

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки

«Затверджую»

Зав. кафедри теоретичної та
прикладної системотехніки

_____ д.т.н., проф. С. І. Шматков

«__» _____ 2024 р

Пояснювальна записка

до кваліфікаційної роботи

бакалавра

на тему: *«Модель комп'ютеризованої Web-системи пошуку транспортних засобів для перевезення вантажів»*

Захищено на засіданні
Атестаційної комісії № 42
протокол № __ від __.06.2024 р.
Оцінка _____ / _____

Голова Атестаційної комісії

_____ Скоб Ю. О.
(підпис) (прізвище та ініціали)

Виконав:
студент 4 курсу, групи КІ-41
Галузь знань: 12 – Інформаційні
технології
Спеціальність: 123 – Комп'ютерна
Інженерія

Добродій Вячеслав Юрійович _____
(прізвище, ім'я та по батькові) (підпис)

Керівник: канд. техн. наук, доцент
кафедри теоретичної та прикладної
системотехніки

Лабенко Дмитро Петрович _____
(підпис)

Рецензент:
професор кафедри ОВ ППО СВ ХНУПС
імені Івана Кожедуба к.т.н, доцент

Наконечний Олександр
Анатолійович _____
(підпис)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, 4 розділів, висновків, списку використаних джерел і 4 додатків. Загальний обсяг роботи складає 72 сторінок, із яких 50 сторінок основної частини з 68 рисунками, 6 таблиць, 6 найменуваннями списку використаних джерел та 4 додатками.

Метою кваліфікаційної роботи є спрощення процесу пошуку транспортних засобів для перевезення вантажів з використанням сучасних web-технологій на ринку автоперевезень.

Об'єкт дослідження – процес пошуку транспортних засобів для перевезення вантажів.

Предмет дослідження – методи та технології розробки веб-систем для пошуку послуг перевезення вантажу.

Проблема, що досліджується в даній кваліфікаційній роботі, полягає в розробці та впровадженні ефективної моделі комп'ютеризованої веб-системи пошуку транспортних засобів для перевезення вантажів.

Область застосування – розроблений програмний продукт може широко застосовуватись у сфері транспортного бізнесу, включаючи автоперевезення. Веб-система може бути корисною як для індивідуальних перевізників, так і для груп перевізників або транспортних компаній. Завдяки своїй гнучкості, система може бути адаптована для використання різними учасниками ринку вантажоперевезень, включаючи малий та середній бізнес, а також великі логістичні компанії.

Ключові слова: API, SQLite, Python, Django, комп'ютеризована веб-система, пошук транспортних засобів, перевезення вантажів.

ABSTRACT

The explanatory note to the bachelor's thesis consists of an introduction, 4 chapters, conclusions, a list of used sources and 4 appendices. The total volume of the work is 72 pages, of which 50 pages are the main part with 68 figures, 6 tables, 6 titles of the list of used sources and 4 appendices.

The purpose of the qualification work is to simplify the process of finding vehicles for cargo transportation using modern web technologies in the road transportation market.

The object of research is the process of finding vehicles for cargo transportation.

The subject of the research is the methods and technologies of developing web systems for searching cargo transportation services.

The problem investigated in this qualification work is the development and implementation of an effective model of a computerized web-based vehicle search system for cargo transportation.

Field of application - the developed software product can be widely used in the field of transport business, including road transport. The web system can be useful both for individual carriers and for groups of carriers or transport companies. Thanks to its flexibility, the system can be adapted for use by various participants in the freight transportation market, including small and medium-sized businesses, as well as large logistics companies.

Keywords: API, SQLite, Python, Django, computerized web system, vehicle search, freight transportation.

ЗМІСТ

ВСТУП.....	5
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
РОЗДІЛ 1. АНАЛІЗ ІСНУЮЧИХ WEB-СИСТЕМ.....	7
1.1 Огляд сучасних Web-систем.....	7
1.2 Аналіз переваг та недоліків сучасних Web-систем.....	8
1.3 Аналіз та вивчення вимог користувачів	12
Висновки за розділом 1	14
РОЗДІЛ 2. ОПИС МОДЕЛІ КОМП'ЮТЕРИЗОВАНОЇ WEB-СИСТЕМ	15
2.1 Визначення функціональних вимог.....	15
2.2 Проектування архітектури системи.....	21
2.3 Вибір технологій та інструментів розробки	22
Висновки за розділом 2.....	24
РОЗДІЛ 3. РЕАЛІЗАЦІЯ WEB-СИСТЕМИ.....	25
3.1 Розробка інтерфейсу користувача.....	25
3.2 Реалізація основних функцій.....	28
3.3 Тестування та відлагодження.....	33
Висновки за розділом 3.....	43
РОЗДІЛ 4. ОЦІНКА ЕФЕКТИВНОСТІ ТА ПОРІВНЯННЯ З ІСНУЮЧИМИ СИСТЕМАМИ.....	44
4.1 Проведення тестування залежно від різних критеріїв.....	44
4.2 Аналіз результатів.....	45
4.3 Порівняння отриманих даних з існуючими системами.....	46
Висновки за розділом 4.....	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТКИ.....	55

ВСТУП

У сучасному світі розвиток логістики та транспорту має велике значення для економічного зростання та глобалізації. Впровадження комп'ютеризованих систем для пошуку транспортних засобів, які можуть перевезти вантажі, стає все більш актуальним. Веб-системи дозволяють автоматизувати процес пошуку, бронювання та моніторингу транспорту.

Актуальність розробки: В умовах зростаючої конкуренції на ринку логістики, всі учасники логістичних процесів, включаючи компанії, індивідуальних перевізників та клієнтів, шукають шляхи оптимізації своїх процесів управління перевезеннями, планування маршрутів, обробки замовлень та взаємодії з клієнтами. Веб-системи для пошуку транспортних засобів дозволяють швидко знайти доступні варіанти перевезень. Це сприяє зменшенню часу простою транспорту, ефективнішому використанню ресурсів та підвищенню задоволеності клієнтів. Впровадження таких систем допомагає мінімізувати витрати, підвищити оперативність та конкурентоспроможність на ринку логістичних послуг.

Мета розробки: Спрощення процесу пошуку транспортних засобів для перевезення вантажів з використанням сучасних web-технологій на ринку автоперевезень. Завдання розробки:

1. Аналіз існуючих систем пошуку транспортних засобів для перевезення вантажів.
2. Аналіз та вивчення вимог користувачів.
3. Опис моделі комп'ютеризованої Web-системи.
4. Реалізація Web-системи.
5. Оцінка ефективності та порівняння з існуючими системами.

Методи дослідження: включає вивчення принципів роботи комп'ютеризованих Web-систем для пошуку транспортних засобів для перевезення вантажів

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

- Фреймворк (Framework) – це набір інструментів, бібліотек та правил, який використовується для створення програмних додатків. Він зазвичай являє собою структуру, яка визначає, як компоненти програми повинні взаємодіяти між собою, які шаблони використовувати для створення інтерфейсів і які методи використовувати для роботи з базами даних та іншими зовнішніми ресурсами.
- Python – об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією.
- API (Application Programming Interface) – це набір правил та протоколів, які визначають спосіб взаємодії між програмами або компонентами програмного забезпечення. Він встановлює набір функцій, методів, структур даних та конвенцій, які використовуються для комунікації між різними програмами або сервісами.

РОЗДІЛ 1

АНАЛІЗ ІСНУЮЧИХ WEB-СИСТЕМ

В останні десятиліття розвиток інформаційних технологій значно вплинув на всі сфери економіки, включаючи логістику та вантажоперевезення. Сучасні веб-системи для пошуку послуг вантажоперевезень дозволяють компаніям і приватним користувачам швидко знаходити необхідний транспорт для перевезення вантажів, оптимізуючи логістичні процеси та знижуючи витрати. Цей розділ присвячений аналізу існуючих веб-систем, їх переваг та недоліків, а також виявленню проблем, які потребують вирішення.

1.1 Огляд сучасних Web-систем.

На ринку існує велика кількість веб-систем для пошуку транспортних засобів для перевезення вантажів. До найбільш популярних належать:

- Freightos [4]: Глобальна система для пошуку та бронювання транспортних послуг, що охоплює всі види транспорту. Основні функції:
 - Пошук та бронювання транспортних послуг.
 - Порівняння цін і термінів доставки.
 - Розрахунок вартості перевезення.
- Uber Freight [6]: система для пошуку вантажівок, що використовує аналогічну модель, як і сервіс Uber для пасажирських перевезень.
- Основні функції:
 - Пошук вантажівок.
 - Швидке бронювання транспортних послуг.
 - Відстеження вантажів.
- Freightera [3]: Web-система для пошуку та бронювання вантажних перевезень з акцентом на екологічні рішення. Основні функції:
 - Пошук та бронювання вантажних перевезень.
 - Порівняння тарифів.
- Freightquote [5]: Система для онлайн-бронювання вантажних перевезень. Основні функції:

- Пошук та бронювання вантажних перевезень.
- Порівняння цін.
- Розрахунок вартості перевезення.

1.2 Аналіз переваг та недоліків сучасних Web-систем.

При огляді цих Web-систем для пошуку транспортних засобів для перевезення вантажів мають ряд переваг та недоліків:

1 Freightos [4]:

Таблиця 1.1

Переваги та недоліки системи

Переваги:	Недоліки:
Широкий вибір послуг	Вартість послуг
Зручність та швидкість бронювання	Легковажність служби підтримки
Автоматизація та інтеграція з іншими системами	Якість послуг
Відстеження вантажів у режимі реального часу	
Підтримка клієнтів через різні канали	



21 вересня 2021 р

Не користуйтеся цією послугою

Не користуйтеся цією послугою. Вони візьмуть ваші гроші та передадуть вас нічному експедитору, який не відправить ваш матеріал і попросить більше грошей. У кращому випадку ви отримаєте пізню доставку за більшу суму, ніж рекламується. Крім того, вони затримують будь-які кредити на місяці, використовуючи відмовки, щоб не повернути ваші гроші. Найгірший досвід!!!!

Рис 1.1 – Коментар користувача щодо оплати послуг



Оновлено 31 грудня 2021 р

Це найгірша компанія, яку ви можете...

Це найгірша компанія, з якою ви можете зв'язатися щодо своїх потреб у доставці. Підтримка, яку вони надають, це жарт. Я б із задоволенням поставив їм мінус 5 зірок, якби міг. Китайський вантажовідправник намагається стягнути з нас завищену плату. Вантажовідправник намагався стягнути з нас понад 700 доларів США за 4 порожні піддони, на яких був розміщений вантаж! Я зв'язався зі службою підтримки frightos і надав усі чіткі докази, але вони навіть не перевірили та не прочитали подробиці. Вони просто копіюють та вставляють відповіді та намагаються стягнути з нас плату без сторонньої допомоги! . Ця компанія є шахраєм. Я зв'язуся з BVB, офісом підтримки Alibaba у США, будь-якими соціальними мережами, які я зможу, щоб переконатися, що у вас не виникне таких проблем. Уникайте їх будь-якою ціною. він не був закритий вони намагаються стягнути плату за мою візу без нашого дозволу, що є шахрайством!

Рис 1.2 – Коментар користувача щодо якості служби підтримки



Оновлено 2 дні тому

Колись я любив Freightos

Раніше я високо поважав Freightos, але їхня остання робота була просто катастрофою. Їхня обробка вантажу, яка мала коштувати 1800 доларів США, включаючи митні збори, зрештою обійшлася мені в приголомшливі 5700 доларів США, і, на додачу до образи, увесь мій вантаж було знищено.

Проблему з митницею США можна було легко вирішити за допомогою простого доручення, яке я, як офіційний імпортер, не міг надати. Митниця США пояснила, що надати довіреність на виконання покладається на митного брокера Clearit. Незважаючи на мої зусилля повідомити про це Freightos і Clearit, вони не взяли жодної відповідальності за ситуацію.

Лі Картер, представник Clearit, компанії, що належить Freightos, продемонстрував повну відсутність співчуття та відповідальності, просто заявивши: «Митниця США знищила ваш вантаж. Мені шкода, що ви розчаровані». Ця відповідь є неприйнятною, особливо коли я втрачаю понад 15 000 доларів через їхню недбалість.

Прикро бачити, як Freightos перекладає всю відповідальність на Clearit, якого вони обрали своїм єдиним митним брокером за замовчуванням. Clearit неправдиво рекламує 5-зірковий рейтинг на Trustpilot, хоча насправді вони мають лише 3,8-зірковий рейтинг.

Рис 1.3 – Коментар користувача щодо якості послуг



✓ Перевірено

4 квітня 2022 р

Freightos об'єднав усе, що ми...

Freightos об'єднав усе, що нам було потрібно для безперебійного імпорту в США з Китаю. Ми змогли відстежити кожен аспект нашого вантажу, включаючи документи, митний процес, отримання, LCL океанський транзит, доставку тощо. Ми обов'язково будемо використовувати платформу знову!

Рис 1.4 – Коментар користувача щодо вибору послуг



29 лютого 2024 р

Великий досвід

Freightos чудовий. Вони спрощують бронювання відправлень, і більшість експедиторів дійсно хороші. Зазвичай посилки приходять вчасно або близько до нього.

Рис 1.5 – Коментар користувача щодо бронювання

2 Uber Freight [6]:

Таблиця 1.2

Переваги та недоліки системи

Переваги:	Недоліки:
Низька вартість	Скасування замовленого авто
Простота використання	Послуги підтримки
Безпека вантажу	
Швидкість пошуку та бронювання вантажів	

Плюси: найбезпечніший процес перевезення вантажів, який забезпечує доставку вчасно.

Мінуси: ми все ще не знайшли найменшого варіанту щодо цього.

Рис 1.6 – Коментар користувача щодо безпеки

Плюси: низька вартість порівняно з альтернативами та проста у використанні платформа для бронювання.

Мінуси: скасовані завантаження без попередження. Мало інформації чи допомоги від персоналу, щоб повторно забронювати скасовані вантажі - вони майже просто кажуть, що повідомлять вам, коли знайдуть водія, а потім радіомовчання. ЖАХЛИВИЙ процес розгляду претензій, коли є проблема. Буде на боці перевізника, навіть якщо у вас є докази, а вони мають лише слово водія, який створив проблему. Можна припустити, що ви не отримаєте ні копійки.

Рис 1.7 – Коментар користувача щодо вартості та скасування послуг

Загальний рейтинг	★★★★★ 5.0	Переваги: Мені подобається, тому що він швидкий, безпечний і має дуже доступну ціну.
Простота використання	★★★★★ 5.0	
Обслуговування клієнтів	★★★★☆ 4.0	Мінуси: кілька разів траплялося, що послуги скасовували, і це погано для тих, хто чекає, або особливо для тих, хто поспішає.

Рис 1.8 – Коментар користувача щодо швидкості та скасування послуг

Обслуговування клієнтів	★☆☆☆☆ 1.0	Плюси: Нічого, дуже непрофесійно, персонал служби підтримки має погану англійську мову, і вони не можуть вирішити проблеми, які створили самі
особливості	★☆☆☆☆ 1.0	
Співвідношення ціни та якості	★☆☆☆☆ 1.0	Мінуси: Вміння спілкуватися. Нерозуміння інструкцій із завантаження

Рис 1.9 – Коментар користувача щодо якості підтримки

3 Freightera [3]:

Таблиця 1.3

Переваги та недоліки системи

Переваги:	Недоліки:
Широкий вибір перевізників та логістичних компаній	Можливість втрати контролю над вантажем під час перевезення
Онлайн-бронювання та керування вантажами через веб-платформу	Повільне обслуговування
Якісна служба підтримки	Затримка доставки

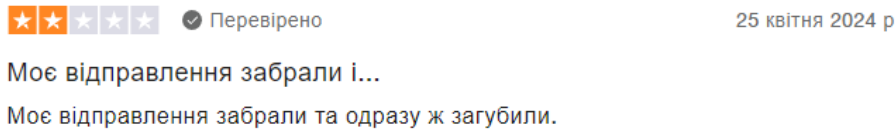


Рис 1.10 – Коментар користувача щодо втрати вантажу

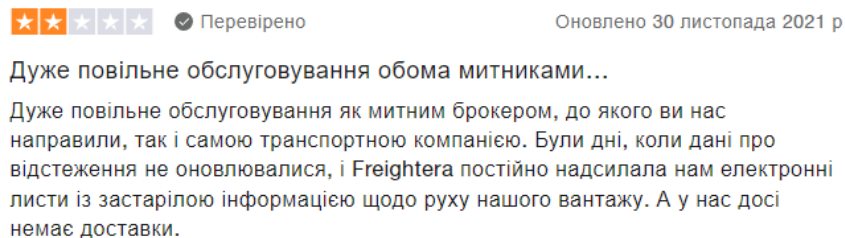


Рис 1.11 – Коментар користувача щодо повільного обслуговування

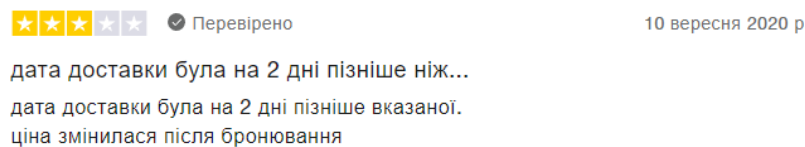


Рис 1.12 – Коментар користувача щодо затримок доставки

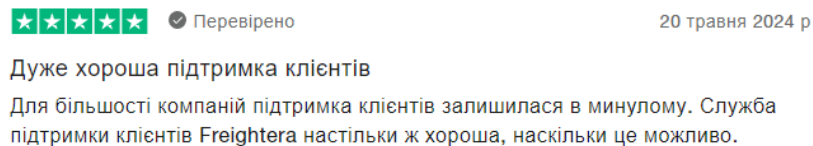


Рис 1.13 – Коментар користувача щодо підтримки

4 FreightQuote [5]:

Таблиця 1.4

Переваги та недоліки системи

Переваги:	Недоліки:
Широкий вибір послуг	Розбіжність заявленої ціни від фактичної
Простота використання	Затримка доставки
Якісна служба підтримки	Можливість втрати контролю над вантажем під час перевезення



Оновлено 2 січня 2024 р

Жахливе обслуговування LTL

Жахливе обслуговування LTL. Незважаючи на моє розважливе судження, я думаю, що дам їм додаткові шанси після втрати вантажу. Перш за все, як ви втрачаєте вантаж? Це не те, що це щось маленьке, що можна легко доставити не за адресою. Я користувався цією компанією понад 15 років без будь-яких проблем, але лише останні 4 місяці у мене виникли лише кошмарні проблеми від надзвичайно високих витрат до втрати вантажу, пізнього самовивозу та пізньої доставки, не просто день чи два, а 7 - Запізнився на 10 днів! Крім того, вони починають надсилати мені додаткові комісії, як-от плата за доставку в помешкання та плату за підйомні двері, за товари, які відправляються компаніям у бізнес-зонах за допомогою вилкових навантажувачів, без підйомних воріт, підприємствам, яким я відправляв раніше. Ми створюємо обладнання на замовлення та встановлюємо терміни, щоб доставити його нашим клієнтам, і коли речі взагалі не з'являються, і нам доводиться робити замовлення повторно, це коштує набагато більше, ніж просто втрачена вартість продукту. Тепер це створює навантаження на наш уже витриманий графік виробництва та відстає від наших клієнтів. Я знайшов більш компетентну компанію для доставки моїх продуктів і більше не буду користуватися цією компанією.

-Оскільки компанія отримала відповідь: я все ще не маю \$\$\$, і вони все ще намагаються стягнути з мене плату за непотрібні послуги.

Рис 1.14 – Коментар користувача щодо втрати замовлення, додаткових витрат та затримки



Запрошений

22 березня 2024 р

Зрозумілий і простий у використанні

Freightquote зробила процес пропозиції, створення та планування наших вантажних перевезень простим і легким. Завдяки покроковим інструкціям і численним тарифам для перегляду та вибору доставки нашого вантажу в дорогу є швидкою, зручною та економічною.

Рис 1.15 – Коментар користувача щодо послуг та їх простоти



Запрошений

21 березня 2024 р

я вперше скористався послугою і...

я вперше скористався цією послугою, і служба підтримки провів мене через кожен крок, була дуже терплячою та корисною. дуже ціную це.

Рис 1.16 – Коментар користувача щодо якості підтримки

1.3 Аналіз та вивчення вимог користувачів.

Для успішної розробки моделі комп'ютеризованої веб-системи пошуку транспортних засобів для перевезення вантажів важливо ретельно проаналізувати і вивчити потреби та вимоги користувачів. Це дозволить

створити продукт, який максимально задовольнить їх очікування та забезпечить високу конкурентоспроможність на ринку.

Основні категорії користувачів системи:

- Клієнти
- Перевізники (Водії)

Вимоги ватажовідправників:

1. Простота використання:

- Інтуїтивно зрозумілий інтерфейс.
- Легкий процес реєстрації та створення замовлення.

2. Надійність та безпека:

- Гарантії безпеки вантажу.
- Гарантії безпеки персональних даних.

3. Оптимізація витрат:

- Автоматичні розрахунки вартості перевезень.
- Знижки для постійних клієнтів

4. Персоналізація:

- Налаштування інтерфейсу відповідно до потреб користувача.
- Збереження історії замовлень та автоматичне заповнення форм на основі попередніх даних.

5. Функціонал:

- Можливість створювати замовлення на перевезення вантажу.
- Можливість самостійного пошуку транспортного засобу для перевезення.

Вимоги перевізників:

1. Зваження та планування:

- Інструменти для оптимального планування маршрутів.
- Можливість знаходження зворотних вантажів для мінімізації порожніх пробігів.

2. Прозорість і довіра:

- Компенсацій у разі проблем.
- Верифікація клієнтів та замовлень.
- Історія взаємодії з кожним клієнтом.

3. Фінансова стабільність:

- Чіткі умови оплати за послуги.
- Можливість отримання передоплати або гарантії оплати.

4. Підтримка:

- Наявність служби підтримки для вирішення технічних проблем.

6. Функціонал:

- Можливість додавати транспортні засоби.
- Можливість робити публікації доданих транспортних засобів.
- Можливість самостійного пошуку привабливого замовлення

Висновки за розділом 1

В процесі аналізу існуючих веб-систем для пошуку транспортних засобів для перевезення вантажів було виявлено, що на ринку представлено широкий спектр систем, які задовольняють різноманітні потреби як клієнтів, так і перевізників. Проведене дослідження дозволило виділити основні особливості та функціональні можливості цих платформ, переваги та недоліки, які треба врахувати для створення власної системи. Крім того, аналіз ринкових тенденцій та конкурентного середовища показав, що ринок швидко розвивається, для підвищення ефективності та безпеки. Важливим аспектом є зручність користування платформами, включаючи інтуїтивно зрозумілий інтерфейс, інтеграцію з іншими бізнес-системами та високий рівень підтримки клієнтів.

РОЗДІЛ 2

ОПИС МОДЕЛІ КОМП'ЮТЕРИЗОВАНОЇ WEB-СИСТЕМ

2.1 Визначення функціональних вимог

У рамках розробки комп'ютеризованої Web-системи для пошуку транспортних засобів для перевезення вантажів необхідно визначити ряд функціональних вимог, які забезпечать необхідний функціонал.

1. Реєстрація користувача

Система повинна забезпечувати зручний та інтуїтивно зрозумілий процес реєстрації для нових користувачів, що включає як клієнтів, так і перевізників. При реєстрації користувачі повинні мати можливість створювати свої облікові записи, вводячи основні дані, такі як ім'я, прізвище, електронна пошта та пароль. Щоб підтвердити особистість водіям треба відправляти до технічної підтримки фотографії паспорту, ліцензії та написати про свій досвід.

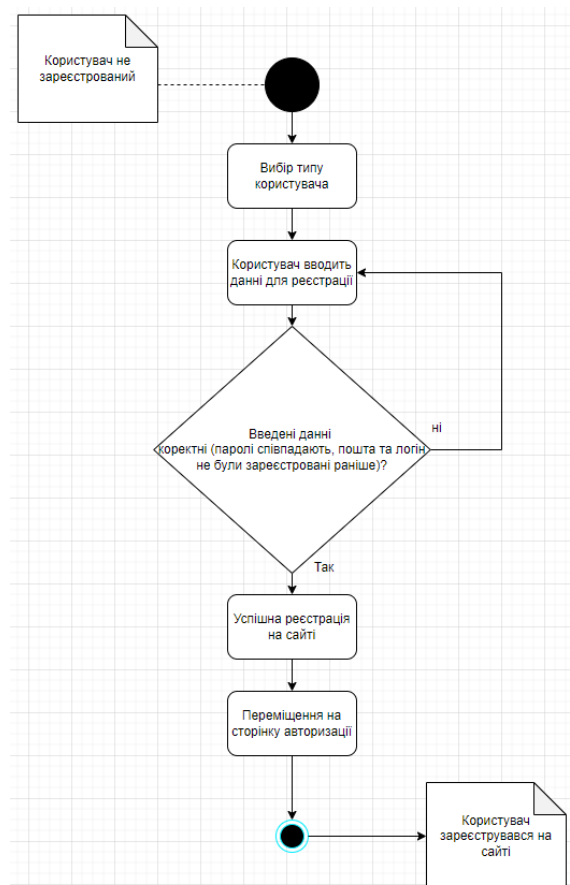


Рис 2.1 – UML діаграма реєстрації

2. Авторизація користувача

Для входу в систему користувачі повинні мати можливість використовувати свої облікові дані, створені під час реєстрації, а саме електронну пошту або логін та пароль. При введенні цих даних система повинна перевіряти їх на відповідність інформації, що зберігається в базі даних. Якщо дані введені правильно, користувач повинен бути перенаправлений на головну сторінку системи.

На сторінці входу має бути передбачена функція "Забули пароль?", яка дозволить користувачам отримати інструкції з відновлення доступу через електронну пошту.

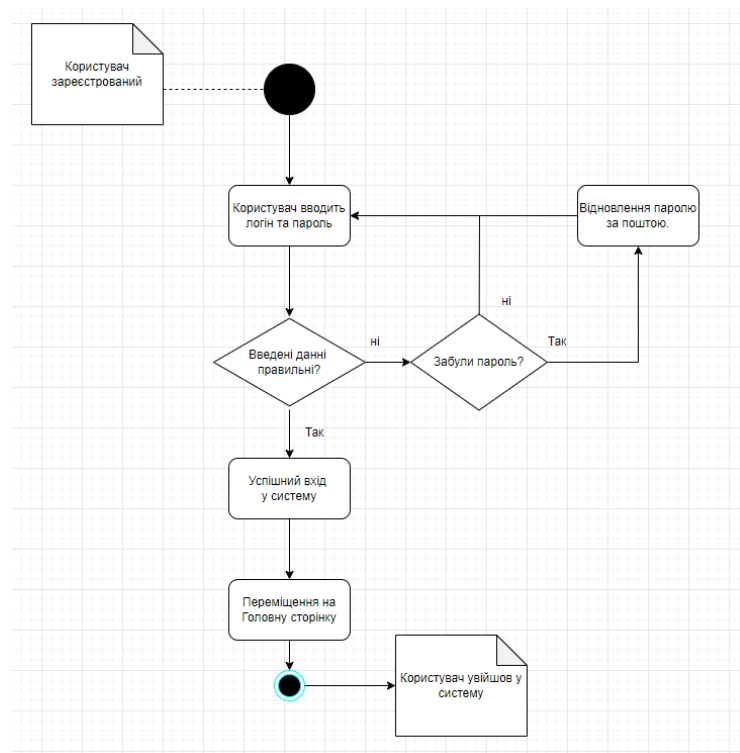


Рис 2.2 – UML діаграма авторизації

3. Створення замовлення та пошук транспорту

Процес створення замовлення включає такі кроки:

1. Додавання основної інформації про вантаж:

Користувачі повинні мати можливість вводити детальну інформацію про вантаж, який потрібно перевезти. Це може включати назву

вантаж, його вагу, об'єм, кількість одиниць, габарити, тип упаковки та інші характеристики.

2. Визначення місць відправки та доставки:

Користувачі повинні вказати міста відправки та доставки вантажу. Система може включати інтеграцію з картографічними сервісами для автоматичного заповнення адреси та підказок під час введення.

3. Розрахунок вартості та оформлення замовлення:

Система повинна автоматично розраховувати орієнтовну вартість перевезення на основі відстані між містами. Після цього користувач повинен мати можливість підтвердити та оформити замовлення.

Процес пошуку транспорту включає такі кроки:

1. Інтерфейс для перегляду доступних оголошень:

Система має включати в себе показ усіх оголошень зроблених водіями.

2. Фільтри та сортування:

Система повинна включати фільтрацію замовлень за основними критеріями таку як місто відправки та доставки, мінімальна та максимальна ціна за перевезення.

3. Детальний перегляд та прийняття оголошень:

Кожне оголошення повинно мати окрему сторінку з детальною інформацією. Користувач повинен мати можливість обрати транспорт який йому підходить та отримати данні водія для подальшого обговорення перевезення.

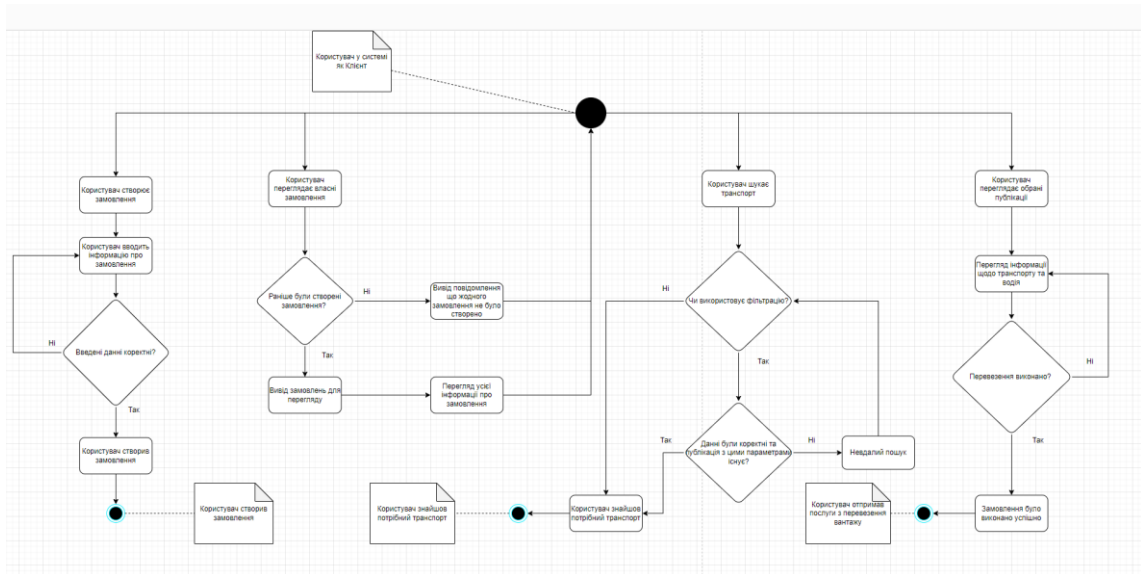


Рис 2.3 - UML діаграма випадків використання функціоналу клієнта

4. Пошук замовлень перевезення вантажу та створення оголошень про транспортні засоби.

Система повинна надати перевізникам можливість ефективно знаходити та вибирати замовлення на перевезення вантажів, що відповідають їхнім критеріям та можливостям. Також створювати оголошення про доступність вільного транспорту та відкритість для нових замовлень.

Функціональність пошуку замовлень включає такі аспекти:

1) Інтерфейс для перегляду доступних замовлень:

Система повинна мати зручний та інтуїтивно зрозумілий інтерфейс для відображення списку доступних замовлень. На головній сторінці для перевізників може бути відображений список активних замовлень із короткою інформацією про кожне з них, такою як місце відправки та доставки, опис вантажу а також орієнтовна вартість перевезення.

2) Фільтри та сортування:

Для зручності пошуку система повинна надати можливість застосовувати різноманітні фільтри та сортування замовлень. Перевізники повинні мати можливість фільтрувати замовлення за такими критеріями, як місце відправки, місце доставки та оплата за замовлення.

3) Детальний перегляд замовлення та прийняття замовлення:

Кожне замовлення повинно мати окрему сторінку з детальною інформацією. Перевізник повинен мати можливість переглянути всі деталі замовлення, включаючи повний опис вантажу, міста відправки та доставки. Також після прийняття замовлення з'являться контактні дані клієнта для подальшого з ним зв'язку.

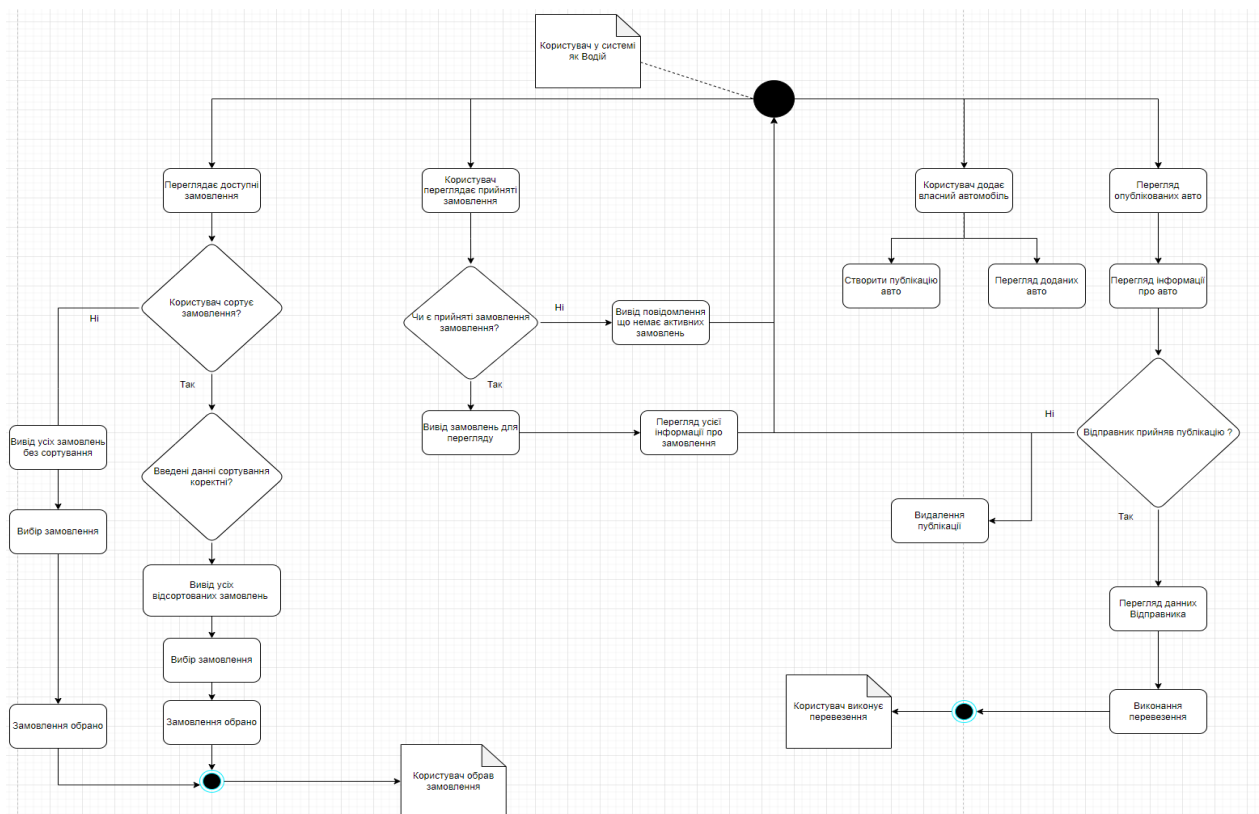


Рис 2.4 - UML діаграма випадків використання функціоналу водія

5. Технічна підтримка

Система повинна забезпечувати наявність технічної підтримки для користувачів. Це включає надання допомоги з технічних питань, вирішення проблем, що виникають при користуванні системою.

1. Цілодобова підтримка:

Користувачі повинні мати доступ до технічної підтримки 24/7. Це важливо, оскільки вантажні перевезення можуть відбуватися у будь-який час доби.

Опис зв'язків таблиць бази даних.

Модель Order зберігає інформацію про кожне замовлення. Кожне замовлення пов'язане з користувачем через поля customer та driver, які вказують на клієнта, що зробив замовлення, і водія, який призначений для його виконання.

Модель User відображає користувачів системи, які можуть бути клієнтами або водіями. Вона має різні поля, такі як фотографія, номер телефону та дата народження, які допомагають ідентифікувати користувачів.

Модель Cars відображає дані про автомобілі, які використовуються для доставки. Кожен автомобіль пов'язаний з власником через поле driver_pk, яке вказує на користувача, який є власником цього автомобіля.

Модель PostedCars відстежує інформацію про автомобілі, які були опубліковані. Вона має зв'язок з моделями Cars та User, щоб вказати водія автомобіля та клієнта.

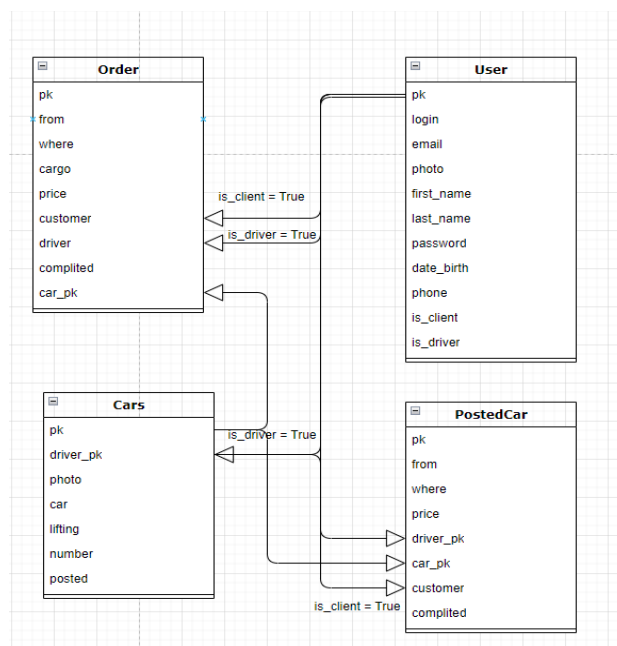


Рис 2.5 - UML діаграма бази даних

Отже, загальна модель Web-системи, враховуючи всі попередні етапи, має наступний вигляд.

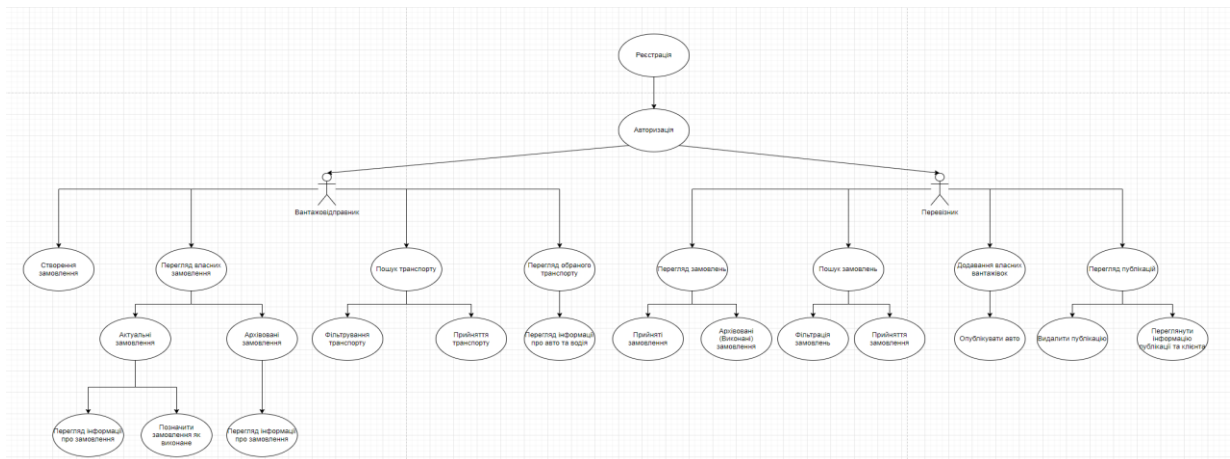


Рис 2.6 - UML діаграма випадків використання загального функціоналу системи

2.2 Проектування архітектури системи

Проектування архітектури комп'ютеризованої Web-системи для пошуку транспортних засобів для перевезення вантажів є ключовим етапом у створенні надійної, масштабованої та ефективної платформи. Архітектура системи визначає її основні компоненти, взаємодію між ними, а також технології, які будуть використані для реалізації системи.

Основні аспекти проектування архітектури включають:

1. **Взаємодія компонентів:** Компоненти системи повинні взаємодіяти між собою для обробки запитів користувачів та забезпечення необхідного функціоналу. Це може бути реалізовано через API або через внутрішній логічний шар системи.

Наприклад:

- Взаємодія між модулем автентифікації та модулем управління замовленнями:

Коли користувач намагається створити нове замовлення, система спочатку перевіряє, чи він аутентифікований. Для цього модуль управління замовленнями звертається до модуля автентифікації, який перевіряє, чи вірно введені дані входу. Якщо користувач вже автентифікований, модуль управління замовленнями продовжує обробку створення нового замовлення.

- Взаємодія між модулем управління замовленнями та модулем пошуку вантажів:

Коли вантажовідправник створює нове замовлення, він вводить параметри вантажу та потрібні місця відправлення та доставки. Модуль управління замовленнями передає ці дані модулю пошуку вантажів, який виконує пошук доступних перевізників та відповідних вантажів для запиту.

2. Безпека: Архітектура системи повинна бути забезпечена механізмами безпеки, щоб захистити конфіденційні дані користувачів, такі як особиста інформація та платіжні дані. Це може включати шифрування даних, захист від SQL-ін'єкцій, захист від перехоплення сесій і т. д.

Наприклад:

- Шифрування даних:

Система повинна використовувати шифрування для захисту конфіденційної інформації, такої як паролі користувачів. Наприклад, перед зберіганням паролів в базі даних вони повинні бути зашифровані за допомогою сучасних алгоритмів шифрування.

- Захист від SQL-ін'єкцій:

Система повинна бути захищена від атак SQL-ін'єкцій, які можуть дозволити зловмиснику виконати небажаний SQL-код на сервері баз даних.

2.3 Вибір технологій та інструментів розробки

Для розробки комп'ютеризованої Web-системи пошуку транспортних засобів для перевезення вантажів були обрані сучасні та надійні технології, які забезпечують високу продуктивність, безпеку та зручність використання. Вибір технологій був обумовлений потребою у масштабованості, легкості інтеграції з іншими сервісами, а також швидкістю розробки та підтримки.

Основні технології та інструменти включають:

Django - це фреймворк, необхідний для розробки веб-додатків на стороні сервера, що дозволяє програмістам створювати дуже швидко, не занурюючись у складні деталі [2]. Сенс веб-фреймворку в 21 столітті полягає в тому, щоб зробити виснажливі аспекти швидкої веб-розробки. Django має забезпечувати неймовірно швидке з'єднання інтернету розробка [1].

Обрання Django зумовлено кількома факторами:

Швидка розробка:

Django забезпечує швидку розробку завдяки великому набору вбудованих інструментів і функцій. Наприклад, включений ORM (Object-Relational Mapping) дозволяє працювати з базою даних через об'єкти Python, що робить роботу з даними більш простою та ефективною.

Безпека:

Django має вбудовані захисти від багатьох типових загроз веб-додатків, таких як SQL-ін'єкції, XSS (Cross-Site Scripting) і CSRF (Cross-Site Request Forgery). Використання вбудованих засобів безпеки дозволяє розробникам покращити безпеку своїх додатків і захистити їх від потенційних атак.

Мова програмування Python:

Python був обраний як основна мова програмування для розробки веб-додатків через свою потужність. Його зрозумілий синтаксис робить процес програмування більш доступним, особливо для початківців, та дозволяє швидко створювати функціональні веб-додатки. Python має велику кількість бібліотек, які можуть допомогти з різними аспектами веб-розробки, включаючи обробку даних, машинне навчання, веб-скрапінг та багато іншого [2].

База даних SQLite:

SQLite - це легковагова реляційна база даних, що відрізняється простотою використання та інтеграцією безпосередньо у додаток. Її вибір для проекту обумовлено декількома факторами:

Простота використання: SQLite не потребує складного налаштування сервера бази даних, що робить процес розробки та тестування додатків більш простим і швидким.

Фронтенд технології включають HTML, CSS та JavaScript і використовуються для створення інтерактивного та зручного інтерфейсу користувача.

Таким чином, вибір технологій та інструментів для розробки Web-системи дозволяє забезпечити її високу продуктивність, безпеку та зручність використання, а також швидко і ефективно розробку і підтримку.

Висновки за Розділом 2

Розділ 2 охоплює основні аспекти проектування та розробки моделі комп'ютеризованої Web-системи для пошуку транспортних засобів для перевезення вантажів. У рамках цього розділу були визначені ключові функціональні вимоги, які включають реєстрацію та авторизацію користувачів, створення замовлень та пошук транспорту, а також технічну підтримку. Ці вимоги забезпечують необхідний функціонал.

Проектування архітектури системи передбачало визначення взаємодії між основними компонентами, з акцентом на забезпечення безпеки та ефективності. Архітектура системи включає механізми шифрування даних, захист від SQL-ін'єкцій та інші заходи для захисту конфіденційної інформації користувачів.

Вибір технологій та інструментів розробки був зроблений на основі їх здатності забезпечити високу продуктивність, безпеку та зручність використання. Основними технологіями є Django, мова програмування Python, база даних SQLite, а також фронтенд технології HTML, CSS та JavaScript. Використання цих технологій дозволяє швидко та ефективно розробляти і підтримувати систему.

Таким чином, Розділ 2 визначає стратегічний курс акредитаційної роботи, надаючи міцну базу для подальшої розробки та впровадження системи. Завдяки ретельному визначенню функціональних вимог, проектуванню архітектури та обранню відповідних технологій, система буде надійною, масштабованою та здатною задовольнити потреби користувачів.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ WEB-СИСТЕМИ

3.1 Розробка інтерфейсу користувача

Розробка інтерфейсу користувача є ключовим етапом у створенні веб-системи, який визначає спосіб взаємодії користувачів з програмним продуктом.

Інтерфейс користувача було вирішено зробити у темних тонах з метою створення сучасного та стильного дизайну, який допоможе залучити увагу користувачів та покращити їхній враження від веб-сайту. Темний дизайн може надати веб-сайту елегантності, а також забезпечити комфортне використання вночі або в умовах слабкого освітлення.

Переваги темного дизайну включають:

- Сучасний вигляд: Темний дизайн є популярним у сучасному веб-дизайні і може зробити ваш веб-сайт більш привабливим для молодого покоління та тих, хто цінує сучасні тенденції.
- Підкреслення контенту: Темний фон може використовуватися для підкреслення контенту, так як яскраві кольори та текст виглядають більш контрастно на темному тлі, що допомагає залучити увагу користувача до важливої інформації.
- Зменшення очного напруження: Використання темного дизайну може зменшити напруження очей користувачів, особливо в умовах низького освітлення, що робить веб-сайт більш комфортним для перегляду.

- Естетика: Темний дизайн може надати веб-сайту елегантності та солідності, що підвищує загальний враження від користування сайтом.

Реалізація основних функцій

На базовому HTML шаблоні розташована панель навігації, яка забезпечує зручний доступ до основних розділів сайту та від нього вже наслідуються шаблони усіх сторінок. Для кожного виду користувачів панель навігації особиста:

1. Панель навігації клієнтів:

- Для відправника: Перехід на сторінку зі створенням замовлень та корисними рекомендаціями щодо застосування.
- Пошук вантажівки: Перехід на сторінку для пошуку публікацій вантажівок з фільтруванням публікацій.
- Ваші замовлення: Активні замовлення та архів замовлень.
- Прийняті вантажіки: Вивід прийнятих публікацій вантажівок

2. Панель навігації водіїв:

- Пошук замовлень: Перехід на сторінку з пошуком замовлень та докладним описом роботи пошукової системи.
- Опубліковані авто: Перехід на сторінку зі списком опублікованих автівок.
- Обрані замовлення: Перехід на сторінку зі списком та архівом прийнятих замовлень клієнта.
- Мої авто: Перехід на сторінку зі списком доданих авто.

Ця панель навігації забезпечує зручний та швидкий доступ до ключових функцій сайту, що допомагає користувачам легко зорієнтуватися та здійснити потрібні дії.

Приклад дизайну:



Рис 3.1 – Дизайн головної сторінки

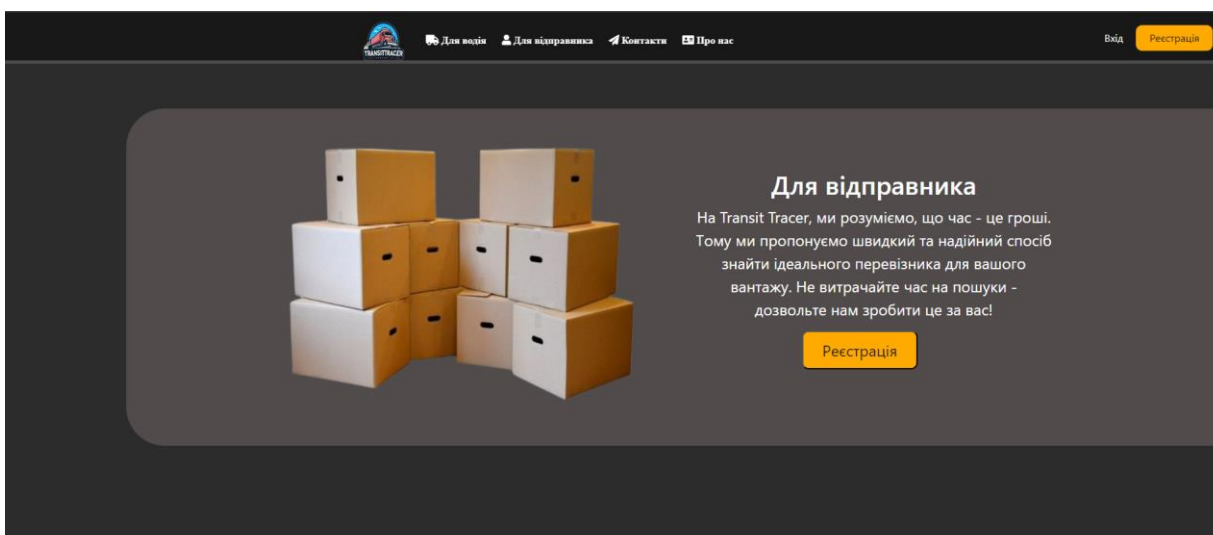


Рис 3.2 – Дизайн сторінки “Для відправника”

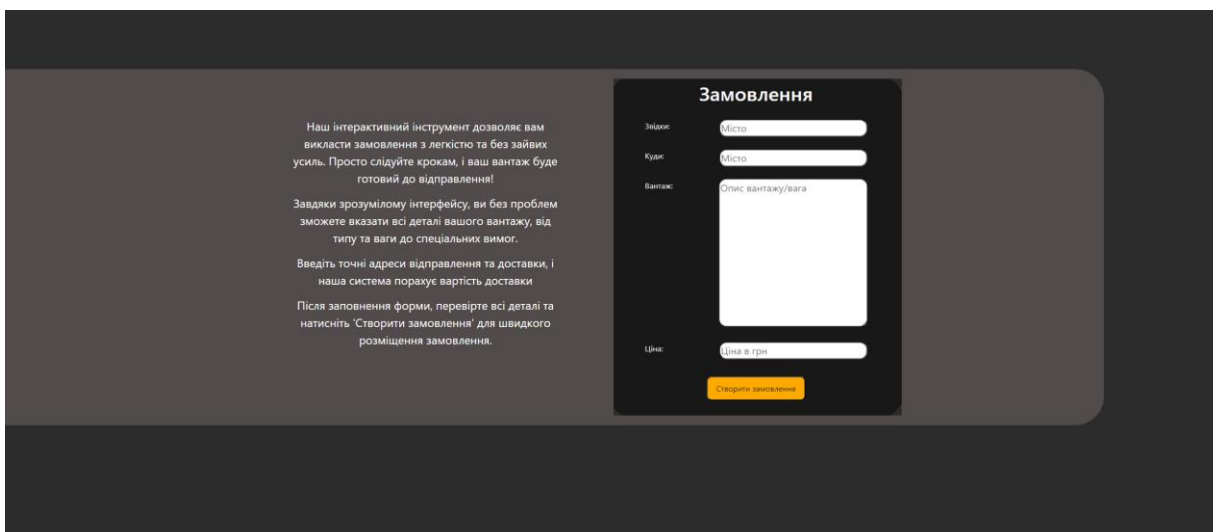


Рис 3.3 – сторінки “Для Водія”

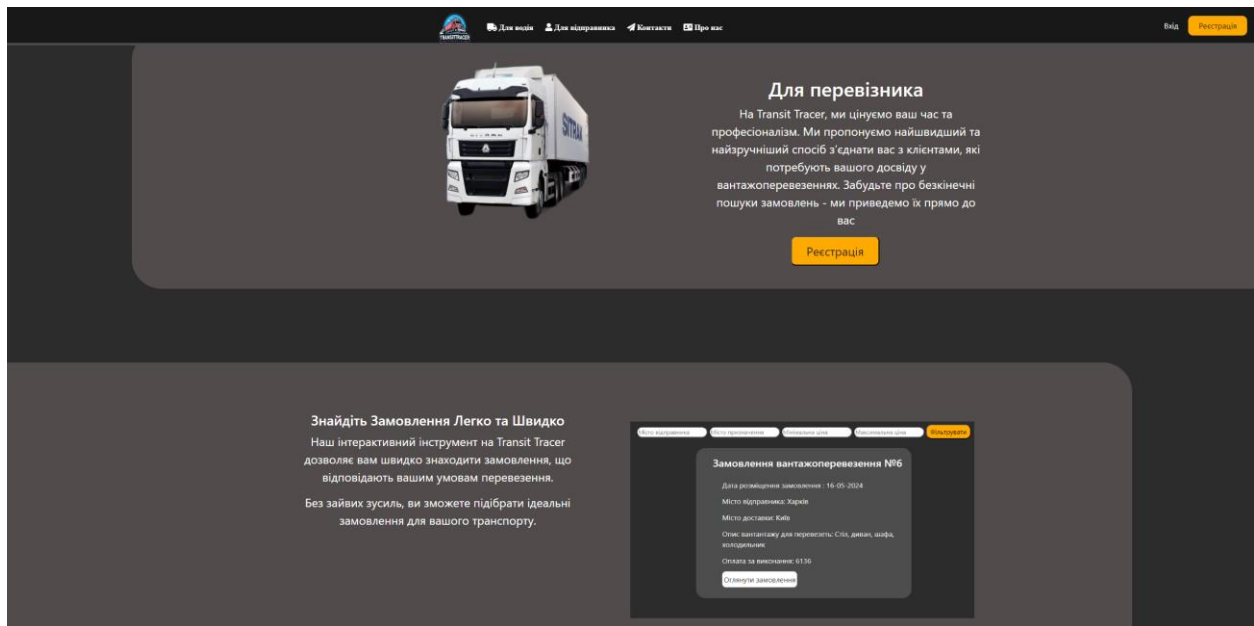


Рис 3.4 – сторінки “Для водія”

3.2 Реалізація основних функцій

Реалізація основних функцій – це один з ключових етап у розробці будь-якої веб-системи. На цьому етапі ми зосереджуємось на перетворенні концепцій та дизайну в реальні функціональні можливості, які користувачі зможуть використовувати.

Для відображення HTML файлів у веб-додатку на Django використовуються файли «urls.py» та «views.py», які працюють разом для обробки запитів користувачів і рендерингу відповідних сторінок.

Файл urls.py визначає маршрутизацію (URL-шляхи) для додатку. Кожен маршрут вказує, яке представлення (view) слід викликати для певного URL.

```

urlpatterns = [
    path('', index, name='home'),
    path('about/', about, name='about'),
    path('register/client/', RegisterUser.as_view(), name='register-client'),
    path('register/driver/', RegisterDriver.as_view(), name='register-driver'),
    path('login/', LoginUser.as_view(), name='login'),
    path('logout/', logout_user, name='logout'),
    path('profile/client/', ProfileUser.as_view(), name='profile-client'),
    path('profile/driver/', ProfileDriver.as_view(), name='profile-driver'),
    path('gruz/', gruz, name='gruz'),
    path('new/order/', AddAnnounce.as_view(), name='order'),
    path('password/change/form/', UserPasswordChange.as_view(), name='password_change'),
    path('password/change/done/', PasswordChangeDoneView.as_view(), name='password_change_done'),
    path('choise/', choise, name='choise'),
    path('contact/', contact, name='contact'),
    path('password-reset/',
        PasswordResetView.as_view(
            template_name='main/password_reset_form.html',
            email_template_name='main/password_reset_email.html',
            success_url=reverse_lazy('password_reset_done')
        ), name='password_reset'),
    path('password-reset/done/',
        PasswordResetDoneView.as_view(
            template_name='main/password_reset_done.html'
        ), name='password_reset_done'),
    path('password-reset/<uidb64>/<token>/',
        PasswordResetConfirmView.as_view(
            template_name='main/password_reset_confirm.html',
            success_url=reverse_lazy('password_reset_complete')
        ), name='password_reset_confirm'),
    path('password-reset/complete/',
        PasswordResetCompleteView.as_view(
            template_name='main/password_reset_complete.html'
        ), name='password_reset_complete'),
    path('orders/', orders, name='orders'),
    path('trans/', trans, name='trans'),
    path('filtred/orders/', filtered_orders, name='filtered_orders'),
    path('user/orders/', user_orders, name='user_orders'),
    path('user/orders/archive/', archive_orders, name='archive_orders'),
    path('order/<int:pk>', ShowOrder.as_view(), name='order'),
]

```

Рис 3.5 – Файл маршрутизації “urls.py”

Файл views.py містить функції або класи, які обробляють запити до відповідних URL і рендерять HTML шаблони з використанням контекстних даних. Таких як:

1. Створення замовлень вантажовідправником:

```

class AddAnnounce(DataMixin, CreateView):
    form_class = AnnounceAddForm
    template_name = 'main/announced.html'
    success_url = reverse_lazy('home')

    def form_valid(self, form):
        order = form.save(commit=False)
        order.customer = self.request.user.pk
        order.save()
        return super().form_valid(form)

    def get_context_data(self, *, object_list=None, **kwargs):
        context = super().get_context_data(**kwargs)
        c_def = self.get_user_context(title='Рестрація')
        return dict(list(context.items()) + list(c_def.items()))

class ShowOrder(View):
    def get(self, request, pk):
        order = get_object_or_404(Order, pk=pk)
        client = get_object_or_404(User, pk=order.customer)
        driver = None
        if order.driver:
            driver = get_object_or_404(User, pk=order.driver)
            car = get_object_or_404(Cars, pk=order.car_pk)
            form = CarSelectionForm(user=request.user)

        return render(request, 'main/order_page.html', {'order': order, 'client': client, 'driver': driver, 'form': form, 'car': car})

    def post(self, request, pk):
        order = get_object_or_404(Order, pk=pk)
        form = CarSelectionForm(request.POST, user=request.user)

        if form.is_valid():
            if request.user.is_authenticated and request.user.is_driver:
                order.driver = request.user.pk
                car = form.cleaned_data['car']
                order.car_pk = car.pk
                order.save()
                return HttpResponseRedirect(request.path_info)

            elif request.user.is_authenticated and request.user.is_client:
                order.completed = True
                order.save()
                return HttpResponseRedirect(request.path_info)

        return self.get(request, pk)

```

Рис 3.6 – Клас представлення форми створення замовлення та показ замовлення.

2. Вивід замовлень водію:

```

def orders(request):
    post_order = Order.objects.all().filter(complited=False).filter(driver__isnull=True)
    return render(request, 'main/orders.html', {'post_order': post_order})

```

Рис 3.7 – Функція представлення замовлень водію

3. Додавання та опублікування авто водієм:

```

class AddCar(DataMixin, CreateView):
    form_class = AddCarForm
    template_name = 'main/add_car.html'
    success_url = reverse_lazy('car')

    def form_valid(self, form):
        car = form.save(commit=False)
        car.driver_pk = self.request.user.pk
        car.save()
        return super().form_valid(form)

    def get_context_data(self, *, object_list=None, **kwargs):
        context = super().get_context_data(**kwargs)
        c_def = self.get_user_context(title='Додавання автівки')
        return dict(list(context.items()) + list(c_def.items()))

class PostCar(CreateView):
    form_class = PostCarForm
    template_name = 'main/post_car.html'
    success_url = reverse_lazy('home')

    def form_valid(self, form):
        post_car = form.save(commit=False)
        post_car.driver_pk = self.request.user.pk

        car = form.cleaned_data['car']

        post_car.car_pk = car

        post_car.driver = self.request.user.pk
        car.posted = True
        car.save()

        post_car.save()

        return super().form_valid(form)

    def get_form_kwargs(self):
        kwargs = super().get_form_kwargs()
        kwargs['user'] = self.request.user
        return kwargs

```

Рис 3.8 – Клас представлення усіх замовлень водію

Файл `models.py` у Django містить визначення моделей, які представляють структуру даних додатку. Кожна модель є класом, який успадковується від `django.db.models.Model` і визначає поля та поведінку даних, що зберігаються в базі даних. Моделі є основою для ORM (Object-Relational Mapping) у Django, що дозволяє взаємодіяти з базою даних за допомогою об'єктно-орієнтованого підходу. У випадку цієї системи було використано дві таблиці, це:

1. Розширення базової моделі Користувача для додавання нових полів таблиці:

Ця модель містить як стандартні поля такі як Ім'я, Прізвище, електронна пошта, пароль та підтвердження паролю також містить доповнення у вигляді полів фото, номер телефону, дату народження та статус відправника чи перевізника.

```

class User(AbstractUser):
    photo = models.ImageField(upload_to='main/%Y/%m/%d/', blank=True, null=True, verbose_name='Фотографія')
    phone = models.IntegerField(null=True, verbose_name='Номер телефону', max_length=50)
    date_birth = models.DateTimeField(blank=True, null=True, verbose_name='Дата народження')
    is_driver = models.BooleanField(default=False)
    is_client = models.BooleanField(default=False)

```

Рис 3.9 – Клас користувачів з додатковими полями

2. Модель Замошень користувачів:

Ця модель містить поля такі як дата створення замовлення, звідки перевезення, куди доставити, опис вантажу, вартість перевезення, ID вантажовідправника, ID перевізника, ID автівки та поле чи виконано замовлення

```

class Order(models.Model):
    date = models.DateField(auto_now_add=True)
    fromw = models.CharField(max_length=50)
    where = models.CharField(max_length=50)
    cargo = models.TextField()
    price = models.IntegerField()
    customer = models.IntegerField(blank=True)
    driver = models.IntegerField(blank=True, null=True)
    complited = models.BooleanField(default=False)
    car_pk = models.IntegerField(blank=True, null=True)

    def get_absolute_url(self):
        return reverse('order', kwargs={'pk': self.pk})

```

Рис 3.10 – Клас замовлень

3. Модель Автівок водіїв:

Ця модель містить поля такі як ID власника авто, фото авто, марка та модель, вантажопідйомність, номер авто та чи опубліковано авто.

```

class Cars(models.Model):
    driver_pk = models.IntegerField()
    photo = models.ImageField(upload_to='main/%Y/%m/%d/', verbose_name='Фотографія автомобіля')
    car = models.CharField(max_length=50)
    lifting = models.IntegerField()
    number = models.CharField(max_length=50)
    posted = models.BooleanField(default=False)

    def __str__(self):
        return self.car

```

Рис 3.11 – Клас автомобілей

4. Модель публікацій авто:

Ця модель містить поля такі як звідки перевезення, куди доставити, вартість перевезення, ID вантажовідправника, ID перевізника, ID автівки та поле чи виконано замовлення

```
class PostedCars(models.Model):
    fromwe = models.CharField(max_length=50)
    where = models.CharField(max_length=50)
    price = models.IntegerField()
    driver_pk = models.IntegerField()
    car_pk = models.ForeignKey(Cars, on_delete=models.CASCADE)
    customer = models.IntegerField(blank=True, null=True)
    driver = models.IntegerField(blank=True, null=True)
    complited = models.BooleanField(default=False)
```

Рис 3.12 – Клас публікацій авто

Файл forms.py - це ключовий компонент у веб-розробці з використанням фреймворка Django. У ньому визначаються класи форм, які дозволяють веб-сайту взаємодіяти з користувачами через введення та відправку даних.

Приклад форм для додавання авто та створення публікації:

```
class AddCarForm(forms.ModelForm):
    photo = forms.ImageField(label='Фото автомобіля', widget=forms.FileInput(attrs={'class': 'form-input'}))
    car = forms.CharField(label='Назва Автомобіль', widget=forms.TextInput(attrs={'class': 'form-input', 'placeholder': 'Марка та модель авто'}))
    lifting = forms.DecimalField(max_digits=10, decimal_places=2, label='Вантажопідйомність', widget=forms.NumberInput(attrs={'class': 'form-input', 'placeholder': 'Вантажопідйомність в кг'}))
    number = forms.CharField(label='Номер автомобіля', widget=forms.TextInput(attrs={'class': 'form-input', 'placeholder': 'Номер'}))

    class Meta:
        model = Cars
        fields = ['photo', 'car', 'lifting', 'number']

class PostCarForm(forms.ModelForm):
    fromwe = forms.CharField(max_length=50, label='Звідки', widget=AutoCompleteWidget(attrs={'placeholder': 'Місто', 'id': 'fromwe'}))
    where = forms.CharField(max_length=50, label='Куди', widget=AutoCompleteWidget(attrs={'placeholder': 'Місто', 'id': 'where'}))
    price = forms.DecimalField(max_digits=10, decimal_places=2, label='Ціна', widget=forms.NumberInput(attrs={'class': 'form-input', 'placeholder': 'Ціна в грн', 'id': 'price'}))
    car = forms.ModelChoiceField(queryset=None, label='Вибірять автомобіль', widget=forms.Select(attrs={'class': 'form-input'}))

    class Meta:
        model = PostedCars
        fields = ['fromwe', 'where', 'car', 'price']

    def __init__(self, *args, **kwargs):
        user = kwargs.pop('user', None)
        super(PostCarForm, self).__init__(*args, **kwargs)
        if user:
            self.fields['car'].queryset = cars.objects.filter(driver_pk=user.pk, posted=False)
```

Рис 3.13 – Класи для відображення форм

3.3 Тестування та відлагодження.

Тестування та відлагодження є критично важливими етапами у процесі розробки веб-системи. Вони забезпечують стабільність, надійність і безпеку

системи, гарантують, що всі функціональні можливості працюють належним чином, і допомагають виявити та виправити помилки.

Потрібні тести для веб-системи:

Реєстрація

Для перевірки функціоналу реєстрації необхідно протестувати кілька сценаріїв:

I. Успішна реєстрація:

- ❖ Заповнення всіх обов'язкових полів форми реєстрації (логін, ім'я, прізвище, електронна пошта, пароль).

Рис 3.14 – Тестування форми реєстрації

id	password	last_login	is_superuser	username	first_name	last_name	email	is_staff	is_active	date_joined	photo	phone	car	date_birth	is_driver	is_client
1	pbkdf2_sha256\$720000\$WRrYWRDZF4P4lsatc4A...	NULL	0	SlaveAn	Вячеслав	Добродорний	slavikdobrodniy7@gmail.com	0	1	2024-05-20 15:53:20.843424		NULL		NULL	0	1

Рис 3.15 – Перевірка запису нового користувача у базу даних

Авторизація

Тести для авторизації повинні охоплювати:

I. Успішний вхід:

- ❖ Використання правильних облікових даних для входу.
- ❖ Переконатися, що користувач перенаправляється на головну сторінку або особистий кабінет.

The screenshot shows a user profile form on a dark background. At the top, the title 'Профіль' is centered. Below it is a circular profile picture placeholder with a blue gradient. Underneath the profile picture, there are several input fields: 'Фотографія:' with a file selection button 'Вибрати файл' and the text 'Файл не вибрано'; 'Логин:' with a greyed-out input field; 'E-mail:' with a greyed-out input field; 'Ім'я:' with the value 'Вячеслав'; 'Прізвище:' with the value 'Доброродній'; 'Дата народження:' with a date picker showing 'October 26, 2002'; and 'Номер телефону:' with an empty input field. At the bottom, there are two yellow buttons: 'Змінити пароль' and 'Зберегти'.

Рис 3.16 – Перевірка авторизації (Профіль користувача)

Створення замовлення

Тести для створення замовлення повинні охоплювати:

I. Успішне створення замовлення:

- ❖ Заповнення всіх необхідних полів (опис вантажу, міста відправки та доставки).
- ❖ Переконатися, що замовлення успішно створюється і з'являється в списку замовлень.

The screenshot shows an order creation form on a dark background. The title 'Замовлення' is centered at the top. Below it are several input fields: 'Звідки:' with the value 'Харків'; 'Куди:' with the value 'Львів'; 'Вантаж:' with a text area containing the text 'Перевірка створення замовлення для перевізників'; and 'Ціна:' with the value '13089'. At the bottom, there is a yellow button labeled 'Створити замовлення'.

Рис 3.17 – Перевірка створення замовлення

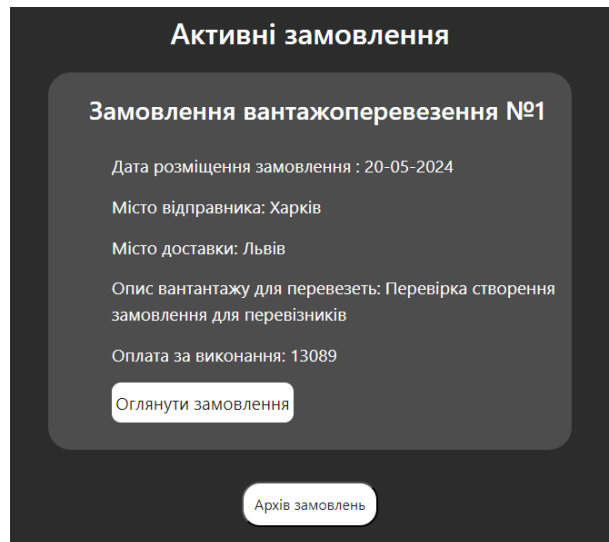


Рис 3.18 – Вивід створеного замовлення

Вивід усіх замовлень перевізнику

Для цього функціоналу потрібно перевірити:

I. Відображення замовлень:

- ❖ Перевірка, що перевізник може бачити список всіх доступних замовлень.
- ❖ Переконатися, що інформація про кожне замовлення відображається коректно.

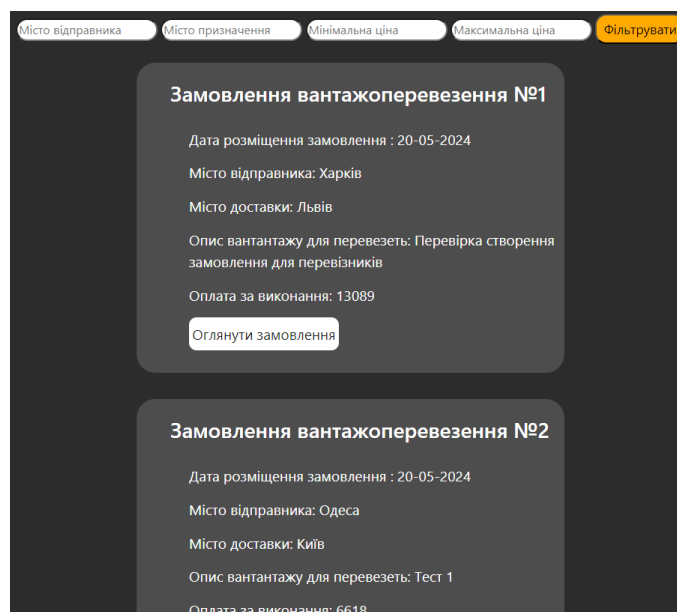


Рис 3.19 – Перевірка виводу усіх замовлень перевізнику

Прийняття замовлення перевізником

Тести повинні включати:

I. Успішне прийняття замовлення:

- ❖ Вибір замовлення зі списку і підтвердження його прийняття.
- ❖ Переконалися, що замовлення позначається як прийняте і зникає з списку доступних для інших перевізників.

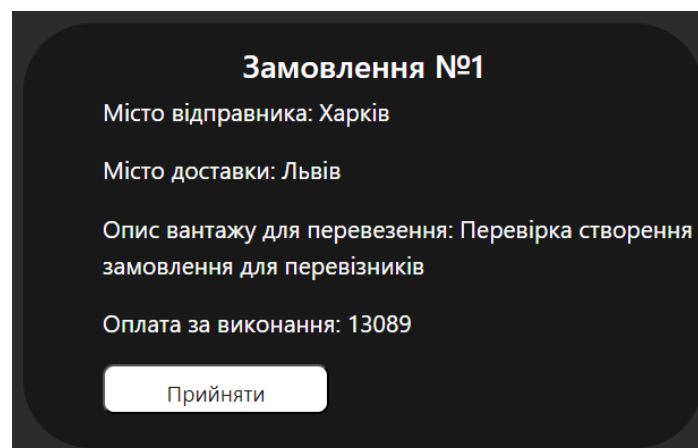


Рис 3.20 – Сторінка з описом замовлення для перевірки прийняття

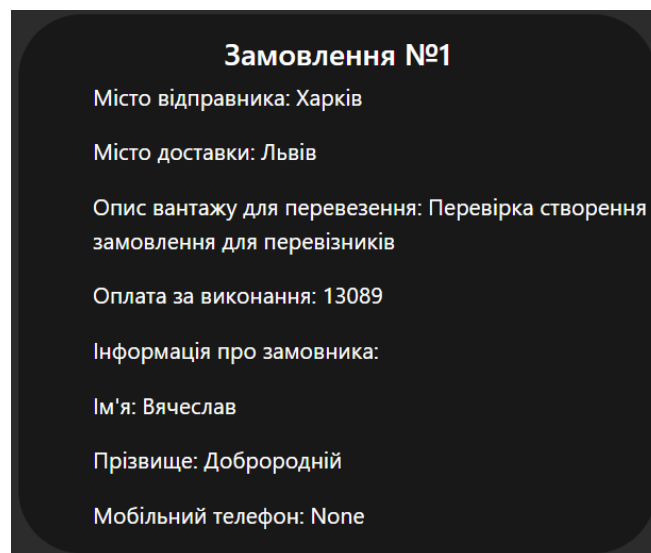


Рис 3.21 – Перевірка прийняття замовлення пройшла успішно

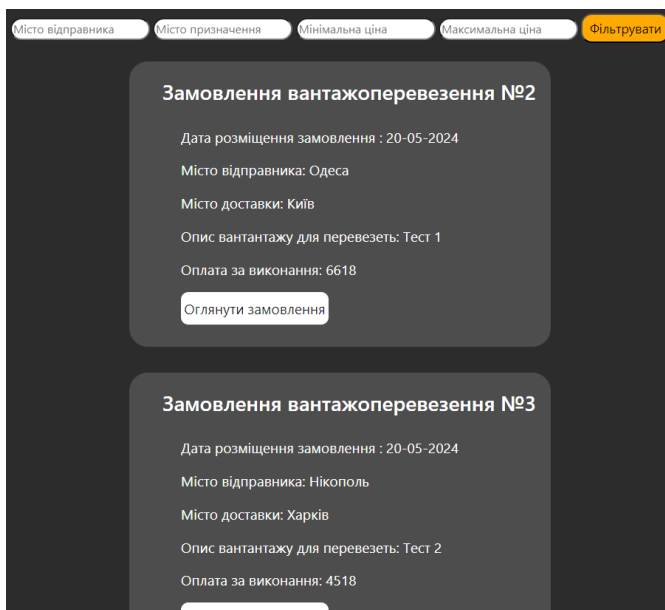


Рис 3.22 – Перевірка зникнення замовлення з пошуку

Помітка відправника про виконання замовлення

Цей функціонал повинен бути перевірений наступним чином:

I. Успішна помітка про виконання:

❖ Відправник відмічає замовлення як виконане після доставки вантажу.

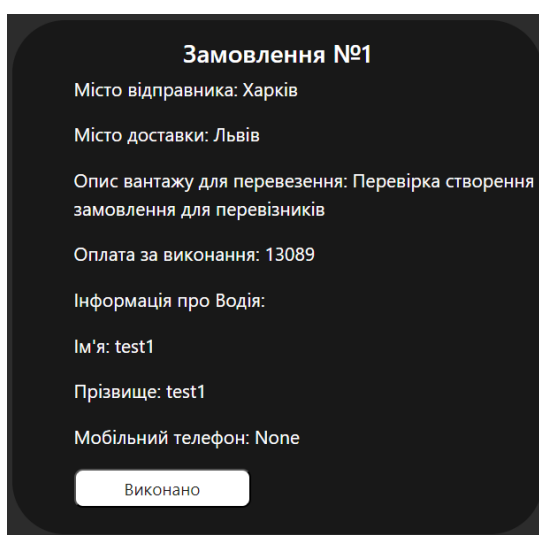


Рис 3.23 – Сторінка з описом замовлення для перевірки виконання
замовлення

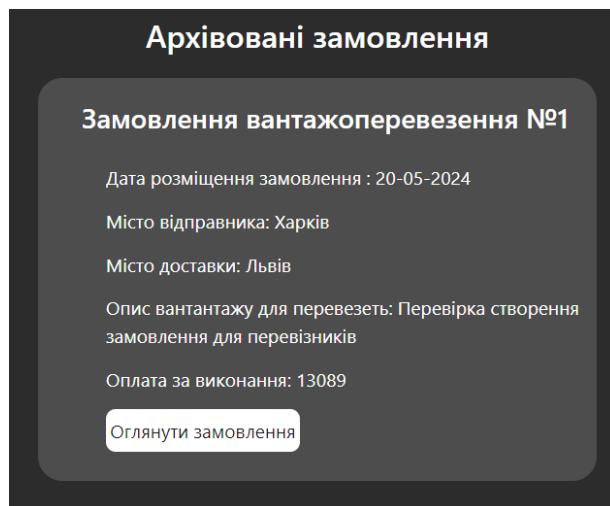


Рис 3.24 – Замовлення було архівовано (тобто виконано)

Тестування кожного з цих аспектів гарантує, що система працює належним чином, і користувачі можуть безпечно та ефективно виконувати всі необхідні дії.

Додавання авто

Тести для створення замовлення повинні охоплювати:

II. Успішне додавання авто:

- ❖ Заповнення всіх необхідних полів (фото, модель та марка авто, вантажопідйомність та номер).
- ❖ Переконатися, що замовлення успішно створюється і з'являється в списку замовлень.

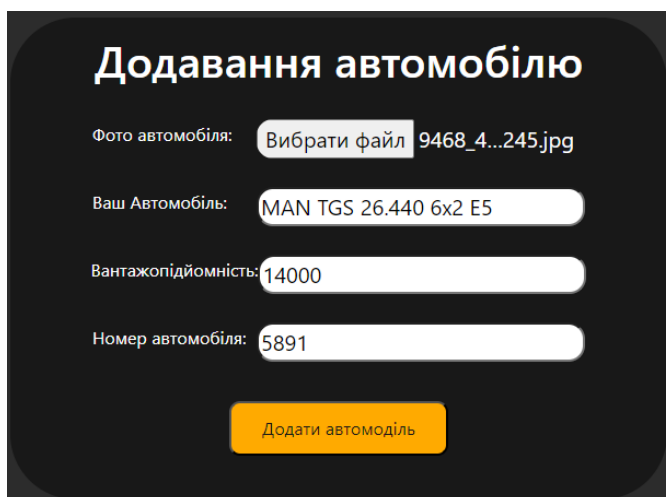


Рис 3.25 – Додавання автомобілю



Рис 3.26 – Успішне додавання нового авто

Створення публікації

Тести для створення замовлення повинні охоплювати:

III. Успішне створення публікації:

- ❖ Заповнення всіх необхідних полів (міста відправки та доставки, обрати авто зі списку та вказати ціну за послугу).
- ❖ Переконатися, що замовлення успішно створюється і з'являється в списку замовлень.

Рис 3.27 – Створення публікації авто

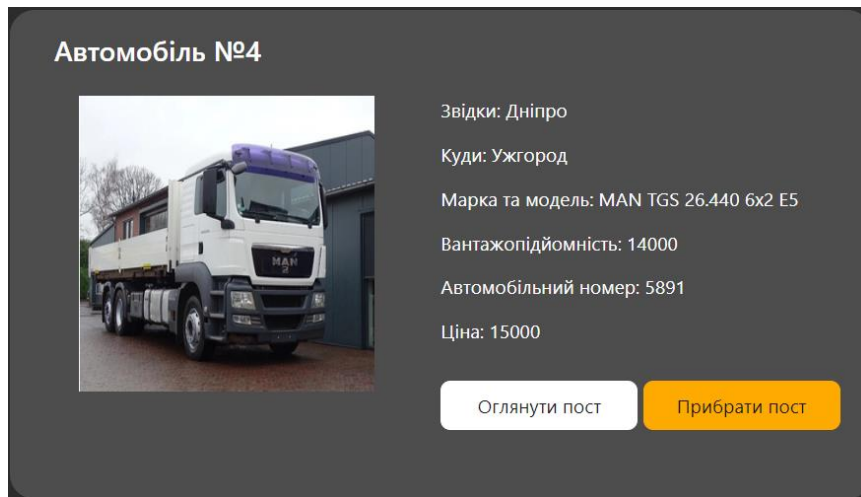


Рис 3.28 – Успішне створення публікації авто

Вивід усіх публікацій клієнту

Для цього функціоналу потрібно перевірити:

II. Відображення замовлень:

- ❖ Перевірка, що клієнт може бачити список всіх доступних автівок.
- ❖ Переконатися, що інформація про кожне замовлення відображається коректно.
- ❖ Перевірити сортування замовлень.

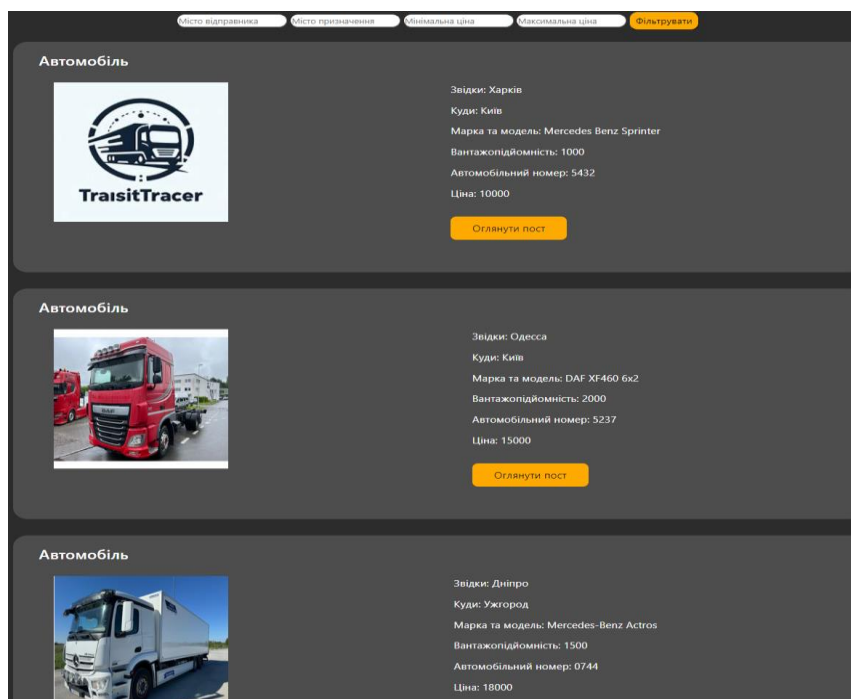


Рис 3.29 – Вивід усіх доступних автівок клієнту

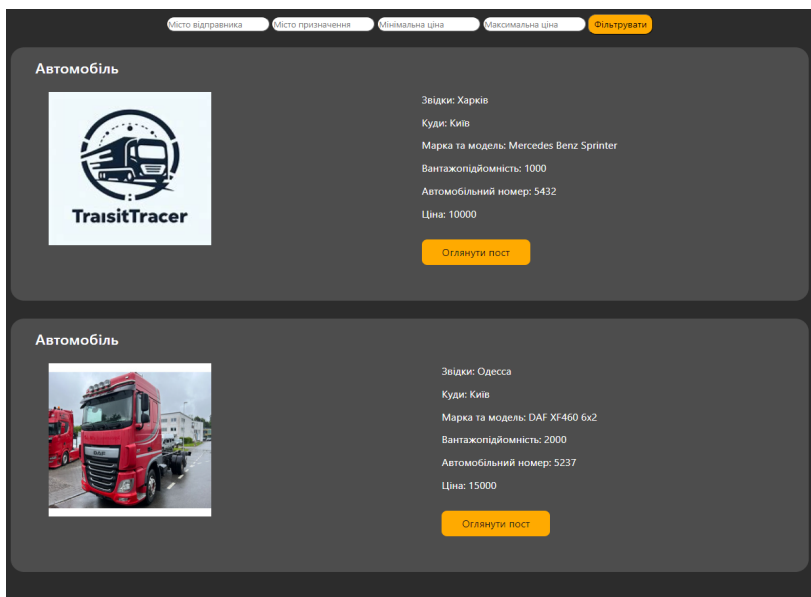


Рис 3.30 – Успішна фільтрація за містом доставки

Прийняття публікації користувачем

Тести повинні включати:

II. Успішне прийняття замовлення:

- ❖ Вибір публікації зі списку і підтвердження його прийняття.
- ❖ Перевірка кнопки «Виконано»

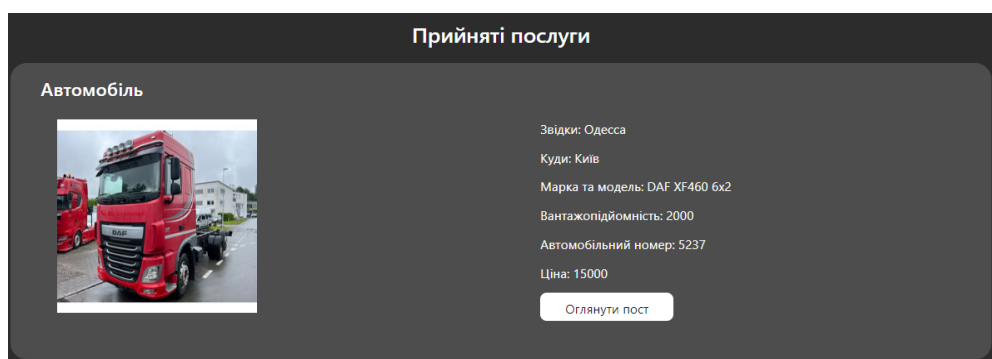


Рис 3.31 – Успішне прийняття транспорту

id	fromwe	where	price	driver_pk	customer	driver	complited	car_pk_id
Філ...	Фільтр	Фільтр	Філ...	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	Харків	Київ	10000	1	NULL	1	0	1
2	Одеса	Київ	15000	1	2	1	1	2

Рис 3.32 – Успішний запис що послуга виконана

Висновки за Розділом 3

Розділ 3 охоплює процес реалізації функціональності та тестування системи. Під час розробки важливо враховувати потреби користувачів.

Перш за все, було розроблено функціонал, який відповідає вимогам та потребам користувачів системи. Це включало в себе реєстрацію, авторизацію, створення та відображення замовлень для відправників та перевізників, додавання авто та їх публікацію, пошук потрібного авто клієнтом, а також взаємодію з цими замовленнями та публікаціями.

Після цього важливим етапом було проведення тестування, яке включало в себе перевірку коректності реалізації функціональності, виявлення та виправлення помилок, а також забезпечення стабільності та надійності системи.

У цьому розділі також працював над вдосконаленням дизайну та інтерфейсу користувача, зробивши його більш зручним та привабливим для користувачів.

В результаті ця веб-система стала більш функціональною, надійною та зручною для використання, що дозволить задовольнити потреби користувачів і забезпечити їм позитивний досвід використання системи

РОЗДІЛ 4

ОЦІНКА ЕФЕКТИВНОСТІ ТА ПОРІВНЯННЯ З ІСНУЮЧИМИ СИСТЕМАМИ

4.1 Проведення тестування залежно від різних критеріїв

Оцінка ефективності веб-системи за допомогою Lighthouse

Lighthouse — це потужний інструмент, який надає комплексну оцінку продуктивності веб-сайтів за різними критеріями. Інструмент інтегрований у Chrome DevTools і дозволяє провести детальний аналіз веб-сторінки.

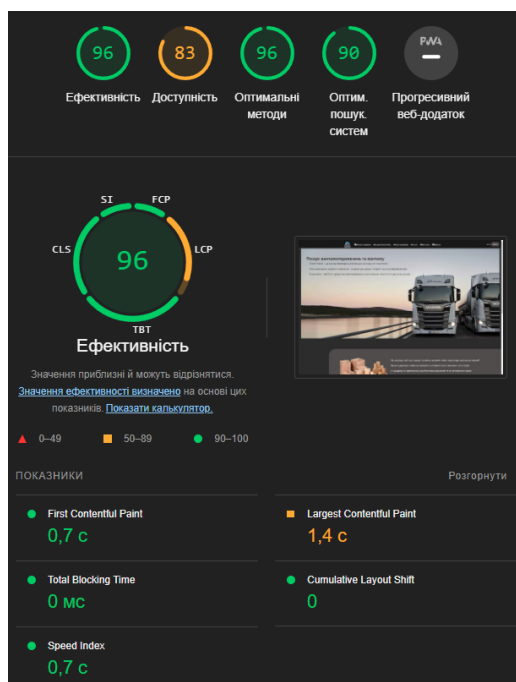


Рис 4.1 – Результати перевірки Lighthouse однієї сторінки

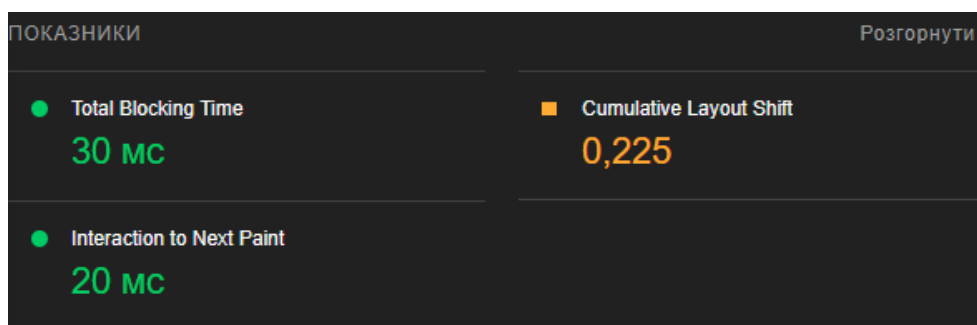


Рис 4.2 – Результати перевірки Lighthouse при користуванні системою

Таблиця 4.1

Результати перевірки однієї сторінки та користуванням системою

Метрика	Значення однієї сторінки	Значення при користуванні
First Contentful Paint (FCP)	0.7 с	Н/Д
Largest Contentful Paint (LCP)	1.4 с	Н/Д
Total Blocking Time (TBT)	0 мс	30 мс
Cumulative Layout Shift (CLS)	0	0.225
Speed Index	0.7 с	Н/Д
Interaction to Next Paint (INP)	Н/Д	20 мс

4.2 Аналіз результатів

Розглянемо кожну метрику з поданого переліку більш детально:

First Contentful Paint (FCP): Ця метрика вказує на час, який потрібно вашому веб-системі для того, щоб відобразити перший елемент змісту під час завантаження. Низьке значення FCP, таке як отримане 0,7 с, свідчить про швидкість відображення вмісту, що може сприяти позитивному першому враженню користувача.

Largest Contentful Paint (LCP): Ця метрика вказує на час, який потрібно системі для того, щоб повністю завантажити найбільший елемент. Отримане значення LCP, яке складає 1,4 с, також вказує на швидкість завантаження контенту, що є дуже позитивним для користувачів.

Total Blocking Time (TBT): Ця метрика вказує на загальний час, протягом якого система блокується і не реагує на введення користувача. Отримане значення TBT становить 0 мс, що свідчить про відсутність блокуючих часів, що є дуже позитивним для користувачів.

Cumulative Layout Shift (CLS): Ця метрика вказує на стабільність макета сторінки під час завантаження. Отримане значення CLS відсутнє для однієї сторінки і становить 0,225 при користуванні. Це означає, що інтерфейс системи

досить стабільний, але може виникати певний зсув макета під час взаємодії користувача.

Speed Index: Ця метрика вимірює швидкість, з якою вміст сторінки стає видимим для користувача. Отримане значення Speed Index становить 0,7 с, що є дуже позитивним, оскільки швидке відображення вмісту може покращити сприйняття користувачем.

Interaction to Next Paint (INP): Ця метрика вимірює час між взаємодією користувача з сайтом і моментом початку наступного оновлення візуального представлення сторінки. Цей показник при користуванні вказує на те, що система майже миттєво реагує на дії користувача, що є дуже позитивним для загального враження про сайт.

Отже, хоча загальна продуктивність системи висока, існують можливості для покращення, зокрема в усуненні незначних зсувів макета та оптимізації часу завантаження найбільшого контенту.

4.3 Порівняння отриманих даних з існуючими системами

У процесі збору інформації про ефективність існуючих систем використовувалася методика Lighthouse. Цей інструмент надає можливість об'єктивно оцінити такі параметри, як ефективність та доступність, шляхом автоматизованого аналізу веб-сторінок. Використання Lighthouse дозволило систематизувати та зіставити дані щодо ефективності систем, надаючи зрозумілі та об'єктивні результати порівняння. Отримані результати аналізу:

Freightos [4]:

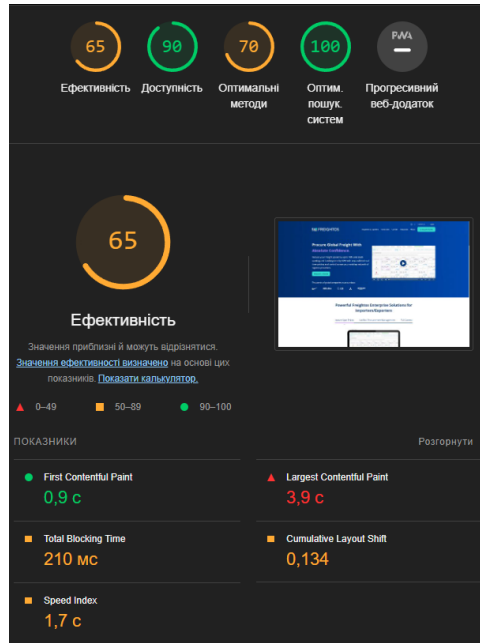


Рис 4.3 – Результати перевірки Lighthouse однієї сторінки Freightos

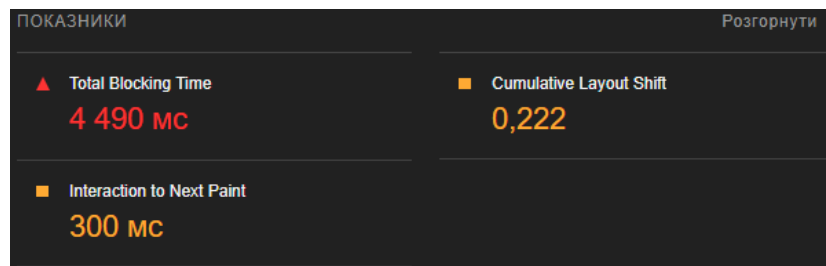


Рис 4.4 – Результати перевірки Lighthouse при користуванні системою Freightos

Uber Freight [6]:

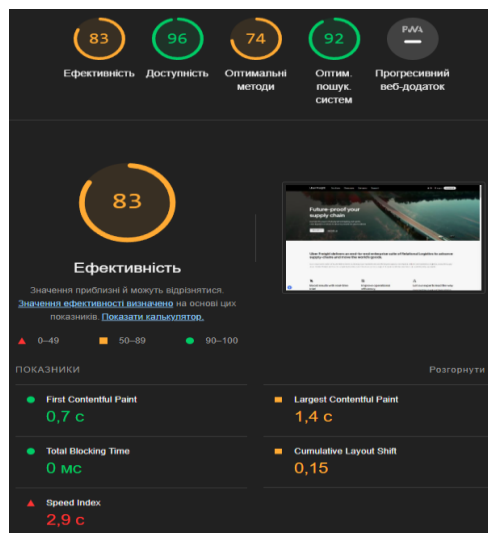


Рис 4.5 – Результати перевірки Lighthouse однієї сторінки Uber Freight

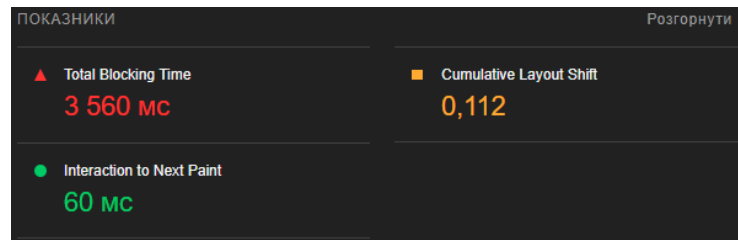


Рис 4.6 – Результати перевірки Lighthouse при користуванні системою Uber Freight

Freightera [3]:

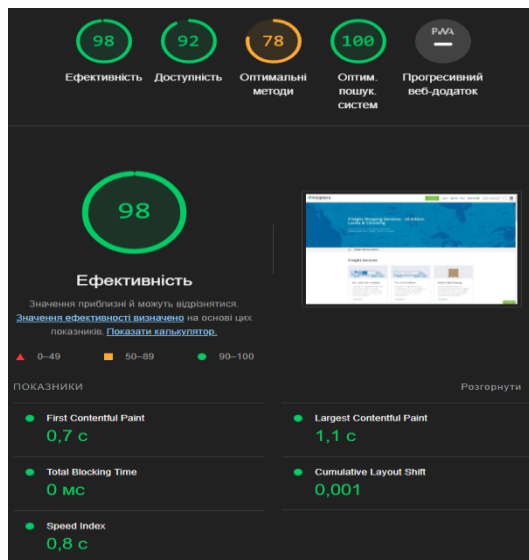


Рис 4.7 – Результати перевірки Lighthouse однієї сторінки Freightera

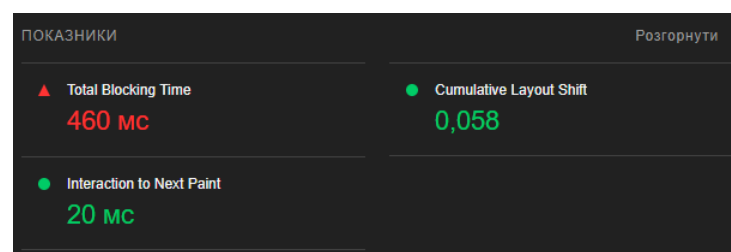


Рис 4.8 – Результати перевірки Lighthouse при користуванні системою Freightera

Freightquote [5]:

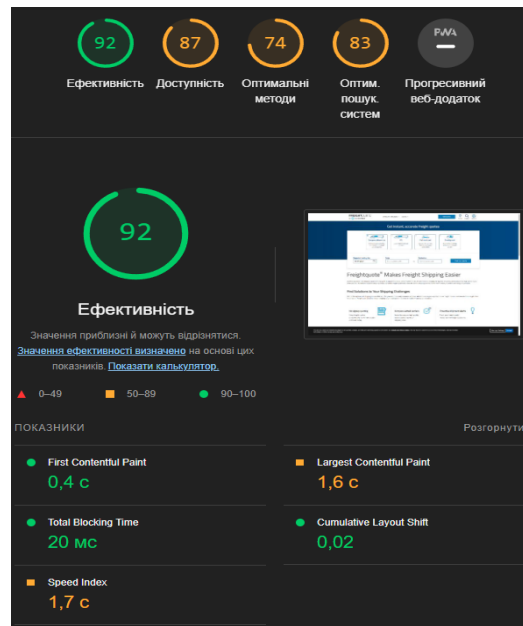


Рис 4.9 – Результати перевірки Lighthouse однієї сторінки Freightquote

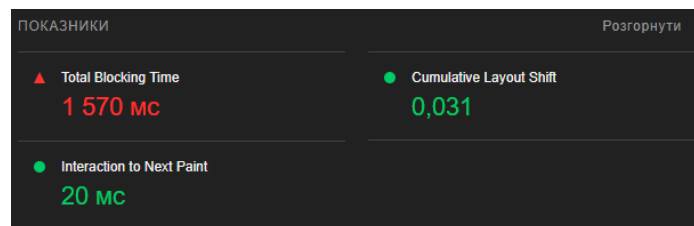


Рис 4.10 – Результати перевірки Lighthouse при користуванні системою Freightquote

Таблиця 4.2

Результати перевірки існуючих системам

	Freightos		Uber Freight		Freightera		Freightquote	
	Одна сторінка	Користування	Одна сторінка	Користування	Одна сторінка	Користування	Одна сторінка	Користування
FCP	0.9 с	Н/Д	0.7 с	Н/Д	0.7 с	Н/Д	0.4 с	Н/Д
LCP	3.9 с	Н/Д	1.4 с	Н/Д	1.1 с	Н/Д	1.6 с	Н/Д
TBT	210 мс	4490 мс	0	3560 мс	0	460 мс	20 мс	1570 мс
CLS	0.134	0.222	0.15	0.112	0.001	0.058	0.02	0.031
SI	1.7 с	Н/Д	2.9 с	Н/Д	0.8	Н/Д	1.7 с	Н/Д
INP	Н/Д	300 мс	Н/Д	60 мс	Н/Д	20 мс	Н/Д	20 мс

Графіки порівняння результатів перевірки:

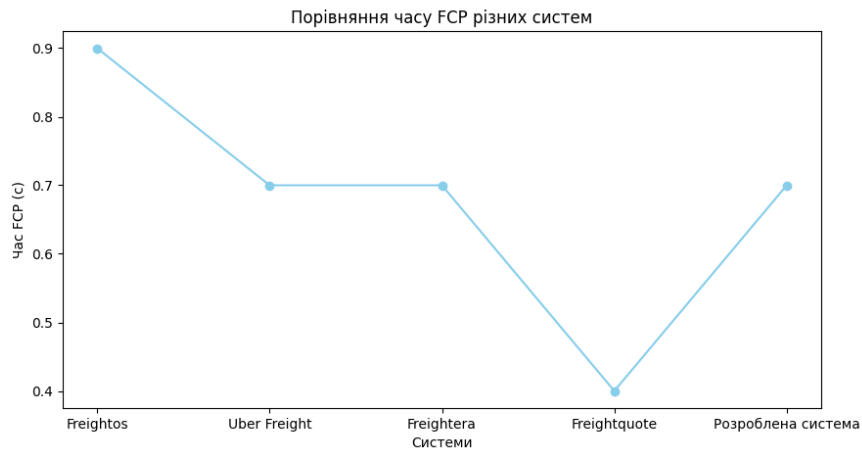


Рис 4.11 – Графік порівняння метрик FCP усіх систем

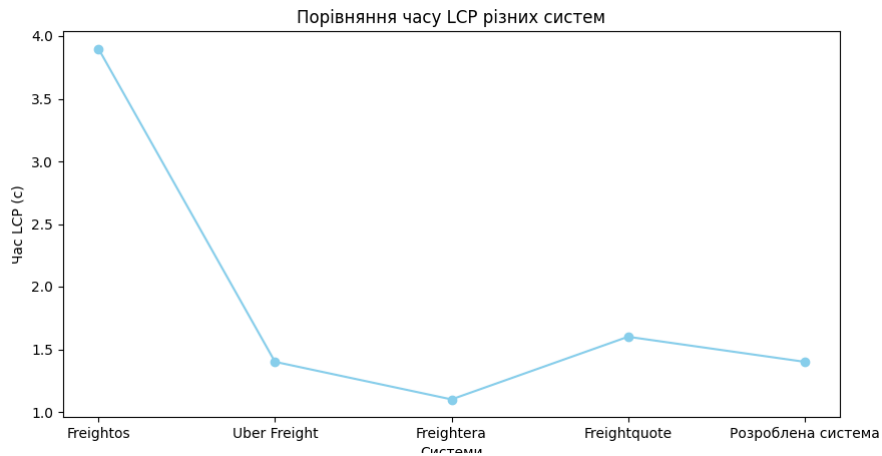


Рис 4.11 – Графік порівняння метрик LCP усіх систем

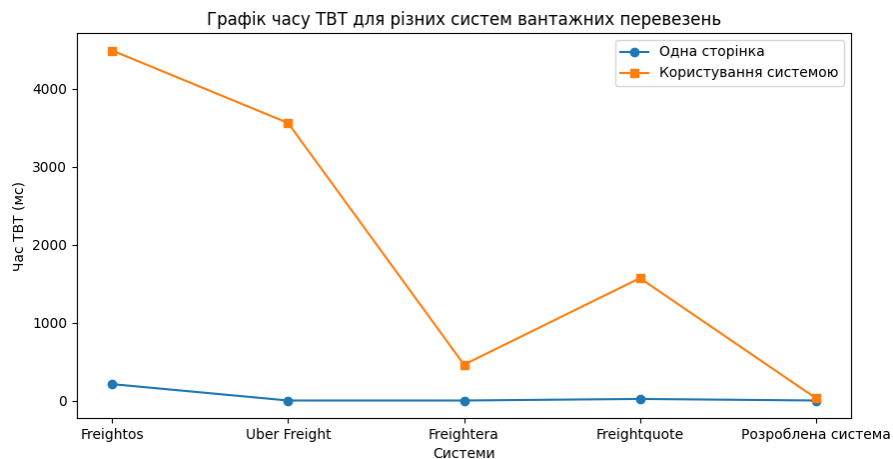


Рис 4.12 – Графік порівняння метрик TBT усіх систем

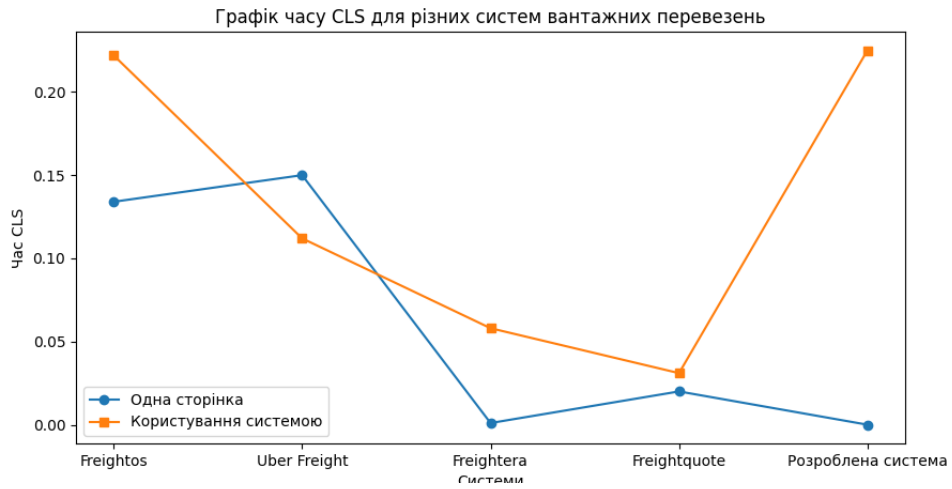


Рис 4.13 – Графік порівняння метрик CLS усіх систем

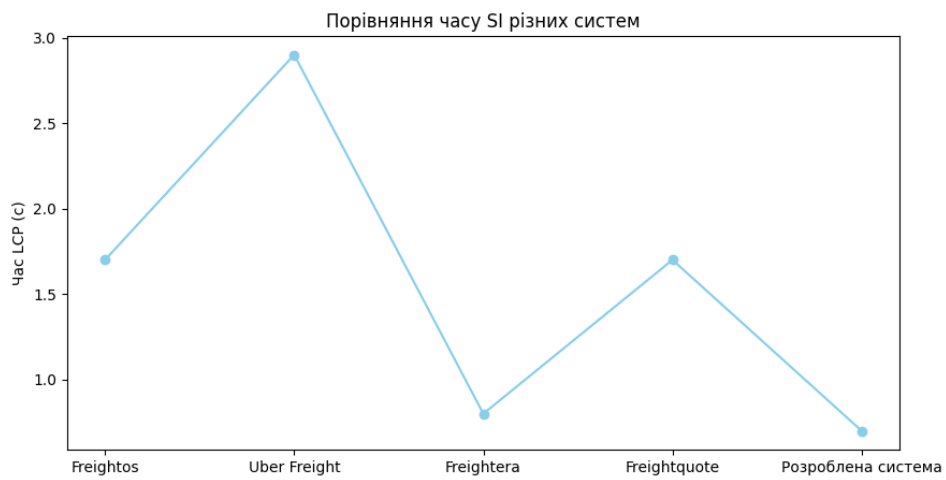


Рис 4.13 – Графік порівняння метрик SI усіх систем

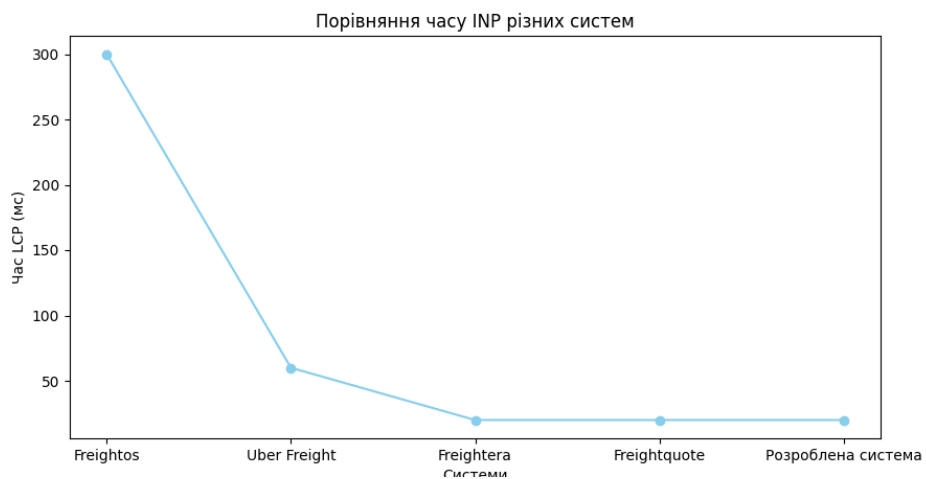


Рис 4.14 – Графік порівняння метрик INP усіх систем

Порівнюючи аналіз розробленої Web-системи та існуючих систем, можна зробити висновок, що розроблена система демонструє високу продуктивність і конкурентоспроможність. Завдяки використанню сучасних технологій та інструментів.

Висновки за Розділом 4

Розділ 4 охоплює оцінку ефективності веб-системи та порівняння її з існуючими системами. Під час аналізу використовувався інструмент Lighthouse, який надає комплексну оцінку продуктивності веб-сайтів за різними критеріями, такими як ефективність, доступність, передові веб-технології та SEO.

Перш за все, було проведено тестування системи за допомогою Lighthouse, що включало оцінку ключових метрик: First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), Cumulative Layout Shift (CLS), Speed Index (SI) та Interaction to Next Paint (INP). Результати тестування показали, що система демонструє високу продуктивність.

Далі результати розробленої системи були порівняні з існуючими аналогами, такими як Freightos, Uber Freight, Freightera та Freightquote. Отримані дані дозволили зробити об'єктивне порівняння та виявити переваги та недоліки нашої системи. Зокрема, наша система показала конкурентоспроможні результати за метриками FCP та LCP, а також відмінні показники TBT та SI. Проте, деякі аспекти, такі як Cumulative Layout Shift (CLS), потребують подальшого вдосконалення для зменшення зсувів макета під час користування системою.

ВИСНОВКИ

Ця робота охоплює всебічний аналіз та розробку комп'ютеризованої Web-системи, включаючи порівняння з існуючими рішеннями.

У першому розділі було здійснено огляд сучасних Web-систем, їхніх переваг і недоліків, а також вивчено вимоги користувачів. Аналіз показав необхідність створення системи з інтуїтивно зрозумілим інтерфейсом та високою продуктивністю.

Другий розділ присвячений створенню моделі, визначенню функціональних вимог, проектуванню архітектури та вибору технологій. Було розроблено функції, які забезпечують масштабованість і гнучкість системи, використовуючи сучасні технології для забезпечення високої продуктивності та надійності.

Третій розділ охоплює реалізацію функціональності та тестування системи. Розроблено інтерфейс користувача та основні функції, проведено тестування для забезпечення коректності роботи, стабільності та надійності системи.

Четвертий розділ присвячений оцінці ефективності системи за допомогою інструменту Lighthouse та порівнянню з існуючими системами. Результати тестування показали високу продуктивність нашої системи, конкурентоспроможність за ключовими показниками та можливості для подальшого вдосконалення.

Загалом, створена Web-система є високопродуктивною, надійною та зручною у використанні, що дозволить задовольнити потреби користувачів і забезпечити їм позитивний досвід.


СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Філософія дизайну. Документація Django : [Електронний ресурс]. – режим доступу: URL: <https://docs.djangoproject.com/en/5.0/misc/design-philosophies/> (дата звернення: 20.03.2024).
- 2) Deery M. 7 переваг вивчення Python для веб-розробки : [Електронний ресурс]. – режим доступу: URL: <https://careerfoundry.com/en/blog/web-development/reasons-to-learn-python/> (дата звернення: 15.03.2024).
- 3) Freightera. Пошук вантажу та транспорту : [Електронний ресурс]. – режим доступу: URL: <https://www.freightera.com/> (дата звернення: 26.02.2024).
- 4) Freightos. Пошук вантажу та транспорту : [Електронний ресурс]. – режим доступу: URL: <https://www.freightos.com/> (дата звернення: 17.02.2024).
- 5) Freightquote. Пошук вантажу та транспорту : [Електронний ресурс]. – режим доступу: URL: <https://www.freightquote.com/> (дата звернення: 03.03.2024).
- 6) Uber Freight. Пошук вантажу та транспорту : [Електронний ресурс]. – режим доступу: URL: <https://www.uberfreight.com/> (дата звернення: 20.02.2024).

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Факультет комп'ютерних наук
Кафедра теоретичної та прикладної системотехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **бакалавр**
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 – Комп'ютерна інженерія.

ЗАТВЕРДЖУЮ
Завідувач кафедри теоретичної
та прикладної системотехніки
д.т.н., проф. Шматков С. І.
«21» грудня 2024 року



З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

ДОБРОРОДНЬОГО ВЯЧЕСЛАВА ЮРІЙОВИЧА

1. Тема роботи **«МОДЕЛЬ КОМП'ЮТЕРИЗОВАНОЇ WEB-СИСТЕМИ ПОШУКУ ТРАНСПОРТНИХ ЗАСОБІВ ДЛЯ ПЕРЕВЕЗЕННЯ ВАНТАЖІВ»**

керівник роботи **Лабенко Дмитро Петрович, доцент, к.т.н.**
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «03» травня 2024 року № 4101-5/909

2. Строк подання студентом роботи 31 травня 2024 року

3. Перелік питань, які потрібно розробити

- 1) Аналіз існуючих систем пошуку транспортних засобів для перевезення вантажів.
- 2) Вивчення вимог користувачів.
- 3) Опис моделі комп'ютеризованої Web-системи.
- 4) Реалізація Web-системи.
- 5) Оцінка ефективності та порівняння з існуючими системами.
- 6) Розробка пояснювальної записки.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Аналіз існуючих систем пошуку транспортних засобів для перевезення вантажів	14.02.2024 - 10.03.2024
2	Вивчення вимог користувачів.	20.02.2024 - 6.03.2024
3	Опис моделі комп'ютеризованої Web-системи.	2.03.2024 - 20.03.2024
4	Реалізація Web-системи.	2.03.2024 - 24.03.2024
5	Оцінка ефективності та порівняння з існуючими системами.	24.03.2024 - 10.04.2024
6	Розробка пояснювальної записки.	10.04.2024 – 20.05.2024
7	Оформлення звіту за результатами переддипломної практики.	15.05.2024 – 31.05.2024
8	Представлення кваліфікаційної роботи керівнику та рецензенту.	31.05.2024
9	Оформлення пояснювальної записки та підготовка презентації.	31.05.2024

5. Дата видачі завдання 21.12.2023

Студент

В.Ю. Доброродній

ініціали, прізвище


 підпис

Керівник роботи

Д.П. Лабенко

ініціали, прізвище


 підпис

ДОДАТОК Б

Затверджую

« _____ » _____ 2024 р.

Технічне завдання
на розробку програмного виробу
«Модель комп'ютеризованої Web-системи пошуку транспортних засобів
для перевезення вантажів»

Назва розділу	Назва і зміст підрозділу
1. Введення	1.1. Назва програмного виробу – Модель комп'ютеризованої Web-системи пошуку транспортних засобів для перевезення вантажів 1.2. Галузь застосування – 12 – Інформаційні технології
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № _4101-5/909 від «_03_» Травня_2024 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3. Призначення розробки	3.1. Мета розробки: Спрощення процесу пошуку транспортних засобів для перевезення вантажів з використанням сучасних web-технологій на ринку автоперевезень. 3.2. Призначення розробки: полягає в створенні інструменту, який сприятиме оптимізації логістичних процесів, зменшенню витрат та підвищенню ефективності перевезень вантажів для різних учасників ринку. 3.3. Початкові дані для розробки: дані про: клієнтів та їх потреби, ціни та умови перевезення, конкурентів та ринок

<p>4.</p> <p>Технічні вимоги до програмного виробу</p>	<p>4.1. Вимоги до функціональних характеристик:</p> <ol style="list-style-type: none"> 1) Пошук транспорту 2) Фільтрація результатів 3) Реєстрація та авторизація користувачів <p>4.2. Вимоги до надійності: Забезпечення того, щоб система працювала безперервно та стабільно навіть при великому навантаженні.</p> <p>4.3. Вимоги до умов експлуатації: немає</p> <p>4.4. Вимоги до складу і параметрів технічних засобів: Персональний комп'ютер у повній комплектації (ноутбук)</p> <p>4.5. Вимоги до інформаційної та програмної сумісності: забезпечити сумісність з усіма обчислювальними засобами.</p> <p>4.6. Вимоги до маркування та упаковки: відсутні.</p> <p>4.7. Вимоги до транспортування і зберігання: відсутні.</p> <p>4.8. Спеціальні вимоги: відсутні.</p>
<p>5.</p> <p>Вимоги до програмної документації.</p>	<p>Програмою документацією до виробу «Модель комп'ютеризованої Web-системи пошуку транспортних засобів для перевезення вантажів» вважати:</p> <ol style="list-style-type: none"> 1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до дипломної роботи). 2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до дипломної роботи). 3) Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).

<p>6.</p> <p>Техніко-економічні показники</p>	<p>В даному розділі можуть бути представлені:</p> <p>1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до дипломної роботи).</p> <p>2) Методику розрахунку інформативності змінних стану (у вигляді глав 3.2 та 3.3 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>													
<p>7.</p> <p>Стадії і етапи розробки</p>	<table border="1"> <thead> <tr> <th data-bbox="395 954 922 1010">Дата</th> <th data-bbox="930 954 1552 1010">Назва етапу</th> </tr> </thead> <tbody> <tr> <td data-bbox="395 1021 922 1211">14.02.2024 - 10.03.2024</td> <td data-bbox="930 1021 1552 1211">Аналіз існуючих систем пошуку транспортних засобів для перевезення вантажів</td> </tr> <tr> <td data-bbox="395 1223 922 1413">20.02.2024 - 6.03.2024</td> <td data-bbox="930 1223 1552 1413">Вивчення вимог користувачів.</td> </tr> <tr> <td data-bbox="395 1424 922 1615">2.03.2024 - 20.03.2024</td> <td data-bbox="930 1424 1552 1615">Опис моделі комп'ютеризованої Web-системи.</td> </tr> <tr> <td data-bbox="395 1626 922 1816">2.03.2024 - 24.03.2024</td> <td data-bbox="930 1626 1552 1816">Реалізація Web-системи.</td> </tr> <tr> <td data-bbox="395 1827 922 2096">24.03.2024 - 10.04.2024</td> <td data-bbox="930 1827 1552 2096"></td> </tr> </tbody> </table>	Дата	Назва етапу	14.02.2024 - 10.03.2024	Аналіз існуючих систем пошуку транспортних засобів для перевезення вантажів	20.02.2024 - 6.03.2024	Вивчення вимог користувачів.	2.03.2024 - 20.03.2024	Опис моделі комп'ютеризованої Web-системи.	2.03.2024 - 24.03.2024	Реалізація Web-системи.	24.03.2024 - 10.04.2024		
Дата	Назва етапу													
14.02.2024 - 10.03.2024	Аналіз існуючих систем пошуку транспортних засобів для перевезення вантажів													
20.02.2024 - 6.03.2024	Вивчення вимог користувачів.													
2.03.2024 - 20.03.2024	Опис моделі комп'ютеризованої Web-системи.													
2.03.2024 - 24.03.2024	Реалізація Web-системи.													
24.03.2024 - 10.04.2024														

	<p>10.04.2024 – 20.05.2024</p> <p>15.05.2024 – 31.05.2024</p> <p>31.05.2024</p> <p>31.05.2024</p>	<p>Оцінка ефективності та порівняння з існуючими системами.</p> <p>Розробка пояснювальної записки.</p> <p>Оформлення звіту за результатами переддипломної практики.</p> <p>Представлення кваліфікаційної роботи керівнику та рецензенту.</p> <p>Оформлення пояснювальної записки та підготовка презентації.</p>
<p>8. Порядок контролю і приймання</p>	<p>1) Перевірку ходу розробки програмного виробу керівнику робіт виконувати раз в 3 тижні.</p> <p>2) Випробування програмного продукту провести відповідно до програми та методики випробувань на базі комп'ютерного класу.</p> <p>3) Захист розробленої моделі провести на засіданні Атестаційної комісії.</p> <p>4) Пояснювальну записку подати в електронному вигляді в 1 примірнику.</p>	

Виконавець

Студент групи КІ-41

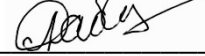
Доброродній В.Ю.



Замовник

к.т.н. доцент

Лабенко Д.П.



ДОДАТОК В

Програма і методика випробувань програмного виробу

«Модель комп'ютеризованої Web-системи пошуку транспортних засобів для перевезення вантажів»

1 Об'єкт випробувань

1.1 Найменування випробуваного програмного виробу «Модель комп'ютеризованої Web-системи пошуку транспортних засобів для перевезення вантажів».

1.2 Галузь застосування : логістика та транспортування вантажів.

1.3 Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

2. Мета випробувань

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

3. Загальні положення

3.1 Підстави для проведення випробувань

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце і тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3.3 Обсяг випробувань

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї Програми і методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програми або програмного виробу

Модель повинна задовольняти наступним вимогам:

Вимоги до функціональних характеристик:

- 1) Пошук транспорту
- 2) Фільтрація результатів
- 3) Реєстрація та авторизація користувачів

4.2. Вимоги до надійності: Забезпечення того, щоб система працювала безперервно та стабільно навіть при великому навантаженні.

4.3. Вимоги до умов експлуатації: немає

4.4. Вимоги до складу і параметрів технічних засобів: Персональний комп'ютер у повній комплектації (ноутбук)

4.5. Вимоги до інформаційної та програмної сумісності: забезпечити сумісність з усіма обчислювальними засобами.

4.6. Вимоги до маркування та упаковки: відсутні.

4.7. Вимоги до транспортування і зберігання: відсутні.

4.8. Спеціальні вимоги: відсутні. Спеціальні вимоги (не пред'являються).

5. Вимоги до програмної документації

Склад програмної документації, що подається на випробування, включає:

1) Технічне завдання на розробку програмного виробу (представлено в Додатку Б до пояснювальної записки до дипломної роботи).

2) Ця Програма і методика випробувань розробленого програмного виробу (представлена в Додатку В до пояснювальної записки до кваліфікаційної роботи).

3) Опис програмного виробу (представлено в розділі 3 пояснювальної записки до кваліфікаційної роботи).

6. Засоби і порядок випробувань

6.1 Засоби випробувань

Для проведення випробувань необхідний проєкт для розробки веб-додатку на мові програмування python з використанням фреймворку Django

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.
2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
3. Перевірку комплектності складу технічних і програмних засобів.
4. Методику проведення перевірок на 1 етапі випробувань.
5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1. перевірку відповідності технічних характеристик програми вимогам технічного завдання;
2. перевірку ступеня виконання функціональних вимог до програми;
3. методику проведення перевірок, що входять до переліку по 2 етапу випробувань.

1. Програма працює відповідно до умов експлуатації операційних систем MS Windows, Linux та MacOS.

2. Для роботи необхідний компілятор мови програмування Python, версії не нижчої ніж 3.0
3. Порядок проведення випробувань:
 1. Запуск програми здійснюється за допомогою запуску веб-серверу на мові програмування python: “python manage.py runserver”
 2. Після запуску програми необхідно у браузері комп’ютеру перейти за посиланням: <http://localhost:8000>;
 3. У вкладці браузеру з’явиться головна сторінка Web-системи для пошуку транспортних засобів для перевезення вантажів з навігацією:
 - Логотип сайту – натиснувши на нього Ви опинетесь на головній сторінці.
 - Для водіїв – містить інформацію щодо користування системою водія
 - Для відправника – містить інформацію щодо користування системою користувачам
 - Контакти – містить форму для зв’язку з техпідтримкою
 - Про нас – містить інформацію про сайт у цілому
 - Вхід – після натискання перенаправляє користувача сторінку “Вхід” щоб увійти у особистий кабінет.
 - Реєстрація - після натискання перенаправляє користувача сторінку “Реєстрація” щоб користувач міг зареєструвати власний особистий кабінет

Для проведення випробувань пропонується тест 1, тест 2 та тест 3.

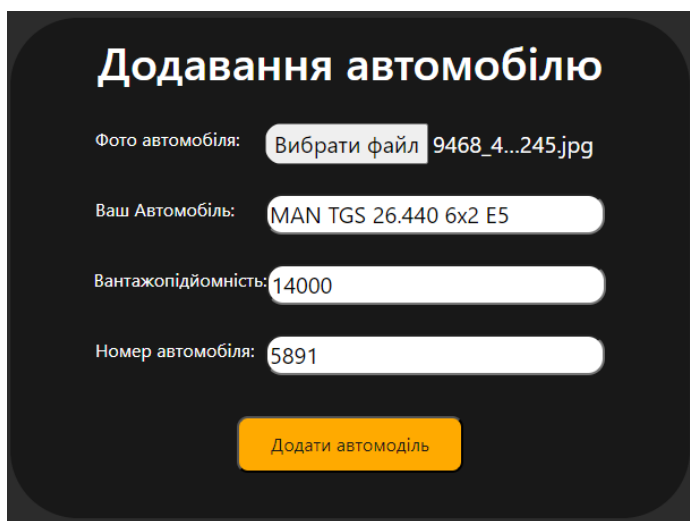
Тест 1: Додавання авто та Створення публікації

Тести для створення замовлення повинні охоплювати:

Успішне додавання авто:

- ❖ Заповнення всіх необхідних полів (фото, модель та марка авто, вантажопідйомність та номер).

- ❖ Переконайтеся, що замовлення успішно створюється і з'являється в списку замовлень.
- ❖ Заповнення всіх необхідних полів (міста відправки та доставки, обрати авто зі списку та вказати ціну за послугу).
- ❖ Переконайтеся, що замовлення успішно створюється і з'являється в списку замовлень.



Додавання автомобілю

Фото автомобіля: Вибрати файл 9468_4...245.jpg

Ваш Автомобіль: MAN TGS 26.440 6x2 E5

Вантажопідйомність: 14000

Номер автомобіля: 5891

Додати автомобіль

Рис В.1 – Додавання автомобілю



Рис В.2 – Успішне додавання нового авто

Рис В.3 – Створення публікації авто

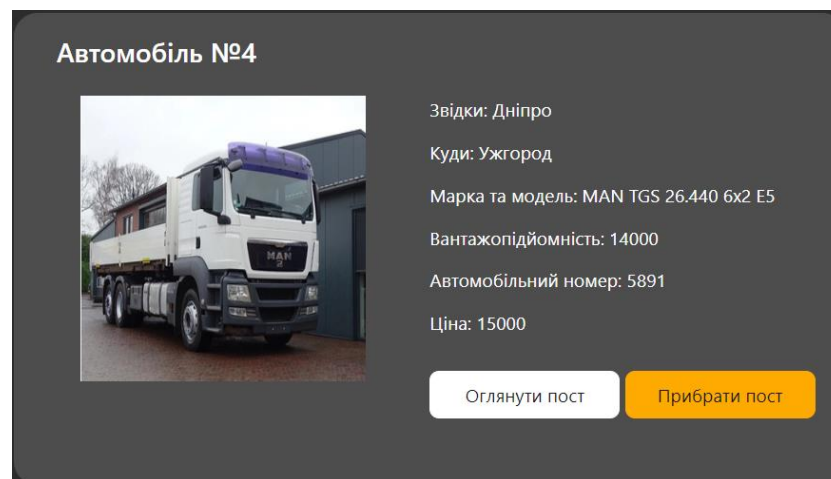


Рис В.4 – Успішне створення публікації авто

Тест 2: Вивід усіх публікацій клієнту та прийняття публікації

Для цього функціоналу потрібно перевірити:

Відображення замовлень:

- ❖ Перевірка, що клієнт може бачити список всіх доступних автівок.
- ❖ Переконатися, що інформація про кожне замовлення відображається коректно (опис вантажу, міста відправки та доставки, орієнтовна вартість).
- ❖ Перевірити сортування замовлень.
- ❖ Вибір публікації зі списку і підтвердження його прийняття.
- ❖ Перевірка кнопки «Виконано»

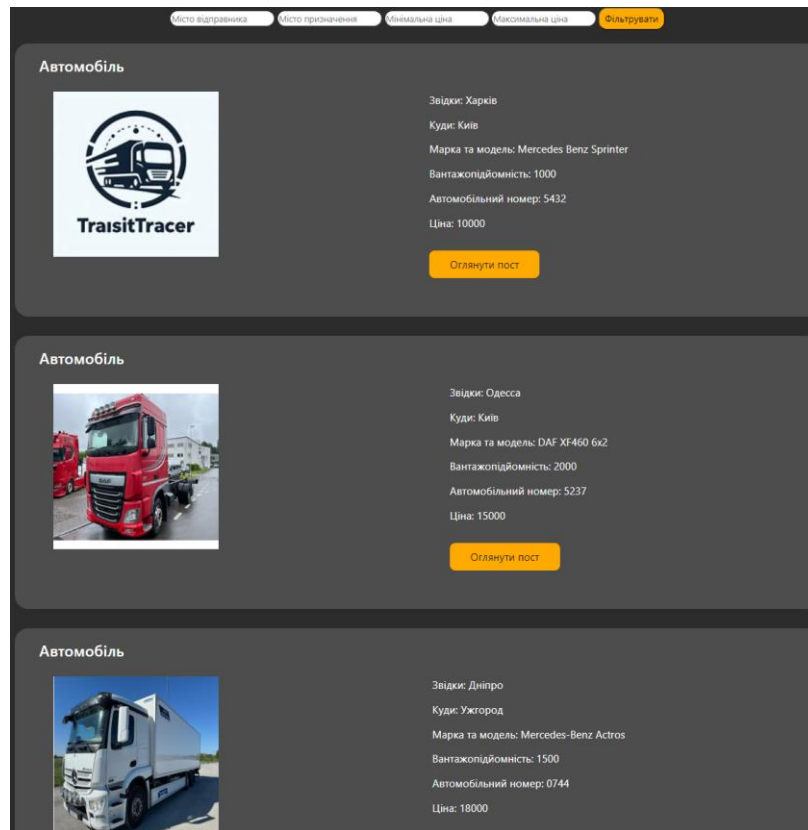


Рис В.5 – Вивід усіх доступних автівок клієнту

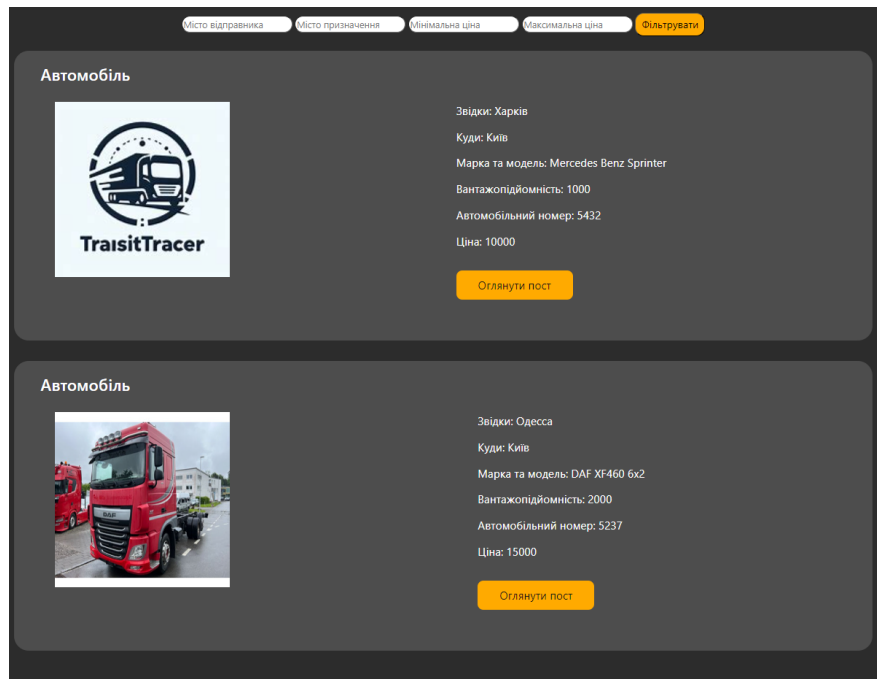


Рис В.6 – Успішна фільтрація за містом доставки

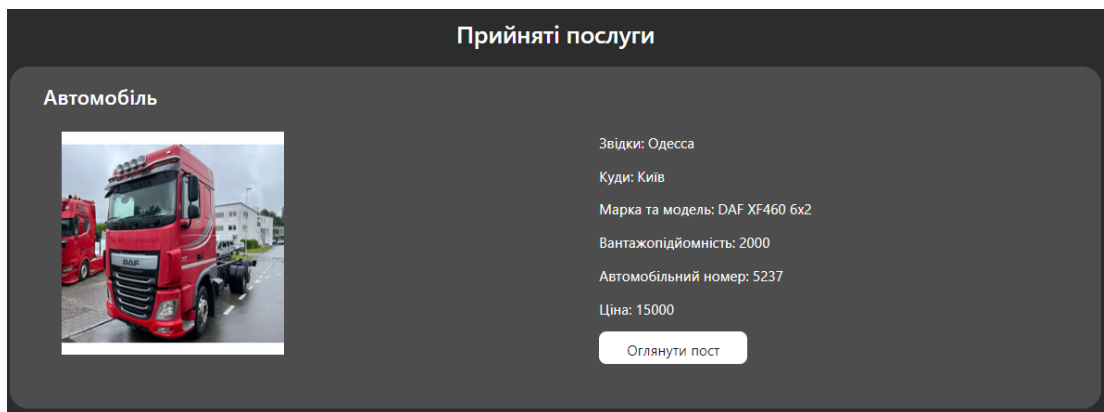


Рис В.7 – Успішне прийняття транспорту

id	fromwe	where	price	driver_pk	customer	driver	complited	car_pk_id
Філ...	Фільтр	Фільтр	Філ...	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	Харків	Київ	10000	1	NULL	1	0	1
2	Одеса	Київ	15000	1	2	1	1	2

Рис В.8 – Успішний запис що послуга виконана

Тест 3:

1. Перевірка виводу усіх замовлень на сайті Водіям
2. Перевірка сортування замовлень
3. Перевірка отримання замовлення водієм
4. Перевірка запису у базу даних

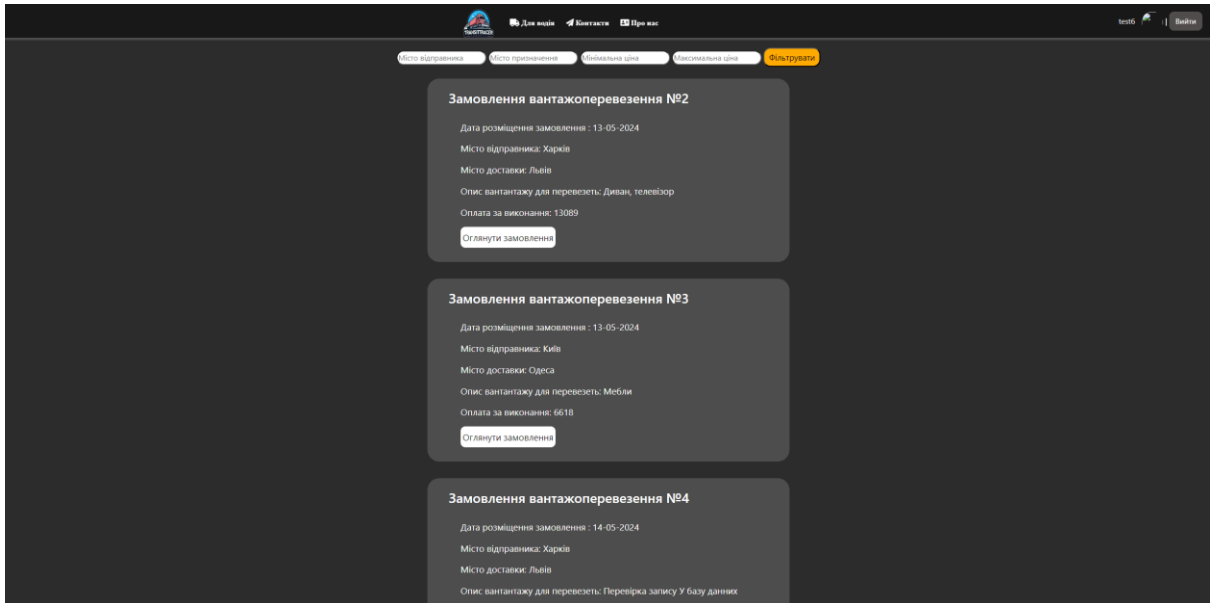


Рисунок В.9 – сторінка усіх замовлень доступних водію

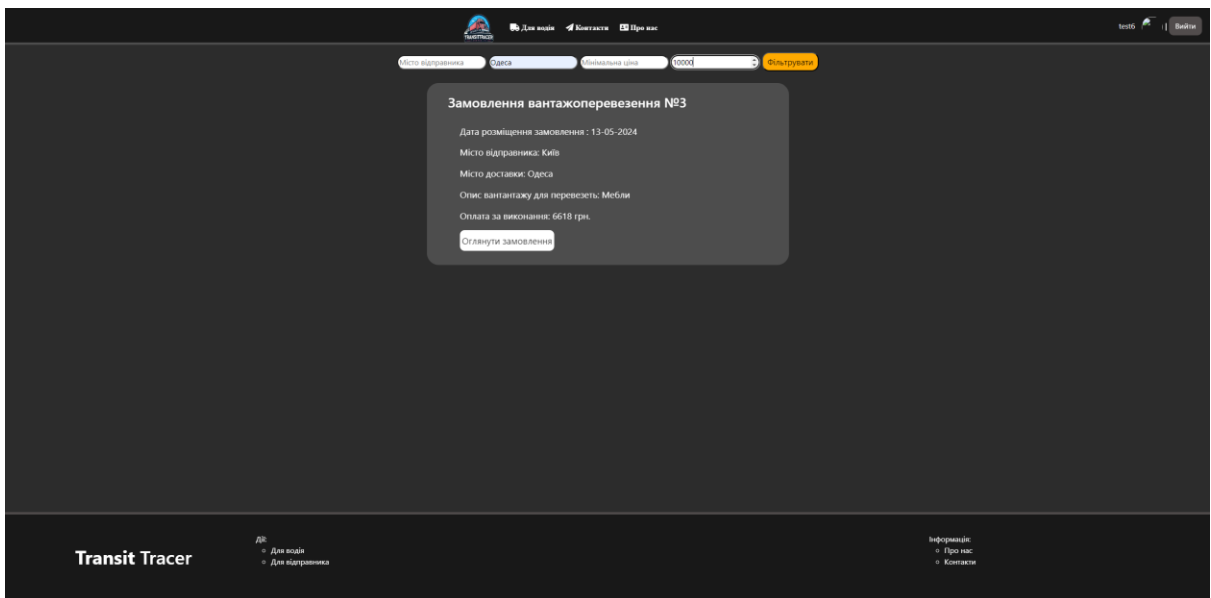


Рисунок В.10 – перевірка сортування замовлень

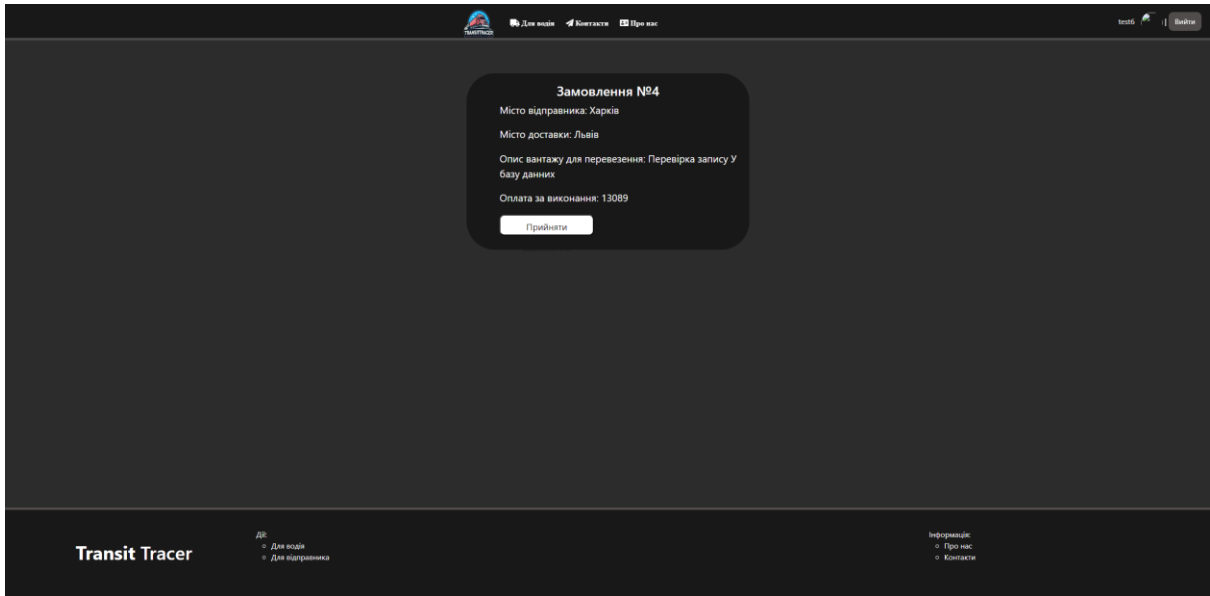


Рисунок В.11 – сторінка Обраного замовлення

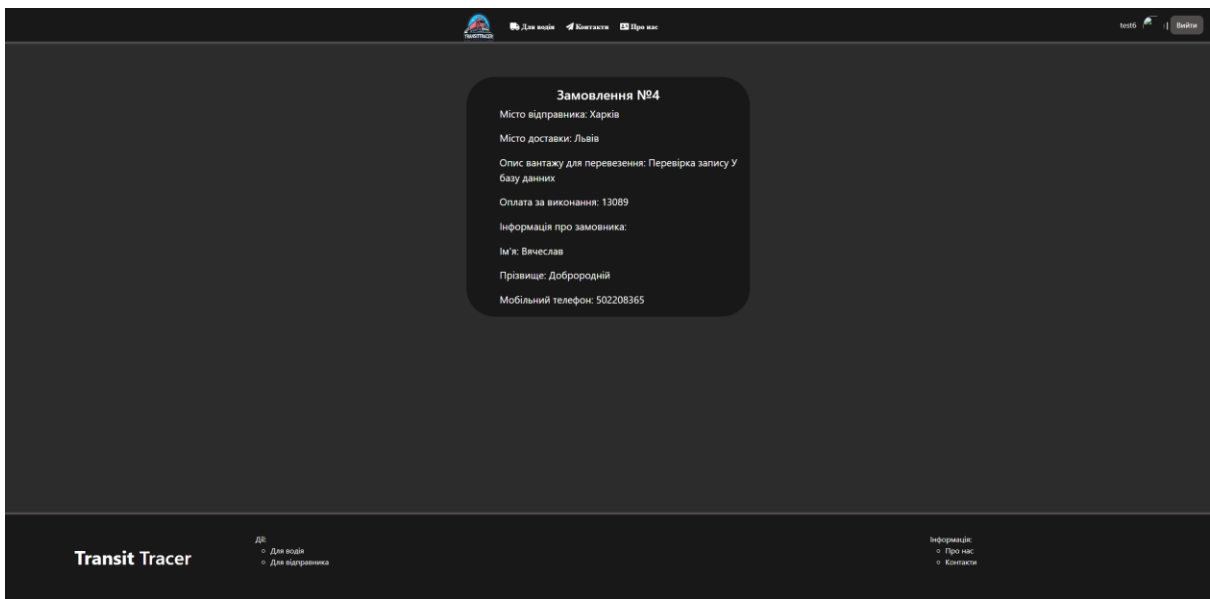


Рисунок В.12 – водій натиснув на кнопку прийняти

id	date	fromw	where	cargo	price	customer	driver	completed
Філ...	Фільтр	Фільтр	Фільтр	Фільтр	Філ...	Фільтр	Фільтр	Фільтр
1	1	2024-05-12	Харків	Львів	qwewqwr	13089	1	NULL 1
2	2	2024-05-13	Харків	Львів	Диван, телевізор	13089	1	2 0
3	3	2024-05-13	Київ	Одеса	Меблі	6618	1	2 0
4	4	2024-05-14	Харків	Львів	Перевірка запису У базу даних	13089	1	2 0

Рисунок В.13 – перевірка запису у базі даних що водій прийняв замовлення

Висновок: Перевірки на вивід усіх замовлень та публікацій, сортування замовлень та публікацій також запис водія за вибраним їм замовлення пройшли успішно

Виконавець: студент групи КІ-41, Доброродній В.Ю. __



__