоспроможність на ринку.

1.4.1. Задачі автоматизації

Для усунення наявних проблем і покращення ефективності бізнес-процесів підприємства з пасажирських перевезень необхідно впровадити низку заходів з автоматизації. Основною задачею є розробка та запуск веб-сайту, який стане основним інструментом для залучення нових клієнтів та покращення якості обслуговування. Веб-сайт повинен включати каталог послуг компанії, детальну

інформацію про автопарк, можливість онлайн-бронювання перевезень, а також форму для залишення контактних даних, щоб клієнти могли легко зв'язатися з компанією. Це зробить послуги доступними для більшої аудиторії та полегшить процес замовлення, забезпечивши зручність для клієнтів.

Однією з ключових функцій майбутньої інформаційної системи буде можливість оформлення замовлення на перевезення. Клієнти зможуть вибирати час, дату, маршрут, тип транспорту та кількість пасажирів, що дозволить їм отримати персоналізовану послугу. Система також автоматично розраховуватиме приблизну вартість перевезення залежно від обраного маршруту, кількості пасажирів та класу транспорту. Це дозволить клієнтам отримати точну інформацію про вартість послуги ще до оформлення замовлення.

Для залучення нових клієнтів необхідно активно використовувати онлайн-інструменти, такі як соціальні мережі, контекстна реклама та електронна пошта. SEO-оптимізація сайту дозволить залучати клієнтів через пошукові системи, що значно розширить аудиторію компанії. Наразі основним джерелом залучення клієнтів є рекомендації, фізична реклама та постійні клієнти, проте впровадження онлайн-інструментів дозволить вийти на новий рівень розвитку.

Для успішного впровадження нових систем необхідно провести навчання персоналу. Працівники повинні навчитися ефективно використовувати нові інструменти та технології, щоб забезпечити безперебійну роботу системи. Для адміністратора компанії буде розроблена адміністративна панель, яка дозволить управляти заявками на перевезення, моніторити статуси замовлень, переглядати та редагувати дані про клієнтів та транспорт. Диспетчери зможуть обробляти та розподіляти заявки серед водіїв.

Також важливим кроком є встановлення механізмів моніторингу та звітності. Це дозволить відстежувати ефективність системи, виявляти можливості для подальшого вдосконалення та адаптуватися до змін у ринкових умовах та потребах клієнтів. Збір даних про замовлення, відгуки клієнтів та інші показники допоможе компанії приймати обґрунтовані рішення для покращення якості послуг.

Розробка сайту та впровадження системи онлайн-замовлення є ключовими кроками для підвищення конкурентоспроможності компанії. Це дозволить не лише залучити нових клієнтів, але й покращити внутрішні процеси, зробивши їх більш ефективними та зручними для клієнтів. Впровадження цих заходів стане важливим кроком у розвитку компанії та забезпечить її успіх на ринку пасажирських перевезень.

1.5. Огляд існуючих рішень для розв'язання виявлених проблем

Аналіз аналогів інформаційних систем є важливим етапом при розробці нових технологічних рішень, оскільки дозволяє оцінити існуючі рішення на ринку, виявити їх переваги та недоліки, а також визначити можливості для вдосконалення. Це дає змогу створити систему, яка відповідає потребам користувачів та забезпечує ефективність бізнес-процесів організації.

Сайт підприємства з пасажирських перевезень на замовлення має бути зручним інструментом для залучення клієнтів, надання інформації про послуги та спрощення процесу замовлення. Ось ключові характеристики та функціональні можливості, які має мати сайт:

1. Каталог послуг: Детальний перелік доступних послуг (оренда автобусів, мікроавтобусів, міжміські та міжнародні перевезення, трансфери тощо) з описами та фотографіями автопарку.
2. Опис послуг: Інформація про типи транспорту, умови оренди, маршрути, ціни та додаткові опції (Wi-Fi, кондиціонер, розетки тощо).
3. Онлайн-консультант: Чат або форма зворотного зв'язку для оперативної консультації клієнтів.
4. Форма замовлення: Можливість залишити заявку на перевезення з вказівкою дати, маршруту, кількості пасажирів та контактних даних.
5. Розділ акцій та пропозицій: Інформація про знижки, сезонні пропозиції чи спеціальні умови для корпоративних клієнтів.

6. Контактна інформація: Адреса, телефони, email, а також карта для легкого зв'язку чи пошуку офісу.

Сайт підприємства «КиївТранс»

The screenshot displays the website for Kievtrans. At the top, there is a navigation bar with the company logo 'KEVTRANS' and menu items: 'Головна', 'Розрахунок вартості поїздки', 'Акції', 'Контакти', and 'Українська'. On the left side, contact numbers are listed: '+38-067-500-45-42' and '+38-063-235-90-84'. Below these are input fields for 'Ваше ім'я' and 'Ваш телефон', followed by a 'Передзвонити мені' button and a 'Відгуки про нас. Залишити відгук' button. The main content area features a text block stating that Kievtrans offers private passenger transport services with its own buses and minibuses, highlighting 7 years of successful operation and a growing fleet. Below this, a note says 'Ми не накручуватимемо ціну - якщо для вашого завдання є більш простий і дешевий варіант ми його запропонуємо:'. Three service categories are presented with images and details:

- Автобуси**: до 76 місць, від 11 грн/км та 300 грн/год. Button: 'Вибрати'.
- Мікроавтобуси**: до 21 місця, від 9 грн/км та 250 грн/год. Button: 'Вибрати'.
- Мінівени**: від 4-х до 8 місць, від 9 грн/км та 250 грн/год. Button: 'Вибрати'.

Рисунок 1.4 – Приклад інтерфейсу Kievtrans

Сайт компанії "КиївТранс" пропонує послуги пасажирських перевезень у межах Києва, по Україні та країнах Європи, позиціонуючи себе як зручний портал для замовлення автобусів і мікроавтобусів для туристичних поїздок, корпоративних заходів чи приватних трансферів [3]. Головна сторінка вітає відвідувачів інформацією про компанію, акцентує увагу на власному автопарку як гарантії якості та має кнопку для швидкого замовлення. Розділ послуг детально описує міжміські, міжнародні та екскурсійні перевезення, доповнені фотографіями автобусів із зазначенням їхньої місткості та зручностей. Функціонал включає можливість залишення онлайн-заявки, де клієнти вказують маршрут, дату, час і кількість пасажирів, а також вибір типу транспорту – від компактних мікроавтобусів до великих автобусів. Є автоматичний розрахунок вартості на основі введених даних, що економить час, а зворотний зв'язок із диспетчером забезпечує оперативне підтвердження. Контакти з телефонами, email і адресою доступні для зв'язку. Додатково є приклади маршрутів, таких як трансфери до

аеропортів, і умови співпраці для бізнес-клієнтів. Сайт зручний і інформативний, але відсутність онлайн-чату обмежує миттєву консультацію.

Сайт підприємства «ПасажирТранс»

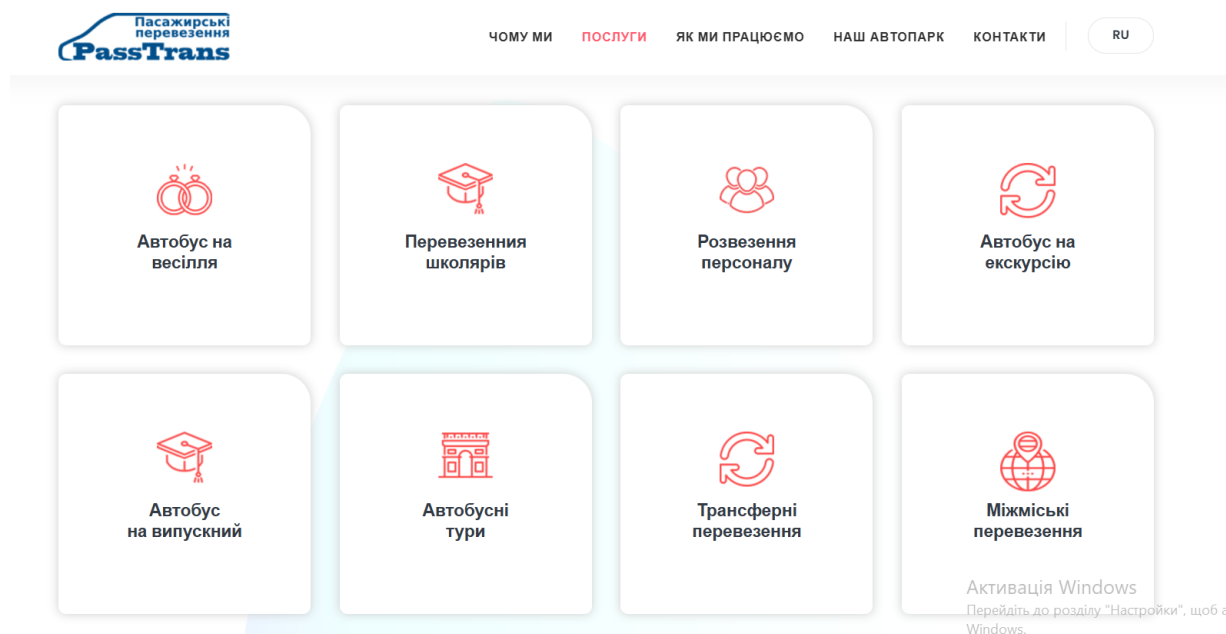


Рисунок 1.5 – Приклад інтерфейсу Passtrans

Сайт компанії "ПасажирТранс" зосереджений на пасажирських перевезеннях по Києву та Україні, наголошуючи на безпеці, доступності та комфорті як основних перевагах. [4] Головна сторінка презентує принципи роботи компанії, демонструє фото автопарку та пропонує форму заявки для швидкого запиту. Розділ послуг описує оренду автобусів, таких як "Богдан", MAN чи YUTONG, із характеристиками, наприклад, кількістю місць чи наявністю кондиціонера, а також пропозиції для корпоративних перевезень. Функціонал включає залишення онлайн-заявки з маршрутом і кількістю пасажирів, але вибір конкретного типу транспорту відсутній – клієнти отримують пропозицію від менеджерів. Розрахунок вартості доступний через форму, а зворотний зв'язок із диспетчером забезпечує підтвердження. Контакти з телефонами та адресою розміщені для зручності. Особливістю є пряма робота без посередників, що знижує ціни. Дизайн сайту

простий і дещо застарілий, а брак інтерактивних функцій обмежує його привабливість.

Сайт підприємства «Mega-Trans»

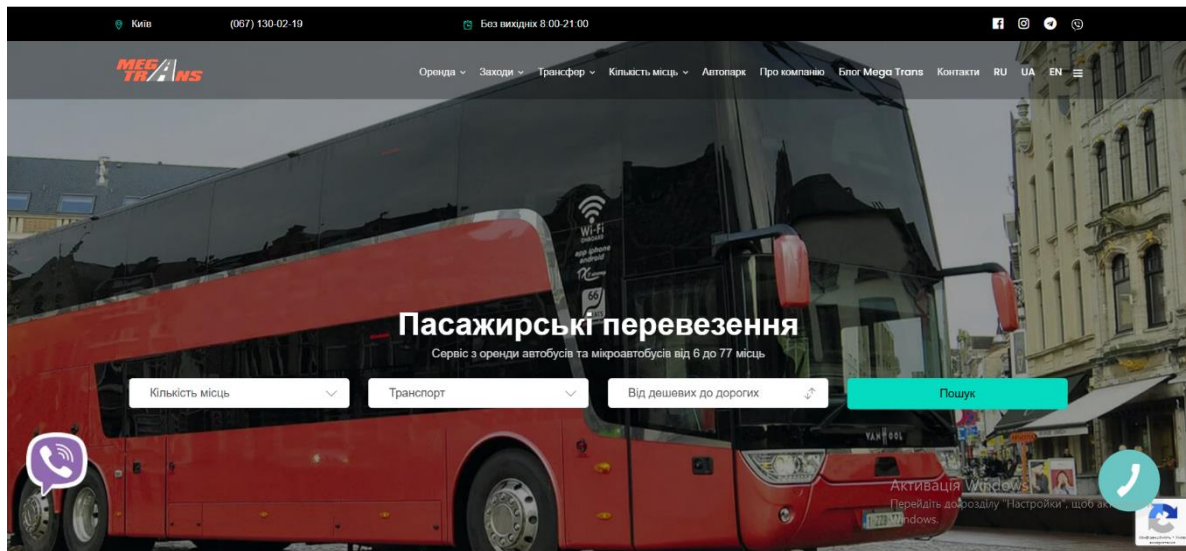


Рисунок 1.6 – Приклад інтерфейсу Mega-trans

Сайт компанії "Mega-Trans" охоплює широкий спектр транспортних послуг, включаючи пасажирські перевезення, і вирізняється сучасним дизайном [2]. Головна сторінка містить інформацію про компанію, слайдер із фото автобусів і контактні дані для зв'язку. Розділ послуг описує перевезення, але деталі про автопарк обмежені – є лише загальні характеристики без фото чи специфікацій моделей. Функціонал включає залишення онлайн-заявки з маршрутом, годинами та кількістю пасажирів, а також вибір типу транспорту – автобуси, мікроавтобуси чи VIP-авто. Розрахунок вартості доступний через форму, а зворотний зв'язок із диспетчером забезпечує оперативність. Є розділ відгуків клієнтів, що додає довіри, але без інтерактивних елементів. Контакти з телефонами легко знайти. Сайт привабливий, але менш спеціалізований на пасажирських перевезеннях через недостатню деталізацію.

Порівняння інформаційних систем аналогів

Таблиця 1.3. Порівняльна таблиця функцій сайтів аналогів

Критерії оцінки / функції	КиївТранс	ПасажирТранс	Mega-Trans
Залишення онлайн-заявки	Наявна	Наявна	Наявна
Вибір типу транспорту	Наявна	Відсутня	Наявна
Розрахунок вартості	Наявна	Наявна	Наявна
Зворотній зв'язок з диспетчером	Наявна	Наявна	Наявна
Оцінка послуг	Відсутня	Відсутня	Відсутня
Дизайн	Сучасний	Застарілий	Сучасний
Інформаційне наповнення	Повне	Середнє	Середнє
Навігація	Зручна	Проста	Зручна
Реєстрація/Вхід	Відсутня	Відсутня	Відсутня

Ця таблиця надає детальніше порівняння веб-сайтів компаній "КиївТранс", "ПасажирТранс" та "Mega-Trans" за параметрами функціональності, дизайну й інформаційного наповнення, висвітлюючи їхні переваги та недоліки. Аналіз дозволяє зрозуміти, як ці сайти вирішують задачі інформування клієнтів і спрощення замовлення послуг, а також виявити аспекти, які можна вдосконалити при розробці нового сайту пасажирських перевезень на замовлення.

Після аналізу сайтів "КиївТранс", "ПасажирТранс" та "Mega-Trans" можна зробити такі висновки. "КиївТранс" вирізняється сучасним дизайном, повною інформацією та зручною навігацією, але бракує онлайн-чату й оцінки послуг. "ПасажирТранс" має просту структуру й доступність, однак застарілий дизайн і відсутність вибору транспорту знижують привабливість. "Mega-Trans" приваблює сучасним оформленням і вибором транспорту, але недостатня деталізація послуг робить його менш спеціалізованим.

Отже, кожен сайт має свої переваги й недоліки. "КиївТранс" кращий для тих, хто цінує зручність і повноту даних, "Mega-Trans" – для любителів сучасного дизайну, а "ПасажирТранс" підходить для простих запитів, але потребує модернізації. Цей аналіз слугуватиме основою для розробки нового сайту, який поєднає найкращі риси аналогів і усуне їхні недоліки.

1.6. Обґрунтування доцільності проектування й розроблення веб-сайту для надання послуг з пасажирських перевезень на замовлення

У сучасних умовах розвитку цифрових технологій наявність веб-сайту є ключовим фактором успішної діяльності будь-якої компанії, що надає послуги з пасажирських перевезень. Відсутність онлайн-платформи значно ускладнює процес взаємодії з клієнтами, зменшує охоплення аудиторії та знижує конкурентоспроможність підприємства. Тому розроблення веб-сайту є необхідним кроком для оптимізації бізнес-процесів і підвищення якості обслуговування.

Основною проблемою відсутності веб-сайту є обмеженість доступу клієнтів до інформації про послуги підприємства та неможливість оформлення замовлень онлайн. У сучасному світі користувачі звикли до швидкого та зручного отримання інформації, тому наявність сайту сприятиме підвищенню рівня довіри клієнтів і збільшенню кількості замовлень. Завдяки веб-сайту потенційні клієнти зможуть оперативно переглядати доступні маршрути, умови перевезень, розраховувати вартість послуг та оформлювати замовлення без необхідності телефонного дзвінка або особистого відвідування офісу.

Додатковим важливим аспектом є автоматизація процесів управління замовленнями. Впровадження веб-сайту дозволить зменшити навантаження на персонал, спростити обробку заявок та підвищити точність виконання замовлень. Усі дані про клієнтів та їхні замовлення будуть систематизовані, що полегшить їх подальшу обробку, аналіз і вдосконалення сервісу.

Ще одним важливим фактором є розширення ринку та залучення нових клієнтів. На сьогодні основним джерелом отримання замовлень є фізична реклама, рекомендації постійних клієнтів та особисті контакти. Веб-сайт дозволить суттєво

розширити охоплення аудиторії через використання інструментів цифрового маркетингу. Завдяки таким підходам, як SEO-оптимізація для покращення видимості в пошукових системах, контекстна реклама для залучення цільової аудиторії, інтеграція з соціальними мережами для розширення охоплення, можна значно збільшити кількість потенційних клієнтів. Оптимізоване ранжування сайту в пошукових системах сприятиме його видимості та доступності для широкого кола користувачів, що дозволить залучати нових клієнтів без додаткових витрат на традиційну рекламу.

Також важливим аспектом є підвищення рівня довіри клієнтів. Завдяки можливості залишати відгуки, ставити запитання та отримувати оперативні відповіді через онлайн-платформу, клієнти будуть більш впевнені у виборі послуг компанії. Крім того, наявність веб-сайту сприятиме формуванню позитивного іміджу компанії, адже сучасні користувачі схильні довіряти брендам, які представлені в онлайн-середовищі.

Таким чином, розроблення веб-сайту для компанії, що надає послуги пасажирських перевезень, є доцільним та необхідним заходом. Це дозволить покращити комунікацію з клієнтами, автоматизувати процеси управління замовленнями, розширити ринок збуту, підвищити рівень довіри клієнтів і зміцнити конкурентні позиції підприємства в галузі пасажирських перевезень.

1.7. Розрахунок економічного ефекту від впровадження системи

Проект розробки веб-сайту для компанії, яка надає послуги пасажирських перевезень на замовлення, має на меті підвищення ефективності управління процесами перевезень, автоматизацію ключових операцій та покращення якості обслуговування клієнтів. Впровадження сайту дозволить компанії оптимізувати бізнес-процеси, знизити операційні витрати та забезпечити зручний спосіб взаємодії з клієнтами. Нижче наведено розрахунок економічного ефекту від впровадження сайту.

Основні етапи проекту

1. Аналіз вимог

- Трудомісткість (ТМ): 5 люд.-днів
 - Чисельність працюючих (ЧП): 1 особа
 - Тривалість (ТР): $ТР = ТМ / ЧП = 5 / 1 = 5$ днів
 - Витрати: 5 днів \times 2000 грн/день = 10 000 грн
2. Проектування
- Трудомісткість (ТМ): 10 люд.-днів
 - Чисельність працюючих (ЧП): 2 особи
 - Тривалість (ТР): $ТР = ТМ / ЧП = 10 / 2 = 5$ днів
 - Витрати: 10 днів \times 2000 грн/день = 20 000 грн
3. Розробка та налаштування системи
- Трудомісткість (ТМ): 30 люд.-днів
 - Чисельність працюючих (ЧП): 3 особи
 - Тривалість (ТР): $ТР = ТМ / ЧП = 30 / 3 = 10$ днів
 - Витрати: 30 днів \times 2000 грн/день = 60 000 грн
4. Тестування
- Трудомісткість (ТМ): 8 люд.-днів
 - Чисельність працюючих (ЧП): 2 особи
 - Тривалість (ТР): $ТР = ТМ / ЧП = 8 / 2 = 4$ днів
 - Витрати: 8 днів \times 2000 грн/день = 16 000 грн
5. Впровадження
- Трудомісткість (ТМ): 10 люд.-днів
 - Чисельність працюючих (ЧП): 2 особи
 - Тривалість (ТР): $ТР = ТМ / ЧП = 10 / 2 = 5$ днів
 - Витрати: 10 днів \times 2000 грн/день = 20 000 грн
6. Навчання персоналу
- Трудомісткість (ТМ): 5 люд.-днів
 - Чисельність працюючих (ЧП): 1 особа
 - Тривалість (ТР): $ТР = ТМ / ЧП = 5 / 1 = 5$ днів
 - Витрати: 5 днів \times 2000 грн/день = 10 000 грн
7. Розвиток проекту

- Трудомісткість (ТМ): 20 люд.-днів
- Чисельність працюючих (ЧП): 2 особи
- Тривалість (ТР): $ТР = ТМ / ЧП = 20 / 2 = 10$ днів
- Витрати: $20 \text{ днів} \times 2000 \text{ грн/день} = 40\,000 \text{ грн}$

Загальні витрати проекту:

Загальні витрати = $10\,000 \text{ грн} + 20\,000 \text{ грн} + 60\,000 \text{ грн} + 16\,000 \text{ грн} + 20\,000 \text{ грн} + 10\,000 \text{ грн} + 40\,000 \text{ грн} = 176\,000 \text{ грн}$

Очікуваний економічний ефект

1. Зниження операційних витрат

До впровадження сайту компанія витрачала 50 000 грн на рік на ручну обробку замовлень та координацію маршрутів.

Автоматизація дозволить скоротити ці витрати на 30%.

Зниження витрат = $50\,000 \text{ грн} \times 0,3 = 15\,000 \text{ грн}$ на рік

2. Збільшення кількості замовлень

До впровадження сайту компанія обслуговувала 500 замовлень на рік.

Після впровадження очікується збільшення на 20%.

Додаткові замовлення = $500 \times 0,2 = 100$ замовлень

Середній дохід з одного замовлення = 1 000 грн

Додатковий дохід = $100 \times 1\,000 \text{ грн} = 100\,000 \text{ грн}$ на рік

3. Економія часу на обробку замовлень

До автоматизації час обробки одного замовлення становив 15 хвилин.

Після автоматизації час зменшився до 10 хвилин.

Економія часу на одне замовлення = 5 хвилин

Загальна економія часу на рік = $600 \text{ замовлень} \times 5 \text{ хвилин} = 3\,000 \text{ хвилин}$
= 50 годин

Вартість години роботи диспетчера = 100 грн

Економія коштів = $50 \text{ годин} \times 100 \text{ грн} = 5\,000 \text{ грн}$ на рік

4. Збільшення кількості клієнтів через SEO-оптимізацію

Очікується, що завдяки SEO сайт відвідуватимуть 500 нових користувачів на місяць, з яких 5% стануть клієнтами.

Нові клієнти на місяць = $500 \times 0,05 = 25$ клієнтів

Нові клієнти на рік = $25 \times 12 = 300$ клієнтів

Середній дохід з одного клієнта = 1 000 грн

Додатковий дохід = $300 \times 1\,000$ грн = 300 000 грн на рік

Загальний річний економічний ефект:

Зниження операційних витрат: 15 000 грн

Збільшення доходу від замовлень: 100 000 грн

Економія часу на обробку замовлень: 5 000 грн

Додатковий дохід від SEO: 300 000 грн

Загальний річний економічний ефект = 15 000 грн + 100 000 грн + 5 000 грн + 300 000 грн = 420 000 грн

Окупність проекту

Термін окупності проекту = Загальні витрати / Річний економічний ефект = 176 000 грн / 420 000 грн $\approx 0,42$ року (≈ 5 місяців)

Висновок:

Впровадження веб-сайту для компанії, яка надає послуги пасажирських перевезень на замовлення, є економічно доцільним. Проект окупається менш ніж за півроку та забезпечує значний економічний ефект у розмірі 420 000 грн на рік. Це дозволяє компанії підвищити ефективність роботи, знизити витрати та забезпечити зручний спосіб взаємодії з клієнтами.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЄКТУВАННЯ

2.1. Загальні положення

2.1.1. Найменування системи: «Розроблення веб-орієнтованої системи управління замовленнями пасажирських перевезень»

2.1.2. Результати робіт зі створення системи оформлюються згідно з вимогами

ДСТУ на відповідні етапи розробки. Порядок оформлення і передачі результатів у даному випадку визначається змістом і календарним планом виконання розробки.

2.1.3. У випадку необхідності на наступних стадіях робіт по створенню системи окремі положення можуть уточнюватися і розвиватися [1].

2.2. Призначення і цілі створення системи.

2.2.1. Призначення систем.

Інформаційна система призначена для автоматизації процесів оформлення, обробки та адміністрування замовлень на пасажирські перевезення. Система надає змогу клієнтам обирати послуги з каталогу, переглядати транспортні засоби автопарку, оформлювати замовлення через веб-інтерфейс, отримувати попередній розрахунок вартості послуг, а також забезпечує зворотній зв'язок з компанією.

2.2.2. Цілі створення системи.

Для усунення наявних проблем і покращення ефективності бізнес-процесів підприємства з пасажирських перевезень необхідно впровадити низку заходів з автоматизації. Основною задачею є розробка та запуск веб-сайту, який стане основним інструментом для залучення нових клієнтів та покращення якості обслуговування. Веб-сайт повинен включати каталог послуг компанії, детальну інформацію про автопарк, можливість онлайн-бронювання перевезень, а також форму для залишення контактних даних, щоб клієнти могли легко зв'язатися з компанією. Це зробить послуги доступними для більшої аудиторії та полегшить процес замовлення, забезпечивши зручність для клієнтів.

Однією з ключових функцій майбутньої інформаційної системи буде можливість оформлення замовлення на перевезення. Клієнти зможуть вибирати

час, дату, маршрут, тип транспорту та кількість пасажирів, що дозволить їм отримати персоналізовану послугу.

2.3. Характеристика об'єкта автоматизації.

Підприємство ФОП «Салавор Р.Т.» спеціалізується на наданні послуг пасажирських транспортних перевезень на замовлення.

Сьогодні підприємство має невеликий, але сучасний автопарк, який включає мікроавтобуси та автобуси різної місткості. Це дозволяє компанії задовольняти потреби клієнтів у різних типах перевезень, від невеликих груп до великих колективів.

Основні види послуг:

- Організація перевезень для корпоративних клієнтів
- Розвезення персоналу
- Щоденні маршрути
- Трансфери на святкові заходи
- Екскурсійні поїздки
- Комфортні поїздки між містами
- Туристичні трансфери

2.4. Вимоги до системи.

2.4.1. Вимоги до системи в цілому

Інформаційна система повинна функціонувати як веб-застосунок із клієнт-серверною архітектурою. Клієнтська частина представлена у вигляді адаптивного веб-інтерфейсу, доступного з ПК, планшетів та мобільних пристроїв. Серверна частина забезпечує обробку запитів, зберігання даних у централізованій базі даних та управління бізнес-логікою. Система має підтримувати багатокористувацький режим із розмежуванням прав доступу (адміністратор, клієнт).

2.4.2. Система повинна передбачати наявність таких модулів:

Каталог послуг (із зазначенням типів перевезень, цінових діапазонів та умов). Інформація про автопарк (опис транспортних засобів, фото, технічні характеристики). Онлайн-бронювання (форма замовлення з параметрами:

маршрут, дата, час, кількість пасажирів, тип транспорту). Контактна форма (зворотній зв'язок із представником компанії).

Усі модулі мають бути реалізовані з урахуванням принципів зручності, інтуїтивної зрозумілості та швидкодії.

Система повинна бути захищеною від несанкціонованого доступу, підтримувати авторизацію користувачів, зберігати резервні копії бази даних та реєструвати ключові дії користувачів у системному журналі. Крім того, необхідно забезпечити кросбраузерність, відповідність стандартам HTML5/CSS3, використання адаптивного дизайну та оптимізацію під мобільні пристрої [16]

2.4.3. Вимоги до персоналу

Персонал, який буде працювати з системою, повинен мати базові навички роботи з ПК, бути ознайомлений з функціоналом веб-сайту, вміти взаємодіяти з адміністративною панеллю для обробки заявок. Персонал повинен пройти попереднє навчання. Користувачі, які мають доступ до адміністративної частини, повинні володіти навичками роботи з базами даних, формами введення, знати основи інформаційної безпеки.

Рівень доступу повинен регламентуватись системою авторизації. Адміністратор має повний доступ до управління даними та структурою сайту. Оператор має доступ до замовлень, але не може змінювати базову інформацію про послуги. Клієнт має доступ лише до клієнтського інтерфейсу, без доступу до внутрішньої логіки системи.

2.4.4. Вимоги до надійності

Інформаційна система повинна бути доступною цілодобово з мінімальним рівнем відмов. Основними показниками надійності є:

Ймовірність безвідмовного виконання задачі. Коефіцієнт готовності системи. Час відновлення після відмови. Період між двома відмовами.

Система повинна підтримувати створення резервних копій автоматично щонайменше один раз на добу, зберігати лог-файли активності користувачів і надавати адміністратору повідомлення про критичні помилки чи спроби несанкціонованого доступу.

2.4.5. Вимоги до безпеки

У майбутній реалізації система повинна забезпечувати надійний рівень захисту даних та обмеження доступу до функціональності, яка вимагає авторизації. Уся взаємодія між клієнтом і сервером має здійснюватися через захищене з'єднання з використанням HTTPS.

Для доступу до персоналізованих функцій користувача (оформлення бронювання, перегляд історії замовлень, надсилання повідомлень) необхідно реалізувати механізм реєстрації та авторизації. Паролі користувачів повинні зберігатися в зашифрованому вигляді з використанням сучасного алгоритму хешування (наприклад, bcrypt).

Потрібно забезпечити перевірку введених даних як на клієнтській, так і на серверній стороні для захисту від XSS-атак. Також варто реалізувати механізми захисту від CSRF-атак (наприклад, з використанням токенів) і забезпечити контроль типу запитів до критичних маршрутів.

У структурі бази даних необхідно виключити можливість SQL- або NoSQL-ін'єкцій шляхом використання ORM-інструментів (наприклад, Mongoose) із попередньо визначеними схемами.

2.4.6. Вимоги до захисту інформації від аварій

Необхідно передбачити:

Автоматичне створення резервної копії бази даних на окремому сервері або хмарному сховищі. Можливість оперативного відновлення системи з резервної копії.

2.4.7. Функціональні вимоги

У таблиці нижче наведено основні функції системи із зазначенням вхідної та вихідної інформації.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

№	Найменування функції	Вхідна інформація	Вихідна інформація
1	Формування каталогу послуг	Дані про типи послуг, тарифи, умови	Список послуг
2	Формування каталогу транспортних засобів	Дані про автопарк: марки, фото, характеристики	Перелік доступного транспорту
3	Оформлення онлайн-замовлення	ПІБ, маршрут, дата, час, кількість пасажирів, тип транспорту, персональні дані	Заявка в БД, підтвердження замовлення
4	Історія бронювань	Запит від користувача	Список зроблених бронювань
5	Надсилання повідомлення	Дані з контактної форми (ПІБ, email, повідомлення)	Повідомлення успішно надіслано
6	Авторизація користувачів	Логін, пароль	Сесія користувача, рівень доступу
7	Адміністрування контенту	Дані про нові послуги, зміни в інформації, фото	Оновлений контент на сайті

2.4.8. Вимоги до видів забезпечення

Математичне забезпечення

Система не потребує складного математичного апарату. Розрахунки здійснюються за допомогою стандартних арифметичних операцій, заснованих на відстані маршруту, класі транспорту та кількості пасажирів. Достатньо можливостей вбудованої математичної бібліотеки обраної мови програмування.

2.4.8.1. Програмне забезпечення

Серверна частина системи повинна бути реалізована з використанням середовища Node.js та фреймворку Express.js для створення REST API. Клієнтська частина має бути розроблена на базі React із застосуванням бібліотек axios для обміну даними з сервером і react-router-dom для маршрутизації. Сторінки повинні бути адаптивними, з використанням CSS-фреймворків, таких як Tailwind CSS або аналогічного. База даних повинна бути побудована на основі MongoDB, із підключенням через Mongoose для створення схем даних.

Інтерфейс користувача має бути розроблений відповідно до стандартів UX/UI з урахуванням ергономіки та доступності. Програмне забезпечення повинне мати модульну структуру для полегшення оновлення та розширення функціоналу.

2.4.8.2. Інформаційне забезпечення

У системі повинні бути реалізовані наступні логічні колекції: "Clients" (користувачі), "Reservations" (бронювання), "Vehicles" (транспортні засоби), "Feedbacks" (повідомлення). Структура даних має відповідати принципам документоорієнтованої моделі з урахуванням оптимізації запитів і швидкого доступу до даних.

Моделі повинні створюватися за допомогою Mongoose-схем. Система повинна підтримувати можливість експорту даних у формат JSON або CSV, а також імпорту структурованих даних при необхідності.

2.4.8.3. Технічне забезпечення

Система має розгортатися на хостингу з підтримкою Node.js-застосунків (наприклад, Render, Railway, Vercel або VPS). Мінімальні серверні характеристики: CPU 2 ядра, RAM 2 GB, SSD 10 GB.

На стороні користувача — підтримка сучасних браузерів (Google Chrome, Firefox, Microsoft Edge), стабільне підключення до Інтернету (не менше 10 Мбіт/с), ПК або ноутбук з операційною системою Windows/Linux/macOS. Інтерфейс має бути зручним для використання як на десктопах, так і на мобільних пристроях.

2.4.8.4. Лінгвістичне забезпечення

Інтерфейс системи українською мовою. Усі повідомлення, меню, інструкції повинні бути написані грамотною, зрозумілою мовою. Враховується можливість майбутньої локалізації англійською чи іншими мовами. Формати формування запитів користувачів — через кнопки, форми та випадаючі списки.

2.4.9. Підготовка до введення системи в дію

До моменту запуску система повинна пройти стадії тестування: модульного, інтеграційного, системного та дослідної експлуатації. Персонал компанії повинен бути навчений роботі з адміністративною панеллю. Замовник забезпечує сервер,

доменне ім'я, сертифікат SSL. Розробник зобов'язується надати документацію, інструкції користувача та опис бази даних.

2.5. Склад і зміст робіт по створенню системи

Таблиця 2.2 Межі та час виконання проекту

Етапи проекту	Початок	Тривалість	Затримка	Кінець
Аналіз вимог	01.04.2025	3	0	03.04.2025
Дослідження об'єкту автоматизації	04.04.2025	2	0	05.04.2025
Розробка технічного завдання	06.04.2025	5	0	10.04.2025
Проектування структури бази даних	11.04.2025	3	0	13.04.2025
Розробка front end частини	14.04.2025	10	0	23.04.2025
Розробка back end частини	14.04.2025	7	0	20.04.2025
Тестування системи	24.04.2025	5	0	28.04.2025
Впровадження системи	29.04.2025	3	0	01.05.2025
Реклама та SEO-просування	02.05.2025	7	0	08.05.2025

2.6. Порядок контролю і приймання системи

Контроль виконання проводиться на основі порівняння реалізованого функціоналу з положеннями технічного завдання. Приймання системи здійснюється на основі тестових сценаріїв. Сторони підписують акт здачі-прийому. У разі виявлення недоліків формується протокол з переліком зауважень і термінами їх усунення.

Введення в дослідну експлуатацію супроводжується веденням журналу помилок і побажань користувачів. Після усунення виявлених недоліків система приймається в промислову експлуатацію.

2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- проводить укомплектування технічних засобів;

- організовує навчання користувачів системи роботі на ПК і вивчення інструкції з її експлуатації;
- проводить дослідну експлуатацію і вводить систему в дію.

2.8. Вимоги до документації

2.8.1. На систему розробляється комплекс документації у складі: технічне завдання та технічний проект.

2.8.2. Документація на систему розробляється у відповідності з вимогами Державних стандартів серії 19 «Єдина система програмної документації» та серії 24 «Єдина система стандартів автоматизованих систем управління».

2.9. Джерела розробки

2.9.1. При розробленні технічного завдання на систему використано наступні документи:

ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;

ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво;

ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом [1]

РОЗДІЛ 3. ПРОЕКТУВАННЯ, СТВОРЕННЯ ТА АПРОБАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1. Опис та обґрунтування вибору програмно-технічних засобів розроблення інформаційної системи

Для розробки інформаційної системи було використано такі програмно-технічні засоби:

- мови програмування: JavaScript;
- фреймворки та бібліотеки: React, Node.js, Express, Mongoose;
- система керування базами даних: MongoDB;
- середовища розробки: visual studio code;
- система розгортання додатків: vercel, render або railway;
- інструменти безпеки та аутентифікації: json web token;
- система контролю версій: git, github;
- інструмент збірки фронтенду: vite;
- фреймворк для стилізації інтерфейсу: Tailwind CSS;

JavaScript — це основна мова програмування для створення інтерактивних веб-застосунків. Вона підтримується всіма сучасними браузерами та використовується як на клієнтській, так і на серверній стороні [5]

React — це безкоштовна бібліотека JavaScript, що дозволяє створювати динамічні інтерфейси на основі компонентів. React забезпечує швидке оновлення сторінок без повного перезавантаження, завдяки використанню віртуального DOM [6].

Node.js — це середовище виконання JavaScript, яке дозволяє запускати серверний код поза межами браузера. Node.js забезпечує неблокуючу, подієво-орієнтовану модель, що робить його ефективним для створення масштабованих веб-сервісів [7].

Express — це мінімалістичний вебфреймворк для Node.js, який значно спрощує побудову серверної частини веб-застосунку. Він використовується для створення REST API в проєкті [8].

MongoDB — документо-орієнтована СКБД, що дозволяє зберігати дані у форматі JSON. Це надає гнучкість у зберіганні структурованих і напівструктурованих даних. У зв'язці з бібліотекою Mongoose забезпечується створення схем, валідація даних та зручне об'єктно-документне моделювання [9].

Visual Studio Code — це сучасне середовище розробки, яке підтримує велику кількість розширень, зокрема для роботи з JavaScript, React, Node.js та MongoDB. Забезпечує зручний інтерфейс, підсвітку синтаксису, автодоповнення коду та контроль версій через Git [10]

Vercel — хмарна платформа для деплою frontend-додатків, яка автоматично виконує CI/CD. Підтримує інтеграцію з GitHub і дозволяє швидко та безкоштовно розгортати React-застосунки [12]

Render / Railway — сервіси для хостингу серверної частини додатка. Підтримують автозапуск Node.js додатків, налаштування змінних середовища та розгортання з репозиторіїв.

JWT (JSON Web Token) — технологія для реалізації автентифікації користувачів у системі. Забезпечує шифрування токенів доступу, що підвищує безпеку обміну даними між клієнтом і сервером.

Git і GitHub — система контролю версій і хостинг для збереження коду проєкту. Git дозволяє ефективно відслідковувати зміни, керувати гілками та проводити колективну розробку, а GitHub — зручно публікувати та обслуговувати віддалені репозиторії.

Vite — це сучасний інструмент для збірки JavaScript/TypeScript-проєктів, який значно прискорює запуск і компіляцію застосунку. Завдяки гарячому перезавантаженню (Hot Module Replacement) Vite дозволяє розробникам миттєво бачити зміни в інтерфейсі без повного перезавантаження сторінки [11]

Tailwind CSS — утилітарний CSS-фреймворк, що дозволяє створювати стильні та адаптивні інтерфейси за допомогою готових класів без написання власного CSS-коду. Це значно пришвидшує процес стилізації, зберігає чистоту HTML-структури та забезпечує єдину візуальну мову всього застосунку [13]

3.2. Проектування та створення бази даних

У процесі розроблення веб-орієнтованої системи управління замовленнями пасажирських перевезень, ми дійшли висновку, що ключову роль у забезпеченні ефективної роботи системи відіграє база даних. На підставі аналізу функціональних вимог було прийнято рішення використовувати документно-орієнтовану базу даних MongoDB. Вона дозволяє зберігати неструктуровані дані у форматі JSON-документів і підтримує масштабованість, необхідну для зростання навантаження в майбутньому.

Для спрощення роботи з базою даних у середовищі Node.js було використано бібліотеку Mongoose. Цей інструмент виконує роль Object Data Modeling (ODM) і надає можливість описувати структуру документів, зв'язки між ними, а також здійснювати валідацію даних [14].

Підключення до бази даних реалізовано у модулі connect.js. Рядок підключення зберігається у змінній середовища MONGO_URI у файлі .env. На рисунку 3.1 наведено фрагмент коду, що виконує підключення до MongoDB:

```
api > db > JS connect.js > ...
1   const mongoose = require('mongoose');
2
3   const DB = process.env.MONGO_URI;
4   mongoose.connect(DB).then(
5     ()=>{
6       console.log("DB connected...")
7     })
```

Рисунок 3.1 – підключення до MongoDB у модулі connect.js

Цей модуль імпортується у головному серверному файлі index.js через команду:

```
require("../db/connect");
```

У межах нашої системи було створено три основні моделі: Client (користувачі), Feedback (повідомлення) та Reservation (бронювання).

Модель Client описує користувача та реалізує плагін passport-local-mongoose для автоматичної автентифікації. Структура моделі представлена на рисунку 3.2.

```

api > models > JS clientDB.js > ...
1  const mongoose = require('mongoose');
2  const passportLocalmongoose = require('passport-local-mongoose');
3
4  const clientSchema = new mongoose.Schema({
5    username: String,
6    password: String,
7    phone: String,
8  });
9
10 clientSchema.plugin(passportLocalmongoose);
11
12 const Client = mongoose.model('Client', clientSchema);
13
14 module.exports = Client;

```

Рисунок 3.2 – модель користувача Client

У моделі передбачено такі атрибути: username (логін користувача), password (хешований пароль), phone (номер телефону) наведено в таблиці 3.1. Завдяки використанню плагіна до моделі автоматично додаються методи реєстрації та авторизації.

Таблиця 3.1 ClientDB

Назва	Тип	Значення
username	String	Логін користувача
password	String	Пароль
phone	String	Номер телефону

Модель Reservation відповідає за збереження даних про бронювання, зокрема контактної інформації, маршрутів і часових параметрів. Ця модель містить посилання на користувача, що здійснив замовлення. Її структура наведена на рисунку 3.3.

```
api > models > JS reservationDB.js > ...
1   const mongoose = require('mongoose');
2
3
4   const reservationSchema = new mongoose.Schema({
5     owner: {type:mongoose.Schema.Types.ObjectId, ref: 'Client'},
6     firstname: String,
7     lastname:String,
8     age:String,
9     phone:String,
10    email:String,
11    address:String,
12    city:String,
13    zipcode:String,
14    carType: String,
15    pickPlace: String,
16    dropPlace: String,
17    pickDate: String,
18    dropDate: String,
19    pickTime: String,
20    dropTime : String
21  })
22
23
24
25  const Reservation = mongoose.model('Reservation',reservationSchema);
26
27  module.exports = Reservation;
```

Рисунок 3.3 – схема моделі Reservation

У полі owner зберігається ObjectId користувача, що виконав бронювання. Інші поля містять персональні дані, контактну інформацію, тип транспортного засобу, місця відправлення й прибуття, а також дату і час поїздки. У таблиці 3.1 наведено назви полів, типи даних та їхнє значення.

Таблиця 3.2 ReservationDB

Назва	Тип	Значення
firstname	String	Ім'я клієнта
lastname	String	Прізвище клієнта
phone	String	Номер телефону
email	String	Адреса електронної пошти
address	String	Адреса проживання
city	String	Місто проживання
zipcode	String	Поштовий індекс
carType	String	Тип обраного транспорту
pickPlace	String	Місце відправлення
dropPlace	String	Місце прибуття
pickDate	String	Дата початку поїздки
dropDate	String	Дата завершення поїздки
pickTime	String	Час подачі транспорту
dropTime	String	Час прибуття

На підставі аналізу бізнес-процесів було реалізовано зв'язок між моделями Client та Reservation за принципом "один до багатьох" — один користувач може здійснювати декілька бронювань.

Модель Feedback відповідає за збереження зворотного зв'язку та повідомлень від користувачів, зокрема — імені, електронної адреси та змісту повідомлення. Її структура наведена на рисунку 3.4.

```

api > models > JS feedbackDB.js > ...
1  const mongoose = require('mongoose');
2
3  const feedbackSchema = new mongoose.Schema({
4    fullname: String,
5    email: String,
6    message: String,
7    createdAt: {
8      type: Date,
9      default: Date.now
10   }
11 });
12
13 const Feedback = mongoose.model('Feedback', feedbackSchema);
14
15 module.exports = Feedback;
16

```

Рисунок 3.4. – схема моделі Feedback

Модель містить типову інформацію для обробки запитів клієнтів, зокрема вільний текст, у якому користувач може залишити коментар, зауваження чи пропозицію. У полі `createdAt` зберігається дата і час створення повідомлення. У таблиці 3.3 наведено назви полів, типи даних та їхнє значення.

Таблиця 3.3 *FeedbackDB*

Назва	Тип	Значення
<code>fullname</code>	String	Повне ім'я користувача
<code>email</code>	String	Електронна адреса користувача
<code>message</code>	String	Текст повідомлення запит, відгук, побажання
<code>createdAt</code>	Date	Дата та час створення повідомлення

Таким чином, було створено логічно структуровану та гнучку модель бази даних, яка забезпечує надійне зберігання основних об'єктів інформаційної системи — користувачів, їхніх бронювань та повідомлень зворотного зв'язку. Архітектура побудована з урахуванням подальшого масштабування, що дозволяє розширювати функціональність без необхідності суттєвого перепроектування.

Реалізована структура забезпечує стабільні зв'язки між користувачами та пов'язаними з ними діями в системі, такими як створення замовлень або залишення відгуків.

3.3. Реалізація функцій системи

Для реалізації фронтенд-частини системи було обрано бібліотеку `React.js`. Це одна з найпопулярніших JavaScript-бібліотек для побудови динамічних користувацьких інтерфейсів. `React` дозволяє створювати компонентно-орієнтовану архітектуру, яка сприяє повторному використанню коду, спрощує відлагодження та масштабування застосунку.

Як серверну платформу (бекенд) використано `Node.js` у поєднанні з фреймворком `Express.js`. `Node.js` забезпечує високопродуктивне оброблення

одночасних запитів завдяки неблокуючій моделі введення/виведення, що є особливо важливим для онлайн-сервісів із багатьма користувачами. Express.js використовується для зручної побудови REST API, яка забезпечує комунікацію між клієнтською частиною та базою даних.

Клієнтська частина застосунку взаємодіє з бекендом через HTTP-запити, а бекенд, у свою чергу, звертається до бази даних MongoDB для читання та запису інформації про користувачів, повідомлень та бронювання .

Таким чином, вибраний стек технологій (MERN: MongoDB, Express.js, React, Node.js) є сучасним і популярним для побудови SPA (Single Page Application) і забезпечує як зручність у розробці, так і масштабованість системи в майбутньому.

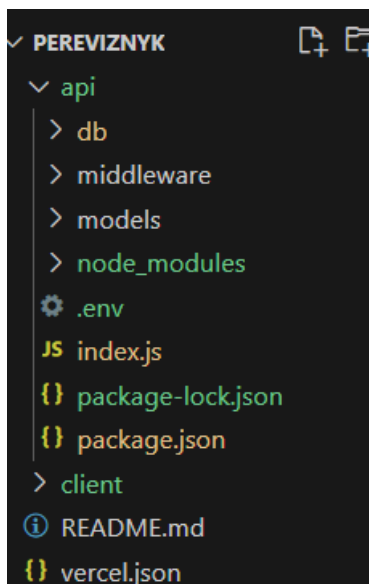


Рисунок 3.5. – структура папки програми

3.3.1. Реєстрація та авторизація користувача

Функція реєстрації та авторизації користувача є необхідною для надання персоналізованого доступу до сервісу з пасажирських перевезень. Зареєстровані клієнти отримують можливість створювати бронювання, переглядати власну історію бронювань і надсилати повідомлення адміністрації. Наявність облікового запису дозволяє підвищити зручність користування системою та забезпечити цілісність даних.

Компонент `RegistrationForm.jsx` реалізує двоетапний інтерфейс: на першому етапі користувач вводить електронну адресу; на другому — створює та підтверджує пароль. Форма містить перевірки на валідність email та надійність

пароля (мінімум 8 символів, великі й малі літери, цифри, спеціальні символи). На рисунку 3.6 наведено сторінку реєстрації.

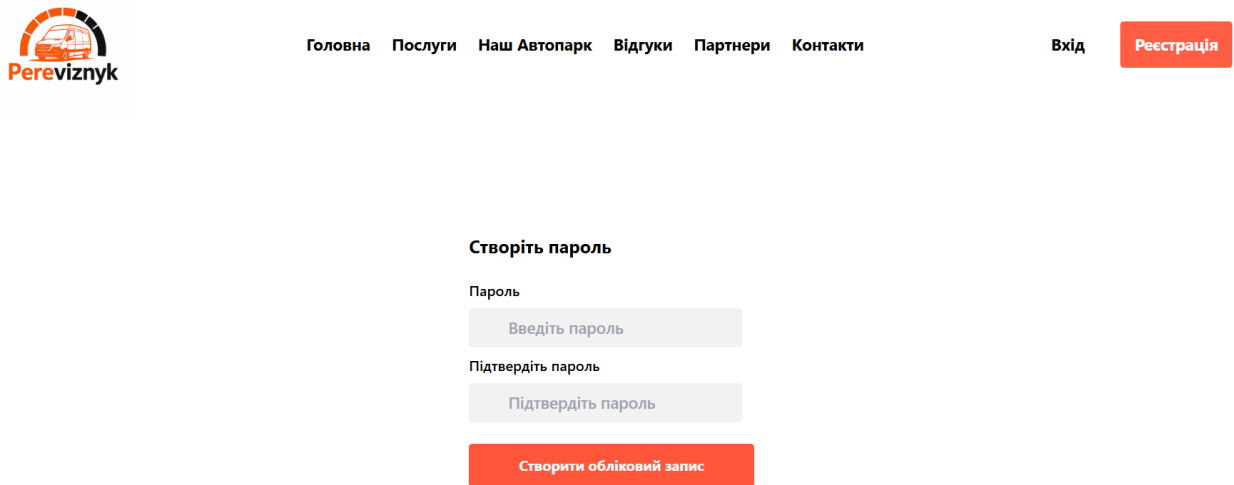


Рисунок 3.6 – сторінка реєстрації користувача

Для контролю стану компонент використовує React-хуки `useState` та `useEffect`, а також `useNavigate` для переадресації після успішної реєстрації. Валідність пароля перевіряється за допомогою регулярного виразу, а електронна адреса — стандартним `emailRegex`. Після підтвердження пароля користувач натискає кнопку «Створити обліковий запис», що викликає функцію `Submit`.

На стороні бекенду реалізовано обробку запитів через бібліотеку `passport-local-mongoose`, яка автоматично додає функції реєстрації, логіну, серіалізації користувача та збереження пароля у хешованому вигляді. На рисунку 3.7 зображено реалізацію запити на реєстрацію.

```

const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
const emailResult = emailRegex.test(username);
const passwordRegex = /^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[!@#%&*()-_+={}|;:'.<\/?]).{8,}$/;
const passwordResult = passwordRegex.test(password);

const HandleClick = () => {
  if (page === 0 && emailResult) {
    setPage(page + 1);
  } else if (!emailResult) {
    setIsvalid(true);
  }
}

const Submit = async () => {
  try {
    if (passwordResult && password === confirmPassword) {
      const res = await axios.post('http://localhost:5000/api/register', { username, password });
      if (res.data.username === username) {
        alert("Реєстрацію завершено. Увійдіть у систему");
        navigate('/login');
      }
    } else {
      setValidPassword(true);
    }
  } catch (error) {
    console.log(error);
  }
}

```

Рисунок 3.7 – Реалізація запиту на реєстрацію

Після надсилання запиту дані користувача зберігаються в базі MongoDB зображено на рис. 3.8, через серверну частину /api/register, яка використовує Mongoose-модель Client. Зв'язок між обліковим записом користувача та його замовленнями або повідомленнями реалізовано через зовнішній ключ `_id`, який зберігається у полі `owner` відповідних моделей `Reservation` та `Feedback`.

```

  _id: ObjectId('68136c0f622dd3931788f9af')
  username: "salavormax2901@gmail.com"
  password: "$2b$05$4S8gfwBgMELQRoyv5jEwP.lRXLxNc/5omNQthge1ZGSjTR7pYqiy."
  __v: 0

```

```

  _id: ObjectId('681521ad622dd3931788f9c7')
  username: "allgirlsoftheworldai@gmail.com"
  password: "$2b$05$4S8gfwBgMELQRoyv5jEwP.nbMzsBblyIsriML3FiP0xb82EZ16/32"
  __v: 0

```

Рисунок 3.8 – дані користувача в базі MongoDB

Після успішної реєстрації користувач може здійснити вхід через форму авторизації, реалізовану в компоненті Login.jsx. Інтерфейс містить поля для введення електронної пошти та пароля зображено на рисунку 3.9. Передбачено виведення повідомлення про помилку у разі неправильного введення облікових даних. На рисунку 3.10 зображено запит на авторизацію.

Рисунок 3.9 – сторінка входу користувача

```
const Submit = async (event) => {
  event.preventDefault();

  try {
    const { data } = await axios.post(
      'http://localhost:5000/api/login',
      { username: email, password },
      { withCredentials: true }
    );

    if (data?.username === email) {
      console.log(data);
      setUser(data);
      navigate('/account');
    } else {
      console.log('unauthorized user');
      setCheckUser(true);
      setTimeout(() => {
        setCheckUser(false);
      }, 2000);
    }
  } catch (e) {
    console.log(e);
  }
}
```

Рисунок 3.10 – Запит на авторизацію

Після входу користувач перенаправляється до особистого кабінету. У разі помилки система інформує про невірне ім'я або пароль. Для обміну даними використовується бібліотека `axios` з підтримкою сесій через `withCredentials: true` [19].

Реалізація автентифікації користувача через поєднання `passport-local-mongoose` і `REST API` забезпечує надійний та інтуїтивно зрозумілий процес реєстрації. Логіка обробки запиту дає змогу запобігти дублюванню облікових записів і підвищити безпеку автентифікації.

3.3.2. Бронювання поїздки

Функція оформлення бронювання забезпечує зареєстрованому користувачу можливість створити замовлення на перевезення з урахуванням маршруту, дати, часу, а також вибору типу транспортного засобу. У процесі реалізації нами було передбачено зручну форму, яка охоплює як персональні дані клієнта, так і параметри поїздки. З технічної точки зору, ця функція є центральною в системі, оскільки реалізує головну бізнес-логіку — зв'язок клієнта із сервісом перевезень.

Графічна форма бронювання складається з двох взаємопов'язаних компонентів:

`Reservationbox.jsx` — відповідає за вибір транспортного засобу зображено на рисунку 3.11. Користувачу надається перелік моделей мікроавтобусів і автобусів, таких як Mercedes-Benz Sprinter 318, Ford Transit, Peugeot Expert тощо. Після вибору автомобіля відображається зображення та таблиця з технічними характеристиками: кількість місць, рік випуску, наявність кондиціонера, ціна за годину та кілометр тощо.

The screenshot shows a web form for booking transport. The title is "Забронювати транспорт". There are four main input fields: "Оберіть транспорт" (selected "Mercedes-Benz Sprinter 318"), "Звідки" (selected "Київ"), "Куди" (selected "Львів"), and "Дата виїзду" (selected "25.05.2025"). There is also a "Дата повернення" field (selected "26.05.2025") and a red "Шукати" button.

Рисунок 3.11 – Вибір транспортного засобу для бронювання

Bookcar.jsx — відповідає за введення даних замовника та параметрів поїздки, зображено на рисунку 3.12. Сюди входять поля для імені, прізвища, контактної інформації, адреси, міста, індексу, а також дати та часу виїзду й повернення. Форма відкривається у вигляді модального вікна після натискання кнопки «Забронювати».

Персональні дані

Ім'я	Телефон
<input type="text" value="Ім'я"/>	<input type="text" value="Телефон"/>
Прізвище	
<input type="text" value="Прізвище"/>	
Email	
<input type="text" value="Email"/>	
Адреса	
<input type="text" value="Адреса"/>	
Місто	Поштовий індекс
<input type="text" value="Місто"/>	<input type="text" value="Індекс"/>

Надсилати новини та оновлення

Рисунок 3.12 – Введення персональних даних та підтвердження бронювання

Кнопка «Забронювати» розташована внизу форми. Стилізація виконана за допомогою Tailwind CSS, із адаптацією до різних розмірів екрана.

Введені дані зберігаються у станах через `useState`, а ідентифікатор авторизованого користувача отримується через `UserContext`. У разі незаповнених обов'язкових полів користувач отримує відповідне попередження. Якщо користувач не авторизований — система автоматично перенаправляє його на сторінку входу.

Натискання на кнопку «Забронювати» викликає функцію `ReserveCar()`, яка виконує перевірку введених даних і надсилає POST-запит на маршрут `/reservation`.

Також за допомогою useContext отримується ідентифікатор авторизованого користувача. Він автоматично додається до запиту у вигляді поля owner. Це забезпечує прив'язку кожного бронювання до конкретного користувача. Запит на збереження бронювання зображено на рисунку 3.13.

```

useEffect(() => {
  const car = Cars.find(car => car.name === carType);
  if (car) setSelectedimage(car.image);
}, [carType]);

const ReserveCar = async () => {
  if (isReserveDisabled) {
    setShowwarning(true);
    setTimeout(() => setShowwarning(false), 1500);
    return;
  }
  if (!user) {
    navigate('/login');
  } else {
    try {
      const { data } = await axios.post('/reservation', {
        carType, pickPlace, dropPlace, pickDate, dropDate, pickTime, dropTime,
        firstname, lastname, phone, email, address, city, zipcode
      });
      navigate('/account/bookings');
    } catch (e) {
      console.log(e);
    }
  }
};

```

Рисунок 3.13 – Запит на збереження бронювання

На серверній частині обробка маршруту POST /api/reservation передбачає збереження надісланих даних у базу MongoDB. Для цього використовується модель Reservation, яка зв'язана з користувачем через поле owner, що містить _id з моделі Client, фрагмент коду наведено нижче.

```

app.post('/api/reservation', authenticateJWT, async(req,res)=>{
  const { carType, pickPlace, dropPlace, pickDate, dropDate, pickTime,dropTime
,firstname,lastname ,age ,phone,email,address,city,zipcode} = req.body;
  const clientId = req.user.id;
  try{
    const reservation = new Reservation({

```

```
owner: clientId,  
firstname: firstname,  
lastname: lastname,  
age: age,  
phone: phone,  
email: email,  
address: address,  
city: city,  
zipcode: zipcode,  
carType: carType,  
pickPlace: pickPlace,  
dropPlace: dropPlace,  
pickDate: pickDate,  
dropDate: dropDate,  
pickTime: pickTime,  
dropTime : dropTime  
})  
await reservation.save();  
res.json(reservation)
```

На рисунку 3.14 показано приклад документа, який зберігається у колекції `reservations` після створення замовлення. Поле `owner` містить `ObjectId` користувача, інші поля — маршрут, дату, контактну інформацію тощо.

```

)  _id: ObjectId('681a1942f641679fff2b6eb2')
    owner: ObjectId('68136c0f622dd3931788f9af')
    firstname: "Василь"
    lastname: "Гутник"
    phone: "0988579425"
    email: "gurnykvasil@gmail.com"
    address: "Амосова"
    city: "Київ"
    zipcode: "03341"
    carType: "Peugeot Expert"
    pickPlace: "Амосова 124"
    dropPlace: "Трублаїні 8"
    pickDate: "2025-05-07"
    dropDate: "2025-05-07"
    pickTime: "08:00"
    dropTime: "20:15"

```

Рисунок 3.14 – дані бронювань в базі даних MongoDB

3.3.3. Перегляд історії бронювань

Функція перегляду історії бронювань надає авторизованому користувачу можливість ознайомитися зі своїми попередніми або активними замовленнями. Вона реалізована у вигляді сторінки, на якій виводиться перелік усіх бронювань, зроблених з облікового запису. Кожне бронювання містить детальну інформацію про поїздку, персональні дані, а також тип використаного транспортного засобу.

Функціонал реалізовано в окремому компоненті Bookinghistory.jsx. Після авторизації користувач автоматично отримує доступ до своєї історії бронювань, що відображається у вигляді переліку замовлень із короткими деталями (тип транспорту, дати, пункти поїздки). Кожне замовлення оформлено як картка з інформаційним блоком зображено на рисунку 3.15.

Історія бронювань



Рисунок 3.15 – Інтерфейс перегляду історії бронювань

Після відкриття сторінки запускається перевірка авторизації за допомогою функції `Privateroute()`. Якщо користувач має право доступу, система виконує запит на сервер для отримання усіх його бронювань.

Для цього використовується React-хук `useEffect`, у якому викликається метод `axios.get('/bookings')`, зображено на рисунку 3.16. Отримані дані зберігаються у стані `bookings`, після чого відображаються у вигляді карток на сторінці.

```
useEffect(() => {
  axios.get('/bookings').then((response) => {
    setBookings(response.data)
  });
}, []);
```

Рисунок 3.16 – завантаження бронювань через `useEffect`

Дані, що зберігаються у стані `bookings`, відображаються на сторінці у вигляді сітки з картками. Для кожного бронювання виводяться ім'я, email, телефон, маршрут і дата поїздки. Також у картці виводиться зображення транспорту, яке зіставляється з полем `carType` за допомогою масиву `Cars`. Фрагмент коду виведення бронювань з транспортом та деталями наведено нижче.

```
{bookings?.map((booking, index) => (
  <div key={index} className="shadow-xl py-4 px-2">
    <p>{booking.firstname} {booking.lastname}</p>
    <p>{booking.email}</p>
    <p>{booking.phone}</p>
    <p>{booking.pickDate} - {booking.dropDate}</p>
    <p>{booking.pickPlace} - {booking.dropPlace}</p>
    {Cars.map((car, index) => (
      booking.carType === car.name && (
        <img key={index} src={car.image} alt={car.name} className="max-h-40
object-contain" />
      )
    ))}
  <button onClick={() => handleCancelBooking(booking._id)}>Скасувати</button>
```

```
</div>
  )})
```

Функція скасування бронювання реалізована через кнопку «Скасувати», яка викликає метод `handleCancelBooking`. У середині цієї функції відправляється POST-запит на маршрут `/cancel` з ідентифікатором замовлення. Після успішного скасування сторінка оновлюється, щоб відобразити актуальний список замовлень.

Фрагмент коду скасування наведено нижче.

```
const handleCancelBooking = (bookingId) => {
  axios.post('/cancel', { bookingId }).then((response) => {
    console.log('Booking canceled:', response.data);
    window.location.reload();
  });
};
```

Зв'язок між бронюванням і користувачем реалізовано через поле `owner`, яке є зовнішнім ключем до моделі `Client`.

Користувач бачить перелік своїх замовлень у вигляді списку карток. У разі, якщо замовлень немає, відображається відповідне повідомлення про відсутність даних. Це забезпечує зрозумілу взаємодію з користувачем.

3.3.4. Надсилання зворотного зв'язку

Система передбачає можливість залишення повідомлень від користувачів у вільній формі. Ця функція реалізована для збирання пропозицій, скарг, подяк або технічних питань, що надходять від клієнтів. Зворотній зв'язок є важливою складовою сервісу, орієнтованого на користувача, оскільки дає змогу безпосередньо впливати на якість обслуговування.

Інтерфейс форми зворотного зв'язку реалізовано в компоненті `Contact.jsx`. Він доступний як для авторизованих користувачів (дані заповнюються автоматично), так і для неавторизованих, які можуть ввести їх вручну. Форма зворотного зв'язку наведена знизу на рисунку 3.17.



Потрібна додаткова інформація?

Якщо виникають питання, Ви завжди можете їх задати, скориставшись зручним для Вас засобом зв'язку.

(098) 757 78 47

pereviznyk2025@gmail.com

Київ, Україна

Повне ім'я

Наприклад: "Іван Іванович"

Електронна пошта

salavormax2901@gmail.com

Розкажіть нам більше

Напишіть тут...

Надіслати повідомлення

Рисунок 3.17 – Форма зворотного зв'язку на сайті

Форма містить такі поля: повне ім'я, електронна адреса, текст повідомлення (textarea) та кнопку «Надіслати повідомлення». Поля стилізовано за допомогою Tailwind CSS з адаптацією для різних розмірів екрана.

Після натискання кнопки викликається функція `Handlesubmit`, яка перевіряє заповнення полів. Якщо хоча б одне поле порожнє, виводиться відповідне повідомлення. У разі успішного введення даних відправляється POST-запит на бекенд, і з'являється повідомлення про успішне надсилання. Реалізація функції `Handlesubmit` у `Contact.jsx` зображено на рисунку 3.18.

```

const Handlesubmit = async (ev) => {
  ev.preventDefault();
  if (!fullname || !email || !Feedback) {
    setMessage(true);
    setTimeout(() => setMessage(false), 2000);
    return;
  }

  try {
    const response = await fetch('http://localhost:5000/api/feedback', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      credentials: 'include',
      body: JSON.stringify({
        fullname,
        email,
        message: Feedback
      })
    });

    if (response.ok) {
      setDoneFeedback(true);
      setFullName('');
      setEmail('');
      setFeedBack('');
      setTimeout(() => setDoneFeedback(false), 2000);
    } else {
      console.error('Помилка надсилання повідомлення');
    }
  } catch (e) {
    console.error('Помилка:', e);
  }
};

```

Рисунок 3.18 – Реалізація функції Handlesubmit у Contact.jsx

На серверній частині у index.js реалізовано маршрут POST /api/feedback, який створює новий документ у базі даних у колекції feedbacks. Структура моделі Feedback, яка зображена на рисунку 3.19, містить поля fullname, email, message, createdAt.

```

_id: ObjectId('6831cb35dd0554035635cc79')
fullname: "Дмитро Іванович"
email: "bogdanslotvin@gmail.com"
message: "Зателефонуйте мені!"
createdAt: 2025-05-24T13:35:49.573+00:00
__v: 0

_id: ObjectId('6831cbe3dd0554035635cc7b')
fullname: "Микола Васильович"
email: "mykoladid32@gmail.com"
message: "Добрий день!
        Потрібно два автобуси на весілля."
createdAt: 2025-05-24T13:38:43.884+00:00
__v: 0

```

Рисунок 3.19 – структура моделі Feedback

Після успішної відправки повідомлення користувач отримує текстове підтвердження. У разі помилки або незаповнених полів — повідомлення про необхідність заповнення.

Функція зворотного зв'язку є додатковим, але важливим компонентом системи, що підвищує якість обслуговування і довіру користувачів.

3.3.5. Інтерфейс системи

Навігаційне меню реалізовано через компонент Accountbar.jsx, який відповідає за вивід горизонтального меню на всіх сторінках. У меню розміщено кнопки для переходу на такі сторінки як: Головна, Послуги, Наш Автопарк, Відгуки, Партнери, Контакти, Бронювання. Нижче наведено код масиву пунктів меню в Accountbar.jsx.

```

const menus = [
  { name: 'Головна', href: '/account' },
  { name: 'Послуги', href: '/about' },
  { name: 'Наш Автопарк', href: '/model' },
  { name: 'Відгуки', href: '/testimonials' },
  { name: 'Партнери', href: '/team' },
  { name: 'Контакти', href: '/contact' },
  { name: 'Бронювання', href: '/account/bookings' },

```

];

Навігація адаптована під мобільні пристрої: для менших екранів активується бургер-меню зображене на рисунку 3.20, яке відкривається за допомогою кнопки і виводить ті самі пункти у вертикальному вигляді. Інтерактивність реалізована через стан `isToggle` та клас `open`, що дозволяє плавно відкривати і закривати мобільне меню. Знизу наведено код перемикача мобільного меню.

```
const open = () => {
  setIsToggle(!isToggle);
};
```

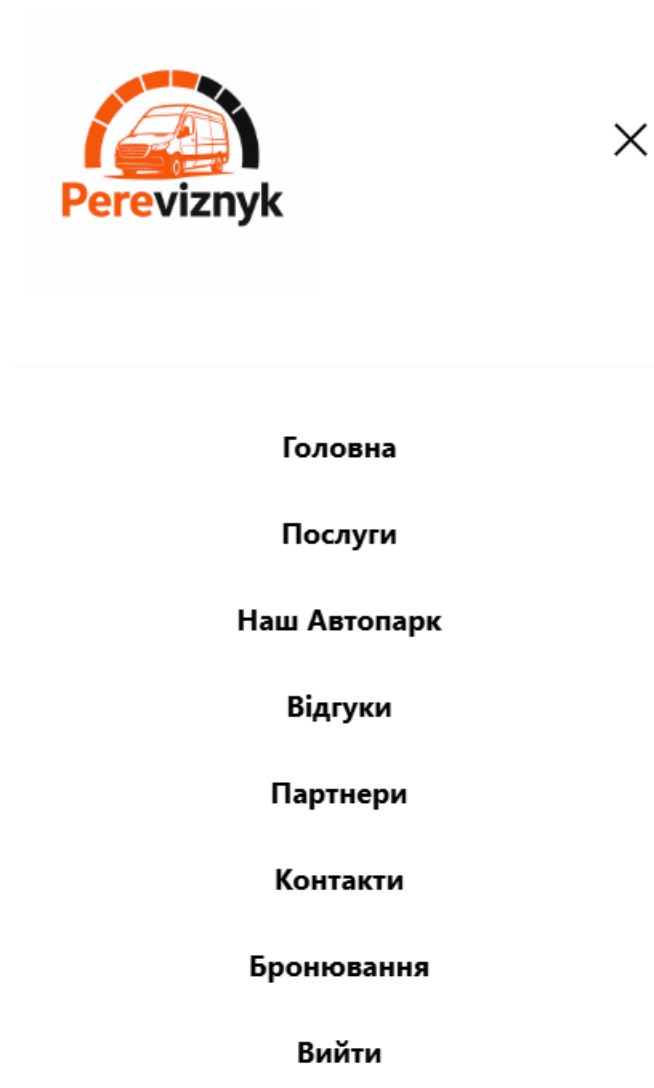


Рисунок 3.20 – Меню

Крім основних пунктів меню, у верхньому правому куті доступна випадаюча панель профілю `Dropdown`, яка містить кнопки для перегляду профілю та виходу з системи. Функція виходу реалізована як `POST`-запит на маршрут `/logout` з

подальшим очищенням користувача в контексті та перенаправленням на головну сторінку. Функцію виходу зображено на рисунку 3.21.

```
const handleOptionClick = async () => {
  await axios.post('/logout').then(()=>{
    setUser(null);
    navigate('/');
  })
}
```

Рисунок 3.21 – Функцію виходу з системи

На рівні основного застосунку всі маршрути визначаються через компонент App зображено на рисунку 3.22, який обгорнуто в BrowserRouter у файлі main.jsx. Це дозволяє здійснювати маршрутизацію між сторінками без перезавантаження, забезпечуючи плавну та зручну навігацію.

```
ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>,
)
```

Рисунок 3.22 – Компонент App у файлі main.jsx

Інтерфейс системи підтримує автентифікацію користувача, адаптацію до різних розмірів екрана, швидке оновлення даних та інтуїтивну структуру. Реалізована система повністю відповідає сучасним вимогам до UX/UI вебзастосунків і забезпечує зручність для користувача на всіх етапах взаємодії.

3.4. Інструкція користувача

Після переходу на головну сторінку користувач бачить логотип, навігаційне меню та рекламний банер із коротким описом послуг. У верхній частині присутні кнопки переходу до основних розділів: «Головна», «Послуги», «Наш Автопарк», «Відгуки», «Партнери», «Контакти». Праворуч знаходяться кнопки «Вхід» і «Реєстрація». Тут також розміщені кнопки «Забронювати» та «Дізнатися більше», що ведуть до відповідних функціональних блоків системи. Головна сторінка зображена на рисунку 3.23.



Комфортний рух Вашого бізнесу - наша мета!

Пасажирські перевезення по Києву та Україні

Замовляйте транспорт з водієм для ваших потреб. Доступні ціни, комфортний транспорт, гнучкі умови бронювання та багато іншого.

[Забронювати](#) ✓

[Дізнатися більше](#) >



Рисунок 3.23 – Головна сторінка

Користувач має можливість переглядати всі публічні сторінки сайту без авторизації. Однак для оформлення замовлення та використання повного функціоналу системи необхідно зареєструватися або увійти в обліковий запис. Далі розглянемо процес реєстрації.

У правому верхньому куті користувач має доступ до кнопок «Вхід» та «Реєстрація». Після натискання на кнопку «Реєстрація» відкривається сторінка створення облікового запису. Користувачу пропонується ввести електронну адресу та натиснути кнопку «Продовжити», зображено на рисунку 3.24.

Створіть обліковий запис

Електронна пошта

sezonivans21@gmail.com

[Продовжити](#)

Рисунок 3.24 – Реєстрація користувача

Після введення електронної пошти користувач переходить до другого етапу реєстрації — створення пароля. На сторінці пропонується ввести новий пароль та

підтвердити його повторним введенням. Для завершення процесу необхідно натиснути кнопку «Створити обліковий запис», яку зображено на рисунку 3.25.

Створіть пароль

Пароль

.....

Підтвердіть пароль

.....

Створити обліковий запис

Рисунок 3.25 – Створення облікового запису

Поля пароля автоматично приховують символи для безпеки. У разі невідповідності паролів або використання слабкого пароля система виводить відповідне попередження. Після реєстрації або за наявності акаунту користувач може увійти до системи через форму авторизації, яку зображено на рисунку 3.26. Для цього вводяться email та пароль. У разі успішного входу користувач автоматично перенаправляється на головну сторінку свого кабінету.

Увійти

Електронна пошта

sezonivans21@gmail.com

Пароль

.....

Увійти

Рисунок 3.26 – Вхід у систему

На сторінці автопарку, який зображений на рисунку 3.27, користувач може переглянути доступні транспортні засоби. Зліва розташований список моделей, при виборі яких оновлюється фото авто та відображається детальна інформація: марка, модель, рік, кількість місць, наявність кондиціонера та ременів безпеки, орієнтовна вартість. Для переходу до замовлення потрібно натиснути кнопку «Забронювати».

Наш автопарк

Сучасні авто від провідних світових виробників
відповідають всім стандартам якості та безпеки

Остаточний розрахунок вартості оренди мікроавтобуса узгоджується з диспетчером після узгодження всіх побажань клієнта.

Детальна інформація	
Модель	Sprinter 318
Марка	Mercedes-Benz
Рік	2015
Місця	21
Кондиціонер	Так
Ремені безпеки	Так
Ціна	500 грн/год 20 грн/км

Рисунок 3.27 – Сторінка автопарку

На головній сторінці розміщено блок із формою пошуку транспорту, яку зображено на рис. 3.28. Користувач має можливість обрати тип транспорту, вказати пункт відправлення та призначення, а також обрати дати поїздки. Після натискання кнопки «Шукати» відкривається сторінка із завершенням бронювання, яке зображено на рисунку 3.29.

Забронювати транспорт

Оберіть транспорт: Mercedes-Benz Sprinter 318

Звідки: Київ

Куди: Львів


Дата виїзду: 25.05.2025

Дата повернення: 25.05.2025

Шукати

Рисунок 3.28 – Форма пошуку транспорту

Куди
Львів



Персональні дані

Ім'я	Телефон
Іван	0987374835
Прізвище	
Сезон	
Email	
sezonivans21@gmail.com	
Адреса	
Польова 4	
Місто	Поштовий індекс
Київ	12143

Надсилати новини та оновлення

Забронювати

Рисунок 3.29 – Завершення бронювання

Після завершення бронювання відкривається сторінка історія бронювань, яка зображена на рисунку 3.30.



Головна Послуги

Історія бронювань

Іван Сезон
 sezonivans21@gmail.com
 0987374835

2025-05-25 - 2025-05-25
 Київ - Львів
 08:00 - 20:00



Скасувати

Рисунок 3.30 – Сторінка історія бронювань

Для кожного бронювання виводиться ім'я, електронна адреса, номер телефону, маршрут поїздки, дати та час виїзду й повернення, а також зображення обраного транспортного засобу. Під карткою доступна кнопка «Скасувати», яка дозволяє скасувати замовлення.

Далі на головній сторінці на зустрічає блок «Чому обирають нас», який зображено на рисунку 3.31. Цей розділ містить ключові переваги сервісу: багаторічний досвід, пряме обслуговування без посередників, а також швидку подачу транспорту. Кнопка «Детальніше» у цьому блоці виконує роль навігаційного елемента, який спрямовує користувача на сторінку «Контакти».

Чому обирають нас

Будьте впевнені, нам можна довіряти.

Безпека, доступність та комфорт - це ті складові, якими ми користуємося щодня у нашій роботі. Також нашою основною перевагою є ще той факт, що ми працюємо без посередників, тобто можемо запропонувати нашим клієнтам найвигідніші ціни.

Детальніше >



Наш досвід

Ми вже понад 10 років на ринку. Нашими послугами скористалися понад 5000 пасажирів та сотні різних замовників.



Працюємо без посередників

Гнучка цінова політика, програма лояльності для постійних клієнтів. Високий рівень сервісу, надійність та професіоналізм.



Подача авто від 30 хвилин

Оперативне надання транспорту у необхідному обсязі. Цілодобова диспетчерська служба. Індивідуальний підхід до потреб клієнта та його побажань.

Рисунок 3.31 – Блок «Чому обирають нас»

На сторінці «Контакти», яка зображена на рисунку 3.32, користувач може заповнити форму для зворотного зв'язку: ввести повне ім'я, електронну пошту та повідомлення. Ліворуч розміщено контактні дані компанії — номер телефону, email і місцезнаходження. Надсилання запиту здійснюється кнопкою «Надіслати повідомлення».



Головна Послуги Наш Автопарк Відгуки Партнери Контакти



Потрібна додаткова інформація?

Якщо виникають питання, Ви завжди можете їх задати, скориставшись зручним для Вас засобом зв'язку.

☎ (098) 757 78 47

✉ pereviznyk2025@gmail.com

📍 Київ, Україна

Повне ім'я

Наприклад: "Іван Іванович"

Електронна пошта

sezonivans21@gmail.com

Розкажіть нам більше

Напишіть тут...

📩 Надіслати повідомлення

Рисунок 3.32 – Сторінка «Контакти»

Після надісланого повідомлення з'явиться коментар, що повідомлення було надіслано, і його отримає адміністрація.

На сторінці «Послуги» представлено повний перелік транспортних послуг, які надає компанія, зображено на рисунку 3.33. Це, зокрема, трансфери, екскурсії, обслуговування весіль, шкільних поїздок, корпоративних заходів і міжміських перевезень.



Головна Послуги Наш Автопарк Відгуки Партнери Контакти



Наші Послуги

Надаємо нашим клієнтам широкий спектр послуг та приймаємо індивідуальні замовлення

- ✓ Автобус на весілля
- ✓ Перевезення школярів
- ✓ Розвезення персоналу
- ✓ Автобус на екскурсію
- ✓ Автобус на випускний
- ✓ Автобусні тури
- ✓ Трансферні перевезення
- ✓ Міжміські перевезення



Ми до ваших послуг. Звертайтеся у будь-який час!

Рисунок 3.33 – Сторінка «Послуги»

На сторінці «Відгуки», яку зображено на рисунку 3.34, відображаються реальні відгуки користувачів про сервіс.

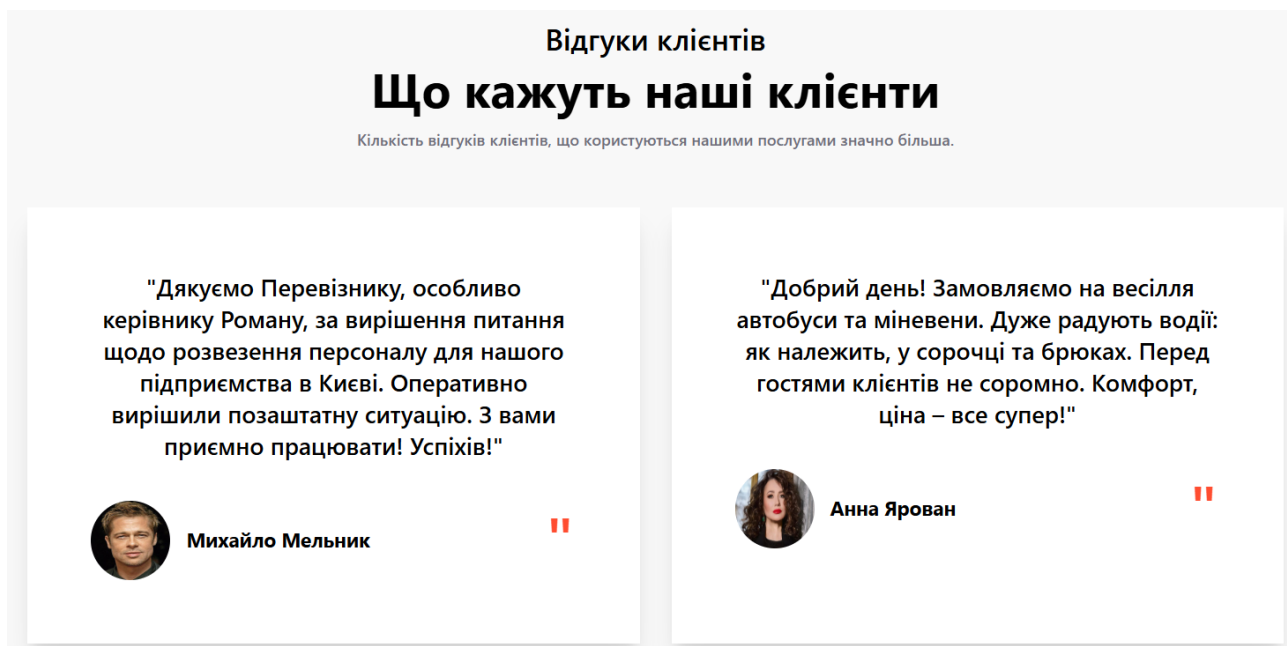


Рисунок 3.34 – Сторінка «Відгуки»

Сторінка «Партнери», яка зображена на рис. 3.35, містить логотипи та назви компаній, які користуються послугами Pereviznyk.

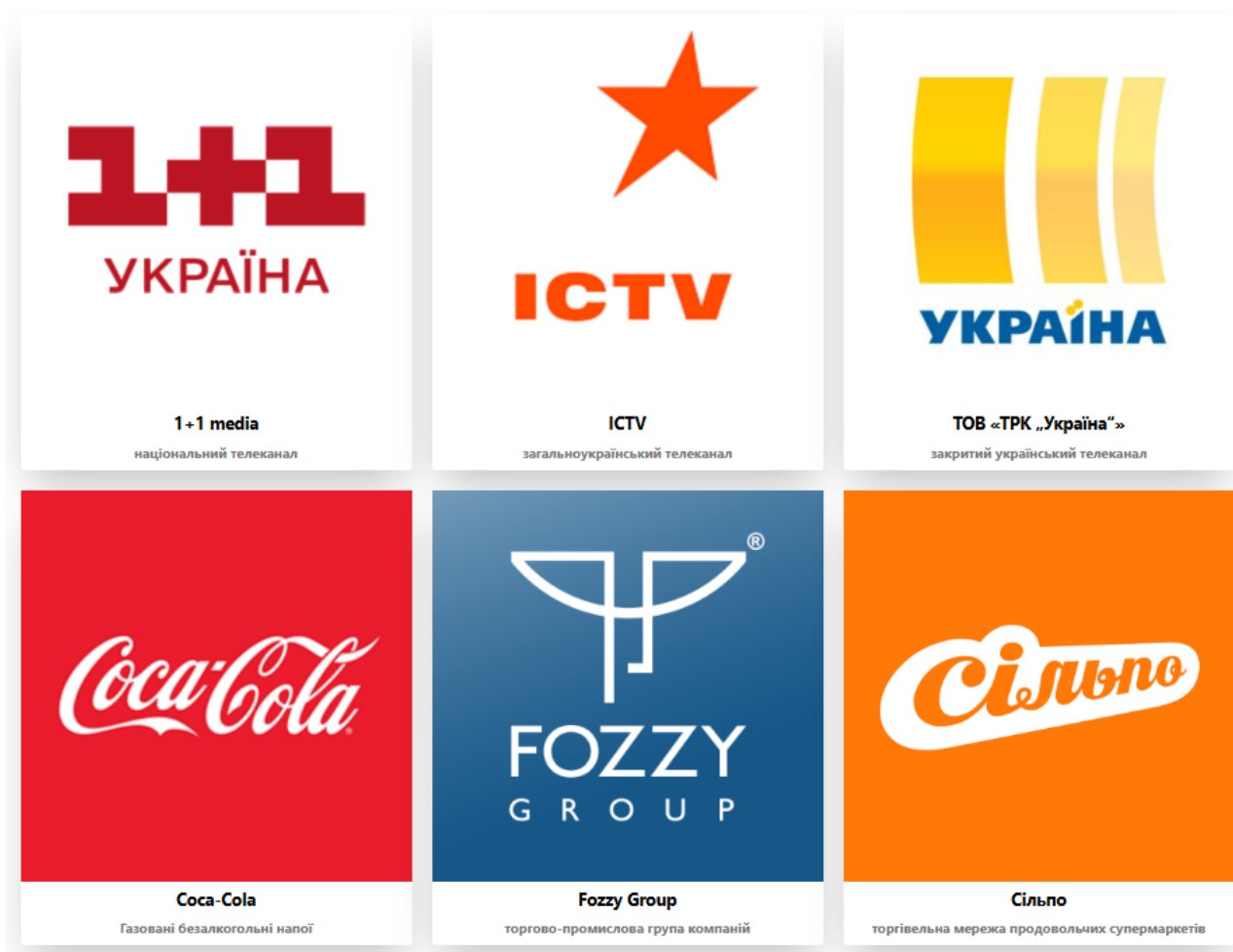


Рисунок 3.35 – Сторінка «Партнери»

Після входу в систему у правому верхньому куті з'являється іконка користувача, зображено на рисунку 3.36. При натисканні відкривається меню з двома опціями: «бронювання» — для перегляду історії замовлень, та «Вийти» — для завершення сеансу.

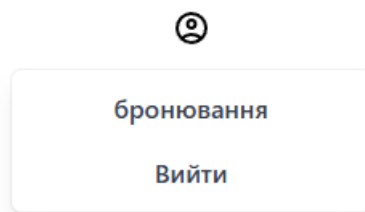


Рисунок 3.36 – Іконка користувача

3.5. Тестування програмного продукту

Тестування проводитиметься в браузерях — Google Chrome, Avast Secure та Microsoft Edge (Браузер 1, Браузер 2, Браузер 3) [15].

У межах роботи було проведено тестування основних функціональних модулів вебсайту системи «Pereviznyk». Перевірці підлягали ключові елементи взаємодії користувача із системою, включаючи реєстрацію, авторизацію, оформлення бронювання, надсилання зворотного зв'язку, перегляд історії замовлень та навігацію в межах особистого кабінету.

Форма реєстрації нового користувача працює стабільно, зображена на рисунку 3.37. Всі поля перевіряються на валідність, помилки у разі некоректного введення даних підсвічуються, а при успішному заповненні формується новий обліковий запис із перенаправленням до сторінки входу. Функція авторизації також виконується коректно — після введення логіна й пароля система переходить до особистого кабінету користувача [20].

1.						
Назва	Реєстрація користувача на сайті					
Опис	Перевірка функції реєстрація користувача					
Вимоги						
Статус	Виконано +					
№	Приоритет	Перевірка	Очікуваний результат	Результат перевірки		
Реєстрація користувача				Google Chrome	Avast Secure	Microsoft Edge
<i>Перевірки для реєстрації користувача</i>						
1	високий	Відкрити сторінку реєстрації	Форма реєстрації відображається	Passed	Passed	Passed
2	високий	Ввести валідні логін, email, пароль	Поля приймаються, не підсвічуються помилками, якщо все введено правильно	Passed	Passed	Passed
3	високий	Натиснути кнопку «Зареєструватися»	Користувач створюється в базі даних	Passed	Passed	Passed
4	високий	Відбувається перехід на Вхід у акаунт	Відображається Вхід у акаунт	Passed	Passed	Passed

Рисунок 3.37 – Результат тестування реєстрації користувача

Оформлення бронювання реалізовано через форму з багатоетапним введенням даних. У процесі тестування встановлено, що форма заповнюється без помилок, при натисканні кнопки «Підтвердити» замовлення зберігається в базі даних, і користувач перенаправляється на сторінку з історією бронювань. Всі створені замовлення відображаються коректно, з актуальними параметрами поїздки та зображенням обраного транспортного засобу. Також було перевірено функцію **скасування бронювання** — після її активації запис видаляється як із бази, так і з інтерфейсу користувача. Результат зображено на рисунку 3.38.

Назва	Бронювання транспорту					
Опис	Перевірка функції бронювання транспорту на сайті					
Вимоги						
Статус	Виконано +					
№	Приоритет	Перевірка	Очікуваний результат	Результат перевірки		
Бронювання туру				Google Chrome	Avast Secure	Microsoft Edge
<i>Перевірки для бронювання</i>						
1	високий	Відкрити сторінку сторінку з автопарком	Відображається весь автопарк	Passed	Passed	Passed
2	високий	Перехід на форму бронювання	Відображається форма з усіма полями	Passed	Passed	Passed
3	високий	Заповнити форму повністю	Форма приймається без помилок	Passed	Passed	Passed
4	високий	Натиснути кнопку "Забронювати"	Бронювання створюється в базі даних	Passed	Passed	Passed
5	високий	Відображення замовлення про в історії бронювань	Всі дані присутні та відображаються правильно	Passed	Passed	Passed

Рисунок 3.38 – Результат тестування функції бронювання транспорту

Надсилання повідомлення через форму зворотного зв'язку на сторінці контактів працює відповідно до вимог. Дані перевіряються на заповнення, а після натискання кнопки «Надіслати повідомлення» виводиться підтвердження про успішну відправку. Повідомлення зберігається у базі для подальшої обробки адміністрацією. Результат тестування функції надсилання повідомлення зображено на рисунку 3.39.

Назва	Надсилання повідомлення					
Опис	Перевірка роботи функції надсилання повідомлення					
Вимоги						
Статус	Виконано +					
№	Приоритет	Перевірка	Очікуваний результат	Результат перевірки		
Відгуки на тури				Google Chrome	Avast Secure	Microsoft Edge
<i>Перевірка роботи функції надсилання повідомлення</i>						
1	високий	Відкрити сторінку Контакти	Відображення форми для зворотного зв'язку:	Passed	Passed	Passed
2	високий	Ввести повне ім'я, електронну пошту та повідомлення	Форма приймається без помилок	Passed	Passed	Passed
3	високий	Натиснути кнопку «Надіслати повідомлення»	Коментар, що повідомлення було надіслано	Passed	Passed	Passed

Рисунок 3.39 – Результат тестування функції надсилання повідомлення

На сторінці «Мій акаунт» було протестовано доступ до історії бронювань, навігацію за допомогою кнопок «Скасувати бронювання» та «Вийти». Усі функції виконуються належним чином — сторінки відображаються без помилок, інтерфейс адаптований для різних екранів, а вихід із системи завершується завершенням сесії

та поверненням на головну сторінку. Результат тестування зображено на рисунку 3.40.

Назва	Мій Акаунт					
Опис	Перевірка роботи функцій акаунту					
Вимоги						
Статус	Виконано +					
№	Приоритет	Перевірка	Очікуваний результат	Результат перевірки		
Фільтрація, сортування, пошук турів за певними параметрами				Google Chrome	Avast Secure	Microsoft Edge
<i>Перевірка роботи функцій акаунту</i>						
1	високий	Відкрити сторінку Акаунт	Відображена сторінка акаунту	Passed	Passed	Passed
2	високий	Кнопка мої бронювання	Список історії бронювань	Passed	Passed	Passed
3	високий	Кнопка Скасувати бронювання	Скасоване бронювання	Passed	Passed	Passed
4	високий	Кнопка Вихід	Успішний вихід з акаунту	Passed	Passed	Passed
5	високий	Кнопка "Скинути" фільтри	Відображаються всі доступні тури	Passed	Passed	Passed

Рисунок 3.40 – Результат тестування “Мій акаунт”

В усіх випадках функціонал працював коректно, жодних критичних збоїв чи помилок виявлено не було.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досягнуто поставленої мети — розроблено веб-орієнтовану інформаційну систему для управління замовленнями пасажирських перевезень, що надаються ФОП Салавор Р. Т. Система створена відповідно до вимог, викладених у технічному завданні, та охоплює ключові етапи життєвого циклу розробки програмного забезпечення: аналіз предметної області, проектування структури бази даних, реалізацію клієнтської та серверної частин, та тестування роботи системи.

У процесі реалізації були розв'язані такі завдання: досліджено предметну область та потреби користувачів, розроблено технічне завдання, спроектовано структуру бази даних MongoDB, реалізовано інтерфейс із використанням бібліотеки React, налаштовано сервер на базі Node.js з Express. Створено механізми реєстрації та авторизації користувачів, обробки бронювань, перегляду історії замовлень, а також надсилання повідомлень. Усі функції протестовано в різних браузерах, критичних помилок не виявлено.

Розроблена система дозволяє ефективно обробляти запити на перевезення, забезпечує зберігання інформації про клієнтів і бронювання, надає користувачам можливість переглядати історію замовлень, залишати повідомлення та взаємодіяти з сервісом у зручній формі. Очікуваний ефект від впровадження полягає у зменшенні часу обробки замовлень, збільшенні кількості клієнтів, підвищенні якості обслуговування та впізнаваності бренду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні рекомендації до виконання кваліфікаційної роботи на здобуття освітнього ступеня «бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки» денної та заочної форми здобуття освіти [Електрон. ресурс] / уклад. С. В. Грибков, Н. В. Ліманська, М. П. Костіков. – К.: НУХТ, 2025. – 43 с.
2. Mega-trans. Офіційний сайт [Електронний ресурс] – Режим доступу: <https://mega-trans.com.ua/> (дата звернення: 20.05.2025).
3. Kievtrans. Офіційний сайт [Електронний ресурс] – Режим доступу: <https://kievtrans.kiev.ua/> (дата звернення: 20.05.2025).
4. Passtrans. Офіційний сайт [Електронний ресурс] – Режим доступу: <https://www.passtrans.kiev.ua/> (дата звернення: 20.05.2025).
5. JavaScript: повне керівництво / Д. Фленаган. – Київ: Діалектика, 2021. – 950 с.
6. React офіційна документація [Електронний ресурс] – Режим доступу: <https://legacy.reactjs.org/docs/getting-started.html> (дата звернення: 20.05.2025).
7. Node.js офіційна документація [Електронний ресурс] – Режим доступу: <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs> (дата звернення: 20.05.2025).
8. Express.js офіційна документація [Електронний ресурс] – Режим доступу: <https://expressjs.com/en/guide/routing.html> (дата звернення: 20.05.2025).
9. MongoDB офіційна документація [Електронний ресурс] – Режим доступу: <https://www.mongodb.com/docs/atlas/> (дата звернення: 20.05.2025).
10. Visual Studio Code офіційна документація [Електронний ресурс] – Режим доступу: <https://code.visualstudio.com/docs/nodejs/nodejs-deployment> (дата звернення: 20.05.2025).
11. Vite офіційна документація [Електронний ресурс] – Режим доступу: <https://vite.dev/guide/> (дата звернення: 20.05.2025).
12. Vercel офіційна документація [Електронний ресурс] – Режим доступу: <https://vercel.com/guides> (дата звернення: 21.05.2025).

13. Tailwind CSS офіційна документація [Електронний ресурс] – Режим доступу: <https://tailwindcss.com/docs/installation/using-vite> (дата звернення: 21.05.2025).

14. Технології проектування No Sql баз даних [Електронний ресурс] : навч. посіб. для здобувачів ступеня магістра за освіт. програмою «Інформаційне забезпечення робототехнічних систем» спец. 126 Інформаційні системи та технології / Є. В. Крилов, В. К. Анікін, ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл). – Київ : КПІ ім. Ігоря Сікорського, 2023. – 117 с

15. Авраменко А.С. Тестування програмного забезпечення: навчальний посібник / А. С. Авраменко, В. С. Авраменко, Г.В. Косенюк– Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с.

16. Проектування інформаційних систем: Загальні питання теорії проектування ІС (конспект лекцій) [Електронний ресурс]: навч. посіб. для студ. спеціальності 122 «Комп'ютерні науки» / КПІ ім. Ігоря Сікорського; уклад.: О. С. Коваленко, Л. М. Добровська. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 192с. Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/c136860d-44cb-4f05-adaf-dcdd20830483/content> (дата звернення: 21.05.2025).

17. Дружиніна В. В., Черніхова О. С. (2024). Управління логістичними процесами на ринку транспортних пасажирських перевезень. Стаття [Електронний ресурс] – Режим доступу: <https://economdevelopment.in.ua/index.php/journal/article/view/896/858> (дата звернення: 21.05.2025).

18. Oracle. What Is the MERN Stack? Guide & Examples. Офіційна документація [Електронний ресурс] – Режим доступу: <https://www.oracle.com/ua/database/mern-stack/> (дата звернення: 21.05.2025).

19. Axios офіційна документація [Електронний ресурс] – Режим доступу: <https://axios-http.com/docs/intro> (дата звернення: 21.05.2025).

20. Канер, С. Тестування програмного забезпечення: фундаментальні принципи : пер. з англ. / С. Канер, Дж. Фолкс, Г. Нгуєн. – Львів : БаК, 2017. – 320 с.

21. Урядовий портал. Правила надання послуг пасажирського автомобільного транспорту [Електронний ресурс] – Режим доступу: <https://www.kmu.gov.ua/npas/93701422> (дата звернення: 21.05.2025).

ДОДАТКИ

Додаток А. Організаційна структура та бізнес процеси ФОП Салавор Р.Т.

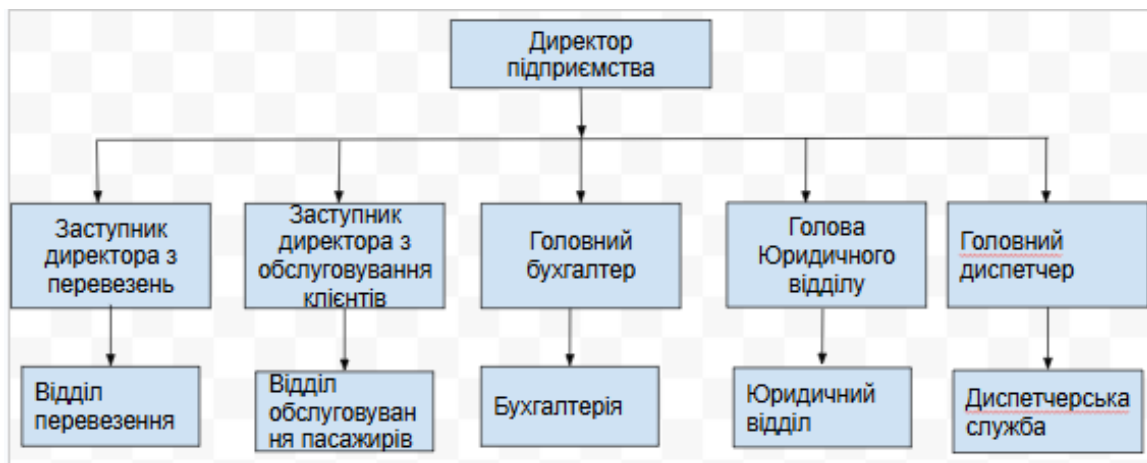


Рисунок А.1 – Організаційно-функціональна схема підприємства з пасажирських перевезень



Рисунок А.2 – Структурна схема відділу перевезень

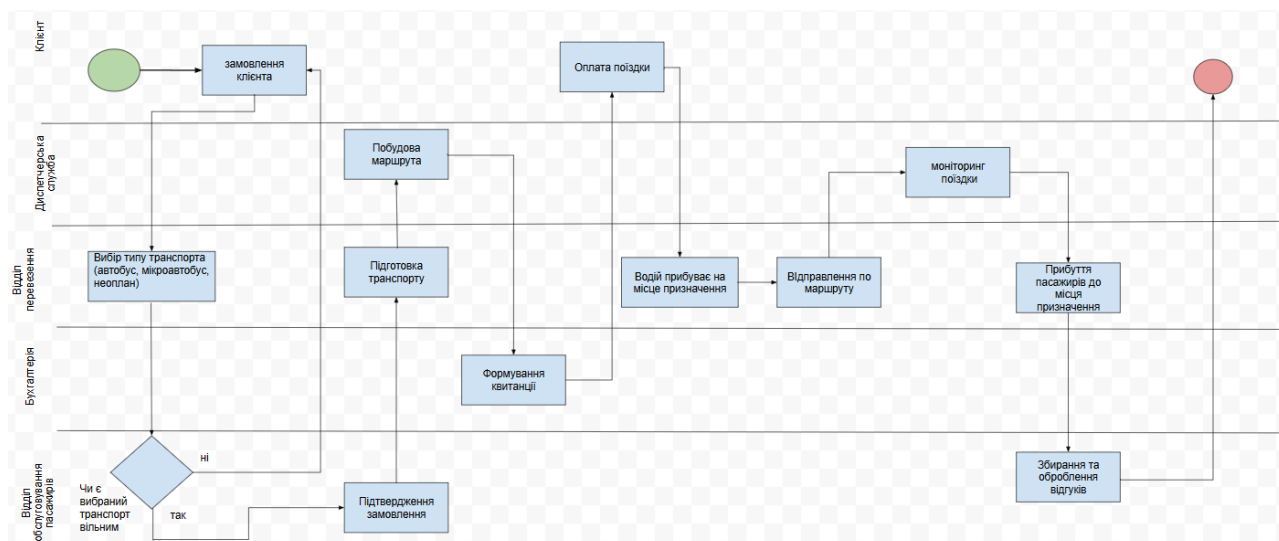


Рисунок А.3 – Бізнес-процеси від замовлення клієнта до завершення замовлення

Додаток Б. Фрагменти коду

Додаток Б.1 – код з файлу aboutsection.jsx

```

import React from 'react';
import About from '../images/about.jpg';
import Truck from "../images/truck.png";
import garrage from "../images/garrage.png";
import outlet from "../images/outlet.png";

export default function AboutSection(){
  return(
    <>
    <div className="container mx-auto p-6 my-20">
    <div className="grid lg:grid-cols-2 ">
    <div className="flex justify-center">
    <img className="z-20 shadow-lg rounded w-[500px]" src={About} alt=""/>
    </div>
    <div className="w-full md:w-3/4 text-center md:text-start">
    <h2 className="font-sans font-bold my-6 text-4xl">Наші Послуги</h2>
    <h1 className="font-sans font-bold my-4 text-2xl">Надаємо нашим клієнтам
широкий спектр послуг
та приймаємо індивідуальні замовлення</h1>
    <ul className="text-lg my-6 text-[#706f7b] font-sans space-y-3 list-none"> <li
className="flex items-start">
    <svg className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor"
viewBox="0 0 20 20">
    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />
    </svg> Автобус на весілля </li> <li className="flex items-start"> <svg
className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor" viewBox="0 0 20
20">

```

```

    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />

```

```

</svg> Перевезення школярів </li> <li className="flex items-start"> <svg
className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor" viewBox="0 0 20
20">

```

```

    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />

```

```

</svg> Розвезення персоналу </li> <li className="flex items-start"> <svg
className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor" viewBox="0 0 20
20">

```

```

    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />

```

```

</svg> Автобус на екскурсію </li> <li className="flex items-start"> <svg
className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor" viewBox="0 0 20
20">

```

```

    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />

```

```

</svg> Автобус на випускний </li> <li className="flex items-start"> <svg
className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor" viewBox="0 0 20
20">

```

```

    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />

```

```

</svg> Автобусні тури </li> <li className="flex items-start"> <svg
className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor" viewBox="0 0 20
20">

```

```

    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />

```

```

</svg> Трансферні перевезення </li> <li className="flex items-start"> <svg
className="w-6 h-6 text-orange mr-2 mt-1" fill="currentColor" viewBox="0 0 20
20">

```

```

    <path fillRule="evenodd" d="M16.707 5.293a1 1 0 00-1.414 0L8 12.586 4.707
9.293a1 1 0 10-1.414 1.414l4 4a1 1 0 001.414 0l8-8a1 1 0 000-1.414z"
clipRule="evenodd" />

```

```

</svg> Міжміські перевезення </li> </ul>

```

```

<div className="flex justify-center md:justify-start gap-6 mt-10">

```

```

<img src={Truck} alt="" className="" />

```

```

<img src={garrage} alt="" className="" />

```

```

<img src={outlet} alt="" className="" />

```

```

</div>

```

```

<p className="text-xl font-bold mt-10 text-[#706f7b]">Ми до ваших послуг.

```

```

Звертайтеся у будь-який час!</p>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

</>

```

```

)

```

```

}

```

Додаток Б.2 – код з файлу Banner.jsx

```

import React from 'react';

```

```

import jeep from "../images/jeep.png";

```

```

export default function Banner() {

```

```

  return (

```

```

    <>

```

```

<div className="w-full bg-[#2d2d2d]">
  <div className="container mx-auto py-12 justify-center text-center">
    <h2 className="text-white font-bold font-sans text-6xl py-3">Регулярне
розвезення ваших працівників</h2>
    <h3 className="text-white font-semibold font-sans text-3xl py-3">Організуємо
розвезення співробітників вашої організації на регулярній основі.
    Вигідна цінова політика для таких послуг, адже в наших інтересах —
стабільна зайнятість транспорту, а у ваших — <span className="text-orange">
Комфорт, Безпека Та Своєчасне Прибуття Персоналу.</span></h3>
  </div>
</div>
<div className="flex justify-center mt-24">
  <img className="z-50 w-[1000px] h-auto" src={jeep} alt=""/>
</div>
</>
)
}

```

Додаток Б.3 – код з файлу Bookcar.jsx

```

import React, { useContext, useEffect, useState } from 'react';
import LocationOnOutlinedIcon from '@mui/icons-
material/LocationOnOutlined';
import CalendarMonthOutlinedIcon from '@mui/icons-
material/CalendarMonthOutlined';
import sprinter318 from "../images/sprinter318.jpg";
import transit from "../images/transit.jpg";
import vito from "../images/vito.jpg";
import expert from "../images/expert.jpg";
import sprinter516 from "../images/sprinter516.jpg";
import tourino from "../images/tourino.jpg";
import { Link, useNavigate } from 'react-router-dom';

```

```

import CloseIcon from '@mui/icons-material/Close';
import axios from 'axios';
import { useContext } from '../Context/Clientcontext';

export default function Bookcar({isDivVisible, setDivVisible, carType,
pickPlace, dropPlace, pickDate, dropDate}){

  const {user} = useContext(UserContext);
  const [pickTime, setPickTime] = useState("");
  const [dropTime, setDropTime] = useState("");
  const [selectedImage, setSelectedimage] = useState();
  const [firstname, setFirstname] = useState("");
  const [lastname, setLastName] = useState("");
  const [phone, setPhone] = useState("");
  const [email, setEmail] = useState("");
  const [address, setAddress] = useState("");
  const [city, setCity] = useState("");
  const [zipcode, setZipcode] = useState("");
  const [showWarning, setShowwarning] = useState(false);
  const navigate = useNavigate();

  const isReserveDisabled = !pickTime || !dropTime || !firstname || !lastname ||
!phone || !email || !address || !city || !zipcode;

  const Cars = [
    {name: "Mercedes-Benz Sprinter 318", image: sprinter318},
    {name: "Ford Transit", image: transit},
    {name: "Mercedes-Benz Vito", image: vito},
    {name: "Peugeot Expert", image: expert},
    {name: "Mercedes-Benz Sprinter 516", image: sprinter516},
  ]

```

```

    {name: "Mercedes-Benz Tourino", image: tourino},
  ];

  useEffect(() => {
    const car = Cars.find(car => car.name === carType);
    if (car) setSelectedimage(car.image);
  }, [carType]);

  const ReserveCar = async () => {
    if (isReserveDisabled) {
      setShowwarning(true);
      setTimeout(() => setShowwarning(false), 1500);
      return;
    }
    if (!user) {
      navigate('/login');
    } else {
      try {
        const { data } = await axios.post('/reservation', {
          carType, pickPlace, dropPlace, pickDate, dropDate, pickTime,
          dropTime,
          firstname, lastname, phone, email, address, city, zipcode
        });
        navigate('/account/bookings');
      } catch (e) {
        console.log(e);
      }
    }
  };

```

```

return (
  <
    {isDivVisible && (
      <div className="container border-4 border-white shadow-2xl
overflow-x-hidden mx-auto bg-white absolute z-50 w-3/5 flex flex-col justify-center
items-center" style={{ top: '170%', left: '50%', transform: 'translate(-50%, -50%)' }}>
        <div className="flex justify-between px-4 py-2 bg-orange w-full">
          <h2 className="font-bold font-sans text-3xl text-
white">ЗАВЕРШЕННЯ БРОНЮВАННЯ</h2>
          <CloseIcon onClick={() => setDivVisible(false)}
className="text-white" style={{ cursor: "pointer" }} />
        </div>
        <div className="bg-[#ffae6] justify-start py-4 px-4 w-full">
          <h2 className="font-bold font-sans text-2xl py-2 text-
orange">Після завершення бронювання ви отримаєте:</h2>
          <p className="font-bold font-sans text-base py-2 text-
[#777]">Ваш ваучер та номер служби підтримки.</p>
        </div>
        <div className="grid lg:grid-cols-2 w-full border-b">
          <div className="justify-start py-4 px-4 w-full bg-white">
            <h2 className="text-orange font-bold font-sans text-
xl">Локація та дата</h2>
            <h3 className="py-2 font-bold font-sans text-
lg"><CalendarMonthOutlinedIcon /> Дата та час виїзду</h3>
            <p className="text-base px-2 text-[#777] font-
sans">{pickDate} / </p><input value={pickTime} onChange={ev =>
setPickTime(ev.target.value)} max={dropTime} className="border-4 focus:outline-
none" type="time" />
            <h3 className="py-2 font-bold font-sans text-
lg"><CalendarMonthOutlinedIcon /> Дата та час повернення</h3>

```

```

        <p className="text-base px-2 text-[#777] font-
sans">{dropDate} </p><input value={dropTime} onChange={ev =>
setDropTime(ev.target.value)} min={pickTime} className="border-4 focus:outline-
none" type="time" />
        <h3 className="py-2 font-bold font-sans text-
lg"><LocationOnOutlinedIcon /> Звідки</h3>
        <p className="text-base px-2 text-[#777] font-
sans">{pickPlace}</p>
        <h3 className="py-2 font-bold font-sans text-
lg"><LocationOnOutlinedIcon /> Куди</h3>
        <p className="text-base px-2 text-[#777] font-
sans">{dropPlace}</p>
    </div>
    <div className="justify-center py-4 px-4 w-full bg-white">
        <h2 className="font-bold font-sans text-lg">Транспорт:
<span className="text-orange font-bold font-sans text-lg">{carType}</span></h2>
        <img src={selectedImage} alt="" />
    </div>
</div>
<h2 className="text-orange py-4 font-bold font-sans text-
xl">Персональні дані</h2>
    <div className="grid lg:grid-cols-2 w-full bg-white">
        <div className="py-2 px-4">
            <label className="text-lg text-[#777] mx-2 font-semibold font-
sans">Ім'я</label><br />
            <input value={firstname} onChange={ev =>
setFirstname(ev.target.value)} type="text" className="focus:outline-none my-2 px-2
w-full py-3 bg-[#dbdbdb]" placeholder="Ім'я" />
            <label className="text-lg text-[#777] mx-2 font-semibold font-
sans">Прізвище</label><br />

```

```

        <input value={lastname} onChange={ev =>
setLastName(ev.target.value)} type="text" className="focus:outline-none my-2 px-2
w-full py-3 bg-[#dbdbdb]" placeholder="Прізвище" />
    </div>
    <div className="py-2 px-4">
        <label className="text-lg text-[#777] mx-2 font-semibold font-
sans">Телефон</label><br />
        <input value={phone} onChange={ev =>
setPhone(ev.target.value)} type="text" className="focus:outline-none my-2 px-2 w-
full py-3 bg-[#dbdbdb]" placeholder="Телефон" />
    </div>
</div>
<div className="grid grid-rows-2 w-full">
    <div className="px-4">
        <label className="text-lg text-[#777] mx-2 font-semibold font-
sans">Email</label><br />
        <input value={email} onChange={ev =>
setEmail(ev.target.value)} type="text" className="focus:outline-none my-2 px-2 w-
full py-3 bg-[#dbdbdb]" placeholder="Email" />
    </div>
    <div className="px-4">
        <label className="text-lg text-[#777] mx-2 font-semibold font-
sans">Адреса</label><br />
        <input value={address} onChange={ev =>
setAddress(ev.target.value)} type="text" className="focus:outline-none my-2 px-2 w-
full py-3 bg-[#dbdbdb]" placeholder="Адреса" />
    </div>
</div>
<div className="grid lg:grid-cols-2 w-full border-b">
    <div className='px-4'>

```

```

<label className="text-lg text-[#777] mx-2 font-semibold font-
sans">Місто</label><br />
<input value={city} onChange={ev => setCity(ev.target.value)}
type="text" className="focus:outline-none my-2 px-2 w-full py-3 bg-[#dbdbdb]"
placeholder="Місто" />
</div>
<div className='px-4'>
<label className="text-lg text-[#777] mx-2 font-semibold font-
sans">ПОШТОВИЙ ІНДЕКС</label><br />
<input value={zipcode} onChange={ev =>
setZipcode(ev.target.value)} type="text" className="focus:outline-none my-2 px-2 w-
full py-3 bg-[#dbdbdb]" placeholder="Індекс" />
</div>
</div>
<div className="px-4 py-4 w-full">
<input type="checkbox" /> <label className="text-lg px-2 text-
[#777] font-sans font-semibold">Надсилати новини та оновлення</label>
</div>
{showWarning && (
<div className="px-4 py-2 bg-[#dbdbdb] w-full flex justify-
center">
<h2 className="text-orange">Будь ласка, заповніть усі
обов'язкові поля.</h2>
</div>
)}
<div className="px-4 py-8 bg-[#dbdbdb] w-full flex justify-
center">
<Link to={!user && '/login'} onClick={ReserveCar}
className="bg-orange px-6 py-3 text-white font-sans font-semibold
rounded">Забронювати</Link>

```

```

        </div>
      </div>
    )}
  </>
);
}

```

Додаток Б.4 – код з файлу Cards.jsx

```

import React from 'react';
import car from "../images/selectcar.png";
import operator from "../images/operator.png";
import drive from "../images/drive.png";

export default function Cards() {
  return (
    <div className="container justify-center mx-auto mb-40">
      <div className="text-center mb-32">
        <h2 className="font-bold text-2xl my-6">
          Сплануйте свою поїздку вже зараз
        </h2>
        <h1 className="font-bold text-5xl font-[poppins,sans-serif] my-6">
          Швидке та зручне бронювання транспорту
        </h1>
      </div>
      <div className="grid lg:grid-cols-3 gap-4">
        <div className="bg-white w-full flex flex-col justify-center items-center">
          <img src={car} alt="" />
          <div className="text-center w-3/5">
            <h2 className="text-3xl font-bold mb-6">Оберіть транспорт</h2>
            <p className="font-[Rubik,sans-serif] text-[#706f7b]">

```

Ми пропонуємо великий вибір автобусів та мікроавтобусів для ваших потреб у перевезеннях. Оберіть відповідний транспорт під вашу подорож.

```
</p>
```

```
</div>
```

```
</div>
```

```
<div className="bg-white w-full flex flex-col justify-center items-center">
```

```
<img src={operator} alt="" />
```

```
<h2 className="text-3xl font-bold mb-6">Погоджуємо замовлення</h2>
```

```
<div className="text-center w-3/5">
```

```
<p className="font-[Rubik,sans-serif] text-[#706f7b]">
```

Наші привітні оператори завжди готові допомогти вам з будь-якими запитаннями чи уточненнями щодо замовлення.

```
</p>
```

```
</div>
```

```
</div>
```

```
<div className="bg-white w-full flex flex-col justify-center items-center">
```

```
<img src={drive} alt="" />
```

```
<div className="text-center w-3/5">
```

```
<h2 className="text-3xl font-bold mb-6">Виконуємо перевезення</h2>
```

```
<p className="font-[Rubik,sans-serif] text-[#706f7b]">
```

Вирушайте в дорогу комфортно! Ми забезпечимо вас надійним транспортом відповідно до вашого маршруту.

```
</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</>
```

```
);
```

```
}
```

Додаток Б.5 – код з файлу Carmodels.jsx

```
import React from 'react';
import sprinter318 from "../images/sprinter318.jpg";
import transit from "../images/transit.jpg";
import vito from "../images/vito.jpg";
import expert from "../images/expert.jpg";
import sprinter516 from "../images/sprinter516.jpg";
import tourino from "../images/tourino.jpg";
import CarCard from './Reactcard';

export default function Carmodel() {

  const cars = [
    {
      name: "Mercedes-Benz Sprinter 318",
      model: "Sprinter 318",
      rent: "$ 90",
      image: sprinter318
    },
    {
      name: "Ford Transit",
      model: "Transit",
      rent: "$ 85",
      image: transit
    },
    {
      name: "Mercedes-Benz Vito",
      model: "Vito",
      rent: "$ 75",
      image: vito
    }
  ]
}
```

```

    },
    {
      name: "Peugeot Expert",
      model: "Expert",
      rent: "$ 70",
      image: expert
    },
    {
      name: "Mercedes-Benz Sprinter 516",
      model: "Sprinter 516",
      rent: "$ 95",
      image: sprinter516
    },
    {
      name: "Mercedes-Benz Tourino",
      model: "Tourino",
      rent: "$ 120",
      image: tourino
    }
  ];

return (
  <div className="container mx-auto md:p-6 mb-16">
    <div className="grid lg:grid-cols-3 gap-8 px-16">
      {cars.map((car, index) => (
        <div key={index}>
          <CarCard
            image={car.image}
            name={car.name}
            model={car.model}

```

```

        rent={ car.rent }
      />
    </div>
  )))
</div>
</div>
);
}

```

Додаток Б.6 – код з файлу downloads.jsx

```

import React from 'react';
import gplay from "../images/gplay.svg";
import appstore from "../images/appstore.svg";
import benz from "../images/benz.jpg";

export default function Downloads() {
  return (
    <>
    <div className="container mx-auto p-6 justify-center mt-28 mb-20">
      <div className="grid lg:grid-cols-2 gap-4 px-16">
        <div className="">
          <img className="" src={benz} alt="" />
        </div>
        <div className="w-3/4 text-center">
          <h1 className='text-4xl font-bold font-sans mb-4'>Завантажте наш
застосунок, щоб отримати максимум можливостей</h1>
          <p className="text-base text-[#706f7b] font-sans">
            Завантажте наш мобільний застосунок та скористайтесь усіма перевагами
онлайн-бронювання, спеціальних пропозицій та зручного керування поїздками.
          </p>
        </div>
      </div>
    </div>
  )
}

```

```

    <a target='_blank'
href="https://play.google.com/store/search?q=carrental&c=apps">
    <img className="transform hover:-translate-x-2 hover:-translate-y-2 transition
duration-300 hover:shadow-xl" style={{ cursor:"pointer" }} src={gplay} alt="Google
Play" />
    </a>
    <a target='_blank'
href="https://www.apple.com/in/search/carrental?src=globalnav">
    <img className="transform hover:-translate-x-2 hover:-translate-y-2 transition
duration-300 hover:shadow-xl" style={{ cursor:"pointer" }} src={appstore} alt="App
Store" />
    </a>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
)
}

```

Додаток Б.7 – код з файлу Footer.jsx

```

import React from 'react';
import PhoneIcon from '@mui/icons-material/Phone';
import EmailIcon from '@mui/icons-material/Email';

export default function Footer(){
  return(
    <>
    <div className="bg-[#f8f8f8] mt-10">
    <div className="container mx-auto p-6">
    <div className="grid lg:grid-cols-4 gap-6">

```

```

<div className="">
  <h1 className="font-sans font-bold text-3xl">PERE <span className='text-xl
font-semibold font-sans">VIZNYK</span></h1>
  <p className="font-sans text-base text-[#706f7b] my-2">Пасажирські
перевезення по Києву та Україні. Замовлення транспорту з водієм. Регулярне
розвезення ваших працівників.</p>
  <h2 style={{ cursor:"pointer" }} className="font-sans font-bold text-lg my-1
hover:text-orange"><PhoneIcon/> (098) 757 78 47</h2>
  <h2 style={{ cursor:"pointer" }} className="font-sans font-bold text-lg my-1
hover:text-orange"><EmailIcon/> pereviznyk2025@gmail.com</h2>
  <h2 style={{ cursor:"pointer" }} className="font-sans text-lg my-1 hover:text-
orange">Дизайн від LazyStudio</h2>
</div>
<div className="">
  <h2 className="font-sans font-bold text-2xl">Про нас</h2>
  <p style={{ cursor:"pointer" }} className="font-sans text-lg mt-1 hover:text-
orange">Наші послуги</p>
  <p style={{ cursor:"pointer" }} className="font-sans text-lg mt-1 hover:text-
orange">Автомобілі</p>
  <p style={{ cursor:"pointer" }} className="font-sans text-lg mt-1 hover:text-
orange">Відгуки</p>
  <p style={{ cursor:"pointer" }} className="font-sans text-lg mt-1 hover:text-
orange">Партнери</p>
  <p style={{ cursor:"pointer" }} className="font-sans text-lg mt-1 hover:text-
orange">Контакти</p>
</div>
<div className="">
  <h2 className="font-sans font-bold text-2xl">ГОДИНИ РОБОТИ</h2>
  <p className="font-sans text-lg mt-1">Без вихідних</p>
  <p className="font-sans text-lg mt-1">Пн - Пт: 8:00 - 20:00</p>

```



```

toggleDiv,
carType,
setCarType,
pickPlace,
setPickPlace,
dropPlace,
setDropPlace,
pickDate,
setPickDate,
dropDate,
setDropDate
}) {
  const [checkFields, setCheckfields] = useState(false);

  const isSearchDisabled = !carType || !pickPlace || !dropPlace || !pickDate ||
!dropDate;

  const cars = [
    'Оберіть авто',
    'Mercedes-Benz Sprinter 318',
    'Ford Transit',
    'Mercedes-Benz Vito',
    'Peugeot Expert',
    'Mercedes-Benz Sprinter 516',
    'Mercedes-Benz Tourino'
  ];

  const handleSearch = (ev) => {
    ev.preventDefault();
    if (isSearchDisabled) {

```

```

    setCheckfields(true);
    setTimeout(() => setCheckfields(false), 1000);
  } else {
    toggleDiv();
  }
};

return (
  <form ref={reservationFormRef}>
    <div className="container justify-center mx-auto mb-40">
      <div id="formbg" className="lg:mx-16 py-10 z-30 shadow-2xl">
        {checkFields && (
          <h2 className="font-sans font-bold text-xl px-12 mb-8 text-orange">
            Будь ласка, заповніть усі поля.
          </h2>
        )}
        <h4 className="mt-6 font-sans font-bold text-xl px-12 mb-
8">Забронювати транспорт</h4>
        <div className="grid lg:grid-cols-3 px-12 gap-4">
          <div className="mb-8">
            <label className="font-semibold text-lg">
              <span className="text-orange"><DirectionsCarOutlinedIcon
/></span> Оберіть транспорт
            </label><br />
            <select
              value={carType}
              onChange={ev => setCarType(ev.target.value)}
              className="mt-4 border border-[#ccd7e6] rounded text-[#ababab]
text-xl py-2 w-full focus:outline-none"
            >

```

```

    {cars.map((car, index) => (
      <option key={index} value={car}>{car}</option>
    ))}
  </select>
</div>
<div className="mb-8">
  <label className="font-semibold text-lg">
    <span className='text-orange'><LocationOnOutlinedIcon /></span>

```

Звідки

```

  </label><br />
  <input
    value={pickPlace}
    onChange={ev => setPickPlace(ev.target.value)}
    placeholder="Наприклад: Київ"
    className="mt-4 border border-[#ccd7e6] rounded text-[#ababab]
text-xl py-2 w-full focus:outline-none"

```

```

  />

```

```

</div>
<div className="mb-8">
  <label className="font-semibold text-lg">
    <span className='text-orange'><LocationOnOutlinedIcon /></span>

```

Куди

```

  </label><br />
  <input
    value={dropPlace}
    onChange={ev => setDropPlace(ev.target.value)}
    placeholder="Наприклад: Львів"
    className="mt-4 border border-[#ccd7e6] rounded text-[#ababab]
text-xl py-2 w-full focus:outline-none"

```

```

  />

```

```

</div>
<div className="mb-8">
  <label className="font-semibold text-lg">
    <span className='text-orange'><CalendarMonthOutlinedIcon
/></span> Дата виїзду
  </label><br />
  <input
    value={pickDate}
    onChange={ev => setPickDate(ev.target.value)}
    max={dropDate}
    type='date'
    className="mt-4 border border-[#ccd7e6] rounded text-[#ababab]
text-xl py-2 w-full focus:outline-none"
  />
</div>
<div className="mb-8">
  <label className="font-semibold text-lg">
    <span className='text-orange'><CalendarMonthOutlinedIcon
/></span> Дата повернення
  </label><br />
  <input
    value={dropDate}
    onChange={ev => setDropDate(ev.target.value)}
    min={pickDate}
    type='date'
    className="mt-4 border border-[#ccd7e6] rounded text-[#ababab]
text-xl py-2 w-full focus:outline-none"
  />
</div>
<div>

```

```

    <button
      onClick={handleSearch}
      className="border border-[#ccd7e6] mt-11 font-semibold font-sans
rounded text-xl bg-orange text-white py-2 w-full"
    >
      Шукати
    </button>
  </div>
</div>
</div>
</div>
</form>
);
}

```

Додаток Б.9 – код з файлу Hero.jsx

```

import React, {useRef} from 'react';
import car from "../images/car.png"
import { Link } from 'react-router-dom';
import ChevronRightIcon from '@mui/icons-material/ChevronRight';
import CheckCircleIcon from '@mui/icons-material/CheckCircle';

export default function Hero({handleBookRideClick}){
  return(
    <>
      <section id="hero">
        <div className="container flex flex-col-reverse md:flex-row items-center px-6
mx-auto mt-10 space-x-0 md:space-y-0">
          <div className='flex flex-col mb-32 space-y-12 md:w-1/2'>
            <div className="lg:px-16">

```

```

    <h2 className="mb-4 max-w-md text-3xl font-bold text-center md:text-left
lg:mt-24">Комфортний рух Вашого бізнесу - наша мета!</h2>
    <h1 className="mb-10 max-w-md text-4xl font-bold text-center md:text-5xl
md:text-left">
        Пасажирські <span className="text-orange">перевезення</span> по Києву та
Україні
    </h1>
    <p className="mb-16 font-sans text-[gray] text-xl text-center md:text-left">
        Замовляйте транспорт з водієм для ваших потреб. Доступні ціни,
комфортний транспорт, гнучкі умови бронювання та багато іншого.
    </p>
    <div className='flex flex-col-reverse md:flex-row items-center gap-4'>
    <Link onClick={handleBookRideClick} className="text-white bg-orange font-
bold px-14 py-5 rounded opacity-95 hover:opacity-100 hover:shadow-lg">
        Забронювати <CheckCircleIcon/>
    </Link>
    <Link to={'/about'} className="text-white border border-black bg-black font-
bold px-12 py-5 rounded hover:bg-white hover:text-black">
        Дізнатися більше <ChevronRightIcon/>
    </Link>
    </div>
    </div>
    </div>
    </div>
    <div className='hidden md:flex flex-col mb-32 space-y-12 md:w-1/2'>
    <img className="" src={car} alt=""/>
    </div>
    </div>
</section>
</>
)

```

```
}

```

Додаток Б.10 – код з файлу linearindertaminate.jsx

```
import * as React from 'react';
import Box from '@mui/material/Box';
import LinearProgress from '@mui/material/LinearProgress';

export default function LinearIndeterminate() {
  return (
    <Box sx={{ width: '100%' }}>
      <LinearProgress />
    </Box>
  );
}
```

Додаток Б.11 – код з файлу linearprogres.jsx

```
import * as React from 'react';
import Box from '@mui/material/Box';
import LinearProgress from '@mui/material/LinearProgress';

export default function LinearIndeterminate() {
  return (
    <Box sx={{ width: '100%' }}>
      <LinearProgress sx={{ backgroundColor: 'orange', color:'orange' }} />
    </Box>
  );
}
```

Додаток Б.12 – код з файлу reactcard.jsx

```
import React, { useContext } from 'react';
import { Link } from 'react-router-dom';
import { UserContext } from '../Context/Clientcontext';
```

```

export default function CarCard({ image, name, rent }) {
  const { user } = useContext(UserContext);

  return (
    <div className="card rounded-none group" style={{ cursor: "pointer" }}>
      <div className="relative">
        <img className='rounded group-hover:opacity-50' src={image}
alt={name} />
        <div className="opacity-0 group-hover:opacity-100 absolute top-1/2
left-1/2 transform -translate-x-1/2 -translate-y-1/2">
          <Link to={user ? '/account' : '/'} className="bg-orange rounded
text-white px-8 py-4 font-bold">Забронювати</Link>
        </div>
      </div>
      <div className="bg-white p-4 text-center">
        <h2 className="text-xl font-bold">{name}</h2>
        <p className="text-orange font-bold text-lg">{rent} / година</p>
      </div>
    </div>
  );
}

```

Додаток Б.13 – код з файлу reservation.jsx

```

import React, { useState } from 'react';
import sprinter318 from "../images/sprinter318.jpg";
import transit from "../images/transit.jpg";
import vito from "../images/vito.jpg";
import expert from "../images/expert.jpg";
import sprinter516 from "../images/sprinter516.jpg";
import tourino from "../images/tourino.jpg";

```

```

export default function Reservationbox({handleBookRideClick}){

  const CarNames = [
    {name: "Mercedes-Benz Sprinter 318", image: sprinter318, data: {model:
    "Sprinter 318", mark: "Mercedes-Benz", year: "2015", doors: "21", ac: "Так",
    transmission: "Так", fuel: "500 грн/год 20 грн/км"}},
    {name: "Ford Transit", image: transit, data: {model: "Transit", mark: "Ford",
    year: "2012", doors: "16", ac: "Так", transmission: "Так", fuel: "400 грн/год 18
    грн/км"}},
    {name: "Mercedes-Benz Vito", image: vito, data: {model: "Vito", mark:
    "Mercedes-Benz", year: "2018", doors: "8", ac: "Так", transmission: "Так", fuel: "500
    грн/год 19 грн/км"}},
    {name: "Peugeot Expert", image: expert, data: {model: "Expert", mark:
    "Peugeot", year: "2013", doors: "8", ac: "Так", transmission: "Так", fuel: "350
    грн/год 15 грн/км"}},
    {name: "Mercedes-Benz Sprinter 516", image: sprinter516, data: {model:
    "Sprinter 516", mark: "Mercedes-Benz", year: "2019", doors: "26", ac: "Так",
    transmission: "Так", fuel: "600 грн/год 21 грн/км"}},
    {name: "Mercedes-Benz Tourino", image: tourino, data: {model: "Tourino",
    mark: "Mercedes-Benz", year: "2014", doors: "35", ac: "Так", transmission: "Так", fuel:
    "650 грн/год 22 грн/км"}},
  ];

  const [selectedCar, setSelectedcar] = useState(CarNames[0].name);
  const [selectedImage, setSelectedimage] = useState(CarNames[0].image);
  const [selectedCardata, setSelectedcardata] = useState(CarNames[0].data);

  const addClass = (carname) => {
    let classes = "bg-[#e9e9e9] font-[Poppins,sans-serif] ml-16 text-center text-
    xl font-semibold py-3 px-4 w-3/5 hover:bg-orange hover:text-white";
  }

```

```

    if(selectedCar === carname){
        classes = "bg-orange text-white font-[Poppins,sans-serif] ml-16 text-
center text-xl font-semibold py-3 px-4 w-3/5 hover:bg-orange hover:text-white"
    }
    return classes;
}

return(
    <>
    <div className="container mx-auto flex justify-center mb-10">
        <div className="text-center w-2/5 ">
            <h1 className="font-bold text-5xl mb-5">Наш автопарк</h1>
            <h2 className="font-bold text-2xl mb-5">Сучасні авто від
провідних світових виробників
                відповідають всім стандартам якості та безпеки</h2>
            <p className="text-base text-[#706f7b] mb-5">Остаточний
розрахунок вартості оренди мікроавтобуса узгоджується з
                диспетчером після узгодження всіх побажань клієнта.</p>
        </div>
    </div>
    <div className="container mx-auto flex justify-center mb-48">
        <div className="grid lg:grid-cols-3 items-center">
            <div className="justify-center">
                <div className="grid grid-cols-1 gap-4 justify-center items-center">
                    {CarNames.map((car, index) => (
                        <div
                            onClick={() => {
                                setSelectedcar(car.name);
                                setSelectedimage(car.image);
                                setSelectedcardata(car.data);
                            }}

```

```

    }}
    key={index}
    className={addClass(car.name)}
    style={{ cursor:"pointer" }}
  >
    {car.name}
  </div>
)}}
</div>
</div>
<div className="mt-4">
  <img src={selectedImage} alt="Фото транспорту" />
</div>
<div className="">
  <div className="grid grid-rows-1 border w-3/5 ml-20">
    <div className="bg-orange py-4 px-2 text-center text-white font-
sans font-bold text-2xl">
      Детальна інформація
    </div>
    <table>
      <thead>
        <tr>
          <td className="py-2 text-center font-bold">Модель</td>
          <td className="py-2 text-center font-
bold">{selectedCardata.model}</td>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td className="py-2 text-center font-bold">Марка</td>

```

```

        <td className="py-2 text-center font-
bold">{selectedCardata.mark}</td>
    </tr>
    <tr>
        <td className="py-2 text-center font-bold">Рік</td>
        <td className="py-2 text-center font-
bold">{selectedCardata.year}</td>
    </tr>
    <tr>
        <td className="py-2 text-center font-bold">Міся</td>
        <td className="py-2 text-center font-
bold">{selectedCardata.doors}</td>
    </tr>
    <tr>
        <td className="py-2 text-center font-
bold">Кондиціонер</td>
        <td className="py-2 text-center font-
bold">{selectedCardata.ac}</td>
    </tr>
    <tr>
        <td className="py-2 text-center font-bold">Ремені
безпеки</td>
        <td className="py-2 text-center font-
bold">{selectedCardata.transmission}</td>
    </tr>
    <tr>
        <td className="py-2 text-center font-bold">Ціна</td>
        <td className="py-2 text-center font-
bold">{selectedCardata.fuel}</td>
    </tr>

```

```

        </tbody>
    </table>
</div>
    <div onClick={handleBookRideClick} style={{cursor:"pointer"}}
className="hover:shadow-2xl opacity-95 hover:opacity-100 py-2 px-2 bg-orange text-
white font-sans font-bold text-center text-2xl w-3/5 ml-20">
        Забронювати
    </div>
</div>
</div>
</div>
</div>
</>
);
}

```

Додаток Б.14 – код з файлу reviews.jsx

```

import React from 'react';
import potter from "../images/men.jpg";
import weisley from "../images/women.jpg";

export default function Reviews() {
    return (
        <>
            <div className="bg-[#f8f8f8] w-full z-10 mt-4">
                <div className="container mx-auto p-6 text-center my-4">
                    <h2 className='font-sans text-3xl font-semibold mb-2'>Відгуки клієнтів</h2>
                    <h1 className="font-sans text-5xl font-bold mb-4">Що кажуть наші
клієнти</h1>
                    <div className='flex justify-center mb-4'>
                        <p className="font-sans text-base font-semibold text-[#706f7b] w-2/4">

```

Кількість відгуків клієнтів, що користуються нашими послугами значно більша.

```
</p>
```

```
</div>
```

```
</div>
```

```
<div className="container mx-auto justify-between w-3/4 pb-16">
```

```
<div className="grid lg:grid-cols-2 gap-8">
```

```
<div className="bg-white font-sans p-16 text-center z-30 shadow-xl">
```

```
<h2 className="text-2xl font-semibold mb-10">"Дякуємо Перевізнику, особливо керівнику Роману, за вирішення питання щодо розвезення персоналу для нашого підприємства в Києві.
```

```
Оперативно вирішили позаштатну ситуацію. З вами приємно працювати! Успіхів!"</h2>
```

```
<div className="flex items-center">
```

```
<img src={potter} alt="Фото користувача" className="w-20 h-20 rounded-full" />
```

```
<span className="ml-4 font-bold text-xl">Михайло Мельник</span>
```

```
<span className="ms-auto text-6xl font-mono text-orange">"</span>
```

```
</div>
```

```
</div>
```

```
<div className="bg-white font-sans p-16 text-center z-30 shadow-xl">
```

```
<h2 className="text-2xl font-semibold mb-10">"Добрий день! Замовляємо на весілля автобуси та міневени. Дуже радують водії: як належить, у сорочці та брюках. Перед гостями клієнтів не соромно. Комфорт, ціна – все супер!"</h2>
```

```
<div className="flex items-center">
```

```
<img src={weisley} alt="Фото користувача" className="w-20 h-20 rounded-full" />
```

```
<span className="ml-4 font-bold text-xl">Анна Ярован</span>
```

```
<span className="ms-auto text-6xl font-mono text-orange">"</span>
```

```
</div>
```

```

</div>
</div>
</div>
</div>
</>
);
}

```

Додаток Б.15 – код з файлу services.jsx

```

import React from 'react';
import { Link } from 'react-router-dom';
import country from "../images/country.png";
import dollar from "../images/dollar.png";
import drive from "../images/drive.png";
import ChevronRightIcon from '@mui/icons-material/ChevronRight';

export default function Services() {
  return (
    <>
    <div className='container mx-auto p-6 mt-32'>
    <div className="grid lg:grid-cols-2 justify-start gap-7 px-16">
    <div className="">
    <h2 className="font-sans font-bold text-2xl mb-2">Чому обирають нас</h2>
    <h1 className="font-sans font-bold text-5xl mb-6">Будьте впевнені,
    нам можна довіряти.</h1>
    <p className="text-base font-semibold text-[#706f7b] font-sans mb-10">
    Безпека, доступність та комфорт - це ті складові, якими ми користуємося
    щодня у нашій роботі.
    Також нашою основною перевагою є ще той факт, що ми працюємо без
    посередників, тобто можемо запропонувати нашим клієнтам найвигідніші ціни.
    </p>

```

```
<Link to={'/contact'} className="px-8 py-4 font-sans mb-10 font-bold rounded
bg-orange text-white opacity-95 hover:opacity-100 hover:shadow-lg">
```

```
Детальніше <ChevronRightIcon />
```

```
</Link>
```

```
</div>
```

```
<div className="grid grid-rows-3 gap-4">
```

```
<div className="grid lg:grid-cols-3">
```

```
<div className="flex justify-center">
```

```
<img src={country} alt="" />
```

```
</div>
```

```
<div className="col-span-2 mt-4">
```

```
<h3 className="font-bold font-sans text-2xl">Наш досвід</h3>
```

```
<p className="text-[#706f7b] font-sans font-semibold">
```

Ми вже понад 10 років на ринку. Нашими послугами скористалися понад 5000 пасажирів та сотні різних замовників.

```
</p>
```

```
</div>
```

```
</div>
```

```
<div className="grid lg:grid-cols-3">
```

```
<div className="flex justify-center">
```

```
<img src={dollar} alt="" />
```

```
</div>
```

```
<div className="col-span-2 mt-4">
```

<h3 className="font-bold font-sans text-2xl">Працюємо без посередників</h3>

```
<p className="text-[#706f7b] font-sans font-semibold">
```

Гнучка цінова політика, програма лояльності для постійних клієнтів.

Високий рівень сервісу, надійність та професіоналізм.

```
</p>
```

```
</div>
```

