

Полтавський університет економіки і торгівлі

Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)

«__» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

«ВЕБ-СИСТЕМА ДЛЯ УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПРОЦЕСАМИ З АЛГОРИТМАМИ ОПТИМІЗАЦІЇ»

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавра

Виконавець роботи Зануда Єсенія Вадимівна
_____ «__» _____ 202_ р.
(підпис)

Науковий керівник к. ф.-м. н., доцент, Ольховська Олена Володимирівна
_____ «__» _____ 202_ р.
(підпис)

Рецензент _____

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	5
1. ПОСТАНОВКА ЗАДАЧ	8
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	10
2.1. Огляд веб-системи Lardi-Trans для управління логістичними перевезеннями	10
2.2. Огляд веб-системи Ant Logistics для управління логістичними процесами 12	
3. ТЕОРЕТИЧНА ЧАСТИНА	17
3.1. Архітектурні підходи до побудови веб-систем: клієнт-серверна взаємодія 17	
3.2. Вибір інструментів для реалізації системи: мови програмування, фреймворки, бази даних	18
3.3. Проектування бази даних для логістичної системи	19
3.4. Теоретичні основи алгоритмів оптимізації маршрутів	21
3.5. Загальна UML діаграма роботи веб-системи	23
4. ПРАКТИЧНА ЧАСТИНА	24
4.1 . Обґрунтування вибору програмних засобів	24
4.2. Опис програмної реалізації	26
4.3. Опис роботи веб-системи для управління логістичними процесами з алгоритмами оптимізації	30
ВИСНОВКИ	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТОК А. КОД ПРОГРАМИ	44

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
API	Application Programming Interface — інтерфейс взаємодії між програмними компонентами
REST API	Архітектурний стиль взаємодії між клієнтом та сервером через HTTP-запити
JSON	JavaScript Object Notation — формат обміну даними
UI	User Interface — користувацький інтерфейс
UX	User Experience — взаємодія користувача із системою
SPA	Single Page Application — односторінковий веб-застосунок
JWT	JSON Web Token — токен авторизації користувача
ORM	Object-Relational Mapping — технологія взаємодії з базою даних через об'єкти
SQL	Structured Query Language — мова запитів до бази даних
PostgreSQL	Реляційна система керування базами даних
Sequelize	ORM-бібліотека для роботи з PostgreSQL у Node.js
React	JavaScript-бібліотека для створення користувацьких інтерфейсів
Node.js	Серверне середовище виконання JavaScript
Express.js	Фреймворк для створення серверної частини веб-застосунків
Axios	Бібліотека для виконання HTTP-запитів
React Router	Бібліотека для навігації між сторінками React-застосунку
Vite	Інструмент для запуску та збірки frontend-проектів
OpenStreetMap	Відкритий картографічний сервіс
Leaflet	JavaScript-бібліотека для роботи з інтерактивними картами
UML	Unified Modeling Language — мова моделювання програмних систем
GPS	Global Positioning System — система супутникової навігації
Dashboard	Головна панель керування веб-системи
Frontend	Клієнтська частина веб-системи
Backend	Серверна частина веб-системи

HTTP	HyperText Transfer Protocol — протокол передачі даних
Validation	Перевірка правильності введення даних
Route Optimization	Оптимізація маршрутів доставки

ВСТУП

У сучасних умовах розвитку цифрових технологій логістика є однією з найдинамічних сфер, де ефективність управління процесами безпосередньо впливає на прибутковість підприємств. Зі збільшенням обсягів товарообігу, кількості транспортних маршрутів і складських операцій виникає потреба у впровадженні автоматизованих веб-систем, здатних оптимізувати логістичні процеси та підвищити точність прийняття управлінських рішень. Використання алгоритмів оптимізації у таких системах дозволяє мінімізувати витрати часу та ресурсів, забезпечити раціональний розподіл вантажів, скоротити простої транспорту та підвищити якість обслуговування клієнтів [17]. Саме тому розробка та впровадження веб-системи для управління логістичними процесами з алгоритмами оптимізації є актуальним завданням сучасної комп'ютерної науки та економічної інформатики.

Перевагами розробленої веб-системи є:

- автоматизація процесів — система дозволяє скоротити вплив людського фактора під час планування маршрутів, обробки замовлень і контролю поставок;
- оптимізація ресурсів — використані алгоритми оптимізації сприяють ефективному розподілу транспорту, скороченню витрат пального та часу перевезень;
- прозорість і контроль — користувачі мають доступ до актуальної інформації про переміщення вантажів та графіки доставки у режимі реального часу;
- гнучкість і масштабованість — веб-платформа забезпечує можливість адаптації системи до різних типів підприємств і сценаріїв логістичних процесів.

Метою роботи є розробка та реалізація веб-системи для управління логістичними процесами з використанням алгоритмів оптимізації, яка

забезпечує ефективне планування маршрутів, управління транспортом і контроль руху товарів.

Об'єкт дослідження — процес управління логістичними операціями підприємства, зокрема планування маршрутів та контроль виконання транспортних завдань.

Предмет дослідження — веб-система, що реалізує функціональні модулі управління логістичними процесами з інтегрованими алгоритмами оптимізації.

У процесі розробки проекту були використані такі методи:

- аналітичний метод — для вивчення існуючих веб-систем, аналізу їхніх функцій і недоліків;
- системний підхід — для формування архітектури веб-системи та визначення взаємодії між її компонентами;
- об'єктно-орієнтоване програмування — для реалізації серверної та клієнтської логіки системи на основі модульного принципу;
- метод моделювання — для побудови UML-діаграм моделювання бізнес-процесів логістики;
- алгоритмічні методи оптимізації — для розрахунку найкоротших маршрутів та мінімізації транспортних витрат;
- експериментальний метод — для тестування працездатності системи, перевірки швидкості дії алгоритмів і точності результатів оптимізації.

Результатом роботи стала розроблена веб-система управління логістичними процесами, яка забезпечує ефективне планування маршрутів, контроль перевезень, управління транспортом та моніторинг доставки товарів. Система дозволяє оптимізувати логістичні процеси підприємства, зменшити витрати ресурсів і підвищити продуктивність роботи.

Розроблений проект демонструє ефективність використання сучасних веб-технологій і алгоритмів оптимізації для вдосконалення

систем управління логістикою та розвитку цифрової інфраструктури підприємств.

Результати роботи апробовані на XLVIX Міжнародної наукової конференції студентів та аспірантів «Актуальні питання розвитку науки та забезпечення якості освіти у XXI столітті», за підсумками науково-дослідних робіт студентів за 2025 рік.

Кваліфікаційна робота складається з чотирьох розділів [1]. У першому розділі сформульовано постановку задачі, що полягає у створенні веб-системи для управління логістичними процесами з використанням алгоритмів оптимізації маршрутів. У другому розділі проведено аналіз сучасних логістичних веб-систем та картографічних сервісів, а також розглянуто функціональні можливості аналогічних платформ для управління доставками та маршрутами. У третьому розділі описано теоретичні основи побудови веб-систем, принципи клієнт-серверної взаємодії, структуру бази даних та алгоритми оптимізації маршрутів. Також наведено UML-діаграму роботи системи. У четвертому розділі висвітлено процес практичної реалізації веб-системи, описано функціональні можливості користувачів, роботу алгоритмів оптимізації, інтеграцію картографічних сервісів та логіку функціонування системи управління логістичними процесами.

Обсяг пояснювальної записки: 44 стор., в т.ч. основна частина – 42 стор., джерела – 21 назва.

1. ПОСТАНОВКА ЗАДАЧ

Метою даної роботи є розробка веб-системи для управління логістичними процесами, призначеної для оптимізації маршрутів перевезень, планування транспортних операцій та контролю виконання поставок. Основна мета системи — забезпечити ефективне управління логістичними потоками за допомогою алгоритмів оптимізації та сучасних веб-технологій.

Визначені основні завдання проекту включають:

1. Розробку інформаційної бази даних, яка зберігатиме відомості про транспортні засоби, маршрути, вантажі та клієнтів.
2. Вибір архітектури та технологій розробки веб-системи, зокрема використання сучасного стеку.
3. Проектування користувацького інтерфейсу, що забезпечить інтуїтивну взаємодію з системою та зручне відображення логістичних даних.
4. Реалізацію функціоналу управління маршрутами, включаючи створення, редагування та оптимізацію перевезень на основі заданих параметрів (відстань, час, витрати).
5. Інтеграцію алгоритмів оптимізації, які дозволять автоматично знаходити найкоротші або оптимальні шляхи транспортування.
6. Розробку модулів аналітики, що надаватимуть звіти про ефективність маршрутів, завантаження транспорту та загальні витрати.
7. Забезпечення безпеки та цілісності даних під час обміну інформацією між користувачами та сервером.

Основна увага в роботі приділяється створенню зручного, ефективного та надійного інструменту для управління логістичними процесами, який дозволить підвищити продуктивність роботи підприємств і зменшити витрати.

Реалізація поставлених завдань забезпечила створення сучасної веб-системи, що підвищує ефективність логістичних операцій, забезпечує раціональний розподіл ресурсів і зменшує витрати підприємств завдяки впровадженню алгоритмів оптимізації.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд веб-системи Lardi-Trans для управління логістичними перевезеннями

Веб-система Lardi-Trans для управління логістикою Lardi-Trans — це міжнародна веб-платформа для логістики вантажоперевезень, призначена для організації перевезень вантажів, пошуку транспорту та взаємодії між перевізниками, експедиторами та замовниками [3]. Система використовується для автоматизації логістичних процесів, контролю доставки вантажів та пошуку оптимальних рішень у сфері перевезень. Щоб почати роботу на цій платформі, користувачеві потрібно відвідати офіційний сайт і зареєструватися. Головна сторінка веб-системи містить меню для пошуку вантажів, транспорту, інформації про перевізників та інструменти взаємодії для учасників сайту. (рис. 2.1).

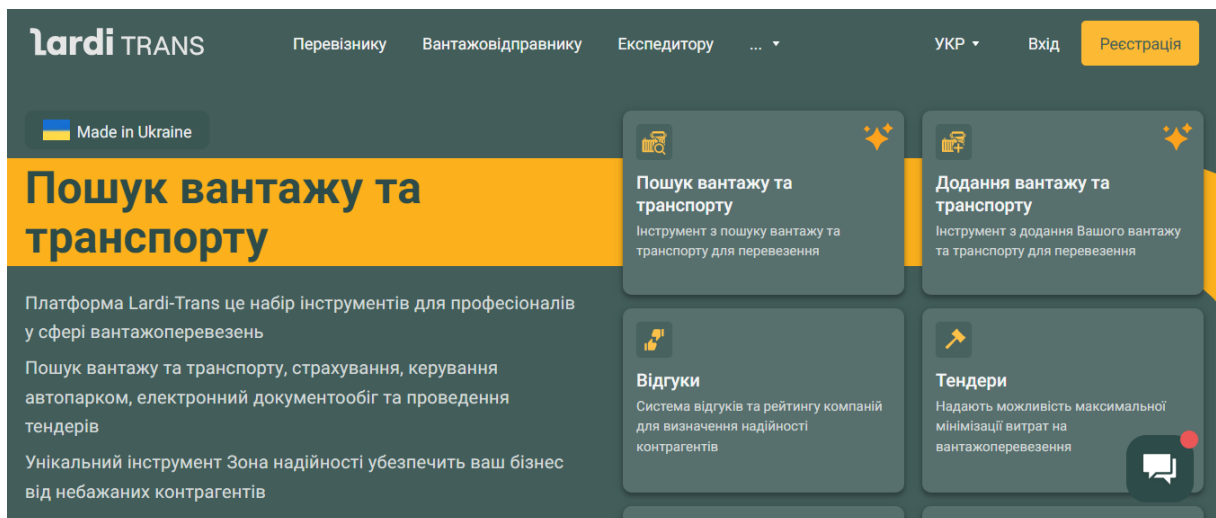


Рисунок 2.1 – Головний інтерфейс веб-системи Lardi-Trans

Однією з основних функцій системи є пошук вантажів та перевезень. Користувач може переглядати доступні заявки, вибирати маршрути перевезень, тип транспорту та аналізувати умови доставки. (рис. 2.2).

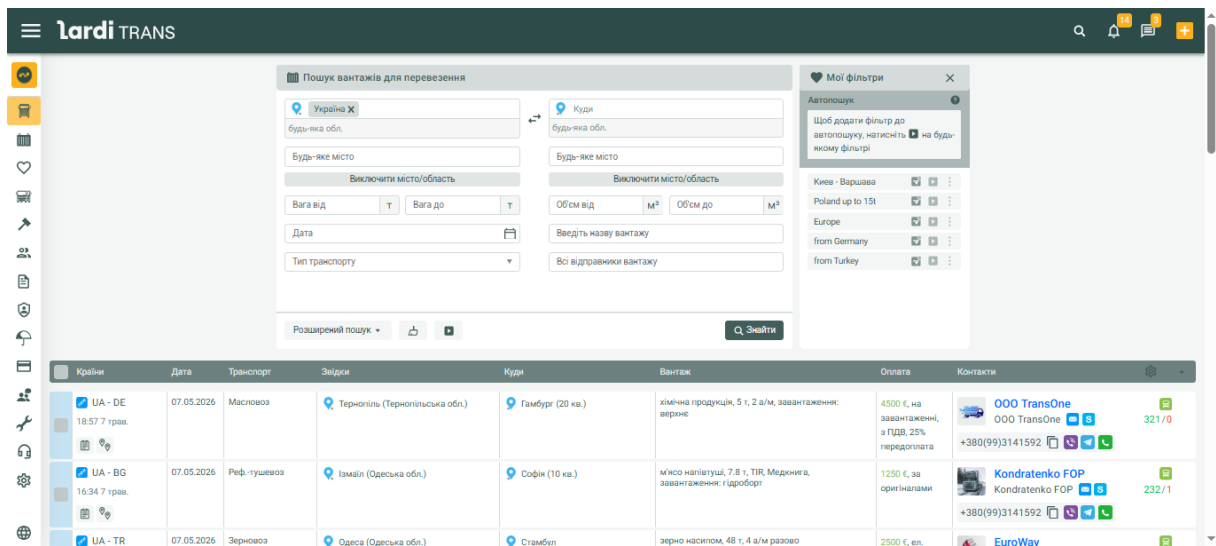


Рисунок 2.2 – Пошук вантажів і транспорту

Для перевезення вантажів система дозволяє користувачеві створювати заявки на перевезення вантажів. Користувач у цій системі може вказати маршрут, характеристики вантажу, дату доставки та інші параметри. (рис. 2.3).

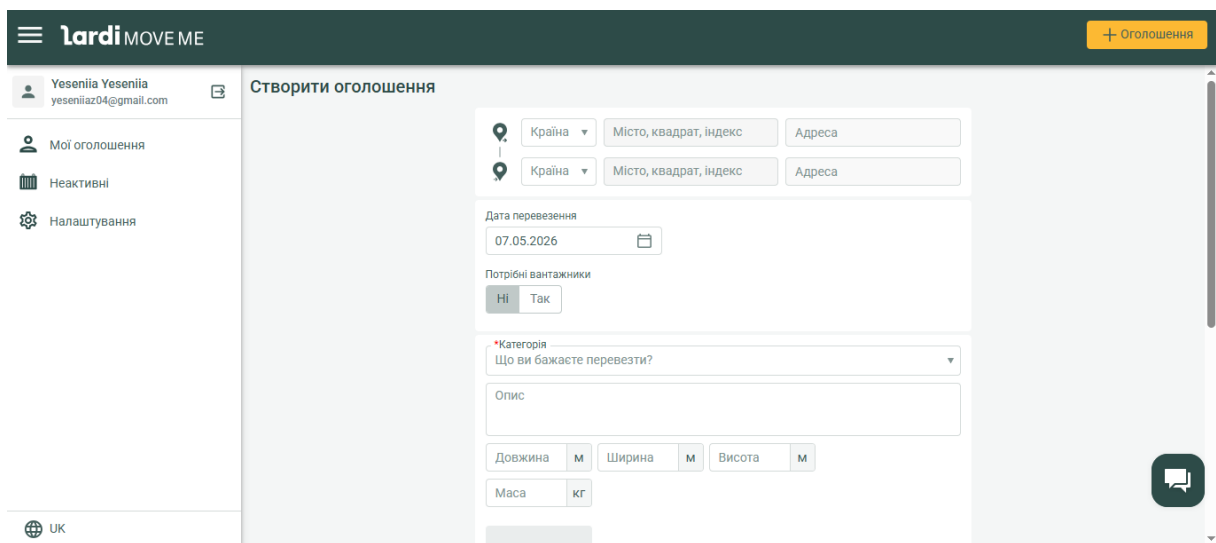


Рисунок 2.3 – Створення заявки на перевезення

Веб-система також надає інформацію про перевізників, рейтинги користувачів та відгуки, що додає безпеки співпраці між сторонами. (рис. 2.4).

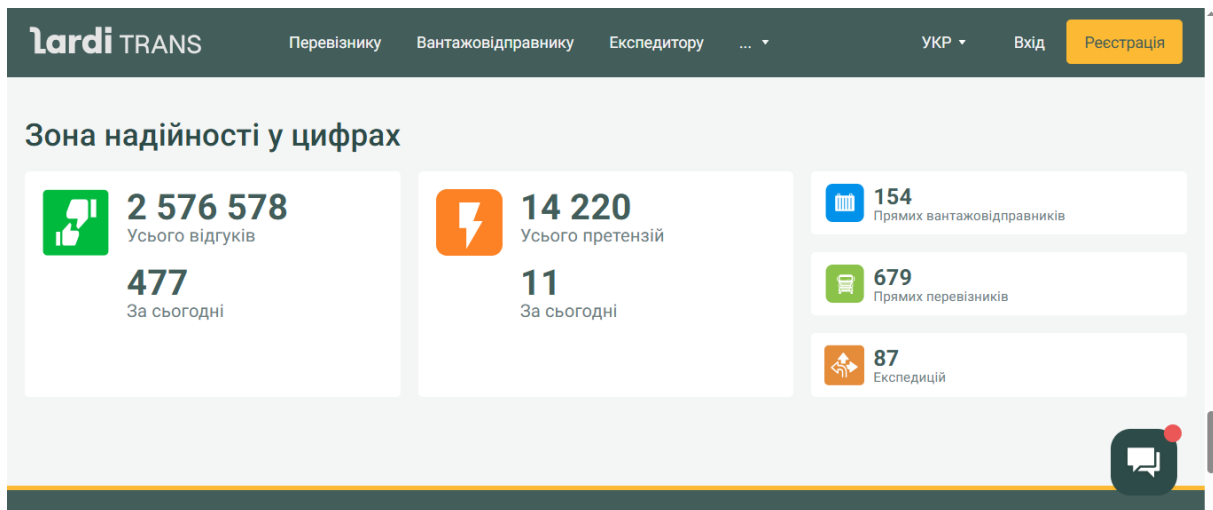


Рисунок 2.4 – Інформація про перевізників та рейтинги

Перевагами системи є велика база даних вантажів та транспортних засобів. Можливість швидкого пошуку перевізників та замовників. Наявність системи рейтингів та відгуків. Доступ до логістичної інформації в реальному часі. Підтримка міжнародних перевезень. Серед недоліків можна виділити відсутність сучасних алгоритмів для автоматичної оптимізації маршрутів. Перевантажений інтерфейс для нових користувачів.

Деякі функції доступні лише після платної підписки. Обмежені можливості аналітики та автоматизації логістичних процесів. Недостатня візуалізація маршрутів та транспортних операцій.

Таким чином, веб-система Lardi-Trans є популярним інструментом для організації вантажоперевезень та пошуку логістичних рішень. Однак система має певні недоліки, пов'язані з обмеженою автоматизацією та недостатнім використанням алгоритмів оптимізації маршрутів.

2.2. Огляд веб-системи Ant Logistics для управління логістичними процесами

Веб-система Ant Logistics для управління логістичними процесами Ant Logistics — це веб-система для автоматизації логістичних процесів, контролю перевезень та планування маршрутів доставки [4]. Система орієнтована на підприємства, що займаються перевезенням вантажів та

логістичними операціями. Щоб використовувати систему, потрібно увійти в веб-середовище. Після входу користувач отримує доступ до головної панелі управління, яка відображає маршрути, транспортні засоби, замовлення та статистику перевезень. (рис. 2.6).

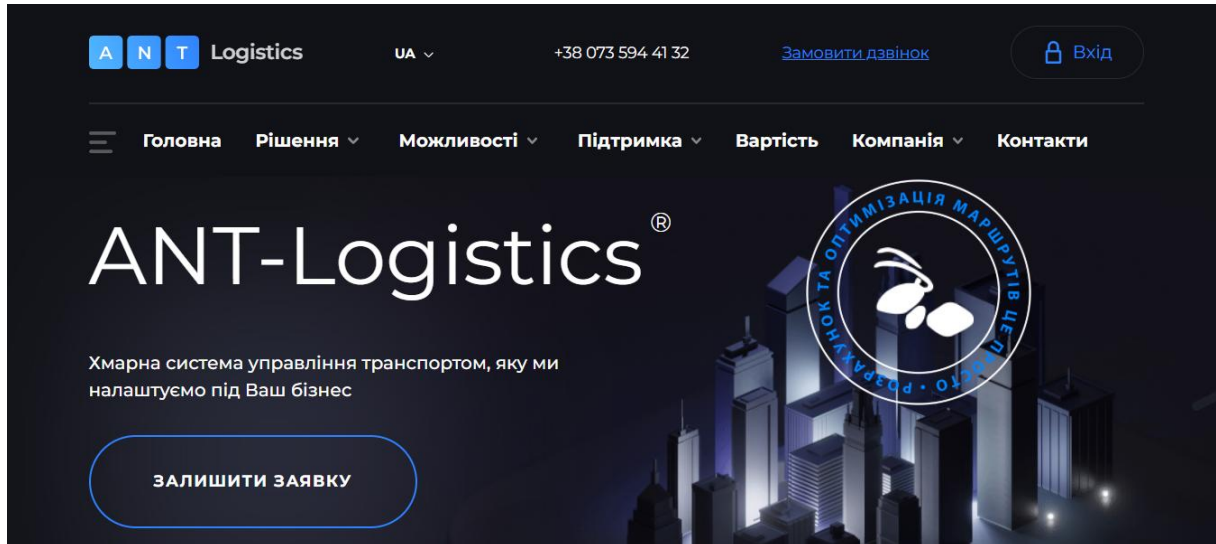


Рисунок 2.6 – Головна панель веб-системи Ant Logistics

Однією з найважливіших функцій системи є автоматизоване планування маршрутів доставки. Користувачі мають можливість визначати маршрути, вказувати точки доставки та контролювати виконання перевезень. (рис. 2.7).

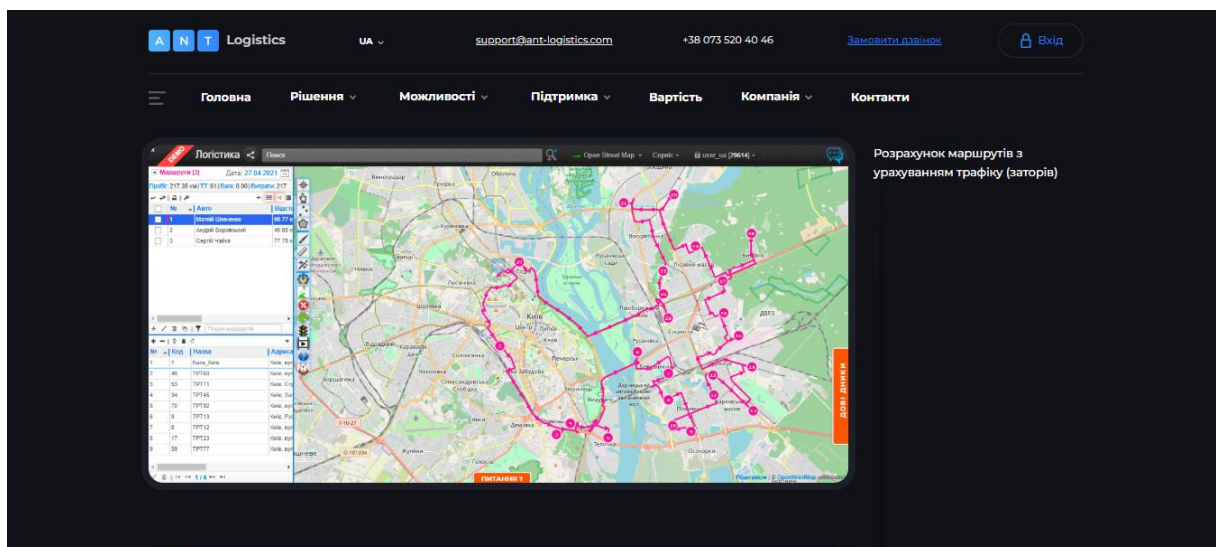


Рисунок 2.7 – Планування маршрутів доставки

Система дозволяє вводити додаткові параметри маршрутизації для полегшення внутрішньої організації та логістичних процесів відповідно до вимог підприємства. Користувач може враховувати тривалість робочого дня, середню швидкість транспорту, дорожній трафік та інші параметри при налаштуванні маршрутизації для доставки. (рис. 2.8).

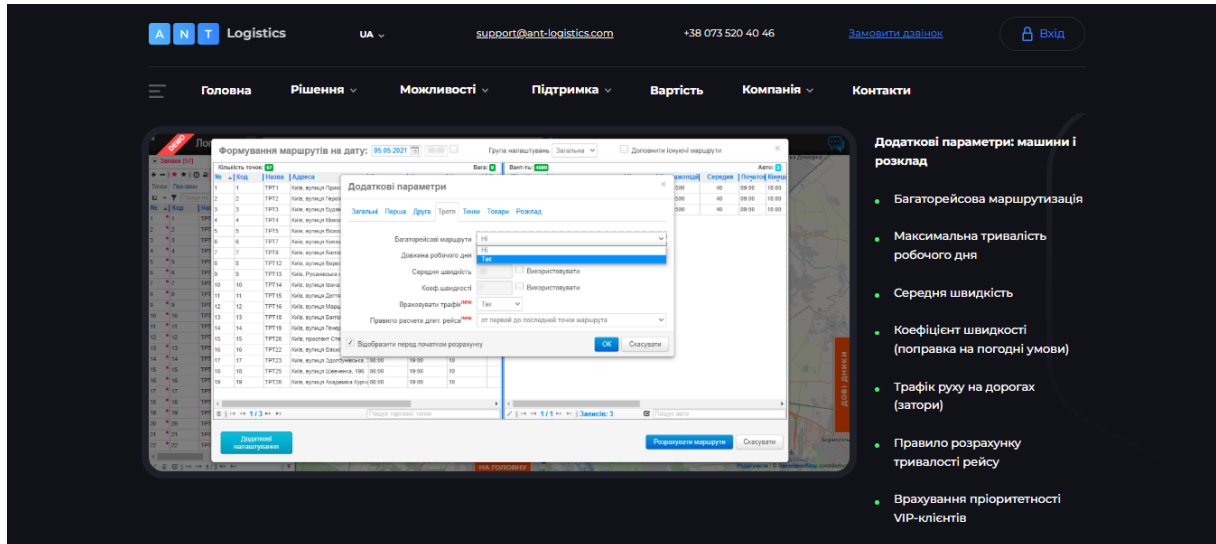


Рисунок 2.8 – Налаштування параметрів маршрутизації

Система підтримує формування аналітичних звітів про ефективність доставки, витрати на паливо та продуктивність логістичних операцій. (рис. 2.9).

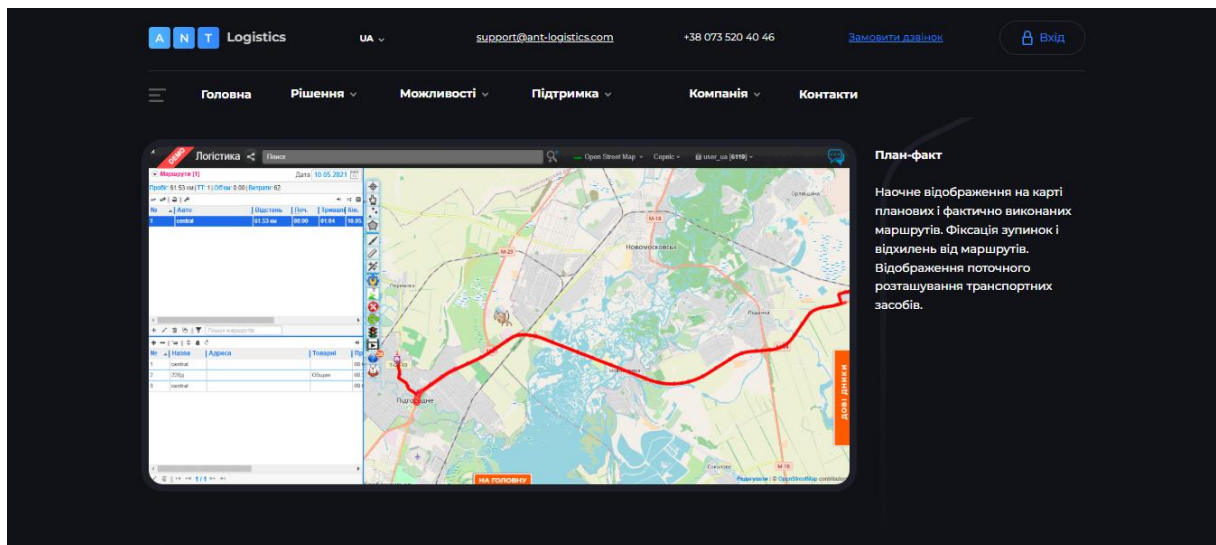


Рисунок 2.9 – Аналітичні звіти логістичної системи

Рисунок 2.11 – Управління замовленнями та доставкою

Переваги системи включають автоматизацію процесів планування доставки, що значно спрощує організацію логістичних процесів та підвищує ефективність роботи. Також включає моніторинг транспорту в реальному часі, що контролює рух транспортних засобів у межах маршруту та дозволяє вчасно реагувати на зміни в маршрутах. Ще однією значною перевагою є можливість генерувати аналітичні звіти та статистику, що спрощує аналіз роботи логістичної системи. Веб-інтерфейс полегшує управління логістичною діяльністю, а підтримка управління транспортом та замовленнями забезпечує централізовану співпрацю з основними процесами системи.

Деякі недоліки цієї платформи включають те, що деякі функції доступні лише у платній версії системи. Кількість функцій та параметрів робить її складною для малих підприємств, тому налаштування та використання системи можуть бути досить складними. Крім того, велика кількість опцій іноді може ускладнювати роботу користувачів із системою. Система працює через GPS та зовнішні служби моніторингу і може мати обмежену адаптивність до індивідуальних потреб користувача.

Таким чином, веб-система Ant Logistics є сучасним інструментом для автоматизації логістичних процесів та управління доставкою. Водночас система має певні недоліки, пов'язані зі складністю використання та обмеженням функцій у безкоштовній версії, що створює потребу в розробці більш гнучких та доступних веб-рішень.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Архітектурні підходи до побудови веб-систем: клієнт-серверна взаємодія

Сучасні веб-системи управління логістикою будуються на основі клієнт-серверної архітектури, яка дозволяє розділити інтерфейс користувача, серверну логіку та роботу з базою даних [21]. Такий підхід забезпечує стабільність роботи системи, зручність підтримки та можливість масштабування.

У розробленому проєкті використовується розділення на frontend та backend частини.

Frontend частина реалізована за допомогою React та відповідає за:

- відображення інтерфейсу;
- взаємодію користувача із системою;
- надсилання HTTP-запитів;
- отримання та відображення даних.

Backend частина реалізована на Node.js та Express.js і забезпечує:

- обробку запитів;
- логіку роботи системи;
- авторизацію користувачів;
- взаємодію з базою даних.

Передача даних між клієнтом та сервером здійснюється через REST API у форматі JSON. Для виконання HTTP-запитів використовується бібліотека Axios.

Принцип роботи системи:

1. Користувач виконує дію в інтерфейсі.
2. Frontend надсилає HTTP-запит на сервер.
3. Сервер обробляє запит.
4. Backend взаємодіє з базою даних.

5. Сервер повертає результат у форматі JSON.

6. Frontend оновлює інформацію на сторінці.

Клієнт-серверна архітектура дозволяє забезпечити незалежність між інтерфейсом користувача та серверною частиною системи, що спрощує розробку та підтримку веб-застосунку.

3.2. Вибір інструментів для реалізації системи: мови програмування, фреймворки, бази даних

Для реалізації веб-системи управління логістикою було обрано сучасний стек технологій, який забезпечує швидкість розробки, гнучкість та зручність підтримки.

Frontend-частина

Для створення інтерфейсу користувача використовуються:

- React;
- Vite;
- Axios;
- React Router.

React використовується для створення динамічного інтерфейсу користувача. Vite забезпечує швидку збірку та запуск проекту. Axios використовується для виконання HTTP-запитів до серверної частини, а React Router відповідає за навігацію між сторінками [7].

Backend-частина

Серверна частина реалізована за допомогою:

- Node.js;
- Express.js.

Node.js дозволяє створювати серверні застосунки на JavaScript, а Express.js використовується для організації REST API та маршрутизації [8, 9].

База даних

Для зберігання інформації використовується PostgreSQL.

Переваги PostgreSQL:

- висока продуктивність;
- підтримка SQL-запитів;
- надійність;
- підтримка зв'язків між таблицями;
- масштабованість.

ORM Sequelize

Для взаємодії з базою даних використовується Sequelize.

Sequelize дозволяє:

- створювати моделі таблиць;
- виконувати SQL-запити через JavaScript;
- налаштовувати зв'язки між таблицями;
- спрощувати роботу з базою даних.

Авторизація та ролі користувачів

У системі реалізована рольова модель доступу користувачів. Кожна роль має власний набір функцій та доступ до окремих модулів системи.

Основні ролі системи:

- адміністратор;
- менеджер;
- водій;
- клієнт.

Розмежування прав доступу дозволяє забезпечити безпеку системи та обмежити доступ користувачів до службових функцій.

3.3. Проектування бази даних для логістичної системи

База даних забезпечує зберігання інформації про користувачів, замовлення, маршрути, транспорт та доставки.

Основними таблицями системи є:

- Users;
- Orders;
- Vehicles;
- Routes;
- Deliveries.

Таблиця Users

Таблиця користувачів містить:

- ім'я користувача;
- email;
- пароль;
- роль користувача.

Таблиця Orders

Таблиця замовлень містить:

- інформацію про доставку;
- статус замовлення;
- дату створення;
- інформацію про клієнта.

Таблиця Vehicles

Таблиця транспорту містить:

- тип транспорту;
- номер транспортного засобу;
- статус;
- вантажопідйомність.

Таблиця Routes

Таблиця маршрутів містить:

- початкову точку;
- кінцеву точку;
- відстань;
- параметри маршруту.

Таблиця Deliveries

Таблиця доставок містить:

- статус доставки;
- інформацію про транспорт;
- прив'язку до маршруту.

Між таблицями реалізовані зв'язки для забезпечення цілісності даних та коректної роботи системи.

3.4. Теоретичні основи алгоритмів оптимізації маршрутів

Оптимізація маршрутів є важливою задачею логістичних систем. У проекті реалізовано алгоритм пошуку оптимального маршруту між точками доставки [16].

Алгоритм аналізує точки маршруту та формує найкоротший шлях між ними.

У системі також реалізовано калькулятор логістичних розрахунків, який дозволяє визначати:

- відстань між точками;
- приблизний час доставки;
- витрати пального;
- орієнтовну вартість маршруту.

Для виконання розрахунків використовуються математичні формули, які враховують довжину маршруту, середню швидкість транспорту та витрати пального.

Розрахунок часу доставки виконується за формулою:

$$t = \frac{s}{v}$$

де:

t — час;

s — відстань;

v — швидкість.

Розрахунок витрат пального:

$$F = \frac{S * R}{100}$$

де:

F — витрати пального;

S — відстань;

R — витрати пального на 100 км.

Такі розрахунки дозволяють автоматизувати логістичні процеси та спрощують планування доставки.

3.5. Загальна UML діаграма роботи веб-системи

Загальна UML діаграма роботи веб-системи для управління логістичними процесами (рис. 3.1) [19].

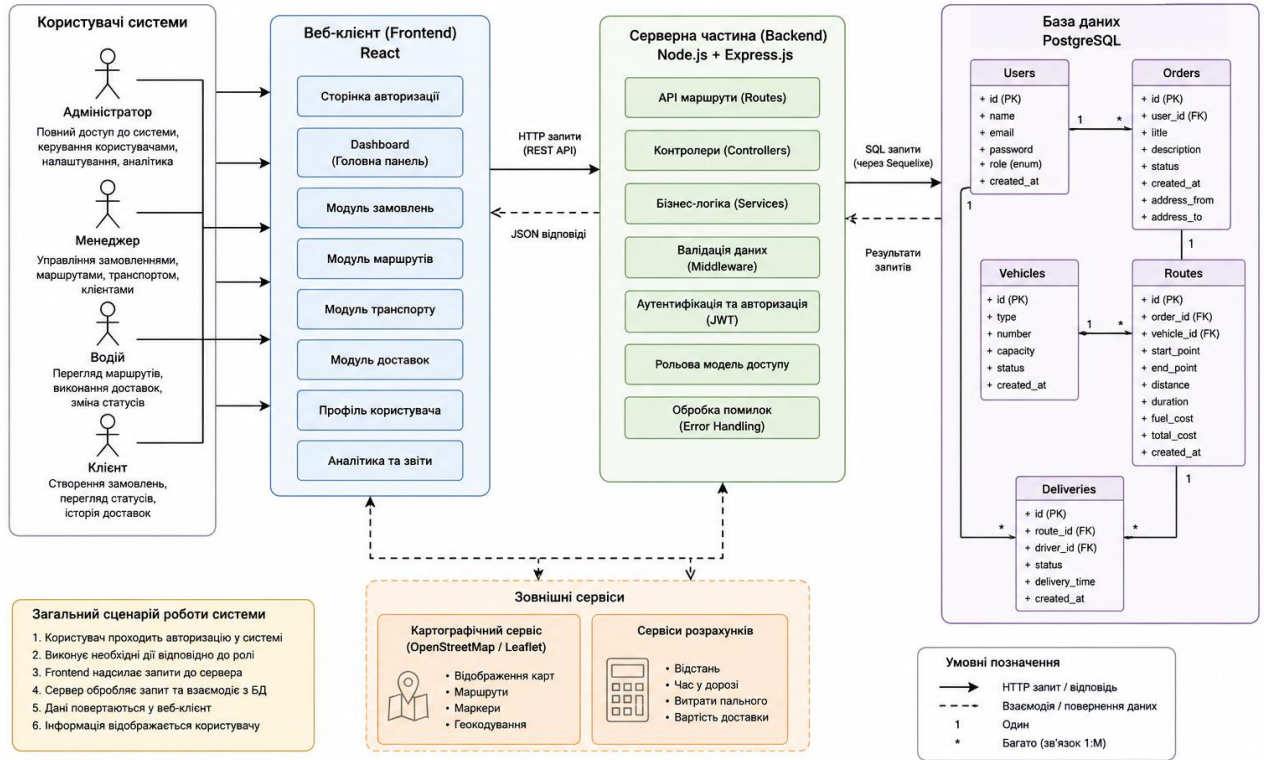


Рисунок 3.1 – загальна UML діаграма

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

Для реалізації веб-системи управління логістичними процесами було обрано сучасний стек веб-технологій, який забезпечує стабільну роботу системи, високу продуктивність, зручність підтримки та можливість подальшого масштабування проекту. Під час вибору програмних засобів враховувались швидкість розробки, продуктивність системи, підтримка сучасних веб-технологій, можливість інтеграції між компонентами системи, підтримка роботи з базами даних та можливість реалізації інтерактивного інтерфейсу користувача.

Для створення клієнтської частини веб-системи було обрано бібліотеку React. React є однією з найпоширеніших JavaScript-бібліотек для розробки сучасних веб-застосунків. Використання React дозволило реалізувати динамічний інтерфейс користувача, забезпечити швидке оновлення сторінок без перезавантаження браузера та організувати зручну структуру компонентів. Компонентний підхід значно спрощує підтримку проекту та дозволяє повторно використовувати окремі елементи інтерфейсу у різних частинах системи.

У розробленій системі React використовується для реалізації сторінок авторизації, dashboard, модулів управління замовленнями, маршрутами, транспортом та доставками. Також React використовується для реалізації калькулятора логістичних розрахунків та інтеграції картографічного функціоналу.

Для створення та запуску frontend-проекту використовується Vite [12]. Даний інструмент забезпечує швидкий запуск застосунку, оптимізацію збірки та високу швидкість оновлення сторінок під час розробки. Використання Vite дозволило значно спростити процес розробки та скоротити час компіляції проекту.

Для обміну даними між клієнтською та серверною частинами системи використовується бібліотека Axios [13]. Axios забезпечує виконання HTTP-запитів та дозволяє передавати інформацію у форматі JSON. За допомогою Axios у системі реалізовано авторизацію користувачів, отримання інформації про замовлення, побудову маршрутів, отримання результатів логістичних розрахунків та взаємодію з серверною частиною системи.

Для реалізації навігації між сторінками використовується React Router [14]. Дана бібліотека дозволяє організувати багатосторінкову структуру SPA-застосунку та забезпечує швидке перемикання між окремими модулями системи без повного перезавантаження сторінки.

Серверна частина системи реалізована за допомогою Node.js та Express.js. Node.js є серверним середовищем виконання JavaScript, яке дозволяє створювати швидкі та масштабовані веб-застосунки. Основною перевагою Node.js є підтримка асинхронної обробки запитів, що дозволяє ефективно працювати з великою кількістю користувачів та забезпечує стабільну роботу системи.

Для організації серверної логіки використовується Express.js. Даний фреймворк дозволяє створювати REST API, реалізовувати маршрутизацію HTTP-запитів, обробляти дані та організовувати взаємодію між frontend і базою даних. За допомогою Express.js у системі реалізовано API для роботи з користувачами, замовленнями, маршрутами, транспортом, доставками та логістичними розрахунками.

Для зберігання інформації використовується PostgreSQL. PostgreSQL є реляційною системою керування базами даних, яка забезпечує високу продуктивність, стабільність та підтримку складних SQL-запитів. Використання PostgreSQL дозволяє ефективно працювати з великим обсягом даних та реалізовувати зв'язки між таблицями бази даних.

У базі даних системи зберігається інформація про користувачів, ролі, замовлення, маршрути, транспорт, доставки та результати логістичних

розрахунків. Для взаємодії з базою даних використовується ORM Sequelize. Дана бібліотека дозволяє працювати з базою даних через JavaScript-моделі, автоматично формувати SQL-запити та спрощує реалізацію CRUD-операцій. Sequelize також забезпечує підтримку зв'язків між таблицями та спрощує структуру серверної частини проекту.

У системі реалізовано авторизацію користувачів та рольову модель доступу. Для авторизації використовується JWT (JSON Web Token), який дозволяє перевіряти користувача під час входу до системи та обмежувати доступ до окремих функцій відповідно до ролі користувача [15]. У системі передбачено ролі адміністратора, менеджера, водія та клієнта. Кожна роль має власний функціонал та окремий рівень доступу до модулів системи.

Для реалізації картографічного функціоналу використовується OpenStreetMap та бібліотека Leaflet. OpenStreetMap є відкритим картографічним сервісом, який дозволяє безкоштовно використовувати картографічні дані у веб-застосунках. Leaflet використовується для відображення інтерактивної карти, побудови маршрутів, роботи з маркерами та візуалізації точок доставки. Інтеграція картографічних сервісів дозволяє підвищити зручність використання системи та забезпечує наочне відображення логістичних процесів [5, 6].

У веб-системі також реалізовано калькулятор логістичних розрахунків, який дозволяє автоматично визначати відстань між точками маршруту, приблизний час доставки, витрати пального та орієнтовну вартість маршруту. Для виконання розрахунків використовуються параметри маршруту та математичні формули. Реалізація калькулятора дозволяє автоматизувати процес планування доставки та спрощує роботу користувачів системи.

4.2. Опис програмної реалізації

У процесі розробки веб-системи було реалізовано frontend та backend частини застосунку. Frontend частина створена за допомогою React та забезпечує відображення інтерфейсу користувача, навігацію між сторінками та взаємодію із серверною частиною системи.

Структура frontend частини побудована за компонентним принципом. Окремі компоненти відповідають за dashboard, сторінки авторизації, модулі замовлень, маршрути, калькулятор логістичних розрахунків та картографічний функціонал.

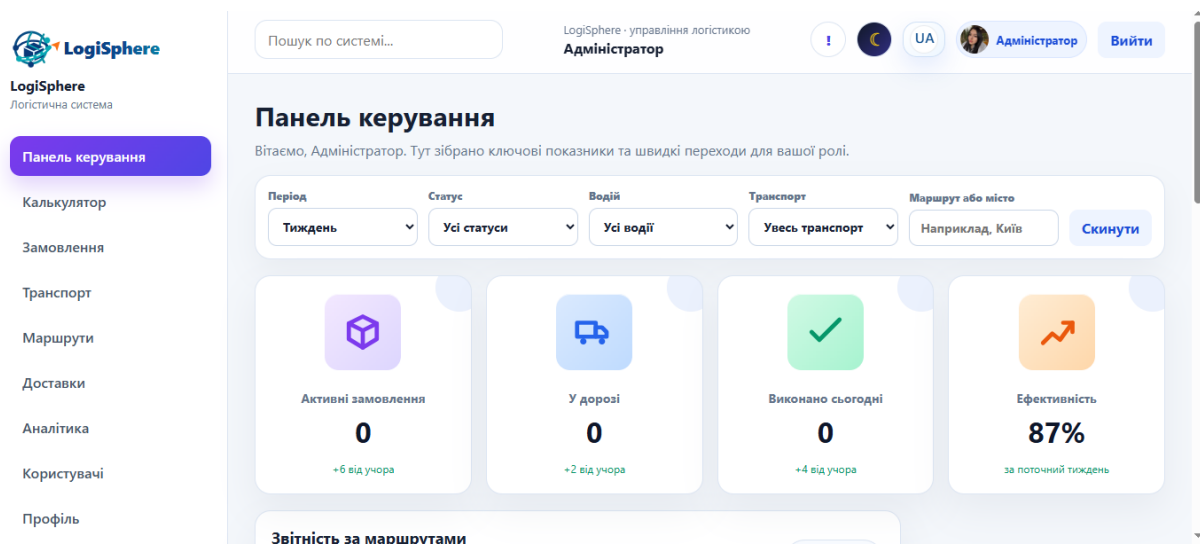


Рисунок 4.1 – Інтерфейс frontend частини системи

Backend частина системи реалізована за допомогою Node.js та Express.js. Серверна частина забезпечує обробку HTTP-запитів, взаємодію з базою даних та реалізацію REST API. У backend частині реалізовано модулі маршрутизації, контролери, middleware, моделі бази даних та алгоритми обробки логістичних процесів.

У системі реалізовано API для авторизації користувачів, роботи із замовленнями, управління маршрутами, виконання логістичних розрахунків та роботи з транспортом.

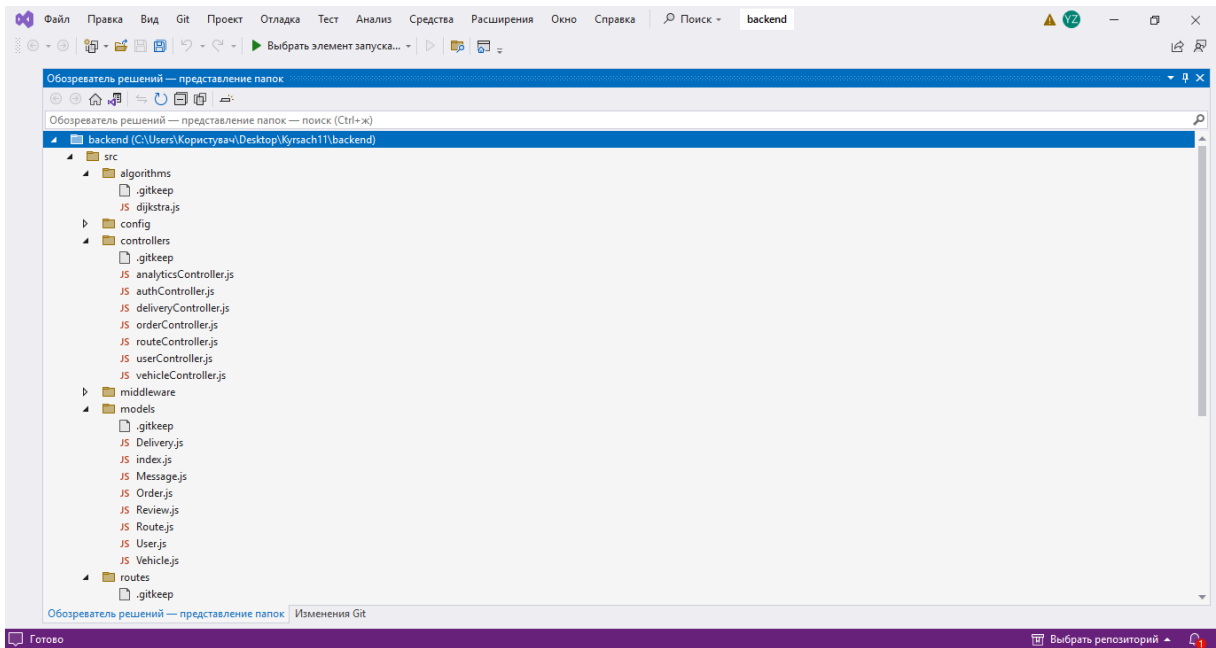


Рисунок 4.2 – Структура backend частини веб-системи

Для зберігання інформації використовується PostgreSQL. У базі даних реалізовано таблиці користувачів, замовлень, маршрутів, транспорту та доставок. Взаємодія з базою даних реалізована за допомогою ORM Sequelize.

У веб-системі реалізовано рольову модель доступу користувачів. Для авторизації використовується JWT. У системі створено ролі адміністратора, менеджера, водія та клієнта.

Для кожної ролі реалізовано окремий набір функцій та доступ до відповідних модулів системи.

Клієнт	client@example.com	Клієнт	Клієнт ▾	Видалити
Водій	driver@example.com	Водій	Водій ▾	Видалити
Менеджер	manager@example.com	Менеджер	Менедже▾	Видалити
Адміністратор	admin@example.com	Адміністратор	Адміністр▾	Видалити

Рисунок 4.3 – Рольова модель користувачів

Для реалізації картографічного функціоналу використовується OpenStreetMap та бібліотека Leaflet. У системі реалізовано відображення маршрутів, точок доставки та інтерактивної карти.

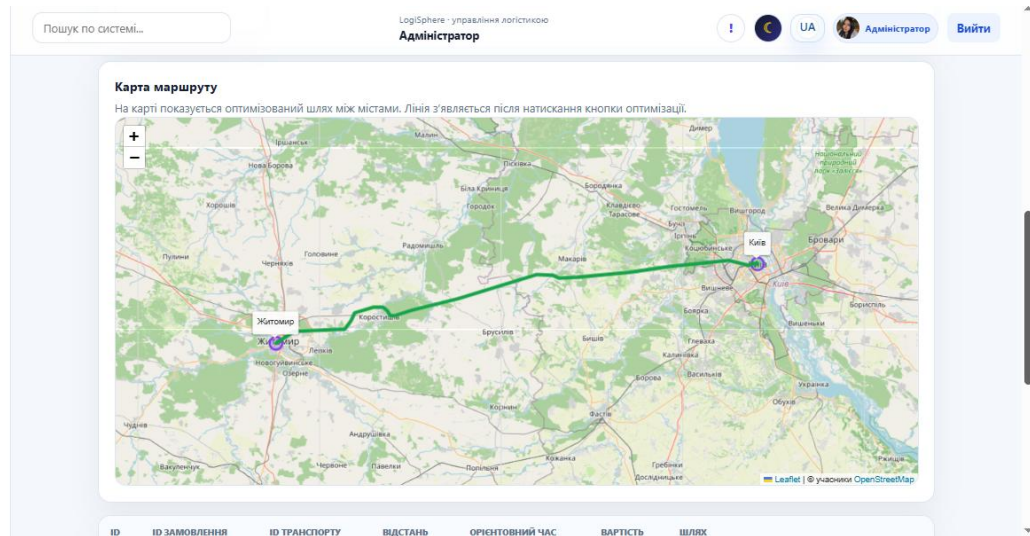


Рисунок 4.4 – Реалізація картографічного функціоналу

У системі також реалізовано калькулятор логістичних розрахунків, який дозволяє автоматично виконувати обчислення відстані, часу доставки, витрат пального та вартості маршруту.

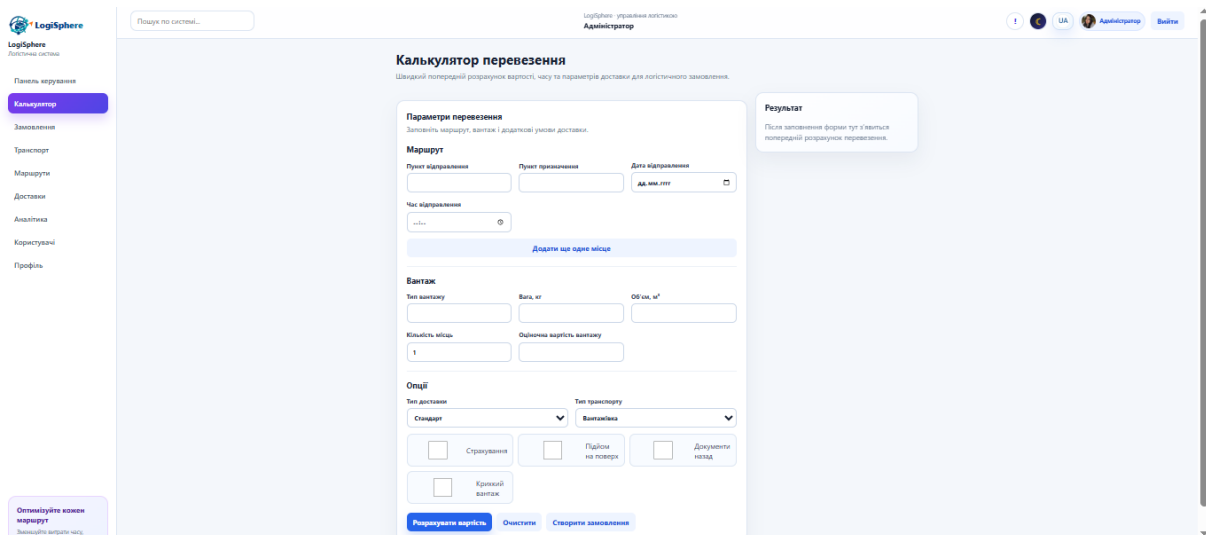


Рисунок 4.5 – Реалізація калькулятора логістичних розрахунків

У веб-системі реалізовано адаптивний дизайн, що забезпечує коректне відображення інтерфейсу на мобільних пристроях. Інтерфейс

автоматично підлаштовується під різні розміри екрана, що дозволяє зручно використовувати систему на смартфонах та планшетах.

Мобільна версія системи підтримує основний функціонал веб-застосунку, включаючи перегляд маршрутів, роботу із замовленнями, логістичні розрахунки та навігацію між сторінками.

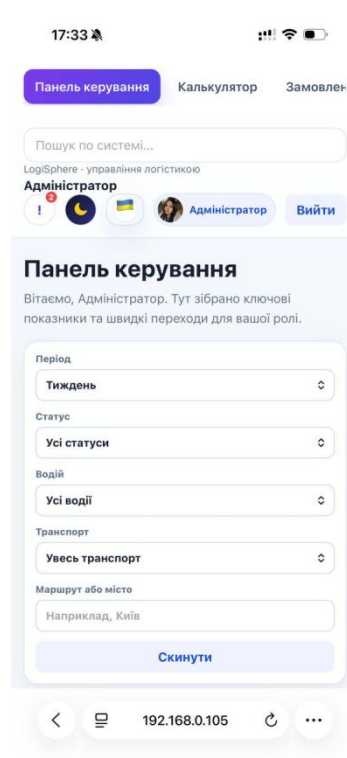


Рисунок 4.6 – Мобільна версія веб-системи

4.3. Опис роботи веб-системи для управління логістичними процесами з алгоритмами оптимізації

Після запуску веб-системи користувач бачить сторінку авторизації, на якій необхідно ввести логін та пароль для входу до системи. Для нових користувачів передбачено можливість реєстрації нового облікового запису.

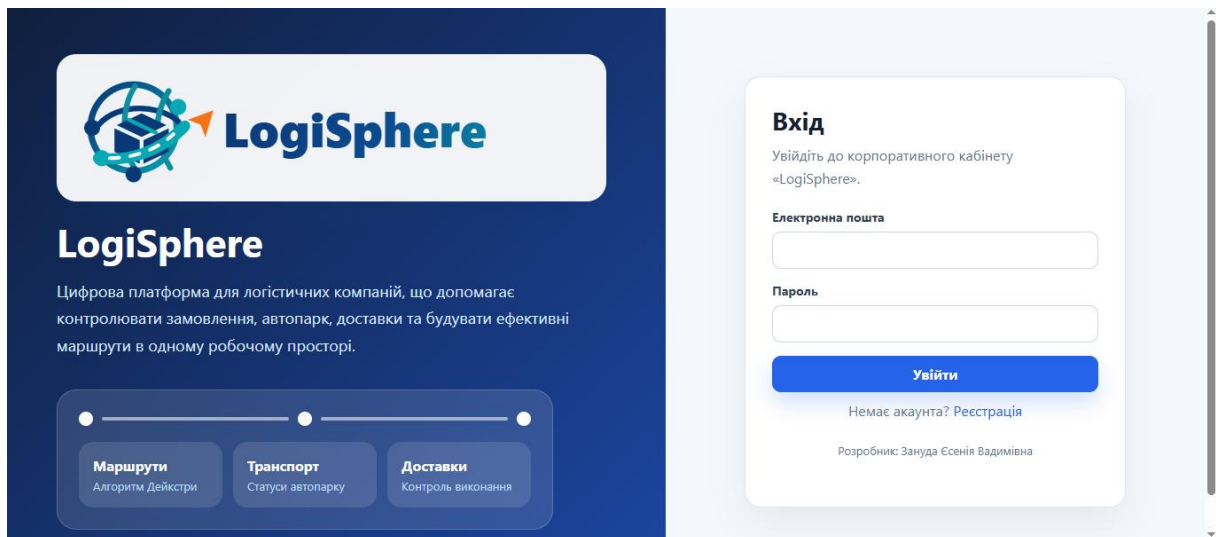


Рисунок 4.7 – Сторінка авторизації користувача

Під час реєстрації користувач може самостійно обрати роль клієнта або водія. Ролі адміністратора та менеджера призначаються системою окремо. Для реєстрації необхідно заповнити всі обов'язкові поля форми.

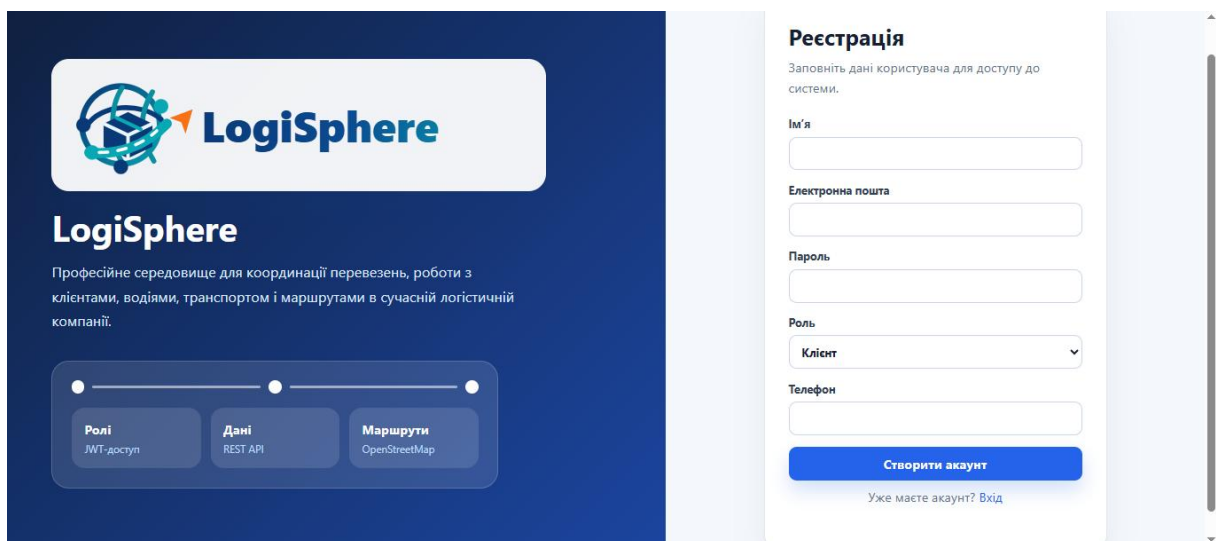


Рисунок 4.8 – Сторінка реєстрації користувача

У веб-системі реалізовано перевірку правильності введення даних. Якщо користувач залишить обов'язкові поля порожніми або введе некоректні дані, система виведе повідомлення про помилку.

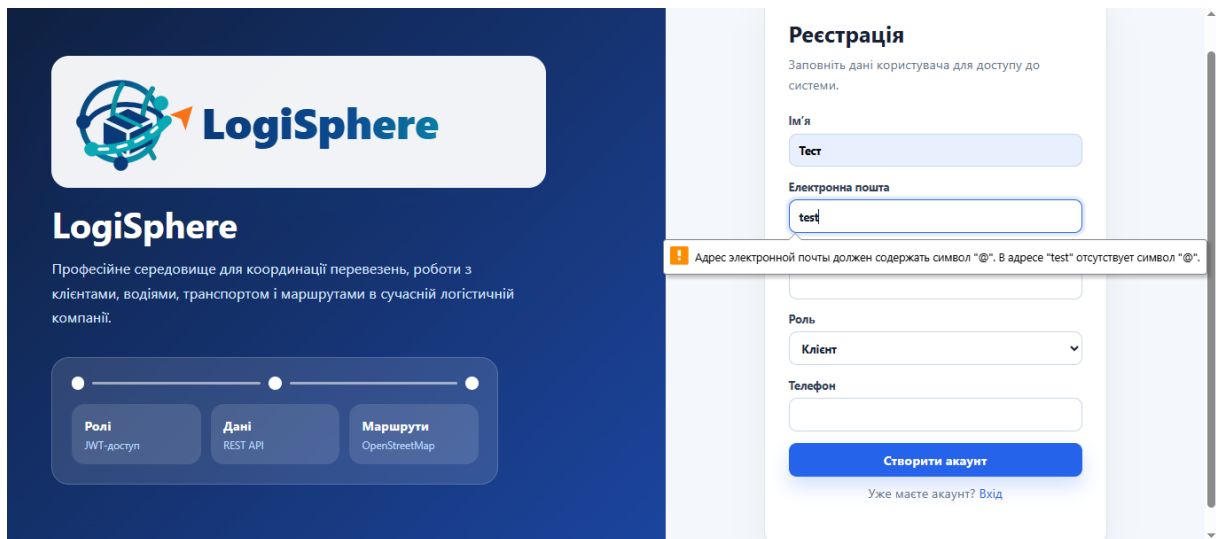


Рисунок 4.9 – Обробка помилок введення даних

Після успішної авторизації користувач переходить до головної сторінки системи. На головному екрані відображаються основні статистичні показники, інформація про замовлення, доставки, транспорт та маршрути. Також користувач отримує доступ до навігаційного меню системи.

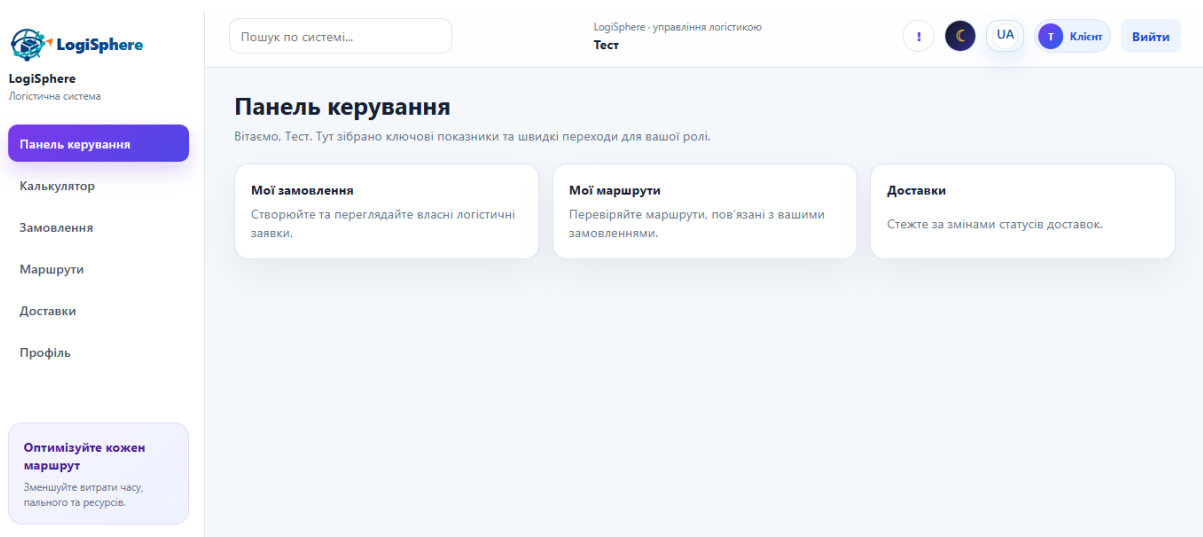


Рисунок 4.10 – Головна сторінка веб-системи

У системі реалізовано рольову модель доступу користувачів. Адміністратор має повний доступ до всіх модулів системи. Адміністратор може переглядати аналітику, редагувати замовлення, управляти

транспорт, додавати водіїв, змінювати ролі користувачів, переглядати маршрути та виконувати оптимізацію доставки.

Клієнт	client@example.com	Клієнт	Клієнт ▾	Видалити
Водій	driver@example.com	Водій	Водій ▾	Видалити
Менеджер	manager@example.com	Менеджер	Менеджер ▾	Видалити
Адміністратор	admin@example.com	Адміністратор	Адміністр ▾	Видалити

Рисунок 4.11 – Рольова модель користувачів системи

Менеджер відповідає за організацію логістичних процесів. Менеджер може створювати нові замовлення, формувати маршрути, користуватися калькулятором логістичних розрахунків, призначати водіїв та контролювати доставки.

Для створення нового замовлення необхідно заповнити інформацію про клієнта, пункт відправлення, пункт призначення, вагу вантажу та інші параметри доставки.

LogiSphere - управління логістикою
Менеджер

Пошук по системі...

Замовлення
Перегляд логістичних замовлень, створення заявок і контроль статусів доставки.

Створити замовлення
Додайте пункт відправлення, пункт призначення та вагу вантажу.

ID клієнта	Пункт відправлення	Пункт призначення	Вага вантажу	Створити замовлення
1111	Кривий Ріг	Запоріжжя	200	Створити замовлення

ID	Клієнт	Пункт відправлення	Пункт призначення	Вага вантажу	Статус	Дата створення	Дії
20	Тест	Миколаїв	Буча	45678	У дорозі	03.05.2026, 22:22:59	Видалити
19	Клієнт	PointA	Client	750	Доставлено	03.05.2026, 22:10:06	Видалити
18	Клієнт	Warehouse	PointC	1200	Призначено	03.05.2026, 22:10:06	Видалити

Рисунок 4.12 – Модуль управління замовленнями

Якщо користувач не заповнить необхідні поля під час створення замовлення, система автоматично виведе повідомлення про помилку та не дозволить зберегти дані.

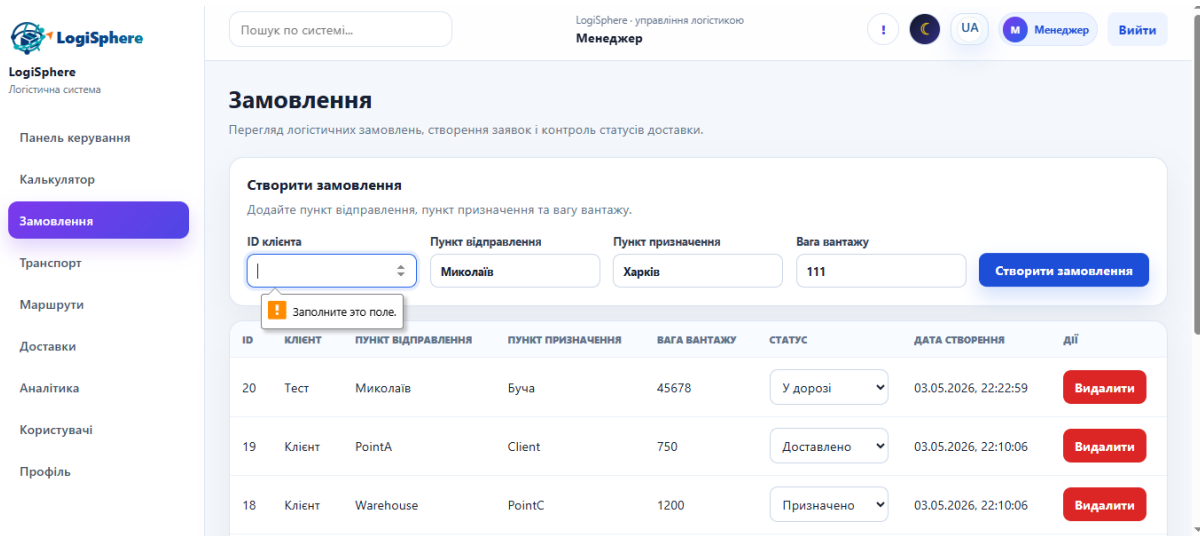


Рисунок 4.13 – Обробка помилок при створенні замовлення

Після успішного створення замовлення інформація автоматично зберігається у базі даних та відображається у таблиці замовлень.

У таблиці замовлень відображаються ID замовлення, клієнт, пункт відправлення, пункт призначення, вага вантажу, статус доставки та дата створення замовлення.

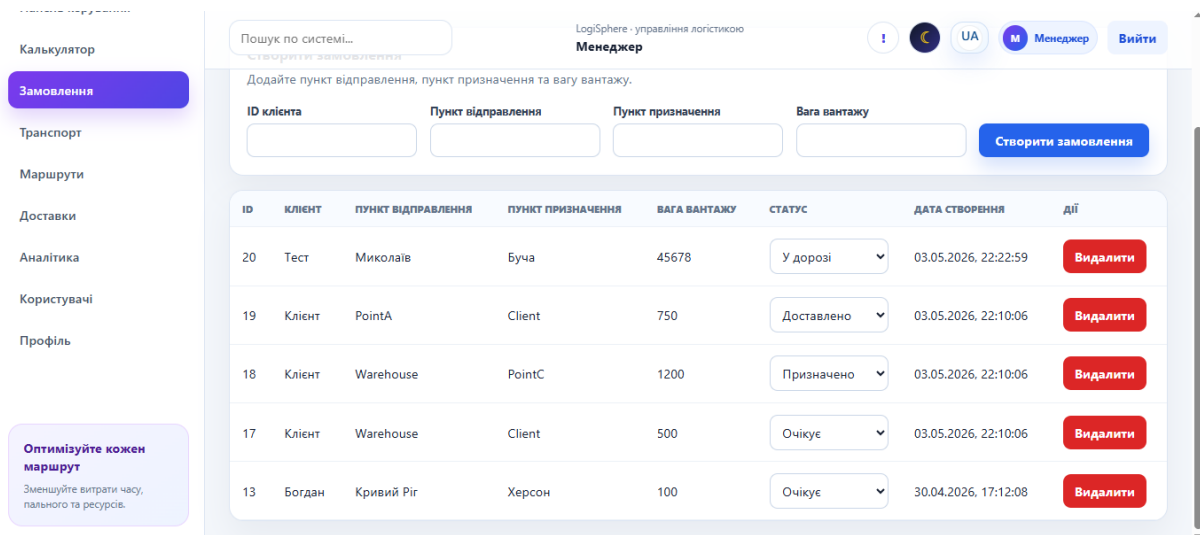


Рисунок 4.14 – Таблиця замовлень системи

Для побудови маршруту у веб-системі реалізовано алгоритм оптимізації доставки. Користувач може обрати замовлення та транспортний засіб, після чого система автоматично формує оптимальний маршрут між вибраними точками.

Після виконання оптимізації система виконує розрахунок відстані, приблизного часу доставки, витрат пального та орієнтовної вартості перевезення.

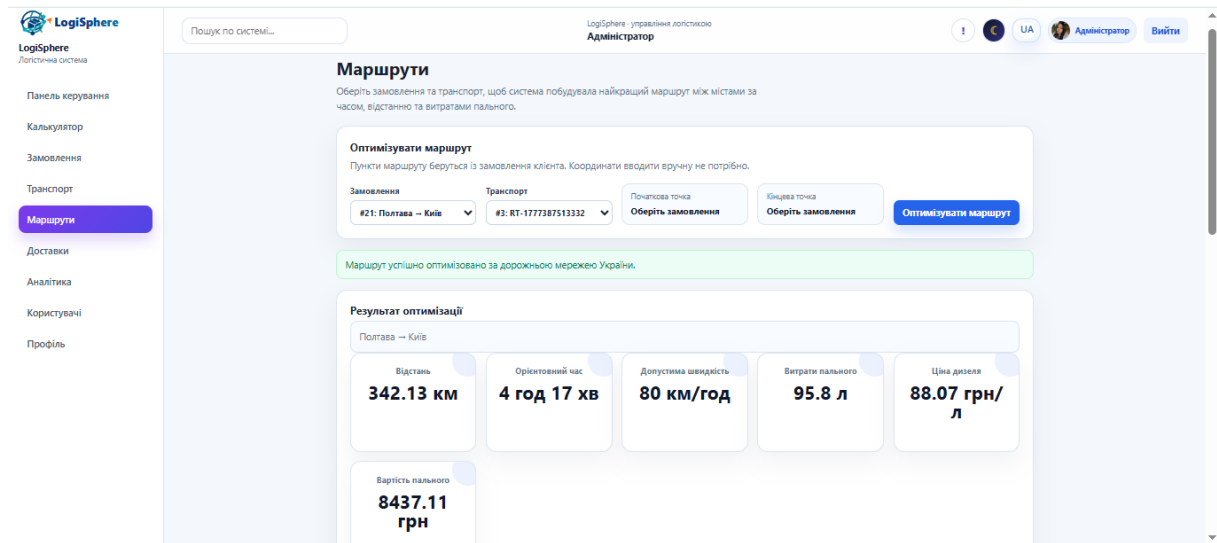


Рисунок 4.15 – Формування маршруту доставки

Для візуалізації маршруту використовується інтерактивна карта OpenStreetMap. На карті відображаються маркери, точки доставки та побудований маршрут між містами.

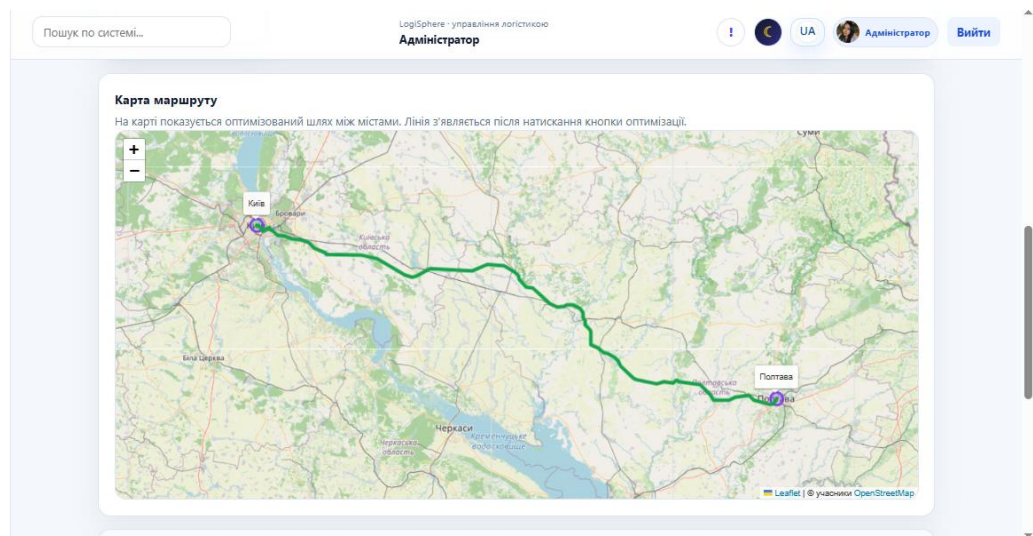


Рисунок 4.16 – Відображення маршруту доставки на карті

У веб-системі реалізовано калькулятор логістичних розрахунків. Користувач може ввести інформацію про маршрут, дату, час доставки, тип транспорту, вагу вантажу, об'єм та додаткові параметри.

Калькулятор підтримує декілька типів транспорту, зокрема вантажівку, фургон, рефрижератор та легковий транспорт. Також користувач може обрати тип доставки: стандартну, експрес або термінову.

Додатково система підтримує такі послуги, як страхування вантажу, підйом на поверх, повернення документів та перевезення крихкого вантажу.

Калькулятор перевезення
Швидкий попередній розрахунок вартості, часу та параметрів доставки для логістичного замовлення.

Параметри перевезення
Задати маршрут, вантаж і додаткові умови доставки.

Маршрут
Пункт відправлення: Пункт призначення: Дата відправлення:

Час відправлення:

[Додати ще одне місто](#)

Вантаж
Тип вантажу: Вага, кг: Об'єм, м³:

Кількість міськ: Оцінена вартість вантажу:

Опції
Тип доставки: Тип транспорту:

Страхування Підйом на поверх Документи назад

Крихкий вантаж

[Розрахувати вартість](#) [Очистити](#) [Створити замовлення](#)

Результат

11 841 грн

Орієнтовна відстань: **379 км**
Орієнтовний час доставки: **6.5 год**
Орієнтовне прибуття: **11.05.2026, 18:30:00**
Обраний транспорт: **Вантажівка**
Рекомендований транспорт: **Фургон**
Тип доставки: **Стандарт**
Додаткові послуги: **Страхування, Крихкий вантаж**

Розрахунок вартості
Базовий тариф: 1 350 грн
Відстань: 6 090 грн
Вантаж: 175 грн
Послуги: 1 220 грн
Коефіцієнт доставки: 0 грн

Маршрут: Полтава – Київ
[Створити замовлення і роздрукувати](#)

Рисунок 4.17 – Калькулятор логістичних розрахунків

Водій у системі має доступ лише до власних доставок та маршрутів. Водій може переглядати інформацію про доставку, маршрут на карті та змінювати статус виконання доставки.

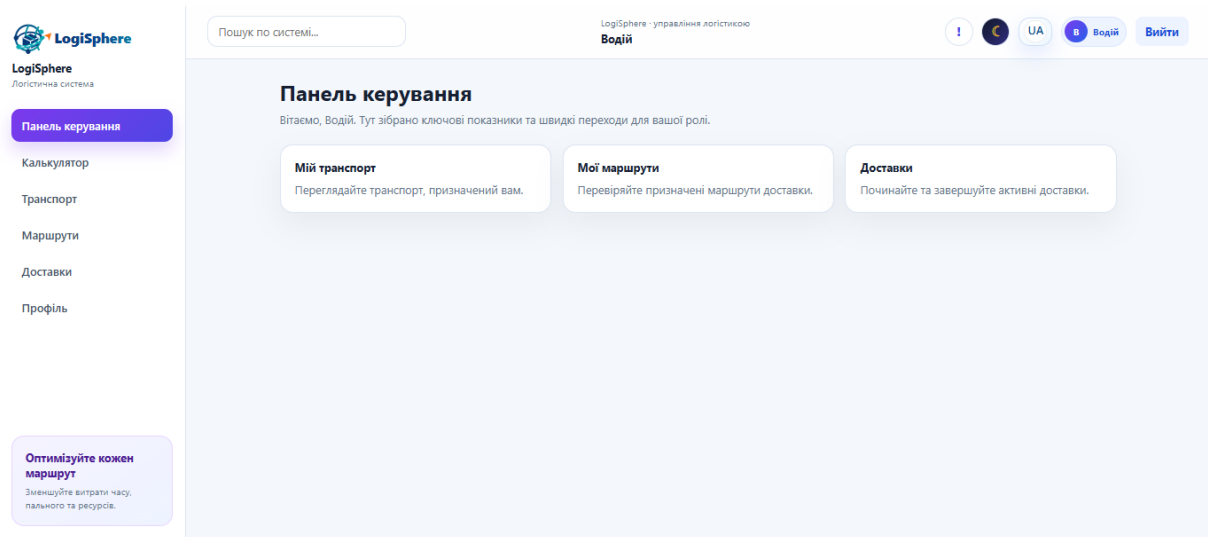


Рисунок 4.18 – Інтерфейс водія системи

Клієнт може створювати замовлення, переглядати історію доставок та контролювати статус виконання замовлення. Інформація про маршрут та доставку доступна у відповідному модулі системи.

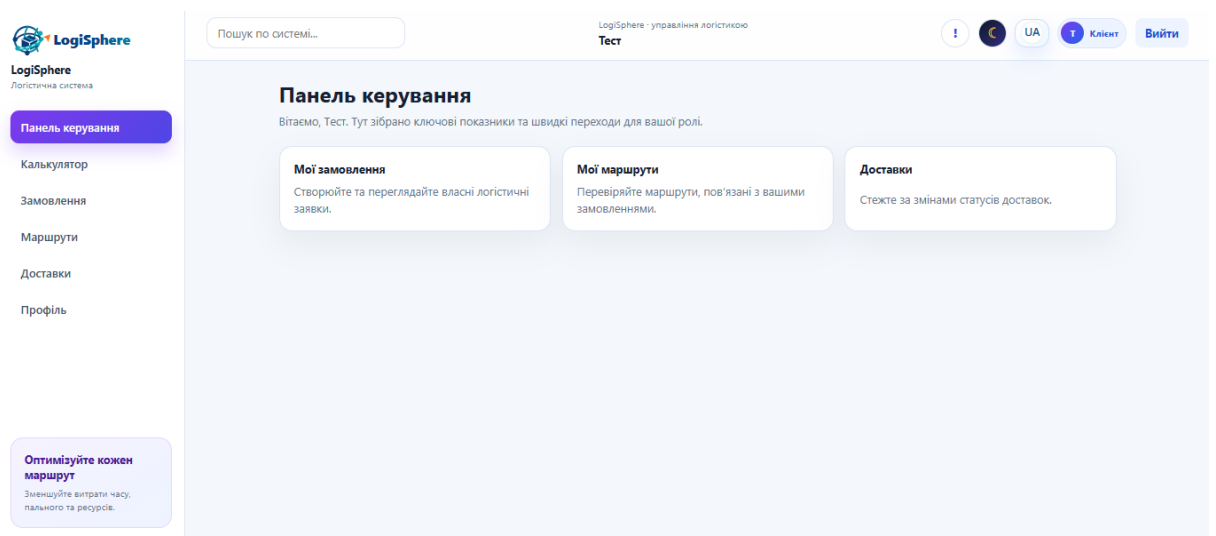


Рисунок 4.19 – Інтерфейс клієнта системи

У системі реалізовано модуль аналітики, який дозволяє переглядати основні показники роботи логістичної системи. Для адміністратора та менеджера доступна інформація про кількість замовлень, доставки, транспорт, маршрути, витрати пального, час доставки та ефективність роботи системи.

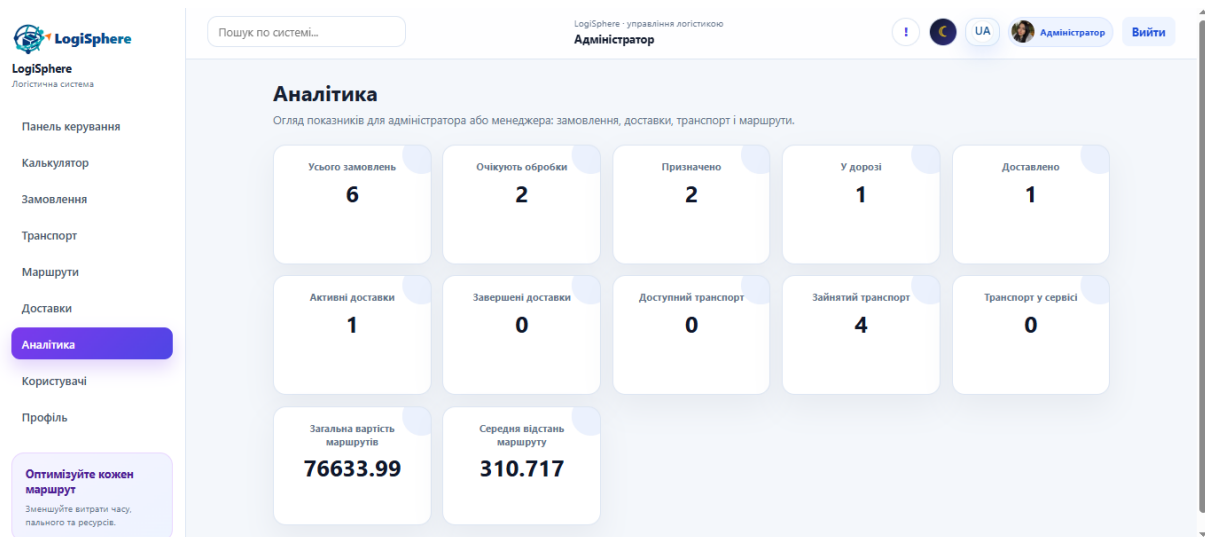


Рисунок 4.20 – Аналітика логістичної системи

У веб-системі реалізовано підтримку темної теми інтерфейсу, а також мультимовність. Користувач може змінювати мову системи між українською та англійською.

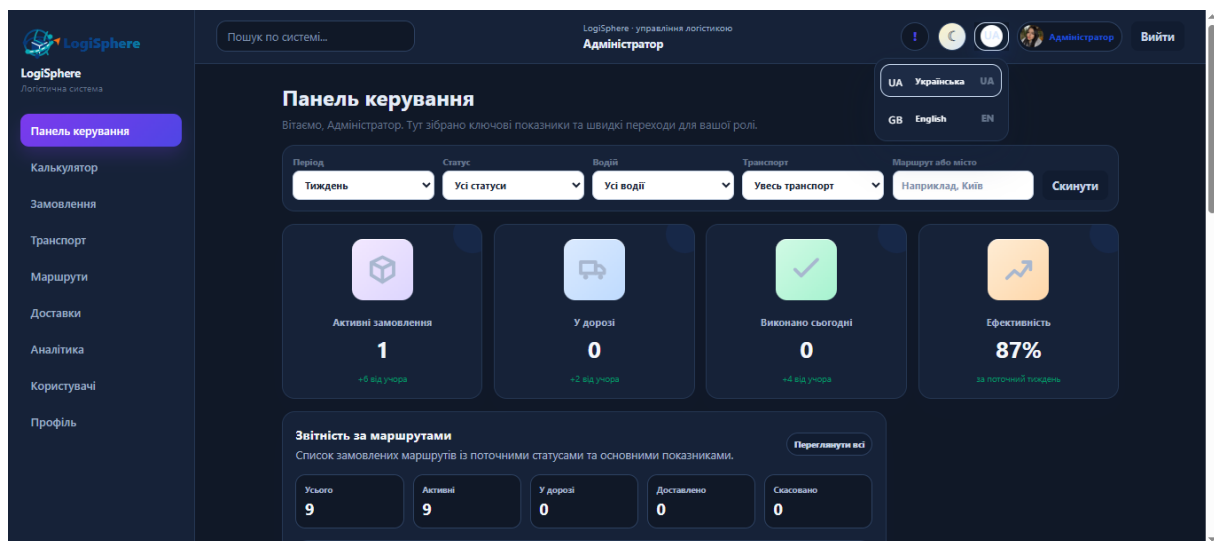


Рисунок 4.21 – Додаткові налаштування інтерфейсу

Додатково у системі реалізовано адаптивний дизайн, що забезпечує коректну роботу веб-застосунку на мобільних пристроях. Мобільна версія підтримує основний функціонал системи та дозволяє працювати із замовленнями, маршрутами та логістичними розрахунками на смартфонах і планшетах.

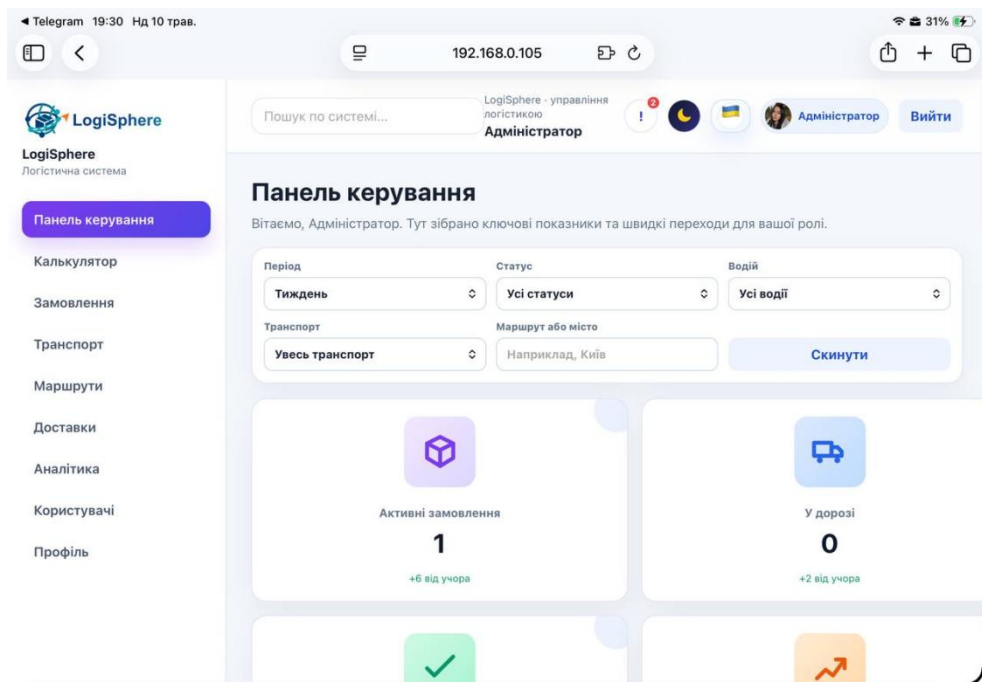


Рисунок 4.22 – Мобільна версія веб-системи

У результаті розробки було створено веб-систему для управління логістичними процесами з алгоритмами оптимізації маршрутів. Система забезпечує автоматизацію роботи із замовленнями, формування маршрутів доставки, логістичні розрахунки, аналітику та контроль транспортних процесів. Реалізовано рольову модель доступу користувачів, інтеграцію картографічних сервісів та адаптивний інтерфейс користувача. Розроблена веб-система є зручною у використанні, підтримує сучасні веб-технології та може бути основою для подальшого розширення функціоналу логістичної платформи.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено аналіз предметної області логістичних процесів, що дозволило визначити основні вимоги до веб-системи управління логістикою з алгоритмами оптимізації маршрутів. Було встановлено, що ефективне управління логістичними процесами потребує автоматизації роботи із замовленнями, транспортом, маршрутами та доставками, а також використання сучасних засобів аналітики й картографічних сервісів.

У процесі роботи було спроектовано архітектуру веб-системи, яка забезпечує взаємодію між клієнтською та серверною частинами застосунку, базою даних і модулем обробки логістичних даних. Розроблена структура системи забезпечує масштабованість, зручність підтримки та можливість подальшого розширення функціоналу.

Frontend частина системи реалізована за допомогою React, що дозволило створити сучасний та адаптивний користувацький інтерфейс. Backend частина реалізована із використанням Node.js та Express.js. Для зберігання даних використовується база даних PostgreSQL, а взаємодія із нею реалізована за допомогою ORM Sequelize. У процесі розробки було використано REST API та реалізовано рольову модель доступу користувачів.

Розроблена веб-система забезпечує управління логістичними замовленнями, формування маршрутів доставки, контроль транспортних процесів та роботу з користувачами різних ролей. У системі реалізовано функціонал адміністратора, менеджера, водія та клієнта, що дозволяє ефективно організувати роботу логістичної платформи.

Однією з ключових особливостей системи є реалізація алгоритмів оптимізації маршрутів, які дозволяють автоматично формувати маршрут доставки та виконувати розрахунок відстані, часу доставки, витрат пального та орієнтовної вартості перевезення. Для візуалізації маршрутів використовується інтеграція з OpenStreetMap та бібліотекою Leaflet [20].

Проведене тестування підтвердило коректність роботи веб-системи, стабільність взаємодії між модулями та правильність виконання логістичних розрахунків. Розроблений інтерфейс є інтуїтивно зрозумілим, підтримує адаптивний дизайн, темну тему та мультимовність, що забезпечує зручність використання системи на різних типах пристроїв.

Таким чином, у результаті виконання роботи було створено функціональну веб-систему для управління логістичними процесами з алгоритмами оптимізації маршрутів, яка дозволяє автоматизувати основні логістичні операції, підвищити ефективність управління доставками та забезпечити зручний контроль транспортних процесів.

Отримані результати підтверджують практичну значущість розробленої веб-системи та були апробовані на XLVIX Міжнародній науковій конференції студентів та аспірантів «Актуальні питання розвитку науки та забезпечення якості освіти у XXI столітті».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ольховська О.В., Черненко О. О. методичні рекомендації до виконання кваліфікаційної роботи для студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра / О. О. Черненко., Ольховська О.В. – Полтава : ПУЕТ, 2025. – 58 с.
2. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.
3. Lardi-Trans : міжнародна логістична платформа для організації перевезень [Електронний ресурс]. – Режим доступу: <https://lardi-trans.com>.
4. Ant Logistics : хмарна система автоматизації транспортної логістики [Електронний ресурс]. – Режим доступу: <https://ant-logistics.com>.
5. OpenStreetMap : відкритий картографічний сервіс та база даних [Електронний ресурс]. – Режим доступу: <https://www.openstreetmap.org>.
6. Leaflet : JavaScript-бібліотека для створення інтерактивних карт [Електронний ресурс]. – Режим доступу: <https://leafletjs.com>.
7. React : бібліотека для створення динамічних інтерфейсів користувача [Електронний ресурс]. – Режим доступу: <https://react.dev>.
8. Node.js : середовище виконання JavaScript на стороні сервера [Електронний ресурс]. – Режим доступу: <https://nodejs.org>.
9. Express.js : веб-фреймворк для побудови REST API в середовищі Node.js [Електронний ресурс]. – Режим доступу: <https://expressjs.com>.
10. PostgreSQL : документація до реляційної системи управління базами даних [Електронний ресурс]. – Режим доступу: <https://www.postgresql.org/docs>.

11. Sequelize : об'єктно-реляційне відображення (ORM) для Node.js [Електронний ресурс]. – Режим доступу: <https://sequelize.org>.
12. Vite : інструментарій для швидкої збірки сучасних веб-проектів [Електронний ресурс]. – Режим доступу: <https://vitejs.dev>.
13. Axios : бібліотека для виконання асинхронних HTTP-запитів [Електронний ресурс]. – Режим доступу: <https://axios-http.com>.
14. React Router : бібліотека для організації навігації в SPA-застосунках [Електронний ресурс]. – Режим доступу: <https://reactrouter.com>.
15. JSON Web Tokens : стандарт безпечної передачі даних між сторонами [Електронний ресурс]. – Режим доступу: <https://jwt.io/introduction>.
16. Алгоритми пошуку найкоротшого шляху : математичні основи та реалізація [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Алгоритм_Дейкстри.
17. Системний підхід у логістиці : основні принципи та методи управління [Електронний ресурс]. – Режим доступу: https://pidru4niki.com/15240525/logistika/sistemniy_pidhid_logistitsi.
18. Об'єктно-орієнтоване програмування : основні концепції та принципи [Електронний ресурс]. – Режим доступу: <https://vseosvita.ua/library/obektno-orientovane-programuvanna-osnovni-koncepcii-567432.html>.
19. UML-моделювання : побудова діаграм класів та бізнес-процесів [Електронний ресурс]. – Режим доступу: <https://itvdn.com/ua/blog/article/uml-diagrams>.
20. Логістика та управління ланцюгами поставок : сучасні технології [Електронний ресурс]. – Режим доступу: <https://econom.univ.kiev.ua/wp-content/uploads/2018/01/logistics.pdf>.
21. Клієнт-серверна архітектура : принципи побудови веб-систем [Електронний ресурс]. – Режим доступу: <https://metanit.com/web/javascript/nodejs/1.1.php>.

ДОДАТОК А. КОД ПРОГРАМИ