

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інтелектуальних інформаційних систем**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних  
інформаційних систем

\_\_\_\_\_Юрій КОНДРАТЕНКО

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**  
**ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ПЛАНУВАННЯ ТА**  
**ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ**  
**АВТОЗАПЧАСТИН**

Спеціальність 122 Комп'ютерні науки  
Освітня програма «Інтелектуальні інформаційні системи»

*Здобувач*

\_\_\_\_\_ Владислав МІЛЄВ

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

*Керівник* д-р техн. наук, професор

\_\_\_\_\_ Олексій КОЗЛОВ

« \_\_\_\_ » \_\_\_\_\_ 2024 р.

**Миколаїв – 2024**

Чорноморський національний університет імені Петра Могили  
(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	122 Комп'ютерні науки
Освітня програма	Інтелектуальні інформаційні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних  
інформаційних систем

\_\_\_\_\_ Юрій КОНДРАТЕНКО

«\_\_\_\_\_» \_\_\_\_\_ 2024 р.

**ЗАВДАННЯ**  
**на кваліфікаційну роботу здобувача**

**Мілева Владислава В'ячеславовича**

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Інформаційна система для планування та оптимізації ланцюгів постачання автозапчастин».

Керівник роботи: Козлов Олексій Валерійович, професор кафедри ІС, д-р техн. наук.

Затверджена наказом ЧНУ ім. Петра Могили від «03» червня 2024 р. № 140/1.

2. Строк представлення кваліфікаційної роботи « 16 » грудня 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: інформаційна система для планування та оптимізації ланцюгів постачання автозапчастин з використанням сучасних алгоритмів машинного навчання та аналітики даних для покращення ефективності ланцюга постачання.

4. Перелік питань, що підлягають розробці: аналіз сучасного стану задачі, щодо розробки інформаційної системи для оптимізації ланцюгів постачання автозапчастин; огляд наявних методів для розв'язування задачі оптимізації маршрутів; моделювання та проєктування інформаційної системи, опис результатів реалізації та тестування.

5. Перелік графічних матеріалів: презентація.

**Керівник роботи**

\_\_\_\_\_

*(Особистий підпис)*

Олексій КОЗЛОВ  
*(Власне ім'я ПРІЗВИЩЕ)*

**Здобувач**

\_\_\_\_\_

*(Особистий підпис)*

Владислав МІЄВ  
*(Власне ім'я ПРІЗВИЩЕ)*

Дата видачі завдання «07» червня 2024 р.

# КАЛЕНДАРНИЙ ПЛАН

## кваліфікаційної роботи

Тема: Інформаційна система для планування та оптимізації ланцюгів постачання автозапчастин

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	03.06.2024	07.06.2024	Виконано
2	Аналіз предметної області та постановка задачі	10.06.2024	20.06.2024	Виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема аналіз публікацій та комерційних систем, щодо оптимізації маршрутів за допомогою алгоритмів	21.06.2024	01.07.2024	Виконано
4	Огляд чинних алгоритмів маршрутизації транспортного засобу для розв'язання поставленої задачі	01.09.2024	25.10.2024	Виконано
5	Впровадження вибраних технологій із демонстрацією результатів	26.10.2024	21.11.2024	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	22.11.2024	22.11.2024	Виконано
7	Коригування роботи за результатами попереднього захисту	23.11.2024	05.12.2024	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	06.12.2024	06.12.2024	Виконано
9	Доробка та остаточне оформлення КР	07.12.2024	10.12.2024	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.12.2024	17.12.2024	Виконано

**Керівник роботи**

\_\_\_\_\_  
(Особистий підпис)

**Олексій КОЗЛОВ**  
(Власне ім'я ПРІЗВИЩЕ)

**Здобувач**

\_\_\_\_\_  
(Особистий підпис)

**Владислав МІЛЄВ**  
(Власне ім'я ПРІЗВИЩЕ)

Дата складання календарного плану  
«19» червня 2024 р.

## АНОТАЦІЯ

до кваліфікаційної роботи  
здобувача (ки) групи 601м ЧНУ ім. Петра Могили

**Мілева Владислава В'ячеславовича**

на тему: **“ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН”**

Підвищений попит на спрощене управління ланцюгом постачання в промисловості автозапчастин підкреслює важливість даного дослідження в умовах глобального розширення автомобільного виробництва та дистрибуції. Кваліфікаційна робота розглядає створення інформаційної системи планування (ІСП), спрямованої на підвищення ефективності процесів постачання автозапчастин.

**Об'єктом** дослідження є процеси постачання автозапчастин.

**Предметом** дослідження є моделі та методи оптимізації управління ланцюгами постачання в промисловості автозапчастин.

**Метою** роботи є підвищення операційної ефективності та зниження витрат в ланцюзі постачання автозапчастин за допомогою впровадження інформаційної системи планування та оптимізації.

Пояснювальна записка містить вступ, 4 розділи, висновки та додаткові матеріали.

У першому розділі подано аналіз поточного стану систем для організації постачання автозапчастин і окреслено цілі дослідження.

У другому розділі розглядаються методи та технології, що застосовуються для проектування ІСП.

У третьому розділі детально описується процес розробки системи, включаючи рішення щодо архітектурного проектування, вибір алгоритмів та впровадження системи.

У четвертому розділі представлено результати дослідження ефективності функціонування інформаційної системи планування та оптимізації, які підтверджують зниження витрат та загальне удосконалення процесів постачання автозапчастин. Загальний обсяг роботи – 71 сторінка. Кваліфікаційна робота містить 41 рисунок, 5 таблиць і 37 джерел посилання.

**Ключові слова:** проблема маршрутизації транспортних засобів, оптимізація ланцюгів постачання.

## **ABSTRACT**

to the qualification work by the student of the group 601m of Petro Mohyla Black Sea  
National University

**Miliev Vladyslav**

### **“INFORMATION SYSTEM FOR PLANNING AND OPTIMIZATION OF AUTO PARTS SUPPLY CHAINS”**

The growing demand for simplified supply chain management in the auto parts industry underscores the importance of this research in the context of the global expansion of automotive manufacturing and distribution. The master's qualification work considers the creation of a planning information system (PIS) aimed at increasing the efficiency of auto parts supply processes.

The **object** of research is the supply processes of auto parts.

The **subject** of the study is models and methods of optimization of supply chain management in the auto parts industry.

The **purpose** of the work is to increase operational efficiency and reduce costs in the auto parts supply chain by implementing an information system for planning and optimization.

The explanatory note contains an introduction, 4 chapters, conclusions and additional materials.

The first chapter provides an analysis of the current state of systems for the organization of the supply of auto parts and outlines the goals of the study.

In the second section, the methods and technologies used for the design of PIS are considered.

The third section describes in detail the system development process, including architectural design decisions, algorithm selection, and system implementation.

The fourth chapter presents the results of the study of the effectiveness of the information system of planning and optimization, which confirm the reduction of costs and the general improvement of the supply processes of auto parts.

The overall scope of the work is 71 pages. Thesis contains 41 figures, 5 tables and 37 references in it.

**Key words:** vehicle routing problem, optimization of supply chains.

## ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ .....	3
ВСТУП.....	4
1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН.....	6
1.1 Опис предметної області.....	6
1.2 Огляд та аналіз комерційних рішень.....	7
1.3 Огляд та аналіз наявних публікацій .....	12
1.4 Огляд наявних технологій для розв’язання поставленої задачі.....	21
1.5 Постановка задачі.....	25
Висновки до розділу 1.....	27
2 МЕТОДИ ТА ТЕХНОЛОГІЇ ДЛЯ ВИРШЕННЯ ЗАДАЧІ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН .....	28
2.1 Інтеграція JavaScript і R для планування маршрутів в реальному часі .....	28
2.2 Методи оптимізації ланцюгів постачання автозапчастин.....	29
2.3 Методи прогнозування: експоненціальне згладжування та метод Холта- Уінтерса .....	32
Висновки до розділу 2.....	37
3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН.....	39
3.1 Проєктування інформаційної системи .....	39
3.2 Структура системи для оптимізації ланцюга постачання автозапчастин .....	45
Висновки до розділу 3.....	47
4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН.....	49
4.1 Опис програмної реалізації.....	49
4.2 Результати та порівняння алгоритмів оптимізації.....	65
Висновки до розділу 4.....	74
ВИСНОВКИ .....	76
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	78
ДОДАТОК А Апробація роботи .....	83
ДОДАТОК Б Лістинг коду інформаційної системи .....	84

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ЛП	– Ланцюг Постачання
ПЗ	– Програмне Забезпечення
ГА	– Генетичні Алгоритми
ІС	– Інформаційна Система
ТЗ	– Транспортний Засіб
SCM	– Supply Chain Management
API	– Application Programming Interface
JWT	– JSON Web Token
HTTP	– HyperText Transfer Protocol
SES	– Simple Exponential Smoothing
TSP	– Travelling Salesman Problem
NN	– Nearest Neighbor



## ВСТУП

В умовах сучасної індустріалізації та глобалізації, управління ланцюгами постачання стало ключовим фактором для забезпечення стабільної роботи галузей, зокрема автомобільної. Ефективність постачання автозапчастин безпосередньо впливає на графіки виробництва, витрати та загальний успіх компаній у цій сфері. З підвищенням попиту на автомобілі у світі збільшується і складність управління ланцюгами постачання, які підтримують цей сектор.

Останнім часом управління взаємовідносинами в системах постачань будь-чого зазнало суттєвих змін під впливом макро- та мікроекономічних факторів як в Україні [1], так і за її межами. Разом з цим з'являються нові виклики, зокрема недостатній розвиток наукових досліджень щодо управління складними економічними системами в українському контексті. Це підкреслює важливість фокусування на практичних та концептуальних аспектах управління ланцюгами постачання.

Швидка передача, збір та обробка інформації сьогодні стають критичними для прийняття ефективних управлінських рішень. Це дозволяє підприємствам швидше реагувати на потреби клієнтів і оперативно передавати необхідні дані про продукцію. Таким чином, інформаційний обмін відіграє важливу роль у забезпеченні швидкого виконання замовлень та ефективної роботи ланцюгів постачання.

Автомобільна промисловість, відома своїми розгалуженими мережами постачальників і дистриб'юторів, потребує високоефективних систем управління ланцюгами постачання. Традиційні підходи, що часто спираються на ручні процеси та застарілі технології, вже не відповідають потребам сучасного виробництва. До цих викликів додаються вимоги скорочення витрат, мінімізації затримок і забезпечення високої якості деталей. Ця система дозволяє користувачам вести облік автозапчастин, оцінювати різні алгоритми оптимізації постачань та розраховувати експлуатаційні

витрати, враховуючи низку факторів, такі як відстань, час доставки та використання ресурсів.

Пояснювальна записка поділена на чотири розділи, кожен з яких охоплює окремий аспект розробки та впровадження системи. Перший розділ аналізує поточний стан управління постачанням автозапчастин, розглядає наукові дослідження та визначає ключові області для покращення. Другий розділ присвячений моделям і методам оптимізації, що стосуються управління ланцюгами постачання, з акцентом на найактуальніші для автомобільної галузі технології. Третій розділ фокусується на технічних аспектах розробки системи, описуючи архітектурні рішення та етапи впровадження. Четвертий розділ містить результати застосування системи та її вплив на ефективність ланцюга постачання, а також можливості для масштабування та адаптації.

# 1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН

## 1.1 Опис предметної області

Ланцюжок постачання автозапчастин – це заплутана мережа, яка об'єднує різноманітні суб'єкти, необхідні для виробництва та розповсюдження автомобільних компонентів. Він починається з постачальників сировини та виробників основних компонентів, які забезпечують базові елементи, необхідні для виробництва транспортних засобів. Ці матеріали та компоненти переміщуються через ряд проміжних постачальників, постачальників логістичних послуг і розподільних центрів, усі вони працюють узгоджено, щоб забезпечити ефективну та вчасну доставку деталей. На останньому етапі ланцюжка постачання ці деталі інтегруються виробниками в транспортні засоби, підкреслюючи важливість безперебійної координації на кожному кроці.

Головна мета цього ланцюжка – забезпечити доступність кожного необхідного компонента в потрібному місці та в потрібний час, узгоджуючи його з виробничими графіками, щоб запобігти затримкам. Збій у будь-якій точці ланцюга, наприклад, відсутність окремої деталі, може призупинити виробництво, що призведе до збільшення операційних витрат, порушення термінів і потенційної втрати доходу. Це підкреслює критичну роль ефективного управління ланцюгом постачання (SCM), яке передбачає не лише забезпечення наявності запчастин, але й оптимізацію процесів для мінімізації неефективності та ризиків.

Управління ланцюгом постачання автозапчастин включає низку взаємопов'язаних дій, кожна з яких має вирішальне значення для загальної ефективності. Прогнозування попиту відіграє ключову роль у прогнозуванні потреби в конкретних запчастинах, дозволяючи виробникам підготуватися заздалегідь. Управління запасами забезпечує збалансованість запасів, щоб уникнути надмірних

запасів або дефіциту, тоді як координація логістики зосереджена на своєчасному транспортуванні деталей у різні регіони. Управління відносинами з постачальниками є ще одним важливим аспектом, який сприяє співпраці та надійності між різними постачальниками для забезпечення безперебійної роботи.

В останні роки автомобільна промисловість трансформувалася через глобалізацію та швидкий технологічний прогрес [2]. У результаті ланцюжки постачання стають дедалі складнішими, часто залучаючи постачальників із різних регіонів світу. Ця глобалізація створює такі виклики, як керування кількома постачальниками, робота з різними валютами, навігація міжнародними нормативними рамками та підтримка незмінних стандартів якості за кордоном. Для вирішення цих складнощів застосовуються більш витончені підходи до управління, розширені інструменти зв'язку та передові технології, такі як штучний інтелект і блокчейн, що гарантує, що сучасні ланцюжки постачання залишаються стійкими, гнучкими та ефективними на глобальному ринку, що швидко змінюється.

## **1.2 Огляд та аналіз комерційних рішень**

Багато компаній покладаються на комерційні системи управління ланцюжками постачання для управління своїми операціями. Ці системи, як правило, є комплексними програмними рішеннями, які інтегрують різні функції (рис. 1.1), включаючи закупівлю, управління запасами, логістику та координацію постачальників.

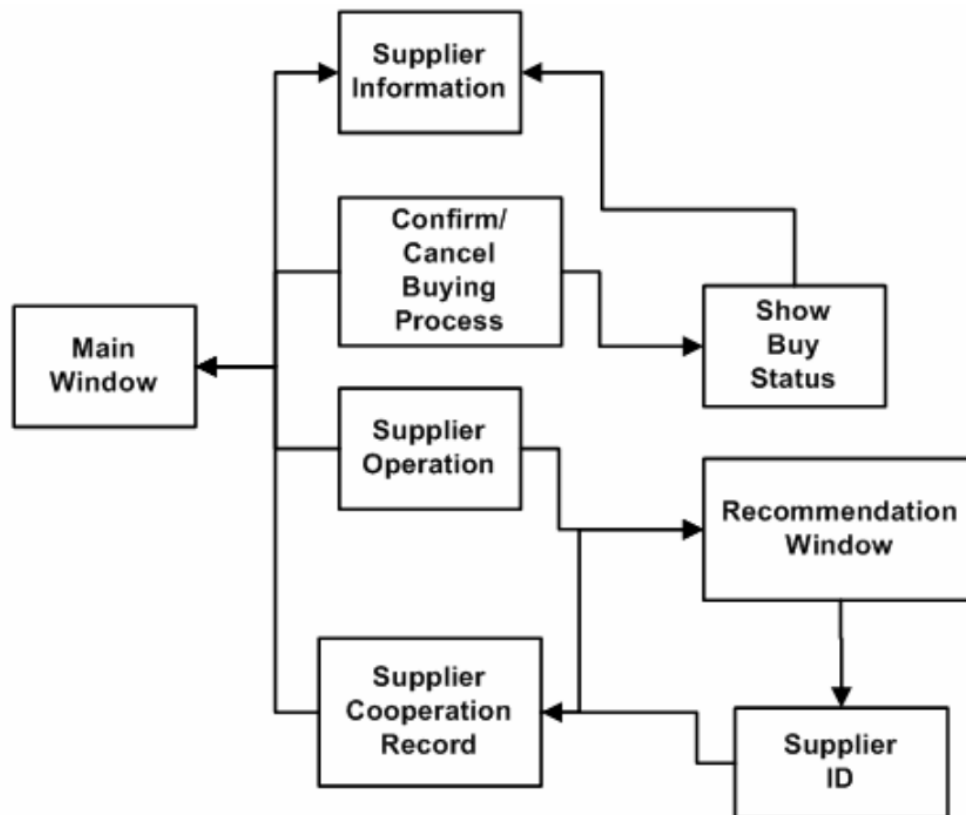


Рисунок 1.1 – Діаграма потоку інтерфейсу системи управління ланцюжками постачання

Системи з процесами, подібними до зображених на схемі, зазвичай зустрічаються в системах управління ланцюгом постачання, планування ресурсів підприємства і систем управління закупівлями. Ці системи розроблені для оптимізації та автоматизації складних взаємодій між різними зацікавленими сторонами, такими як постачальники, покупці та особи, які приймають рішення, з метою покращення операційної ефективності та прийняття рішень.

На глобальному рівні такі системи спрямовані на управління життєвим циклом відносин з постачальниками. Вони об'єднують такі функції, як адаптація постачальників, закупівлі, відстеження замовлень, оцінка ефективності та співпраця. Взаємозв'язок між цими процесами забезпечує безперебійний потік інформації між модулями, зменшуючи ризик неправильного спілкування або затримок. Наприклад,

рішення про покупку в «Процесі підтвердження/скасування покупки» може спиратися на інформацію з «Запису про співпрацю постачальників» і рекомендації, отримані на основі попередніх показників ефективності або рейтингів постачальників.

Подібні системи особливо цінні в галузях зі складними ланцюгами постачання, наприклад у виробництві, роздрібній торгівлі та логістиці. Вони надають централізовані платформи, на яких підприємства можуть відстежувати діяльність із закупівель, керувати відносинами з постачальниками та оптимізувати стратегії закупівель. Поєднуючи всі відповідні процеси, ці системи покращують видимість, зменшують кількість помилок, що виникають вручну, і дозволяють приймати рішення на основі даних. Крім того, їх можна масштабувати, щоб включити глобальних постачальників, вимоги дотримання нормативних вимог і аналітику в режимі реального часу, що є критично важливим для компаній, які працюють у глобалізованій економіці.

По суті, такі системи сприяють більшій співпраці, звітності та ефективності, дозволяючи організаціям досягати своїх операційних цілей, зберігаючи при цьому сильні та надійні мережі постачальників.

Деякі з найпопулярніших систем в автомобільній промисловості включають «SAP Supply Chain Management» [3]. Це програмне забезпечення було розроблено «SAP SE», німецькою багатонаціональною корпорацією програмного забезпечення, заснованою в 1972 році. «SAP SE» відома тим, що створює корпоративне програмне забезпечення для управління бізнес-операціями та відносинами з клієнтами. Широко використовується для планування ресурсів підприємства, яке пропонує інструменти для управління всіма аспектами ЛП, від закупівель до доставки. Воно допомагає компаніям оптимізувати свої запаси, оптимізувати процеси закупівель та покращити загальну прозорість доставки.

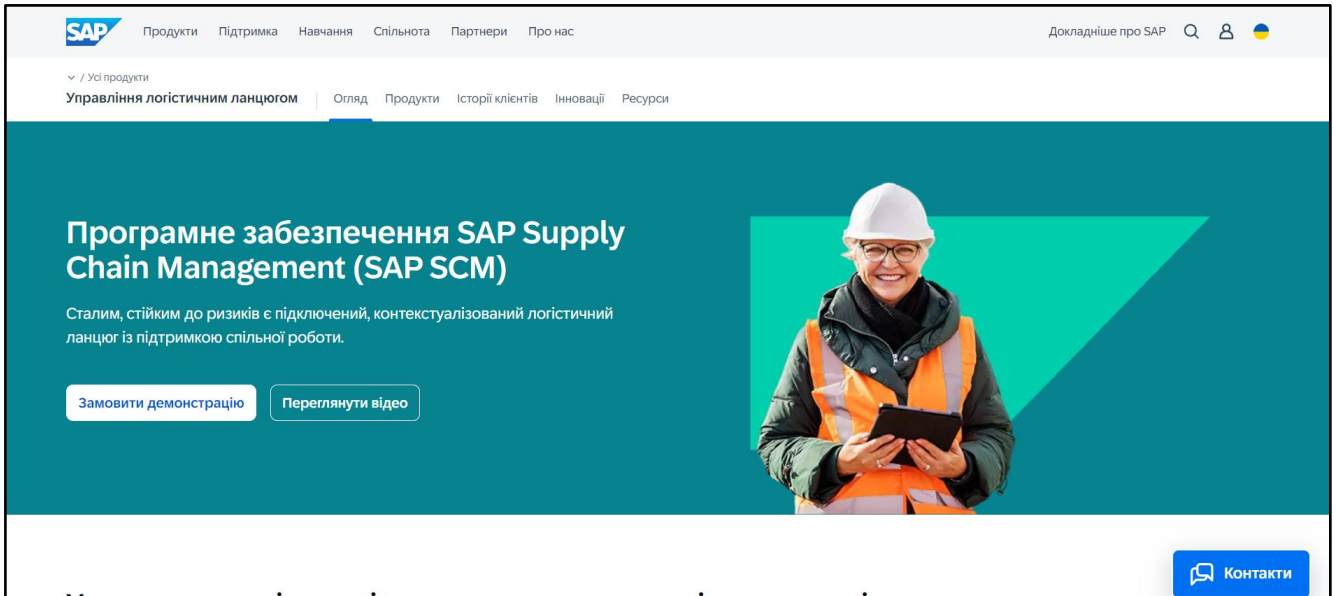


Рисунок 1.2 – Головна сторінка продукту «SAP SCM»

Також існує «Oracle SCM Cloud», хмарне рішення від корпорації «Oracle», яке надає набір програм для керування постачаннями від А до Я [4]. На рис. 1.3 зображено сторінку їхнього продукту, яка досить схожа на попередній приклад, тому неважко здогадатися, що системи досить схожі у своєму функціоналі. Їх рішення також включає модулі для планування постачаннями, управління запасами, управління замовленнями та логістикою в цілому. Oracle SCM Cloud відома своєю здатністю інтегруватися з іншими корпоративними системами, пропонуючи безперебійний потік інформації між різними відділами.

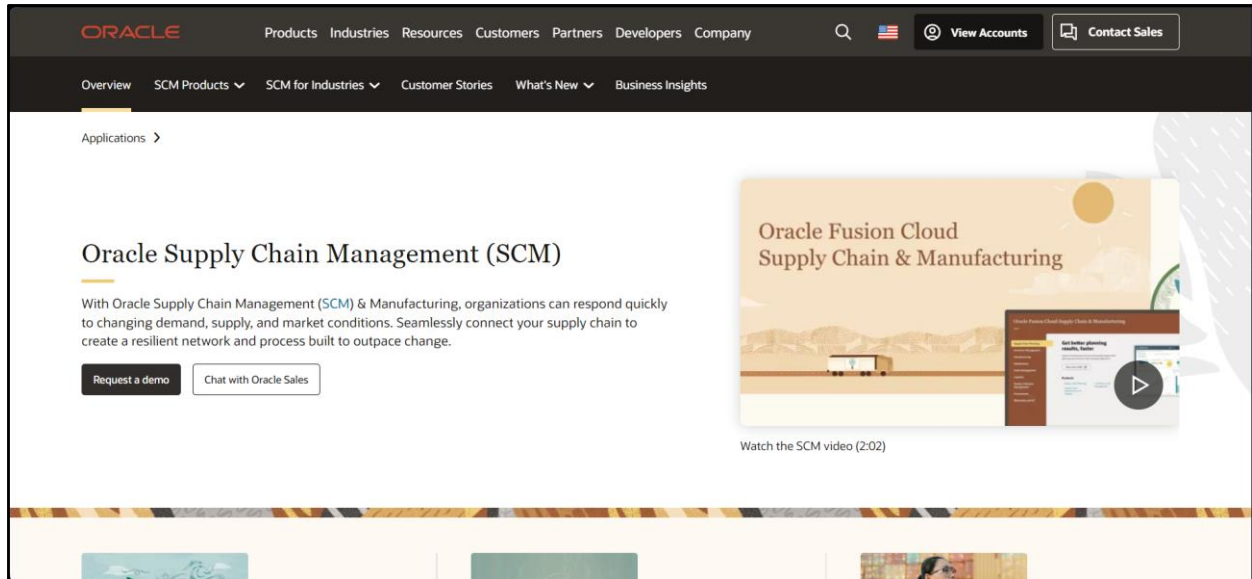


Рисунок 1.3 – Головна сторінка продукту «Oracle SCM»

«Blue Yonder» Supply Chain Management – спеціалізоване програмне забезпечення SCM, орієнтоване на планування попиту та постачання, керування транспортуванням та керування складом [5]. Воно особливо популярно завдяки своїм розширеним аналітичним можливостям, які допомагають компаніям приймати обґрунтовані рішення на основі даних реального часу.

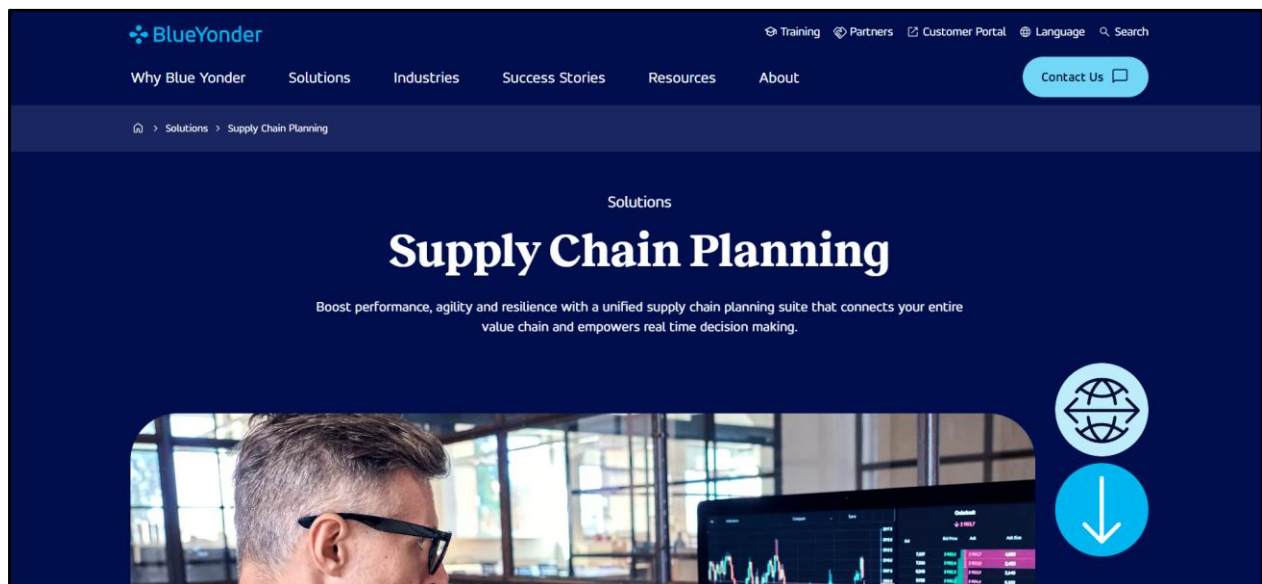


Рисунок 1.4 – Головна сторінка продукту «Blue Yonder»



Остання система для розгляду, «Infor Supply Chain Management», пропонує низку інструментів як попередні приклади [6]. Також відома своїм зручним інтерфейсом та інтеграцією з іншими корпоративними системами.

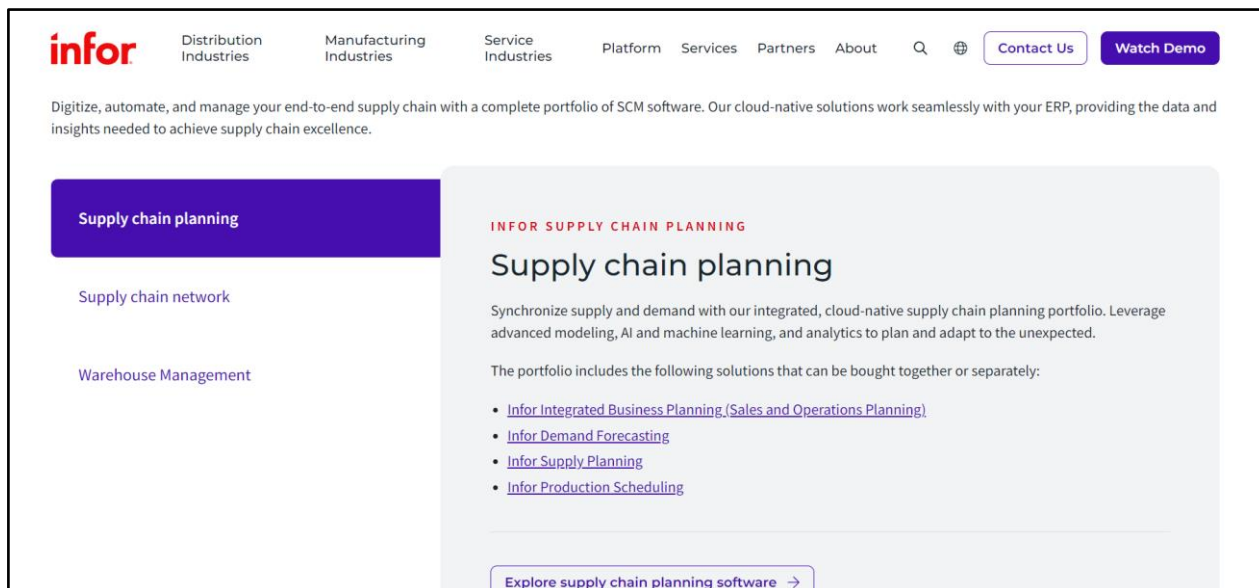


Рисунок 1.5 – Головна сторінка продукту «Infor»

Усі ці комерційні системи розроблені для обробки складнощів сучасних задач щодо логістики та постачання. Однак вони можуть бути дорогими у реалізації та обслуговуванні, особливо для малих та середніх підприємств.

### 1.3 Огляд та аналіз наявних публікацій

Крім комерційних систем, існує безліч академічних досліджень, спрямованих на оптимізацію управління ЛП. У цих дослідженнях часто вивчаються складні алгоритми, які можна інтегрувати в динамічну ІС.

У роботі авторів Anie Lusiani, Euis Sartika, Agus Binarto, Endang Habinuddin та Irfani Azis представляють дослідження для визначення найшвидшого шляху в логістичній дистрибуції логістики з використанням алгоритму Дейкстри [7]. Оптимізація шляху розподілу логістики з використанням алгоритмів Дейкстри та

Прима призводить до того самого найшвидшого маршруту від Чібітунга до Чіанжура, який займає 154 хвилини через Чікампек, Саданг, Чікалонг і Чіпатат, підтверджуючи, що алгоритм Дейкстри є ефективним методом оптимізації шляху.

Дослідження Tarek Sadraoui та Nejib Mchirgui, яке зосереджено на розробці інтегрованої інформаційної системи для управління (ЛП), наголошує на потребі в даних у реальному часі та міцній мережі постачальник-клієнт для розв'язання проблеми глобальної конкуренції та наростаючих вимог клієнтів [8]. Результати показують, що нелінійні методи прогнозування, зокрема нейронні мережі RBF, перевершили лінійні методи, що призвело до кращої точності прогнозування та підвищення ефективності управління ЛП.

Розглянуто в публікації [9], автором якої є Sherif A. Masoud оптимізацію двоетапної задачі автомобільного ЛП, що включає виробництво у постачальника рівня 1 і транспортування до розподільних центрів за допомогою моделей лінійного програмування зі змішаним цілим числом (MILP). Результати демонструють, що запропонований гібридний метаевристичний підхід, який включає схеми евристичних збурень, забезпечує ефективні рішення, майже оптимальні, зі значними покращеннями обчислювальної продуктивності зі збільшенням кількості типів деталей, і пропонує цінну інформацію щодо зниження витрат за допомогою кількох варіантів транспортування.

Дослідження авторів Nelson Chibeles-Martins, Tânia Pinto-Varela, Ana P. Barbosa-Róvoa та Augusto Q. Novais зосереджено на розробці та плануванні зелених ЛП з урахуванням як економічних, так і екологічних цілей [10]. Результати показують, що підхід Bi-objective Simulated Annealing (MBSA) є ефективною та потужною альтернативою точним методам, допомагаючи визначити межу Парето для проєктування та планування мережі постачання, хоча рішення є приблизними та потребують налаштування для кожного нового екземпляра проблеми.

Автори Florin Gheorghe Filip та Luminita Duta розглянули складну теорію прийняття рішень, яку можна використовувати для розв'язування проблем і демонстрації стійкості та інтеграційних рішень для зелених технологій [11]. Результати підкреслюють потребу в системах спільних рішень, які можуть впоратися з мінливістю та невизначеністю в управлінні зворотним ЛП, керованими такими технологіями, як хмарні обчислення, великі дані та кіберфізичні системи.

Наступні автори Abdolhossein Sadrnia, Napsiah Ismail, Norzima Zulkifli, M. K. A. Ariffin, Hossein Nezamabadi-pour та Hamed Mirabi у своїй роботі зосередили увагу на оцінці кількісного співвідношення між викидами CO<sub>2</sub> та витратами на логістику в мережі автомобільного ЛП за допомогою багатоцільового лінійного програмування з екологічними міркуваннями [12]. Результати показують, що модель, розв'язана за допомогою алгоритму MOGSA, надає набір оптимальних рішень за Парето для осіб, які приймають рішення, демонструючи компроміс між вищими витратами на логістику та більшим скороченням викидів CO<sub>2</sub>, з потенційними майбутніми розширеннями для включення зворотної логістики та більш просунутими методами оптимізації.

Також розглянуто публікацію [13], авторами якої є JiSun Shinn, Sungshin Kimn та Jang-Myuung Leen зосереджену на побудові моделі динамічної Баєсової мережі (DBN) для управління виробничими запасами на складальній лінії автозапчастин, спрямованої на управління нерегулярними змінами обсягу поставленої продукції, обсягу виробництва та обсягу запасів. Результати демонструють, що запропонована модель ефективно визначає причини змін обсягу, оптимально коригує виробничі плани для дотримання термінів доставки та підтримує оптимальні рівні запасів, що призводить до зниження витрат на управління запасами порівнюючи сценарії до і після коригування.

Автори Majid Alimohammadi Ardakani та Mehdi Kabiri Naeini публікації [14] сфокусувалися на розробці моделі інтегрованої конкурентоспроможної мережі ЛП,

яка функціонує як багаторівнева, багатопродуктова та багатоперіодна динамічна замкнута система, спрямована на усунення проблем, пов'язаних з обмеженими можливостями та невизначеним попитом і поверненням продуктів. Результати показують, що запропонована модель багатоцільового змішаного цілочисельного нелінійного програмування (MINLP) ефективно мінімізує витрати, максимізує якість продукції та вибирає найбільш бажаних постачальників. Однак ефективність моделі зменшується з розміром проблеми, що спричиняє необхідність використання метаевристичних алгоритмів, таких як PSO, TGA та RDA, для розв'язання більших випадків. Алгоритм TGA демонструє чудову продуктивність як за часом центрального процесора, так і за вартістю для середніх і великих моделей.

Наступне дослідження авторів Junjie Gao, Feng Gu, Ping Hu, Yanan Xie та Baozhen Yao має на меті розв'язати проблему доставки запчастин для технічного обслуговування автомобільних вузлів та мінімізувати загальну відстань доставки, забезпечуючи при цьому прибуття транспортних засобів на станції технічного обслуговування в необхідні терміни [15]. Результати вказують на те, що завдяки застосуванню підходу віртуального центру депо та використання адаптивного алгоритму оптимізації мурашиної колонії (IACO) складність проблеми ефективно зменшується, створюючи відповідні проєкти маршрутів доставки для депо технічного обслуговування в Даляні, що підвищує конкурентоспроможність і задоволеність клієнтів; однак у дослідженні визнаються обмеження, оскільки враховуються лише витрати на транспортування та час у дорозі залежно від відстані, що свідчить про те, що майбутня робота включатиме додаткові чинники реального життя, такі як умови дорожнього руху, для покращення продуктивності моделі.

Seyed Ali Alavikia, Mohammad Taghi TaghaviFard, Maghsoud Amiri та Parham Azimi описали розробку кількісного підходу до управління змінами даних і створення прогностичних та реактивних рішень для пом'якшення в рамках багатопродуктового ланцюга постачання [16]. Результати демонструють, що запропоновані евристичні

методи ефективно дозволяють організаціям переглядати плани ЛП для відомих змін даних, прогнозувати майбутні зміни попиту та швидко адаптуватися до раптових перебоїв у виробництві, що покращує процес прийняття рішень і мінімізує витрати в трирівневому ЛП стратегічних автозапчастин. Ця модель застосовна в різних галузях промисловості, підкреслюючи необхідність адаптації компаній до мінливих ринкових умов і вимог клієнтів для збереження конкурентоспроможності та частки ринку.

Автори Enderson Luiz Pereira Júnior, Miguel Ângelo Lellis Morreira, Carlos Francisco Simões Gomes, Marcos dos Santos, Arthur Pinheiro de Araújo Costa, Stephanye dos Santos Sbrano Chagas, Igor Pinheiro de Araújo Costa, та Emerson Hissao Kojima розглядають проблему оцінку та вибір постачальників автозапчастин у контексті ЛП [17]. Вони використовують метод прийняття рішень під назвою CRITIC-GRA3N для визначення рейтингу різних постачальників на основі різних критеріїв, таких як продуктивність і надійність. Висновки показують, що компанія 1 займає перше місце в рейтингу постачальників, за нею йде компанія 2 як другорядний варіант, що підкреслює їхню важливість у ЛП. Дослідження наголошує на необхідності ретельного розгляду альтернативних постачальників, зокрема Компаній 3, 4 і 5, які демонструють різні рейтинги, що вимагає пильнішої уваги з боку осіб, які приймають рішення, у випадку, якщо основні постачальники недоступні.

Benjamin Rolf, Ilya Jackson, Marcel Müller, Sebastian Lang, Tobias Reggelin та Dmitry Ivanov у своєму дослідженні провели напівсистематичний огляд для вивчення розробки, алгоритмів, застосувань і практичного впровадження навчання з підкріпленням (RL) у сфері управління ЛП (SCM), охоплюючи дослідження з 2000 року [18]. Отримані дані свідчать про стійкий інтерес до додатків RL до 2019 року, після чого відбулося значне збільшення кількості публікацій завдяки вдосконаленню обчислювального обладнання, збільшенню обсягів даних і поширенню методів глибокого навчання. Дослідження пропонує ієрархічну структуру класифікації для класифікації додатків RL у SCM на основі чотирьох ключових критеріїв: драйвери

ЛП, алгоритми, джерела даних і промислові сектори, пропонуючи структурований спосіб зрозуміти висхідну роль RL в оптимізації ЛП.

В опублікованому дослідженні автори Ao Lv та Baofeng Sun запропонували уніфіковану структуру, яка усуває проблему маршрутизації розташування-інвентаризації (LIRP) в логістиці постачання автозапчастин, враховуючи як економічну ефективність, так і стійкість, використовуючи модель багатоцільового змішаного цілочисельного програмування (MIP) [19]. Результати демонструють, що модель ефективно знижує системні витрати та викиди вуглекислого газу, значно покращує використання потужностей транспортних засобів, а також підкреслює, що рішення щодо розташування, запасів і маршрутів дуже чутливі до параметрів вартості, пропонуючи цінну інформацію для оптимізації автомобільних ЛП.

Автори Konstantinos I. Nikolopoulos, M. Zied Babai та Konstantinos Bozos у своїй роботі запропонували концептуальну основу для застосування метода NN (найближчого сусіда) до даних ЛП і дослідили умови, за яких методи NN працюють ефективно [20]. Результати показують, що хоча підходи NN можуть покращити прогнозування спорадичних часових рядів, якщо застосовувати їх вибірково, неконтрольоване використання великих наборів даних може призвести до погіршення точності прогнозування на 12,88%, якщо закономірності не будуть виявлені.

У роботі авторів Pourya Pourhejazy та Oh Kyoung Kwon представлено дослідження, в якому розглядається перехід від звичайного математичного моделювання в оптимізації ланцюга постачання до інтегрованого математичного моделювання та імітаційно-оптимізаційних структур (S-O), які стають дедалі більш важливою областю досліджень [21]. Вони роблять висновок, що структури S-O можуть служити альтернативою інтегрованим моделям, більш ефективно розв'язувати проблеми реального світу шляхом спрощення обчислень, зменшення припущень і включення стохастичних середовищ, хоча для цього необхідна подальша емпірична перевірка.

Автори, Giannakis, Spanaki, and Dubey, представили детальну структуру екосистеми хмарного управління ланцюгами постачання (C-SCM), призначену для покращення реагування ланцюгів постачання (SCR) шляхом використання хмарних обчислень для підвищення прозорості, гнучкості та здатності виявляти зміни та швидко реагувати на них [22]. Результати дослідження показали, що запропонована система C-SCM значно покращує всі три виміри SCR, надаючи цінну інформацію та практичні рекомендації для компаній, які прагнуть перевести свої операції з управління ланцюгами постачання у хмарне середовище.

У наступному дослідженні, проведеному Oana Dumitrascu, Manuel Dumitrascu, та Dan Dobrotă, запропоновано модель оцінки ефективності управління ланцюгами постачання (SCM) на основі алгоритму штучного інтелекту багат шарового перцептрона (MLP) [23]. Модель пов'язує конкретні операційні проблеми з найбільш релевантними ключовими показниками ефективності (KPI), що дозволяє організаціям усувати проблеми, підвищувати ефективність та покращувати управління ризиками. Структуруючи SCM на підсистеми та визначаючи ключові KPI для кожної з них, модель надає цілеспрямовану інформацію для моніторингу ефективності та забезпечення сталого розвитку.

Результати дослідження підтвердили здатність моделі прогнозувати ключові показники ефективності для розв'язання конкретних проблем, підкресливши переваги дашбордів KPI для проактивного управління ризиками та досягнення стратегічних цілей. У дослідженні рекомендується розробити спільну платформу для використання анонімних даних з різних організацій, що підвищить точність і надійність прогнозування KPI. Крім того, майбутні дослідження повинні зосередитися на одночасній оцінці кількох KPI для більш комплексного аналізу. Цей підхід пропонує цінний прогрес у використанні інструментів на основі штучного інтелекту для оптимізації ефективності SCM та забезпечення стійкості.

Дослідження Mohsen Shafiei Nikabadi розробило структуру технологічних факторів в управлінні знаннями в ланцюгах постачання Iran Khodro і Saipa, двох великих іранських виробників автомобілів [24]. Дослідження визначило три ключові фактори: інтеграція інформаційних систем (0,87), інструменти інформаційних технологій (0,85) та управління інформаційною безпекою (0,68), причому найбільше значення надається інтеграції ІТ-систем та мереж. Дослідження також підкреслило, що системи підтримки прийняття рішень та експертні системи, а також інструменти для вилучення даних мають вирішальне значення для ефективного управління знаннями.

Попри обґрунтованість концепції, такі проблеми, як недостатня технологічна зрілість, погане розуміння цінності технологій і недостатня політика в галузі електронного бізнесу, перешкоджали їй успішному впровадженню. Автори дослідження дійшли висновку, що подолання цих проблем є важливим для покращення управління знаннями, і рекомендували провести подальші дослідження для застосування концепції в інших галузях та регіонах.

Дослідження Manisha Seth, D.P. Goyal, та Ravi Kiran досліджує фактори успішного впровадження інформаційних систем управління ланцюгами постачання (SCMIS) в автомобільній промисловості Індії [25]. На основі літературних джерел та думок експертів було визначено 18 критичних змінних успіху та запропоновано модель успішного впровадження SCMIS. Дані, отримані від керівників, постачальників і дистриб'юторів у Північному регіоні Індії, були проаналізовані за допомогою дослідницького факторного аналізу та регресії.

Основні висновки включають важливість підтримки вищого керівництва, залучення користувачів, аналізу перед впровадженням, навчання, управління змінами, точності даних під час конвертації та ефективною комунікації для успіху SCMIS. У дослідженні робиться висновок, що для ефективного впровадження SCMIS необхідно



враховувати технічні, організаційні та людські фактори, підкреслюючи, що такі системи слід розглядати як організаційні перетворення, а не просто ІТ-проекти.

У дослідженні Andries Botha, Jacomine Grobler, та V.S. Sarma Yadavalli оцінюють підходи до управління запасами в ланцюгу постачання автомобільних запчастин, зосереджуючи увагу на доступності запчастин та рівнях запасів [26]. У дослідженні порівнюється чинний метод максимального рівня запасів (MIP) з нещодавно розробленим методом встановлення цільового рівня запасів (STS). Основна мета – оцінити теоретичні та практичні відмінності між цими двома методами та їх ефективність у досягненні цільового рівня заповнення запасів (AFR) на рівні 95,5% при збереженні низького рівня запасів.

Було створено імітаційну модель системної динаміки (SDSM) для моделювання як місцевих, так і імпортованих ланцюгів постачання, що дозволило порівняти методи MIP та STS. Дослідження показало, що метод STS, який включає коефіцієнт демпфування (DF) для запобігання ефекту «батога і пряника», забезпечує кращу амплітудно-частотну характеристику порівняно з теоретичним методом MIP, водночас вимагаючи менших запасів, ніж фактичний метод MIP (MIPActual). Для деталей, що швидко рухаються, обидва методи досягають цільової частоти обертання, але STS у більшості випадків вимагає менших запасів. Для деталей із середньою швидкістю переміщення STS у кількох випадках покращив AFR, водночас зменшивши запаси. Для деталей з повільним рухом STS покращила AFR при меншому збільшенні запасів порівняно з MIPActual. Однак для деталей, що рухаються нестабільно, STS, як правило, вимагав менших або однакових запасів, але не був настільки ефективним у досягненні AFR, як MIPActual.

Отже, метод STS є перспективною альтернативою MIP для управління запасами в автомобільному ланцюгу постачання. Він забезпечує зменшення запасів при збереженні задовільного AFR. Проте, необхідні подальші дослідження для оцінки

його ефективності при різних моделях попиту та підтвердження його довгострокових переваг.

Робота Jie Yao та Wenwen Guo зосереджена на вирішенні проблеми оптимізації запасів у ланцюгу постачання автомобілів, особливо в умовах гострої конкуренції на автомобільному ринку [27]. У дослідженні підкреслюється важливість зниження витрат як у компанії, так і в ланцюзі постачання. Автори розробили математичну модель багаторівневої оптимізації запасів, показуючи, як цей підхід дозволяє мінімізувати рівень запасів у ланцюгу постачання.

Під час аналізу прикладів дослідження вивчається компанія M, яка виробляє запчастини, що постачаються компаніям S і T. Обидві компанії продають по 50 одиниць щодня, вартість зберігання однієї одиниці запасів становить 2 юані на день, а вартість дефіциту – 20 юанів за одиницю. Вартість трансакції складає 2000 юанів за замовлення. На основі цих параметрів визначено, що час циклу замовлення для компанії M становить 4,7 дня, з початковим запасом 426 одиниць і дефіцитом 9,36 одиниць на одиницю часу.

Висновок дослідження показує, що багаторівнева модель оптимізації запасів успішно усуває проблеми з запасами, що виникають через зміни в інформації про попит. Це дозволяє оптимізувати загальні запаси в усьому ланцюгу постачання, що, в свою чергу, підвищує швидкість обороту капіталу та конкурентоспроможність підприємств.

#### **1.4 Огляд наявних технологій для розв'язання поставленої задачі**

MongoDB – це документоорієнтована система керування базами даних NoSQL з відкритим вихідним кодом, яка зберігає дані в гнучкому, схожому на JSON форматі, відомому як BSON (Binary JSON). На відміну від традиційних реляційних баз даних, MongoDB не використовує таблиці з рядками та стовпцями для зберігання даних, а

використовує колекцію документів, які можуть мати складну ієрархічну структуру та можуть бути індексовані й отримані за різними критеріями.

Node.js – це міжплатформове середовище виконання JavaScript-коду, яке дозволяє розробникам створювати швидкі, масштабовані та високопродуктивні мережеві програми. Воно використовує керовану подіями модель вводу та виводу без блокування, що робить його ідеальним для створення застосунків у реальному часі, які мають обробляти велику кількість одночасних з'єднань. Середовище побудоване на основі двигуна V8 JavaScript, який також використовується в браузері Google Chrome. Node.js надає низку вбудованих модулів і бібліотек для роботи з операціями вводу та виводу, мережами, файловими системами тощо.

Express.js є фреймворком вебзастосунків для Node.js, що спрощує процес створення вебзастосунків і програмних інтерфейсів прикладного програмування (API). Він забезпечує набір надійних функцій та інструментів для створення вебзастосунків, включаючи маршрутизацію, проміжне програмне забезпечення та механізми шаблонів. Express.js надає потужну систему маршрутизації, яка дозволяє зіставляти HTTP-запити з відповідними обробниками коду. Крім того, він підтримує різноманітні механізми шаблонів, такі як Pug і Handlebars, для створення динамічних вебсторінок. Додатково Express.js надає проміжне програмне забезпечення для виконання типових завдань, таких як журналювання, розбір HTTP-запитів і обробка помилок.

Chart.js являє собою бібліотеку JavaScript з відкритим вихідним кодом для створення інтерактивних і візуально привабливих діаграм у вебдодатках. Він підтримує різні типи діаграм, такі як стовпчасті, лінійні, секторні, радарні та інші, що робить його універсальним для різноманітних потреб візуалізації даних. Chart.js легко інтегрувати з HTML5 Canvas, що забезпечує плавну візуалізацію та адаптивний дизайн для настільних і мобільних платформ. Бібліотека легко налаштовується, що дозволяє розробникам налаштовувати зовнішній вигляд діаграми, додавати анімацію

та використовувати плагіни для розширених функцій. Його інтуїтивно зрозумілий API робить його доступним як для початківців, так і для досвідчених розробників.

R – це мова програмування та середовище, розроблене для статистичних обчислень і графіки. Він відмінно підходить для аналізу даних, візуалізації та звітності. R має багату екосистему пакетів, таких як `dplyr` для обробки даних, `ggplot2` для візуалізації даних і `caret` для машинного навчання. Завдяки вбудованій підтримці різних типів даних і статистичних методів R широко використовується в академічних дослідженнях, науці про дані та в галузях, які потребують розширеної обробки й аналізу даних.

Бібліотека R, `Plumber API`, дозволяє користувачам перетворювати код R на RESTful API, додаючи прості анотації. Це спрощує розкриття статистичних і аналітичних можливостей R за допомогою вебслужб. Це особливо корисно для додатків, які вимагають інтеграції обробки даних R із сучасними вебтехнологіями, що забезпечують зв'язок між обчисленнями на основі R та вебінтерфейсами.

Ajax (асинхронний JavaScript і XML) – це набір методів веброзробки, які використовуються для створення асинхронних вебдодатків. Це дозволяє обмінюватися даними між клієнтом і сервером без оновлення сторінки. Ajax покращує взаємодію з користувачем, роблячи вебсторінки більш динамічними та чуйними, забезпечуючи взаємодію з даними в реальному часі. Він зазвичай використовується в сучасних вебдодатках для асинхронного завантаження даних або надсилання форм.

JSON (JavaScript Object Notation) являє собою формат даних, який використовується для представлення структурованих даних. Його легко читати людині та легко аналізувати та генерувати машинами, що робить його популярним вибором для обміну даними між клієнтами та серверами. JSON не залежить від мови, що дозволяє використовувати його з різними середовищами програмування. Він широко використовується в API, файлах конфігурації та сучасній веброзробці завдяки своїй простоті та гнучкості.

Leaflet.js – це легка бібліотека JavaScript із відкритим вихідним кодом, яка використовується для створення інтерактивних та адаптивних карт. Вона популярна своєю простотою, пропонуючи такі функції, як масштабування, маркери, спливаючі вікна та користувальницькі накладення, а також розширюваний за допомогою плагінів. Leaflet легко інтегрується з постачальниками плиток карт, такими як OpenStreetMap і Mapbox, що робить його універсальним для програм з картами. Його орієнтація на продуктивність і мобільну сумісність зробила його кращим вибором для розробників у всьому світі.

OpenStreetMap (OSM) – це платформа, керована спільнотою, яка надає безплатні географічні дані та карти, які можна редагувати. Користувачі роблять свій внесок, наносячи на карти дороги, орієнтири та інші географічні об'єкти, забезпечуючи постійне оновлення та точну інформацію. Відкрита ліцензія OSM дозволяє розробникам інтегрувати її дані в додатки для навігації, аналізу або спеціальної візуалізації. Платформа підтримує кілька стилів карт і широко використовується як у комерційних, так і в гуманітарних проєктах.

REST API або Representational State Transfer API – це архітектура вебслужби, яка забезпечує зв'язок між клієнтами та серверами через HTTP. Вона дозволяє користувачам виконувати такі операції, як отримання, створення, оновлення або видалення даних за допомогою стандартних методів HTTP, таких як GET, POST, PUT і DELETE. API REST не мають стану, тобто кожен запит є незалежним і містить всю необхідну інформацію, що робить їх масштабованими та простими для інтеграції. Вони широко поширені завдяки своїй простоті, гнучкості та сумісності з різними мовами програмування та фреймворками.

OSRM (Open Source Routing Machine) – це високопродуктивний механізм маршрутизації, призначений для розрахунку найшвидших і найкоротших шляхів у дорожніх мережах. Він використовує дані OpenStreetMap для забезпечення точної та ефективної маршрутизації для різних видів транспорту, таких як автомобільний,

пішохідний або велосипедний. OSRM широко використовується в додатках, що вимагають розрахунку маршрутів у реальному часі, пропонуючи розширені функції, такі як покрокова навігація та настроювані профілі. Масштабованість і швидкість роботи роблять його придатним як для великомасштабних додатків, так і для індивідуальних проєктів.

Nominatim використовується як потужний інструмент геокодування та пошукова система для даних OpenStreetMap (OSM). Він дозволяє користувачам перетворювати адреси в географічні координати (геокодування) і навпаки (зворотне геокодування), полегшуючи роботу з додатками, що базуються на визначенні місцезнаходження. Nominatim широко використовується для таких завдань, як пошук об'єктів, перевірка адрес або пошук адміністративних кордонів. Він має відкритий вихідний код, добре налаштовується і підтримує кілька мов, що робить його придатним для глобальних застосувань.

## **1.5 Постановка задачі**

Ефективне управління ЛП стало ключовим аспектом сучасних бізнес-операцій, особливо в автомобільній промисловості. Зростаюча складність глобальних ланцюгів у поєднанні зі зростаючим попитом на своєчасну доставку автозапчастин викликала необхідність розробки інноваційних рішень.

Актуальність оптимізації ЛП підкреслюється у звіті [28] Future Market Insights, який вказує на те, що прогнозується, що глобальний ринок управління в цій області досягне приблизно 78,5 мільярдів доларів США до 2033 року, з середньорічним темпом зростання (CAGR) на 11,8% з 2023 по 2033 рік.

У звіті підкреслюється, що ринок управління SCM (система управління ланцюгами постачання) розвивається через зростаючий попит на прозорість і видимість процесів. Крім того, швидке розширення секторів роздрібної торгівлі та електронної комерції загострює конкуренцію, штовхаючи компанії на впровадження

передових рішень SCM. Розвиток Інтернету речей (IoT) ще більше підвищує ефективність складування, виробництва, логістики та управління транспортуванням. Нарешті, збільшення інвестицій як з боку державного, так і приватного секторів сприяє розвитку інфраструктури, підтримуючи загальне зростання ринку.

Тема дослідження – створення інформаційної системи (ІС), призначеної для покращення оптимізації та управління логістикою у сфері автозапчастин за допомогою сучасних технологій та інтеграції наявних алгоритмів. Завдяки інтеграції API (інтерфейсу програмування додатків) та карти міста ця система надасть користувачам можливість вести облік товарів, оптимізувати маршрути доставки та аналізувати ефективність логістичних операцій. Цей підхід спрямований на сприяння спілкуванню та співпраці в режимі реального часу між зацікавленими сторонами ЛП, покращуючи процеси прийняття рішень.

Головна проблема полягає у тому, що чинні методи та інструменти управління логістикою є або дорогими для звичайного користувача, або не забезпечують повноцінного та ефективного управління з використанням сучасних методів та технологій. Багато доступних програм не інтегрують алгоритми оптимізації маршруту або не забезпечують інструменти інтерактивного відображення, що призводить до неоптимальної продуктивності та збільшення операційних витрат. Система, яка поєднує в собі ці елементи, дозволить користувачам приймати обґрунтовані рішення та покращить їхні можливості управління ЛП.

Використовуючи сучасні технології, такі як Plumber API, Node.js, Leaflet.js, це дослідження спрямоване на створення надійної системи для оптимізації ланцюгів постачання автозапчастин. Запропонована програма дозволить користувачам моделювати різні сценарії доставки, аналізувати їх результати та вдосконалювати свої логістичні стратегії.

**Завдання** для досягнення поставленої мети:

- проаналізувати наявні публікації та аналоги, що пов'язані з SCM, їхні функціональні можливості та недоліки;
- дослідити алгоритм Кларка-Райта та інші, які можуть бути використані для реалізації системи;
- створити архітектуру ІС, визначити її основні функції та представити їх за допомогою інформаційних діаграм;
- розробити систему для полегшення обліку автозапчастин і оптимізації маршрутів доставки з використанням даних у реальному часі;
- протестувати систему.

### **Висновки до розділу 1**

Аналіз, представлений у цьому розділі, підкреслює потребу в ефективному управлінні постачаннями в промисловості автомобільних запчастин, особливо з огляду на зростаючу складність виробництва та дистрибуції автомобілів. З розвитком технологій, ланцюги постачання автозапчастин дозволяють значно покращити загальну продуктивність і знизити витрати.

У цьому розділі досліджуються різні методи оптимізації процесів ланцюга постачання, підкреслюючи переваги використання передових алгоритмів і інструментів прийняття рішень на основі даних.

Проведений аналіз наявних рішень за яким можна зробити висновок, що для більш ефективного керування логістикою постачань потрібно більше інтерактивних і адаптованих елементів в інформаційній системі, зосереджених на аналізі даних у реальному часі та залученні користувачів.

Тому було вирішено розробити систему для планування та оптимізації ланцюгів постачання автозапчастин.



## **2 МЕТОДИ ТА ТЕХНОЛОГІЇ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН**

### **2.1 Інтеграція JavaScript і R для планування маршрутів в реальному часі**

Для підвищення ефективності планування маршрутів для ланцюгів постачання автозапчастин розроблено інформаційну панель інвентаризації в реальному часі з використанням JavaScript для інтерактивності та R для розширеного аналізу даних. Ця інтеграція використовує сильні сторони обох технологій для створення централізованої системи відстеження запасів, що дозволяє приймати обґрунтовані рішення щодо поповнення та замовлення.

Сервер, створений за допомогою R, взаємодіє з базою даних MongoDB для отримання поточних даних інвентаризації. Використовуючи Plumber API, R відкриває кінцеві точки, які дозволяють інтерфейсу зовнішнього інтерфейсу запитувати інформацію в реальному часі про рівні запасів, точки повторного замовлення та історичні тенденції продажів [29]. Ця архітектура забезпечує ефективну обробку даних і безперебійний зв'язок між можливостями аналізу даних R та інтерфейсом JavaScript.

У інтерфейсі JavaScript підтримує інтерактивні компоненти інформаційної панелі, використовуючи такі бібліотеки, як Chart.js і D3.js, для візуалізації даних, отриманих з API [30]. Ці візуалізації перетворюють складні набори даних на інтуїтивно зрозумілі діаграми та графіки, що дозволяє користувачам швидко зрозуміти тенденції та стан запасів. За допомогою викликів AJAX інтерфейс отримує оновлення з R API у режимі реального часу, усуваючи потребу в оновленні вручну та гарантуючи, що інформаційна панель завжди відображає останні інвентарні дані [31].

Ця інтеграція JavaScript і R підтримує проактивну систему управління запасами. Наприклад, коли рівень запасів падає нижче попередньо визначеного порогу, R запускає сповіщення через API, а інтерфейс JavaScript відображає сповіщення, щоб

сповістити користувача про дії, такі як зміна замовлення або коригування стратегії постачання. Такий підхід зменшує ризик дефіциту або надлишкових запасів, оптимізуючи витрати на запаси та покращуючи операційну ефективність.

Крім того, характер системи в режимі реального часу дозволяє користувачам динамічно реагувати на ринкові умови. Сервер R обробляє та аналізує тенденції даних, тоді як інтерфейс JavaScript забезпечує представлення цієї інформації в зручному інтерактивному форматі. Ця синергія дає можливість зацікавленим сторонам відстежувати коливання запасів і швидко й стратегічно адаптувати свої стратегії закупівель.

Таким чином, інтеграція JavaScript і R через Plumber API створює надійне рішення для керування запасами. JavaScript обробляє інтерфейс користувача та інтерактивні візуалізації, тоді як R забезпечує потужний аналіз даних і серверну обробку. Разом вони забезпечують оперативний, проникливий і ефективний інструмент для управління складними системами інвентаризації в ланцюгах постачання.

## **2.2 Методи оптимізації ланцюгів постачання автозапчастин**

Одним з ефективних підходів до оптимізації ланцюгів постачання автозапчастин є впровадження аналітики даних у реальному часі в поєднанні з вдосконаленими алгоритмами маршрутизації. Використовуючи дані з різних джерел, як-от рівень запасів, прогнози продажів і умови дорожнього руху, підприємства можуть приймати обґрунтовані рішення щодо управління запасами та логістичних операцій. Поширеним методом у цьому контексті є використання алгоритму економії Кларка-Райта, який розроблено для мінімізації транспортних витрат при оптимізації маршрутів доставки [32].

Алгоритм економії Кларка-Райта працює шляхом обчислення економії, отриманої в результаті консолідації постачання кільком клієнтам в один маршрут. Цей

метод починається з початкового рішення, коли кожен клієнт обслуговується індивідуально. Потім алгоритм ітеративно визначає пари клієнтів, які при спільному обслуговуванні дають найбільшу економію. Завдяки ефективному групуванню доставки алгоритм зменшує загальну відстань подорожі та пов'язані з цим витрати, що забезпечує більш ефективне рішення маршрутизації.

На додаток до оптимізації маршруту, ще одним важливим методом є впровадження систем моніторингу запасів у реальному часі. Ці системи використовують такі технології, як датчики IoT (Інтернет речей) і мітки RFID (радіочастотна ідентифікація), щоб відстежувати рівень запасів у реальному часі [33, 34]. Ці дані можна ввести в аналітичну модель на основі R, яка прогнозує попит і визначає оптимальні точки повторного замовлення, гарантуючи, що підприємства підтримують достатній рівень запасів без зайвих витрат на запаси.

Експоненціальне згладжування – популярний і простий метод, який використовується в аналізі часових рядів для прогнозування майбутнього попиту [35]. Це особливо корисно, коли дані дотримуються відносно стабільної тенденції без великих коливань або сезонності, що робить його добре придатним для ЛП.

Ключова ідея експоненціального згладжування полягає в тому, що воно надає більшій важливості останнім спостереженням, але разом з тим враховує минулі дані. Такий підхід дозволяє прогнозу краще реагувати на останні зміни в моделях попиту. Наприклад, якщо стався раптовий сплеск попиту на певні автозапчастини, метод швидко скорегує свій прогноз з урахуванням нової тенденції, забезпечуючи точніші прогнози.

Однією з причин, чому експоненціальне згладжування широко використовується в прогнозуванні попиту, є його простота та адаптивність. Для цього не потрібні великі історичні дані чи складні обчислення, що робить його ефективним для впровадження та запуску в режимі реального часу. У нашій системі ми застосовуємо експоненціальне згладжування до історичних даних про зміну запасів

автозапчастин, щоб прогнозувати майбутній попит. Це дозволяє компаніям приймати обґрунтовані рішення щодо рівня запасів, графіків поповнення запасів і планування постачання, гарантуючи, що у них немає ані надмірних, ані недостатніх запасів.

В нашій системі процес згладжування використовує постійне значення «альфа» 0,3, щоб визначити, яка вага надається останнім даним про запаси. Початковий згладжений запас встановлюється як перший запис про запас в історії товару, а потім кожне наступне значення запасу комбінується з попереднім згладженим значенням за формулою SES. Цей метод допомагає зменшити вплив короткострокових коливань рівня запасів, забезпечуючи більш стабільну картину запасів у часі. Згладжене значення запасів потім повертається користувачеві, допомагаючи в управлінні запасами.

Цей метод прогнозування особливо цінний у контексті ланцюгів постачання автозапчастин, де раптові зміни попиту можуть відбуватися через зовнішні фактори, такі як поломки транспортних засобів, ринкові тенденції чи економічні умови. Використовуючи експоненціальне згладжування, система гарантує, що прогнози попиту залишаються актуальними та допоміжними, дозволяючи підприємствам підвищити свою операційну ефективність і мінімізувати витрати, пов'язані з браком або надлишком запасів.

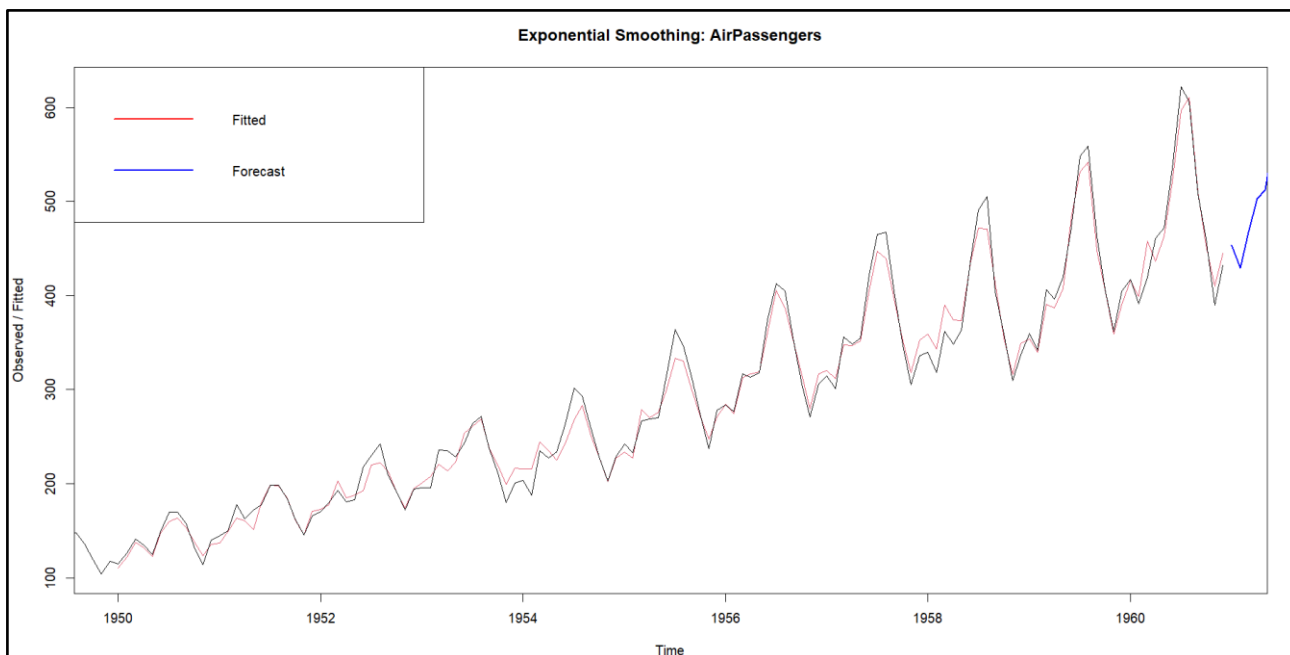


Рисунок 2.1 – Приклад оригінального часового ряду зі згладженими та прогнозованими значеннями з використанням експоненціального згладжування

### 2.3 Методи прогнозування: експоненціальне згладжування та метод Холта-Уінтерса

На відміну від простих ковзних середніх, які обробляють усі минулі дані однаково, експоненційне згладжування надає велику важливість останнім даним, що робить адаптованими до змін у тренді.

Основна формула для простого експоненціального згладжування (SES):

$$S_t = \alpha \cdot D_t + (1 - \alpha) \cdot S_{t-1}, \quad (2.1)$$

де  $S_t$  – згладжене значення (прогноз) на момент часу  $t$ ;

$D_t$  – фактичний попит у час  $t$ ;

$S_{t-1}$  – прогноз на попередній період;

$\alpha$  – коефіцієнт згладжування ( $0 < \alpha < 1$ ), який визначає, скільки ваги надається останньому спостереженню.

Коефіцієнт згладжування,  $a$ , відіграє вирішальну роль у тому, наскільки прогноз реагує на останні зміни. Більше значення  $a$  надає більше ваги нещодавньому попиту, роблячи прогноз більш чутливим до короткострокових коливань. Нижчий  $a$  з іншого боку, згладжує короткострокові коливання, зосереджуючись більше на довгострокових трендах.

Розглянемо приклад (табл. 2.1), коли ми прогнозуємо попит на автозапчастину протягом п'яти періодів часу, використовуючи початковий прогноз у 100 одиниць і коефіцієнт згладжування  $a = 0,5$ .

Таблиця 2.1 – Прогноз попиту на автозапчастину протягом п'яти періодів часу

Період часу ( $t$ )	Фактичний попит ( $D_t$ )	Прогноз ( $S_t$ )
1	100	100 (початковий)
2	120	110
3	130	120
4	90	105
5	110	107,5

Для періоду 1 ми починаємо з початкового прогнозу 100, який може базуватися на історичному середньому.

Для періоду 2 прогноз оновлюється за формулою 2.1:

$$S_2 = 0,5 \cdot 120 + 0,5 \cdot 100 = 110$$

Для періоду 3 прогноз оновлюється знову:

$$S_3 = 0,5 \cdot 130 + 0,5 \cdot 110 = 120$$

Для періоду 4 ми спостерігаємо падіння попиту, і прогноз коригується відповідно:

$$S_4 = 0,5 \cdot 90 + 0,5 \cdot 120 = 105$$

Нарешті, для періоду 5 прогноз трохи змінюється в бік підвищення:

$$S_5 = 0,5 \cdot 110 + 0,5 \cdot 105 = 107,5$$

Цей простий приклад ілюструє, як експоненціальне згладжування реагує на зміни попиту з часом, надаючи оновлений прогноз після кожного періоду. Коефіцієнт згладжування  $a$  можна регулювати залежно від того, яке значення ми хочемо надати останнім змінам порівняно з довгостроковими тенденціями.

Значення  $a$  може істотно вплинути на прогноз. Вищі значення  $a$  (ближче до 1) роблять прогноз більш чутливим до останніх змін, але вони також можуть зробити прогноз більш мінливим. Нижчі значення  $a$  (ближче до 0) призводять до більш стабільного, згладженого прогнозу, але можуть призвести до відставання прогнозу, якщо є раптові зміни попиту. Для кращого розуміння, порівняємо два прогнози з різними значеннями  $a$ : 0,2 і 0,8 (табл. 2.2).

Таблиця 2.2 – Порівняння двох прогнозів з різними значеннями

Період часу ( $t$ )	Фактичний попит ( $D_t$ )	Прогноз ( $a = 0,2$ )	Прогноз ( $a = 0,8$ )
1	100	100 (початковий)	100
2	120	104	116
3	130	109,2	128
4	90	104,36	99,6
5	110	106,29	106,08

Можна побачити, що з  $a = 0,2$  прогноз оновлюється повільно та демонструє меншу волатильність. Водночас при  $a = 0,8$  прогноз швидко реагує на зміни фактичного попиту, але також є більш мінливим.

У той час як просте експоненціальне згладжування ефективно для даних без тенденції чи сезонності, такі варіації, як метод лінійного тренду Холта та метод Холта-Уінтерса, розширюють модель для обробки цих випадків, включаючи в прогноз як трендові, так і сезонні компоненти. Це робить його ідеальним для ситуацій, коли попит повторюється сезонно, як це часто буває в промисловості автозапчастин, де

попит може коливатися залежно від таких факторів, як сезонний ремонт або виробничі цикли. Холт-Уінтерс включає три компоненти:

- рівень ( $S_t$ ): згладжене значення в момент часу  $t$ ;
- тренд ( $T_t$ ): швидкість зміни попиту в момент часу  $t$ ;
- сезонність ( $C_t$ ): повторювана сезонна картина в момент часу  $t$ .

Формули для мультиплікативної версії Холта-Уінтерса наступні:

$$S_t = \alpha \cdot \left( \frac{D_t}{C_{t-L}} \right) + (1 - \alpha) \cdot (S_{t-1} + T_{t-1}) \quad (2.2)$$

$$T_t = \beta \cdot (S_t - S_{t-1}) + (1 - \beta) \cdot T_{t-1} \quad (2.3)$$

$$S_t = \gamma \cdot \left( \frac{D_t}{S_t} \right) + (1 - \gamma) \cdot C_{t-L}, \quad (2.4)$$

де  $S_t$  – згладжений рівнем у час  $t$ ;

$T_t$  – компонент тенденції у час  $t$ ;

$C_t$  – сезонний фактор у час  $t$ ;

$L$  – тривалість сезонного циклу (наприклад, 12 місяців);

$\alpha$ ,  $\beta$  та  $\gamma$  – коефіцієнти згладжування компонентів рівня, тренду та сезонності відповідно.

Розглянемо набір даних із місячним попитом на автозапчастини, який демонструє тенденцію та сезонність (табл. 2.3). Ми використаємо метод Холта-Уінтерса для прогнозування майбутнього попиту. Приклад передбачає 12-місячний сезонний цикл і початкові параметри згладжування  $\alpha=0,3$ ,  $\beta=0,2$  та  $\gamma=0,5$ .



Таблиця 2.3 – Прогнозування попиту у 12-місячному циклі

Місяць	Фактичний попит ( $D_t$ )	Згладжений рівень ( $S_t$ )	Тренд ( $T_t$ )	Сезонний фактор ( $C_t$ )	Прогноз ( $F_t$ )
Січень	120	115	2,5	1,05	121,87
Лютий	130	120	2,7	1,08	128,92
Березень	140	128	2,9	1,10	138,76
Квітень	150	135	3	1,12	151,20
Травень	160	140	3,1	1,15	160,73
Червень	180	148	3,4	1,18	174,33
Липень	190	158	3,6	1,20	189,60
Серпень	170	165	3,8	1,22	180,30
Вересень	160	158	3,7	1,18	165,24
Жовтень	155	150	3,5	1,15	157,50
Листопад	145	145	3,3	1,12	148,40
Грудень	135	138	3,1	1,10	138,30

Початковий рівень ( $S_t$ ) встановлено на перше фактичне значення попиту. Після цього рівень згладжується на кожному кроці з використанням фактичного попиту, сезонного фактора та тенденції.

Початковий тренд ( $T_t$ ) відображає середнє зростання попиту з часом. Тут ми оцінюємо початкову тенденцію з ранніх точок даних.

Сезонний фактор ( $C_t$ ) враховує повторювані моделі попиту. Наприклад, якщо попит має тенденцію до різкого зростання влітку (з червня по серпень), сезонний фактор відповідно скоригується.

Прогноз для кожного періоду розраховується наступним чином:

$$F_t = (S_{t-1} + T_{t-1}) \cdot C_{t-L}, \quad (2.5)$$

де  $S_{t-1}$  – згладжений рівень;

$T_{t-1}$  – тенденція попереднього періоду;

$C_{t-L}$  – сезонний фактор з відповідного місяця в попередньому циклі.

У нашому прикладі в січні фактичний попит становив 120 одиниць, а система прогнозувала 121,87 одиниці на цей місяць. Коефіцієнт сезонності для січня був трохи вищим за 1, що вказує на те, що попит у січні зазвичай вищий за середній.

З часом система коригує як трендову, так і сезонну складову. Наприклад, попит у червні прогнозувався на рівні 174,33 одиниці на основі вищого сезонного фактора (літо) і зростаючої тенденції.

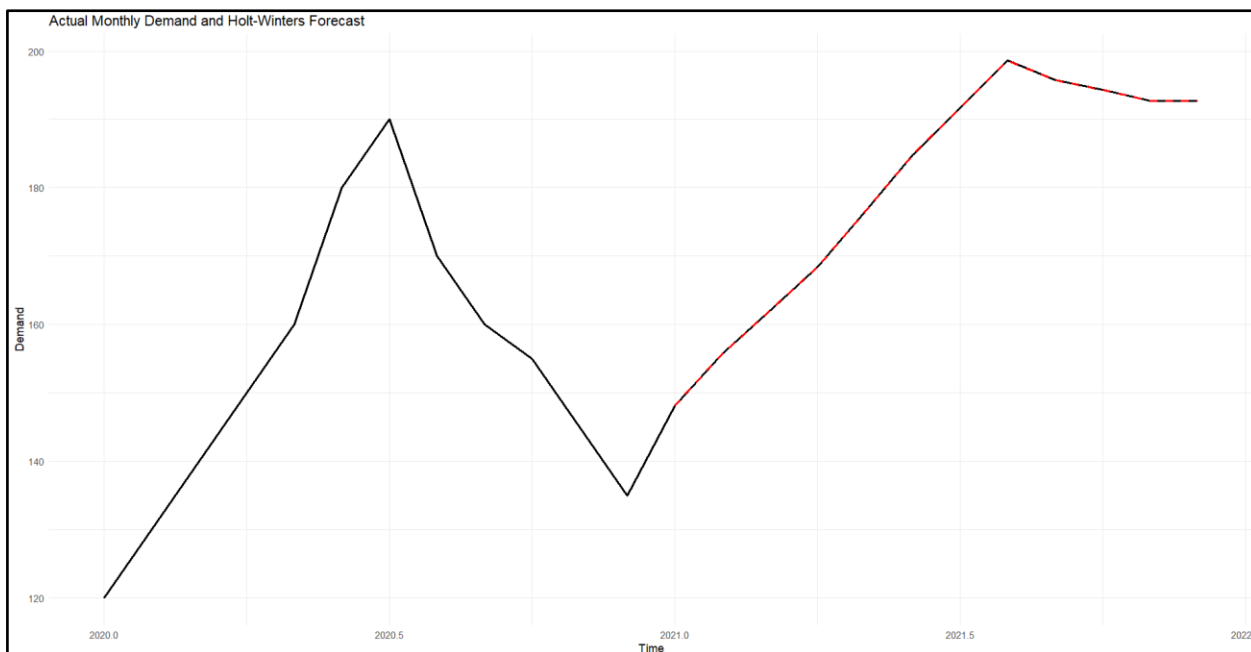


Рисунок 2.2 – Візуалізація нашого приклада використовуючи метод Холта-Уінтерса

## Висновки до розділу 2

У цьому розділі було прийнято рішення про реалізацію простого експоненційного згладжування SES згладжує дані, застосовуючи середньозважене значення до минулих спостережень, причому останнім даним надається більша вага.

Ця методика допомагає прогнозувати майбутні рівні запасів, коригуючи короткострокові коливання, зберігаючи при цьому відносно просту модель. У контексті нашої системи, SES застосовується до історії запасів товару для отримання згладженого значення, яке може бути більш стабільним і корисним для управління

запасами. Константа згладжування (альфа) контролює, наскільки вагомим є останнє значення, балансуючи між оперативністю та стабільністю.

## **3 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН**

### **3.1 Проєктування інформаційної системи**

Для початку розглянемо, як працюють процеси входу та реєстрації в застосунку, що підключений до внутрішнього сервера (рис. 3.1).

Діаграма ілюструє взаємодію між користувачем, браузером та API бекенду під час процесів аутентифікації, таких як вхід у систему та реєстрація. Спочатку користувач взаємодіє з браузером, щоб відправити форму, що містить дані автентифікації. Це відправлення ініціює запит POST, що надсилається браузером до API бекенду, який містить дані форми користувача. Бекенд обробляє цей запит і визначає, чи він відповідає операції входу в систему або реєстрації.

Для входу в систему бекенд звертається до бази даних для перевірки облікових даних користувача. Якщо облікові дані дійсні, бекенд генерує JWT (JSON Web Token) як відповідь. Потім цей токен відправляється назад до браузера, дозволяючи користувачеві пройти автентифікацію для наступних запитів.

У разі реєстрації бекенд перевіряє надані дані форми, щоб переконатися, що вони відповідають необхідним критеріям. У разі успіху бекенд створює новий обліковий запис користувача та зберігає дані користувача в базі даних. Потім у браузер повертається повідомлення про успішне завершення, що підтверджує завершення процесу реєстрації. Цей потік забезпечує безпечну обробку облікових даних користувача та потрібні дії залежно від операції.

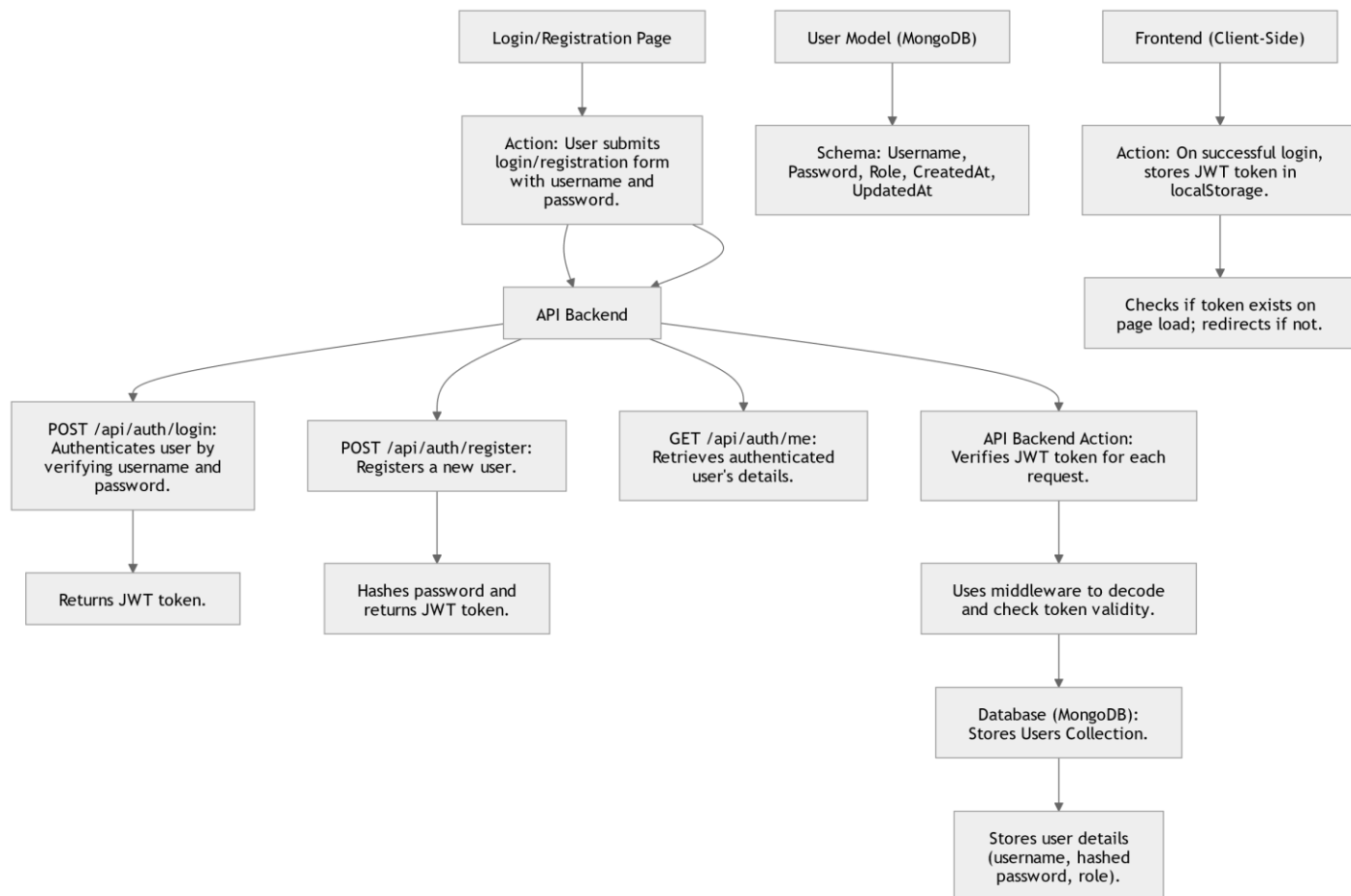


Рисунок 3.1 – Діаграма процесу автентифікації

Далі подивимось, як саме працює сторінка введення обліку автозапчастин в нашій системі (рис. 3.2).

Сторінка обліку розроблена для зручного та ефективного керування товарними запасами. При відкритті сторінки відображається таблиця зі списком усіх продуктів, що містить ключову інформацію. Ми можемо обрати будь-який продукт зі списку, щоб переглянути або відредагувати його деталі, зокрема оновити назву, опис, ціну та наявні запаси. Якщо потрібно додати новий продукт, є кнопка «Додати новий продукт», яка відкриває форму для заповнення необхідних даних.

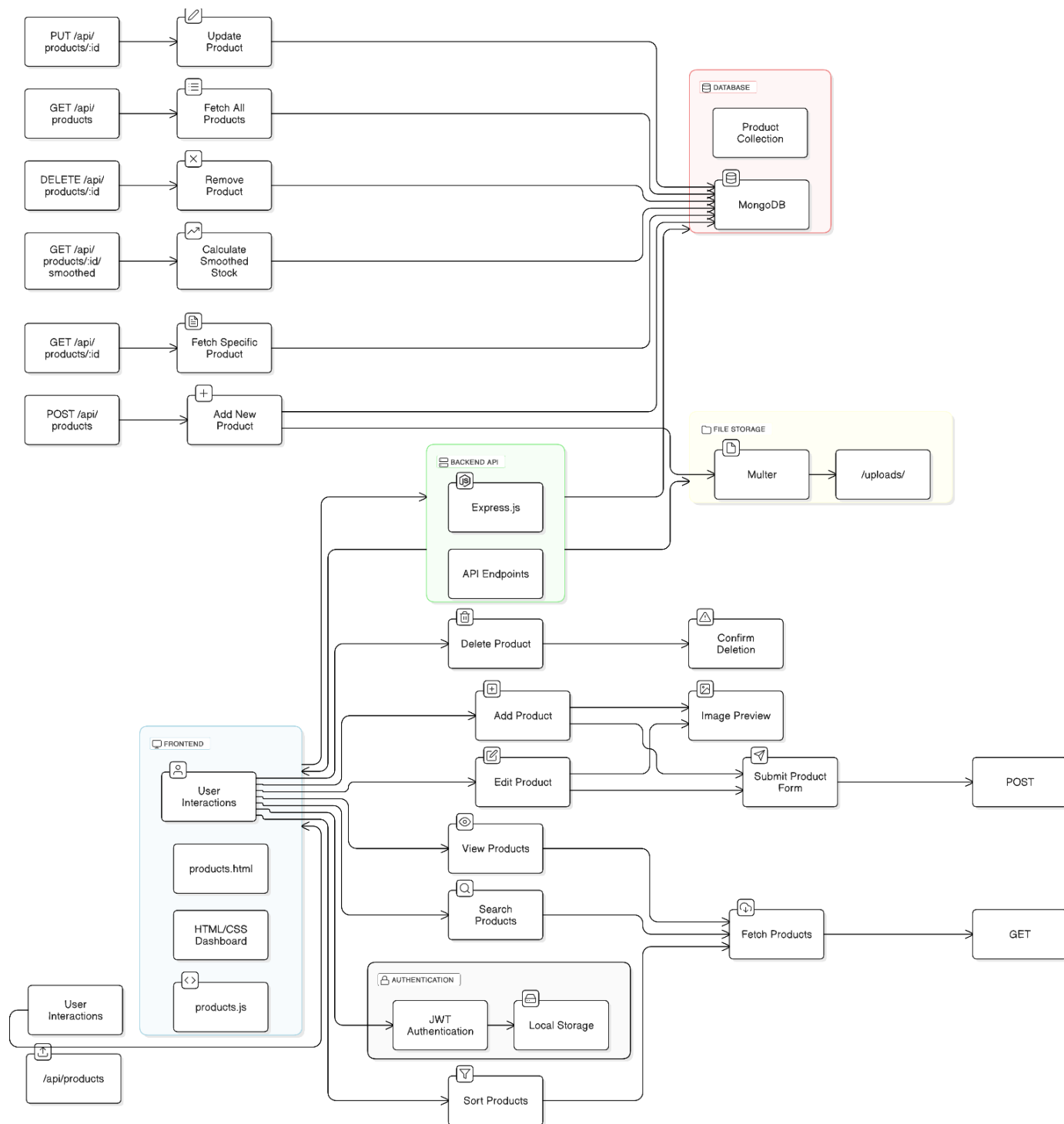


Рисунок 3.2 – Діаграма роботи сторінки керування запасом автозапчастин

Для полегшення навігації ми можемо використовувати рядок пошуку, щоб знайти конкретні продукти за назвою або фільтрувати за категорією. Система також

дозволяє сортувати продукти за такими стовпцями, як назва, ціна або рівень запасів, що полегшує пошук товарів на основі потреб користувача. Якщо товар закінчується або його немає в наявності, користувач побачить значок сповіщення поруч із ним, що допоможе йому відстежувати запаси.

Щоб уникнути випадкових змін, система запитує підтвердження перед виконанням таких дій, як видалення продукту. Є можливість переглядати обмежену кількість продуктів на сторінці з елементами керування розбивкою на сторінки для переходу до наступної чи попередньої сторінки. Це особливо корисно, якщо великий список продуктів. Таблиця налаштовується залежно від розміру пристрою, тому вона має нормальний вигляд як на настільних комп'ютерах, так і на мобільних телефонах. Нарешті, система дозволяє керувати ролями користувачів через MongoDB Compass, тобто різні користувачі можуть мати різні рівні доступу, гарантуючи, що лише авторизовані люди можуть вносити зміни до списку продуктів.

Тепер розглянемо проектування ключових компонентів системи – планування маршрутів, оптимізацію логістичних процесів та візуалізацію доставки (рис. 3.3).

OSRM API використовується у системі для забезпечення швидких і точних розрахунків маршруту та оптимізованої навігації між географічними точками, покращуючи маршрутизацію в реальному часі. API Nominatim використовується для перетворення адрес у географічні координати та навпаки, уможливаючи пошук на основі розташування та функції геолокації, тоді як Leaflet використовується для візуалізації інтерактивних карт, відображення маршрутів та маркерів в інтерфейсі користувача.

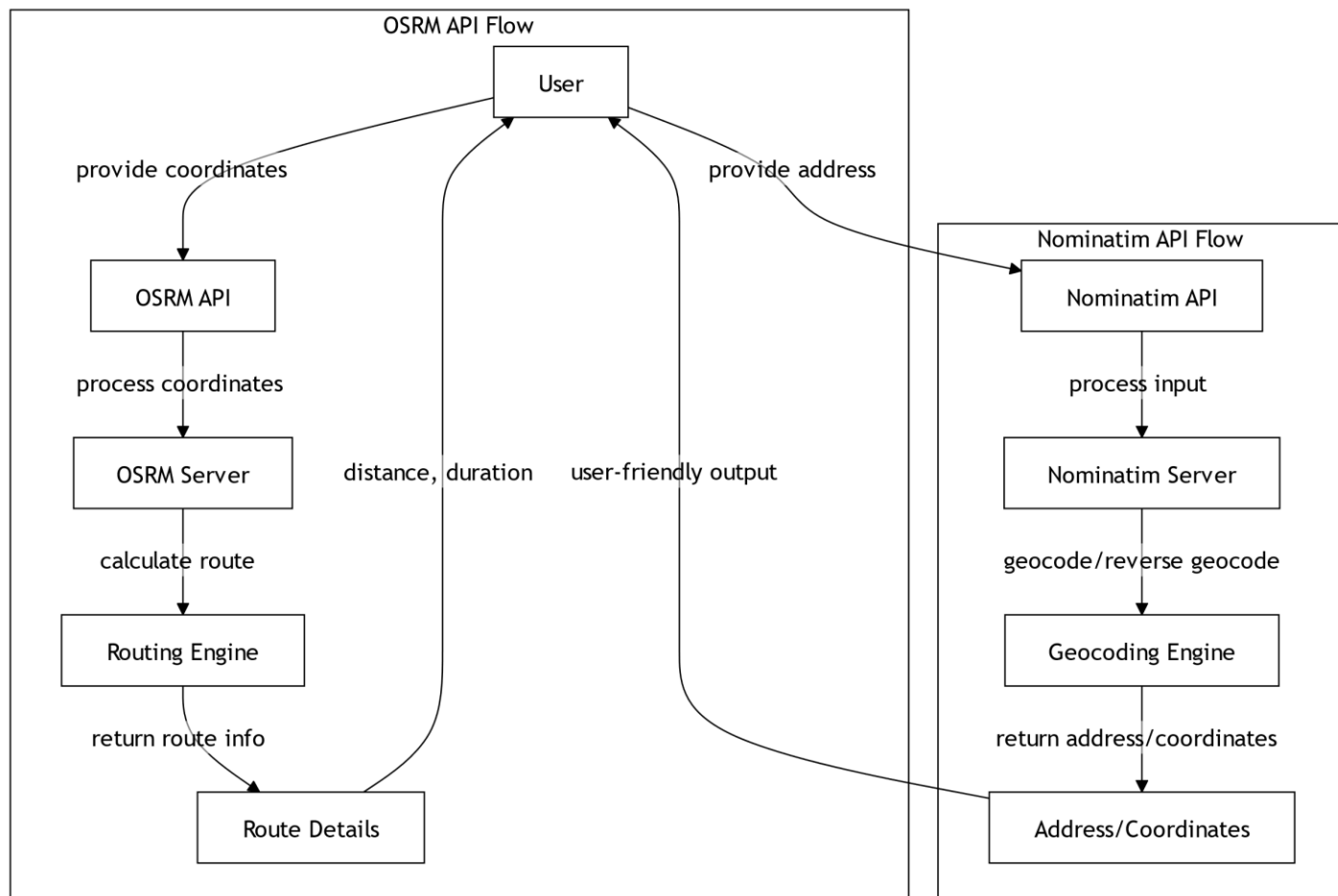


Рисунок 3.3 – Діаграма процесів роботи з API в системі

OSRM API (Сервіс маршрутизації з відкритим вихідним кодом) і Nominatim API є потужними інструментами для геолокації та служб маршрутизації. OSRM зосереджено на забезпеченні швидких рішень для маршрутизації, тоді як Nominatim в основному використовується для геокодування та зворотного геокодування.

OSRM API працює, приймаючи набір координат (широта та довгота) як вхідні дані [36]. Коли користувач запитує маршрут, механізм маршрутизації OSRM обробляє координати, враховуючи такі фактори, як мережа доріг і відстані, щоб обчислити оптимальний шлях. OSRM може повертати інформацію про маршрут, як-от відстань, приблизний час у дорозі та геометрію маршруту (тобто покрокові вказівки або шляхові точки). API зазвичай використовується для прокладання автомобільних,



велосипедних або пішохідних маршрутів, і він відомий своєю швидкістю завдяки використанню попередньо оброблених даних про маршрути.

З іншого боку, Nominatim API використовується для геокодування та зворотного геокодування [37]. Під час геокодування користувачі вказують адресу або назву місця, а Nominatim повертає відповідні координати (широту та довготу). І навпаки, у зворотному геокодуванні користувачі вводять координати, а Nominatim повертає адресу або назву місця, пов'язану з цими координатами. Цей API особливо корисний для програм, таких як картографічні, пошукові та послуги на основі визначення місцезнаходження. Для виконання своїх завдань він покладається на дані OpenStreetMap (OSM).

Обидва API використовують запити на основі HTTP, результати зазвичай повертаються у форматі JSON. Служба маршрутизації OSRM значною мірою покладається на попередньо оброблені дані та алгоритми, оптимізовані для швидкого часу відповіді. Nominatim, навпаки, використовує масштабну базу даних OpenStreetMap, забезпечуючи високу точність і глобальне покриття завдань геолокації.

У той час як OSRM зосереджується на маршрутизації та навігації, Nominatim обслуговує геолокаційні запити, які допомагають ідентифікувати та інтерпретувати географічні розташування на основі зрозумілих людині адрес або координат. Разом ці API є основоположними для створення програм на основі визначення місцезнаходження, картографічних систем і навігаційних служб.

Щоб використовувати Leaflet, розробник ініціалізує об'єкт карти, вказуючи елемент контейнера HTML, і встановлює такі параметри, як початкові географічні координати та рівень масштабування. Потім карта відображається шляхом відображення плиток, які є невеликими файлами зображень, що представляють розділи більшої карти. Leaflet дозволяє розробникам інтегрувати різні текстури карти, наприклад Google Maps або Mapbox, або навіть розміщувати власні (рис. 3.4).

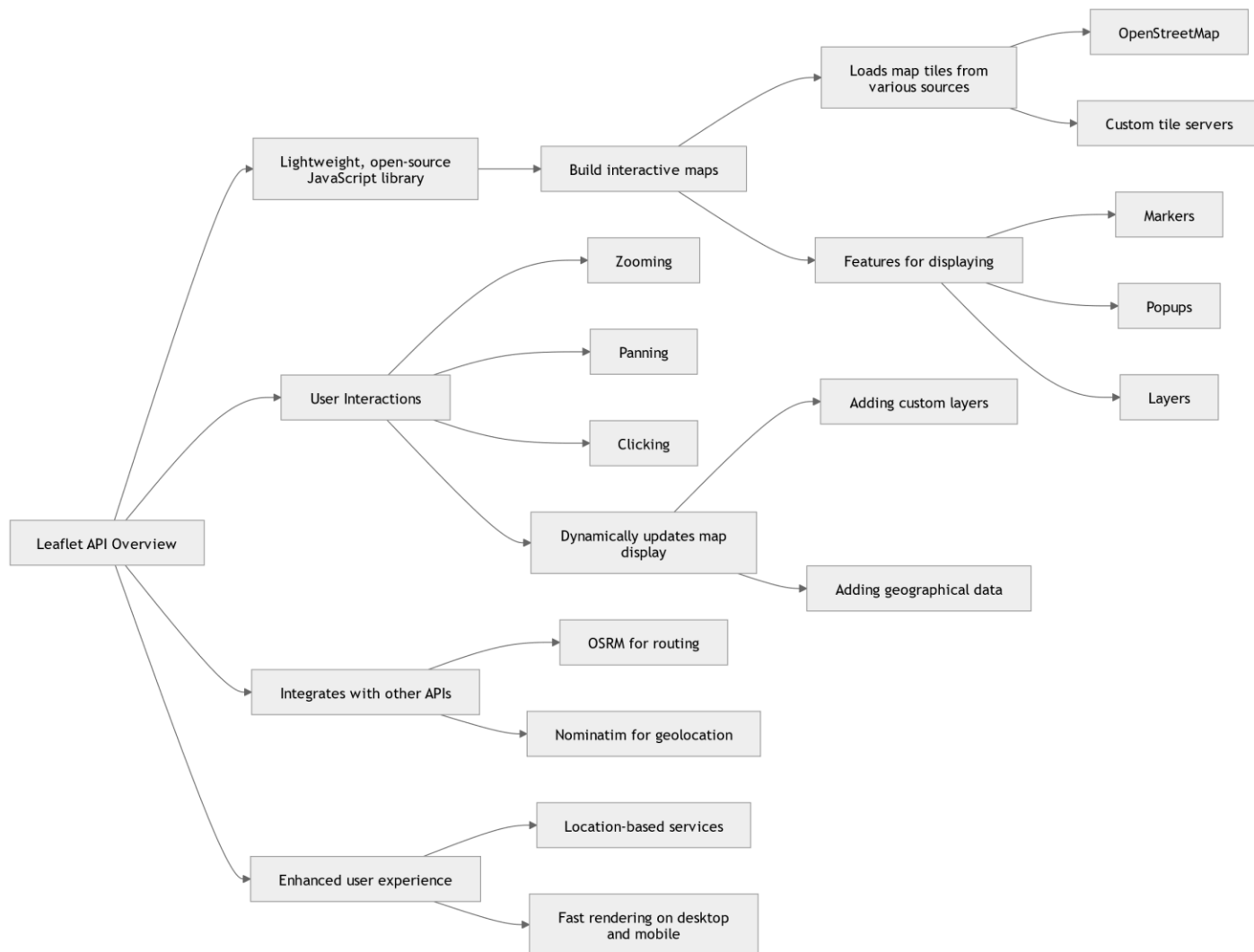


Рисунок 3.4 – Діаграма процесу роботи API карти

### 3.2 Структура системи для оптимізації ланцюга постачання автозапчастин

Розберемо різні компоненти інформаційної системи ланцюга постачання автозапчастин. Наведена схема ілюструє архітектуру системи та взаємодію між кожною ключовою частиною. Нижче зображені компоненти, необхідні для роботи системи (рис. 3.5).

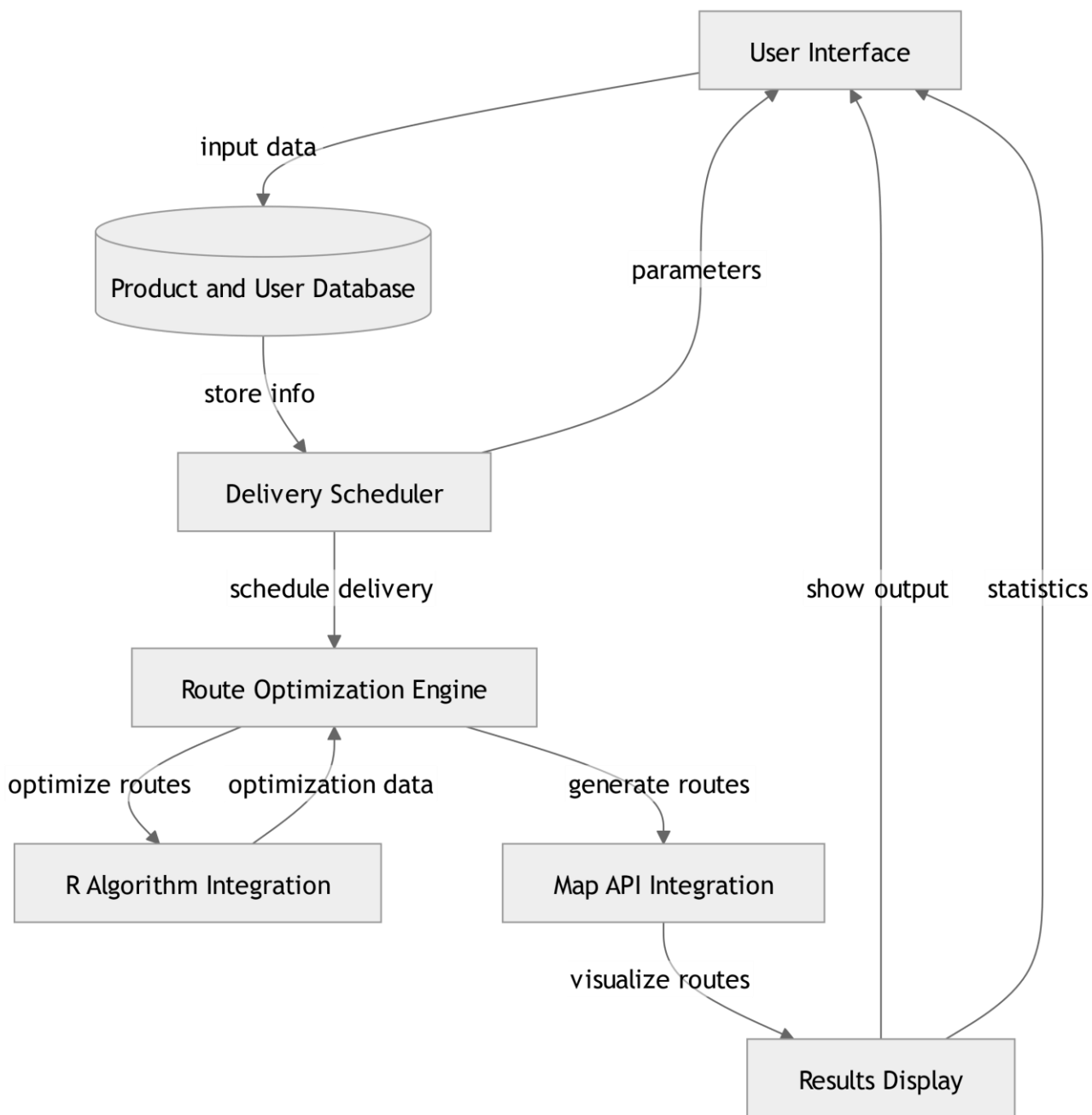


Рисунок 3.5 – Блок-схема архітектури системи

Запропонована система складається з наступних ключових компонентів:

- інтерфейс користувача (User Interface), який дозволяє активним користувачам вводити всі необхідні дані щодо типів та параметрів автозапчастин, встановлювати графіки доставки та ін.;

- база даних (Database), що зберігає всі відомості про обрані продукти, точки доставки та дані користувачів;
- планувальник доставки (Delivery Schedule), який здійснює планування та керування процесами доставки на основі визначених користувачем параметрів;
- механізм оптимізації маршрутів (Route Optimization Engine), що виконує обробку даних процесів доставки та створює оптимізовані маршрути на основі обраних алгоритмів;
- блок інтеграції алгоритмів R (R Algorithm Integration), який здійснює реалізацію алгоритмів оптимізації, використовуючи середовище R для розрахунку поточного маршруту;
- блок інтеграції Map API (Map API Integration), що здійснює візуалізацію створених маршрутів та точок доставки на карті обраного міста;
- блок відображення результатів (Results Display), який показує кінцевий маршрут, статистику та порівняння алгоритмів для користувача.

Запропонована система базується на використанні алгоритмів оптимізації маршрутів для досягнення мінімізації експлуатаційних витрат, включаючи скорочення витрат палива, зменшення часу доставки та підвищення загальної ефективності логістичних операцій.

### **Висновки до розділу 3**

Проектування інформаційної системи забезпечує комплексний і чітко структурований підхід до організації ключових процесів у застосунку для управління автозапчастинами та їх доставки. У цьому розділі були розглянуті основні етапи проектування, які сприяють створенню зручних та ефективних рішень для користувачів.

Розглянуто процеси автентифікації, включаючи входи та реєстрацію, які були представлені через діаграму послідовності. Це дозволяє чітко уявити взаємодію між

користувачем, браузером та бекендом під час обробки облікових даних.

Було побудовано архітектуру системи для управління запасами автозапчастин, зокрема було детально описано функціонал сторінки управління запасами, що включає можливість перегляду, редагування, додавання нових продуктів та фільтрації товарів за різними критеріями. Також були реалізовані функції підтвердження дій, щоб забезпечити безпеку під час управління даними.

Проектування системи доставки було виконано з використанням API OSRM, Nominatim і Leaflet для маршрутизації, геолокації та візуалізації карт. Детально описано, як ці API працюють разом для оптимізації процесів доставки, включаючи перетворення адрес у координати та побудову маршрутів з урахуванням реальних умов.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ДЛЯ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН

### 4.1 Опис програмної реалізації

Автентифікація користувача починається з реєстрації, де користувач вводить ім'я користувача та пароль, які надсилаються бекенду за допомогою POST-запиту на кінцеву точку `/api/auth/register`. Бекенд перевіряє введені дані, хешує пароль і зберігає дані в базі даних, відповідаючи повідомленням про успіх або помилку.

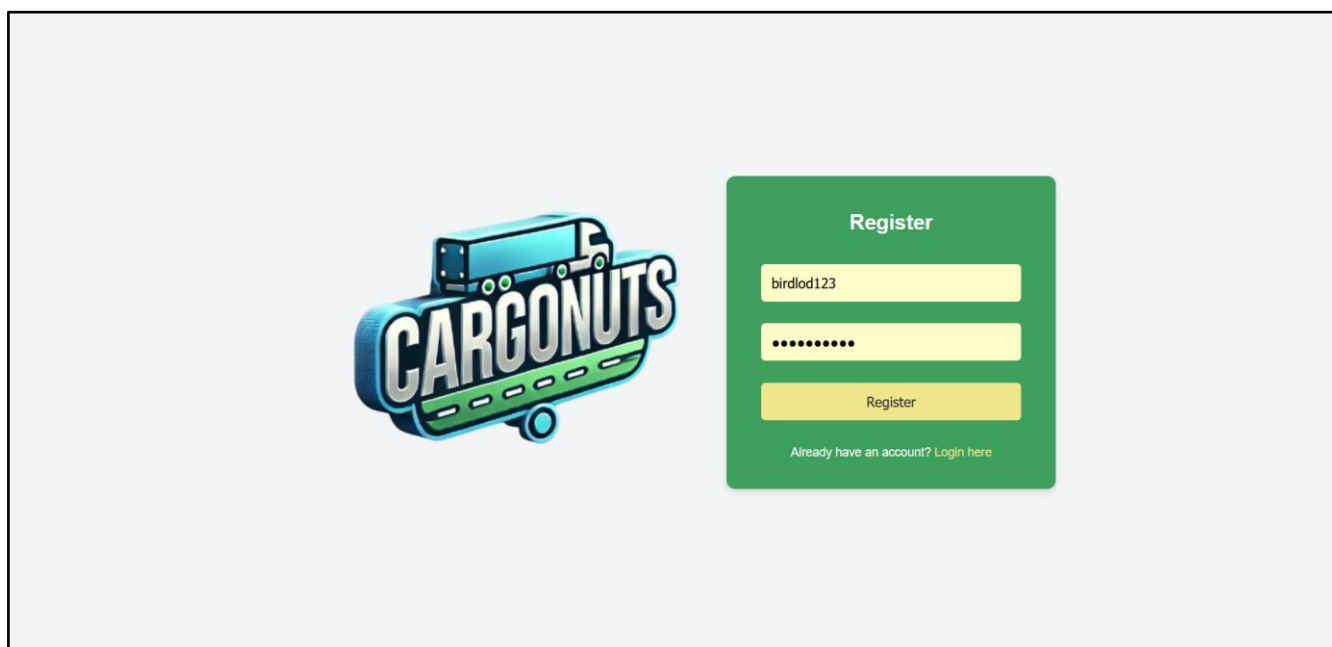


Рисунок 4.1 – Форма реєстрації користувачів

Після успішної реєстрації користувач перенаправляється на сторінку входу в систему (рис. 4.2). Під час входу користувач вводить свої облікові дані, які надсилаються на кінцеву точку `/api/auth/login` бекенду. Бекенд перевіряє ім'я користувача, перевіряє пароль і генерує JSON-токен (JWT) після успішної автентифікації. Цей токен, разом з іменем користувача, зберігається в `localStorage` браузера для використання в наступних аутентифікованих запитах. Для безпечних операцій фронтенд включає JWT в заголовок `Authorization` запитів до API. Бекенд

перевіряє JWT, забезпечуючи ідентифікацію користувача перед наданням доступу до захищених ресурсів. Якщо токен відсутній, недійсний або прострочений, бекенд відмовляє в доступі та видає помилку. Цей процес забезпечує безпечну та ефективну автентифікацію користувачів для вебдодатків.

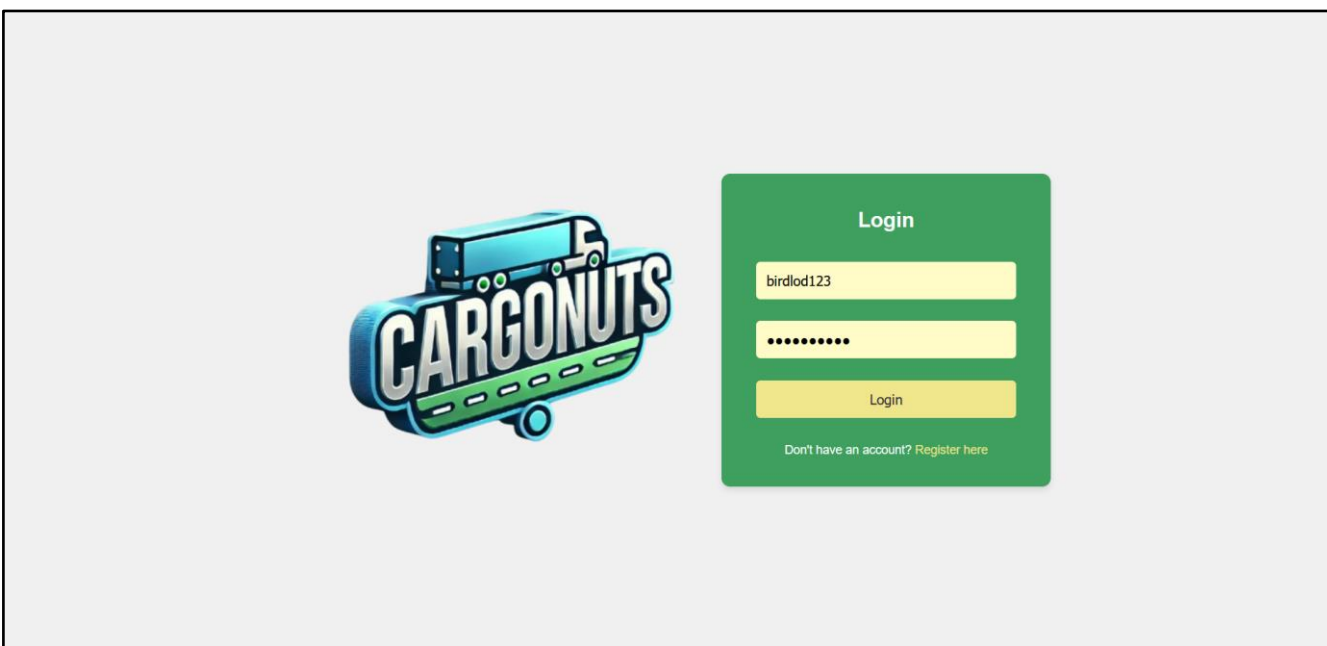


Рисунок 4.2 – Форма входу до системи

Нижче зображено приклад даних користувача, що зберігаються в базі даних MongoDB (рис. 4.3).



Рисунок 4.3 – Дані користувача в базі даних

Після успішного входу в систему статус автентифікації користувача перевіряється на сторінці *index.html* шляхом перевірки наявності токена в *localStorage* браузера. Якщо токен не знайдено, користувач перенаправляється на сторінку входу,

щоб запобігти несанкціонованому доступу. Якщо токен присутній, скрипт отримує ім'я користувача з *localStorage* і відображає персоналізоване привітання в навігаційній панелі разом з посиланням «Вийти». Опція виходу містить діалогове вікно підтвердження, і після підтвердження токен видаляється з *localStorage*, а користувач перенаправляється на сторінку входу в систему.

Основний контент висвітлює мету додатка, зосереджуючись на оптимізації доставки автозапчастин, зберігаючи при цьому чистий і зручний інтерфейс (рис. 4.4). Безпека забезпечується на всіх рівнях, гарантуючи, що неавторизовані користувачі не зможуть отримати доступ до сторінки або її ресурсів.

Таке налаштування забезпечує безпечну та безперешкодну роботу для автентифікованих користувачів, зберігаючи при цьому суворий контроль доступу.

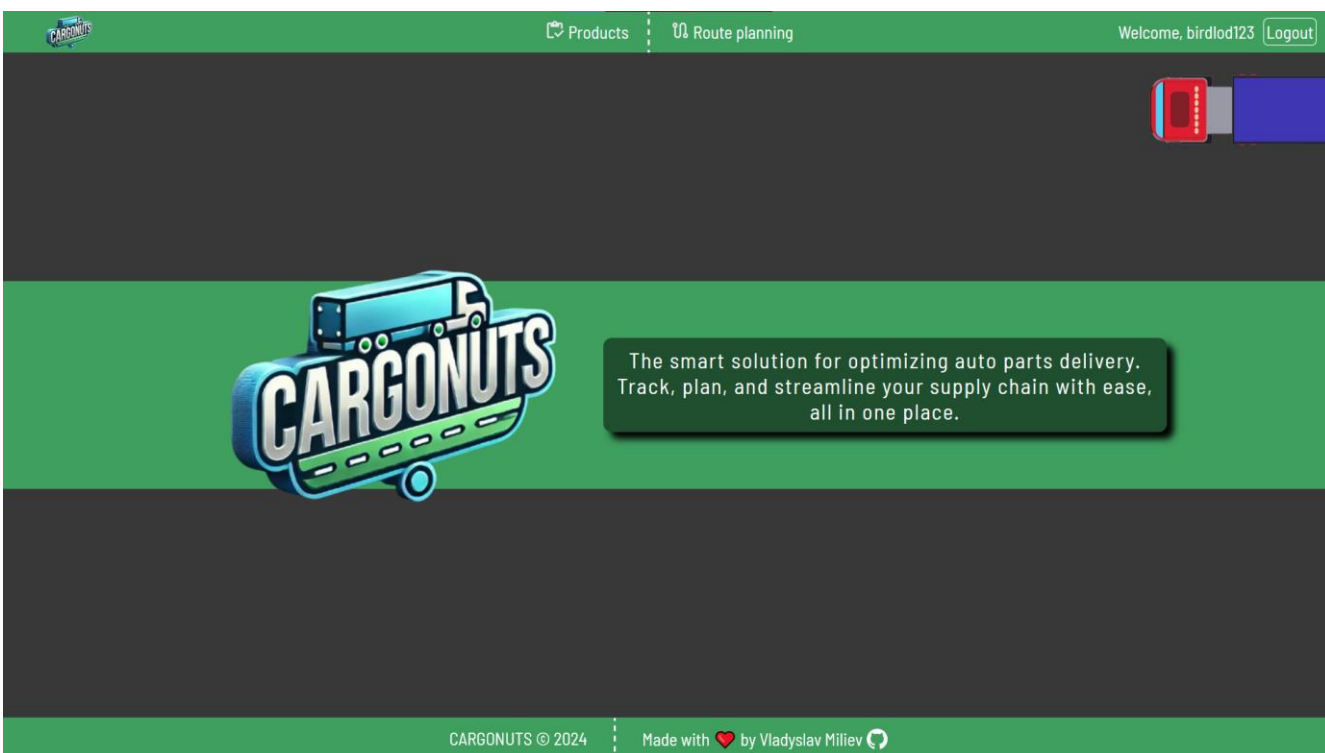


Рисунок 4.4 – Головна сторінка системи

Сторінка, зображена нижче, надає користувацький інтерфейс для управління системою інвентаризації автозапчастин (рис. 4.5). Вона містить панель навігації з



посиланнями на головну сторінку і сторінку планування маршрутів, а також статус автентифікації та функцію виходу з системи. На головній сторінці відображається статистика інвентаризації, наприклад, загальна кількість товарів, низька кількість товарів та відсутність товарів на складі. Рядок пошуку дозволяє користувачам фільтрувати товари за назвою або категорією.

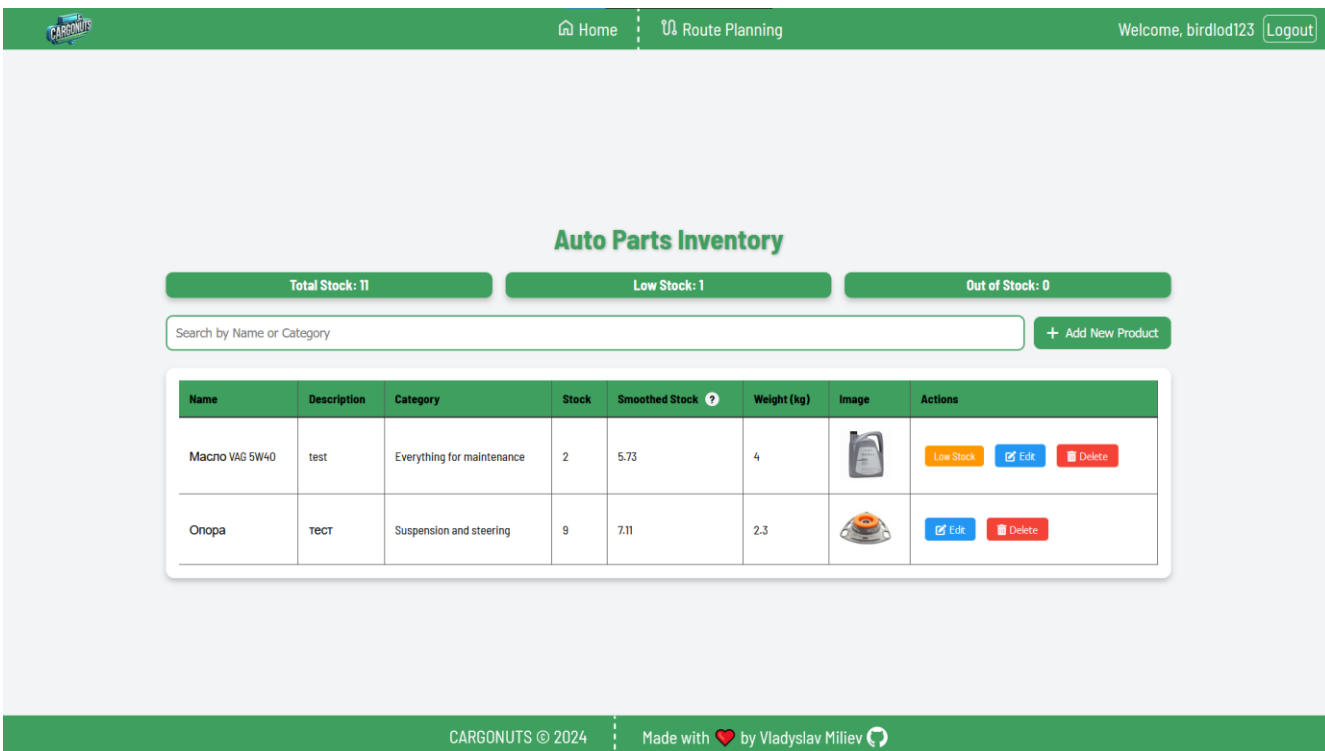


Рисунок 4.5 – Сторінка управління системою інвентаризації автозапчастин

Користувачі можуть додавати нові товари за допомогою кнопки, яка відкриває модальну форму для введення таких даних, як назва, опис, категорія, кількість, вага та зображення (рис. 4.6).

**Add New Product**

Name:

Description:

Category:

Stock Quantity:

Weight (kg):

Image:  
 No file selected.

Рисунок 4.6 – Форма додавання нового товару

Підбір категорій спеціально розроблений під управління складом автозапчастин (рис. 4.7).

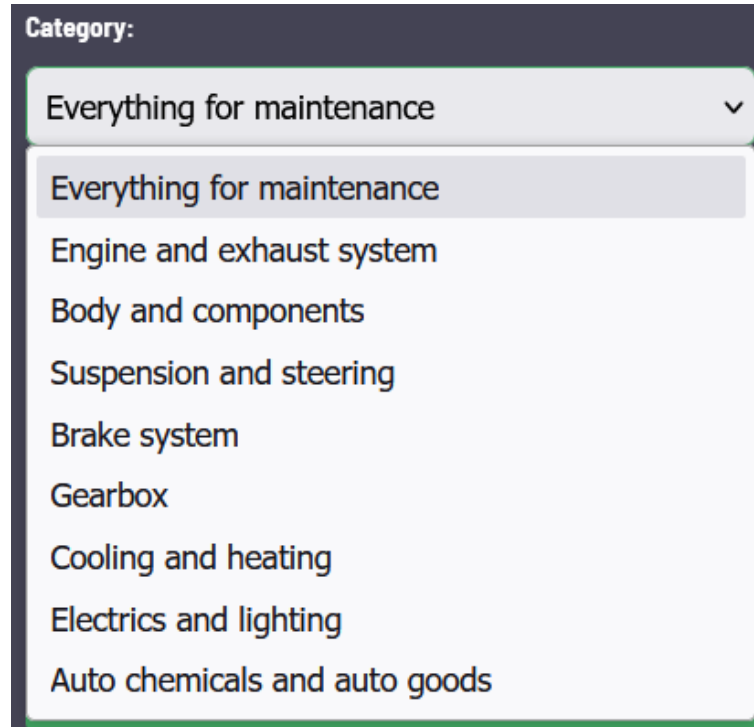
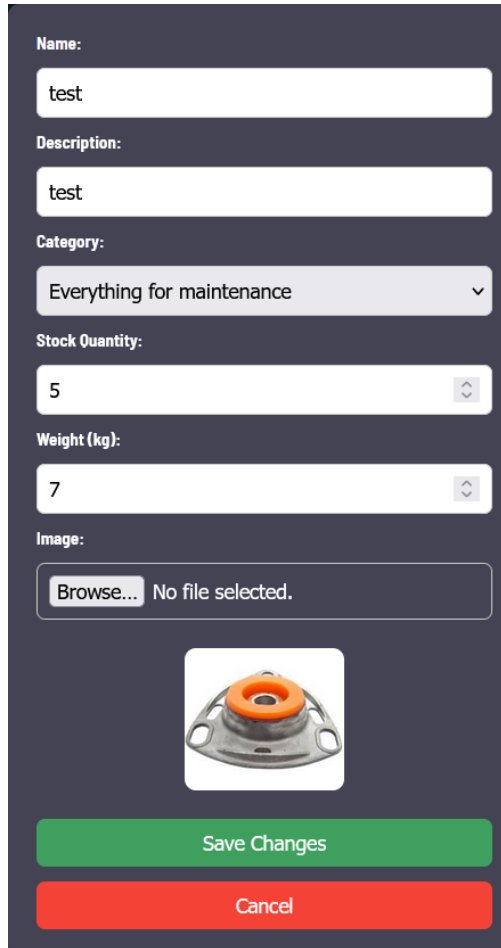


Рисунок 4.7 – Категорії автозапчастин

Модальна форма підтримує редагування вже доданих продуктів, з полями, попередньо заповненими під час додавання (рис. 4.8).



The image shows a dark-themed form for editing a product. It contains the following fields and elements:

- Name:** A text input field containing the word "test".
- Description:** A text input field containing the word "test".
- Category:** A dropdown menu with the selected option "Everything for maintenance".
- Stock Quantity:** A numeric input field with a spinner, containing the value "5".
- Weight (kg):** A numeric input field with a spinner, containing the value "7".
- Image:** A file upload area with a "Browse..." button and the text "No file selected." Below it is a small image of a car part (a bearing).
- Buttons:** A green "Save Changes" button and a red "Cancel" button at the bottom.

Рисунок 4.8 – Форма редагування доданого товару

Таблиця товарів динамічно перераховує позиції запасів, показуючи такі деталі, як назва, опис, категорія, запас і зображення. Користувачі можуть сортувати таблицю, натискаючи заголовки стовпців, або взаємодіяти з товарами за допомогою кнопок редагування і видалення.

Якщо кількість товару менша ніж 5, попередження про низький рівень запасів підсвічуються в таблиці (рис. 4.9).

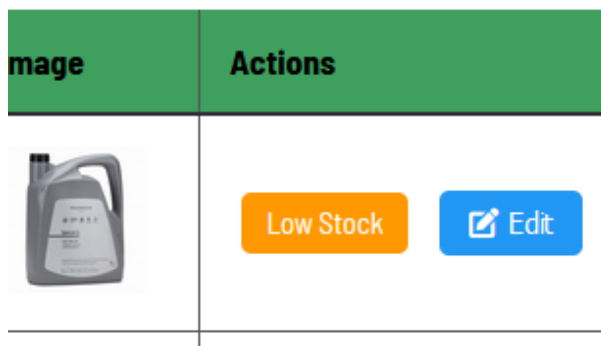


Рисунок 4.9 – Попередження про низький рівень запасів

Функціонал містить візуалізацію історії запасів за допомогою бібліотеки «Chart.js». При наведенні курсора на кількість товару, з'являється графік з динамічно згенерованою лінійною діаграмою, що показує історичну динаміку запасів у часі (рис. 4.10).

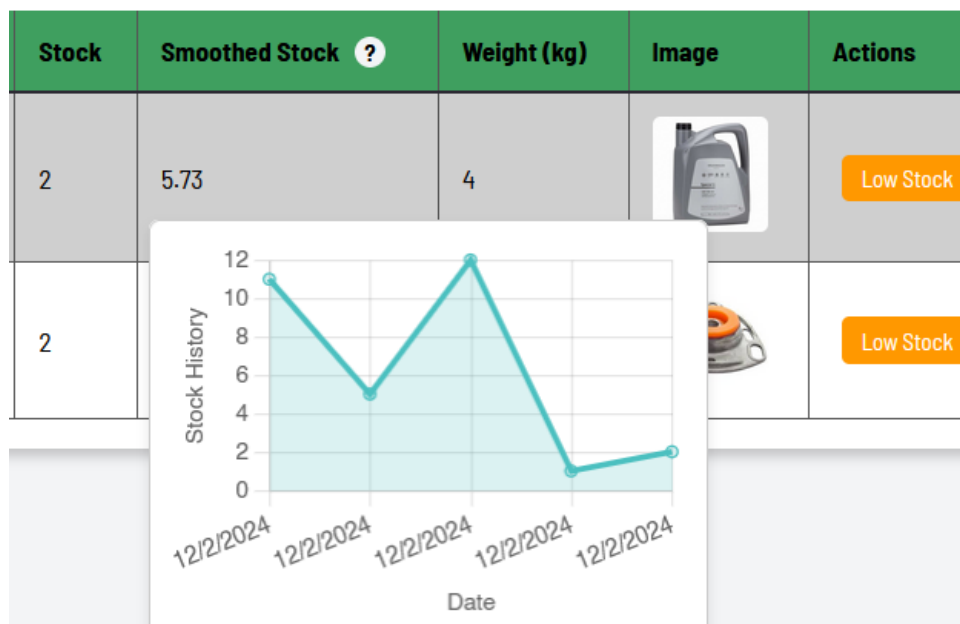


Рисунок 4.10 – Лінійна діаграма зміни запасу з часом

Графік динамічно отримує історичні дані про запаси та візуалізує їх, показуючи тенденцію зміни рівня запасів. Ця функція надає інтуїтивно зрозумілий спосіб аналізу коливань запасів безпосередньо зі списку товарів. Якщо дані історії запасів відсутні,

графік не відображається. При наведенні на знак питання біля колонки «Згладжений запас» користувач побачить спливаючу підказку, яка пояснює, для чого потрібна ця метрика (рис. 4.11).

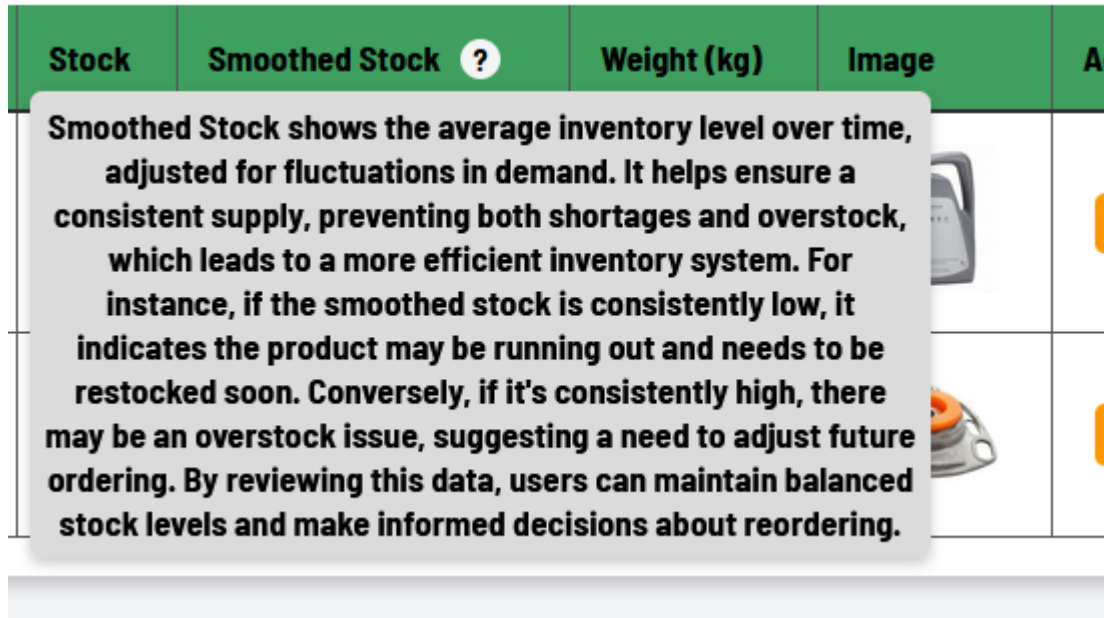


Рисунок 4.11 – Спливаюча підказка з поясненням метрики

Бекенд ініціалізує випадуючий список категорій продуктів і обробляє модальне відкриття і закриття. Заповнення форми додає або редагує продукти за допомогою викликів API, а валідація гарантує, що кількість товарів на складі є невід'ємною. Продукти зберігаються в базі даних MongoDB, що забезпечує надійне рішення для бекенда (рис. 4.12). Запити API для взаємодії з цими даними включають токени автентифікації, що зберігаються в *localStorage*. Якщо користувач не пройшов автентифікацію, він буде перенаправлений на сторінку входу в систему.

```
_id: ObjectId('674c67ba71b75e180690d226')
name: "Опора"
description: "тест"
category: "Suspension and steering"
stock: 9
weight: 2.3
image: "/uploads/1733144692483.jpg"
history: Array (9)
  createdAt: 2024-12-01T13:42:18.887+00:00
  updatedAt: 2024-12-02T13:04:52.539+00:00
  sku: "Suspension and steering-1733060538888"
  __v: 9

_id: ObjectId('674d64f90c9a4876a7ef1bc3')
name: "Масло VAG 5W40"
description: "test"
category: "Everything for maintenance"
stock: 2
weight: 4
image: "/uploads/1733144712490.jpg"
history: Array (5)
  createdAt: 2024-12-02T07:42:49.784+00:00
  updatedAt: 2024-12-02T13:05:12.550+00:00
  sku: "Everything for maintenance-1733125369785"
  __v: 5
```

Рисунок 4.12 – Дані про товар в базі даних

Товари динамічно завантажуються з MongoDB і відображаються в таблиці із застосуванням динамічної фільтрації та сортування. Статистика запасів, така як загальна кількість, низький рівень та відсутність товарів на складі, розраховується та оновлюється в режимі реального часу. Функція пошуку оновлює вигляд таблиці, коли користувачі вводять запит у пошуковому рядку. Користувачі можуть переглядати завантажені зображення перед відправкою. Операції видалення та редагування запускають виклики API для безпосередньої модифікації записів MongoDB.

Далі я хотів би описати реалізацію функціоналу планування маршрутів для інформаційної панелі автозапчастин, який дозволяє користувачам вводити адреси, візуалізувати місця розташування на карті та обчислювати відстані між ними. Ця функціональність побудована з використанням HTML, CSS, JavaScript та бібліотеки

Leaflet для візуалізації (рис. 4.13).

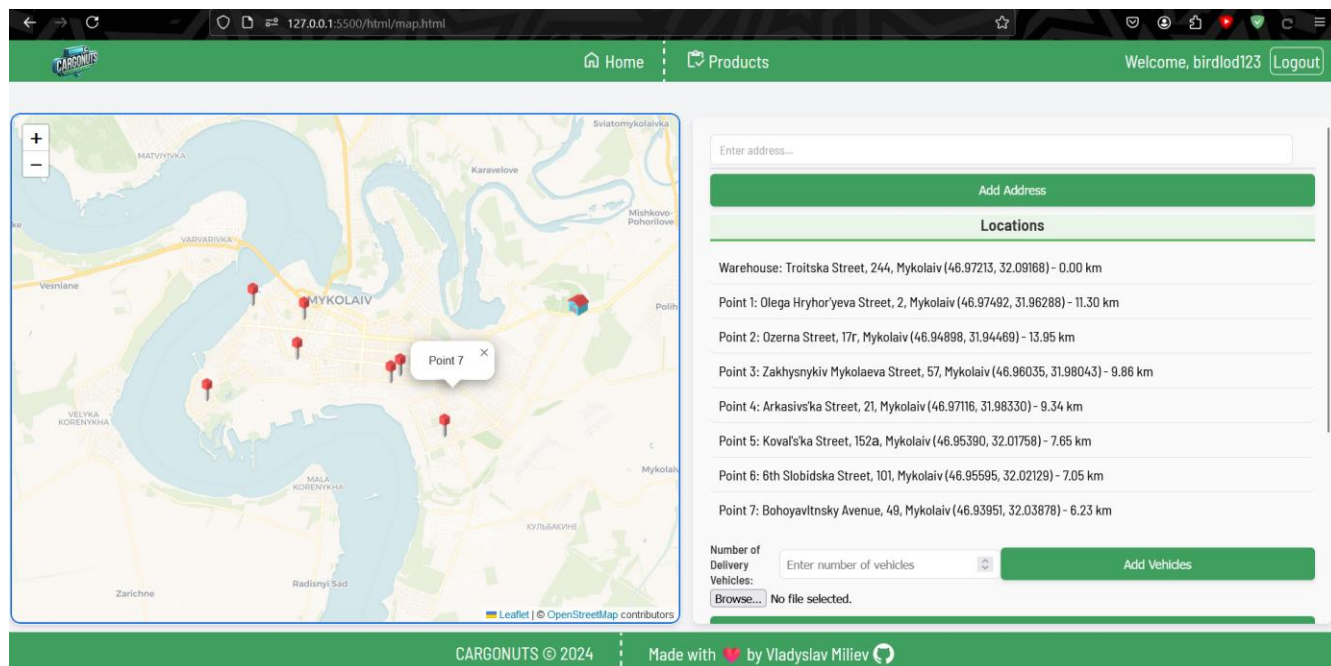


Рисунок 4.13 – Сторінка для планування маршрутів доставки автозапчастин

Файл *map.html* слугує головною сторінкою для цього функціоналу. Він завантажує необхідні таблиці стилів і скрипти, такі як Leaflet для картографічних функцій і Leaflet Control Geocoder для пошуку адрес. Він надає інтуїтивно зрозумілий інтерфейс, де користувачі можуть вводити адреси, візуалізувати їх на карті та взаємодіяти з маркерами. Сторінка також містить панель навігації з посиланнями на інші розділи інформаційної панелі, такі як домашня сторінка та сторінки продуктів, а також функцію статусу автентифікації, яка змінюється залежно від того, чи користувач увійшов до системи, чи ні.

Основною функцією на цій сторінці є інтерактивна карта, створена за допомогою бібліотеки Leaflet. Карта ініціалізується з центром і рівнем масштабування за замовчуванням. Карта дозволяє користувачам додавати маркери для локацій, натискаючи на карту або шукаючи адреси. Коли адреса вводиться в поле введення, система використовує API Nominatim від OpenStreetMap для геокодування адреси,



перетворюючи її в координати широти та довготи, які потім використовуються для розміщення маркера на карті.

Функціонал також підтримує підказки адреси при введенні користувачем, надаючи список можливих збігів з OpenStreetMap (рис. 4.14). Після вибору адреси вона додається до списку локацій, який містить в собі як склад, так і пункти доставки. Перша додана адреса встановлюється як склад, тоді як наступні адреси додаються як пункти доставки.

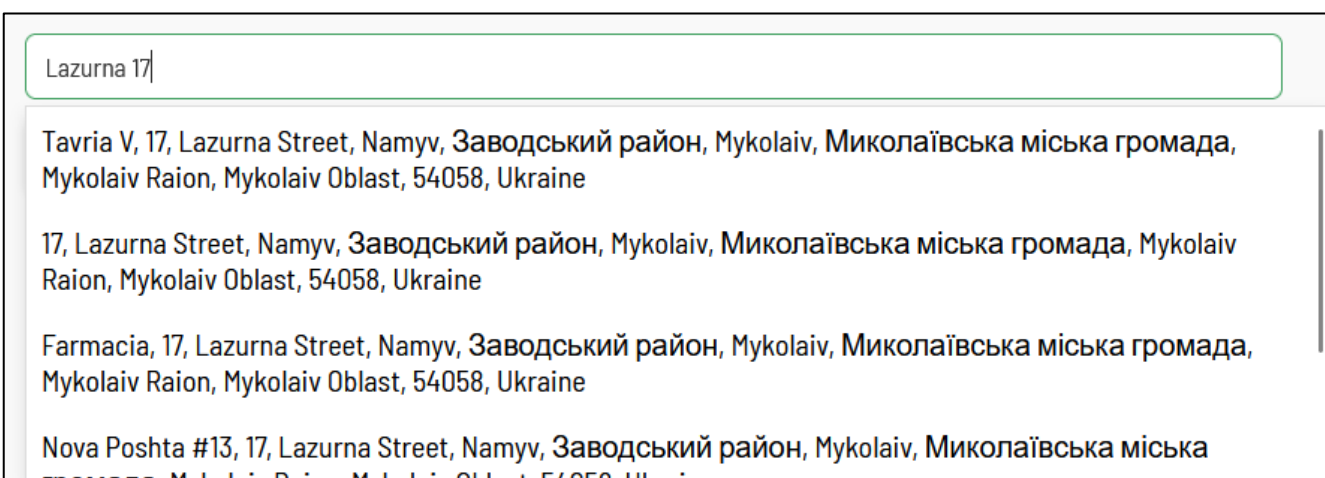


Рисунок 4.14 – Пошукові пропозиції щодо додавання адреси

Логіка планування маршруту реалізована в скрипті *map.js*. Кожного разу, коли додається адреса, система обчислює відстань між новою точкою і складом, використовуючи API Open Source Routing Machine (OSRM). Потім ця відстань відображається в списку локацій поряд з адресою. Система кешує ці розрахунки відстані, щоб оптимізувати продуктивність і зменшити кількість непотрібних викликів API.

Користувачеві також пропонується ввести кількість автомобілів для доставки. Для кожного транспортного засобу система генерує форму, де користувач може ввести інформацію про паливний бак, вантажопідйомність, витрату палива та середню швидкість (рис. 4.15). Ці дані збираються і будуть збережені в JSON-об'єкті, коли користувач вирішить зберегти дані планування маршруту.

Number of Delivery Vehicles: 2

**Vehicle 1 Information**

Fuel Capacity (liters): 20

Truck Capacity (kg): 250

Fuel Consumption (L/100 km): 7.4

Average Truck Speed (km/h): 30

**Vehicle 2 Information**

Fuel Capacity (liters): 20

Truck Capacity (kg): 300

Fuel Consumption (L/100 km): 9.2

Average Truck Speed (km/h): 35

Рисунок 4.15 – Форми для додавання кількості транспортних засобів та їх змінних

Функція матриці відстаней дозволяє користувачам розраховувати відстані між усіма пунктами, включаючи склад і всі пункти доставки. Це робиться шляхом створення матриці, яка містить відстані між кожною комбінацією локацій. Матриця відображається у вигляді таблиці, що дозволяє користувачам візуально порівняти відстані між різними точками (рис. 4.16).

Distance Matrix									
	Warehouse	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	Point 7	Point 8
Warehouse	0.00 km	2.18 km	4.95 km	5.92 km	8.06 km	4.94 km	3.47 km	2.12 km	8.48 km
Point 1	2.18 km	0.00 km	2.97 km	4.10 km	6.25 km	6.02 km	1.96 km	2.70 km	6.68 km
Point 2	4.95 km	2.97 km	0.00 km	3.03 km	4.54 km	8.68 km	4.31 km	5.36 km	5.31 km
Point 3	5.92 km	4.10 km	3.03 km	0.00 km	4.26 km	7.40 km	2.65 km	5.60 km	2.93 km

Рисунок 4.16 – Матриця відстаней

Для забезпечення зручності роботи користувача в системі також передбачена можливість очищення всіх даних. За допомогою кнопки користувач може видалити всі маркери з карти, очистити список локацій і скидати поля введення, надаючи можливість почати планування нового маршруту з самого початку не оновлюючи сторінку.

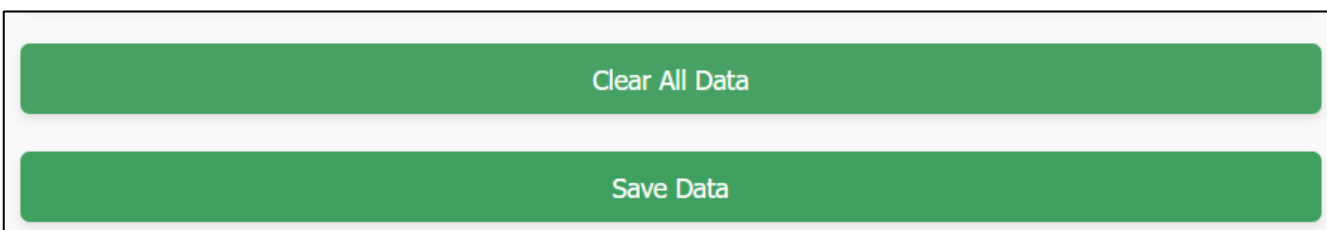


Рисунок 4.17 – Кнопки для очищення всіх даних та збереження вхідних даних

Реалізація включає функцію збереження даних, яка дозволяє користувачеві зберігати дані планування маршруту (місцезнаходження, транспортні засоби та матрицю відстаней) у JSON-файл для персональних задач (рис. 4.18).

```
{ } route_data1.json ×
D: > BSNU > Diploma > json_and_r > { } route_data1.json > ...
18     {
19       "address": "Ozerna Street, 17; 17а, Mykolaiv",
20       "lat": 46.95010084649819,
21       "lng": 31.943435668945316
22     },
23     {
24       "address": "Dalia Street, 30, Mykolaiv",
25       "lat": 46.957603570167194,
26       "lng": 31.990470886230472
27     },
28     {
29       "address": "Spaskyi Descent, Mykolaiv",
30       "lat": 46.976472958156776,
31       "lng": 31.96575164794922
32     }
33   ],
34   "vehicles": [
35     {
36       "fuelCapacity": 40,
37       "truckCapacity": 720,
38       "fuelConsumption": 8,
39       "averageSpeed": 29
40     }
41   ],
42   "distanceMatrix": [
43     [
44       0,
45       3.55,
46       7.1,
47       5.23,
48       1.71,
49       2.32
50     ],
51     [
52       3.55,
53       0,
54       4.54
```

Рисунок 4.18 – Збережені вхідні дані у JSON форматі

Після введення всіх необхідних даних користувач обирає алгоритм оптимізації та натискає кнопку «Оптимізувати маршрут», щоб розпочати процес оптимізації (рис. 4.19).

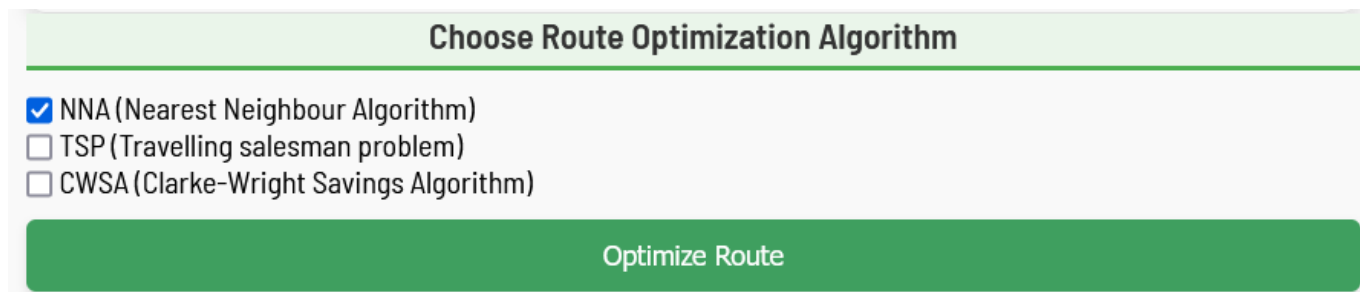


Рисунок 4.19 – Секція вибору алгоритму та кнопка, яка запускає процесу оптимізації

Система пропонує три різні алгоритми, кожен з яких призначений для конкретних сценаріїв доставки.

Алгоритм задачі комівояжера (TSP) призначений для операцій з одним автомобілем. Він визначає найкоротший можливий маршрут, який відвідує всі пункти доставки рівно один раз і повертається до початкової локації. Зосереджуючись на мінімізації загальної відстані, TSP ідеально підходить для забезпечення ефективних маршрутів, коли залучений лише один транспортний засіб.

Алгоритм NN - це простий і швидкий підхід, який обирає найближчу точку доставки як наступну зупинку в послідовності. Хоча цей алгоритм ефективний з погляду обчислень, він не завжди дає оптимальні результати. Він найкраще підходить для невеликих наборів даних або ситуацій, коли швидкість має пріоритет над точністю.

Алгоритм економії Кларка-Райта підходить для оптимізації маршрутів з кількома транспортними засобами. Він групує точки доставки в кластери та розраховує найефективніші маршрути для кожного транспортного засобу, прагнучи скоротити загальну відстань і час роботи. Цей метод особливо ефективний для логістики на базі автопарку, де збалансоване використання транспортних засобів є ключовим фактором.

Після того, як користувач обирає потрібний алгоритм і натискає кнопку «Оптимізувати маршрут», всі введені дані передаються на R-скрипт через локальний

сервер за допомогою Plumber API. Далі скрипт обробляє дані, виконує обраний алгоритм і розраховує оптимальний маршрут доставки.

Потім результати надсилаються назад у вебінтерфейс і відображаються на сторінці. Ці результати включають:

- оптимальну послідовність точок доставки;
- загальна пройдена відстань;
- орієнтовний час водіння та розвантаження;
- загальний час доставки.

Оптимізований маршрут також візуалізується на інтерактивній карті, надаючи користувачам чітке та інтуїтивно зрозуміле уявлення про план доставки. Ця безшовна інтеграція JavaScript і R гарантує, що користувачі можуть ефективно розраховувати та візуалізувати оптимальні маршрути доставки з мінімальними зусиллями.

## **4.2 Результати та порівняння алгоритмів оптимізації**

Тепер ми порівняємо, як з одними й тими самими вхідними даними справляються два різні алгоритми: NN та TSP. Нижче зображена візуалізація маршруту на карті після оптимізації за методом найближчого сусіда (рис. 4.20).



Рисунок 4.20 – Побудований маршрут на карті алгоритмом NN

Route Summary
<b>Order of Travel:</b> Warehouse - Dalia Street - Centralny Avenue - Petra Saranchuka Lane - Spaskyi Descent - Ozerna Street - Warehouse
<b>Total Distance:</b> 29.5 km
<b>Fuel Needed:</b> 2.36 liters
<b>Driving Time:</b> 1.0172 hours
<b>Unloading Time:</b> 7 minutes
<b>Total Time:</b> 1.1339 hours

Рисунок 4.21 – Результати розрахунків

Алгоритм NN вибирає найближчу точку доставки на кожному кроці. Хоча він обчислювально простий і швидкий, він призводить до збільшення загальної відстані



та часу в порівнянні з алгоритмом TSP. Він не завжди може створити найефективніший маршрут через те, що рішення приймаються локально на кожному кроці без урахування загального маршруту.

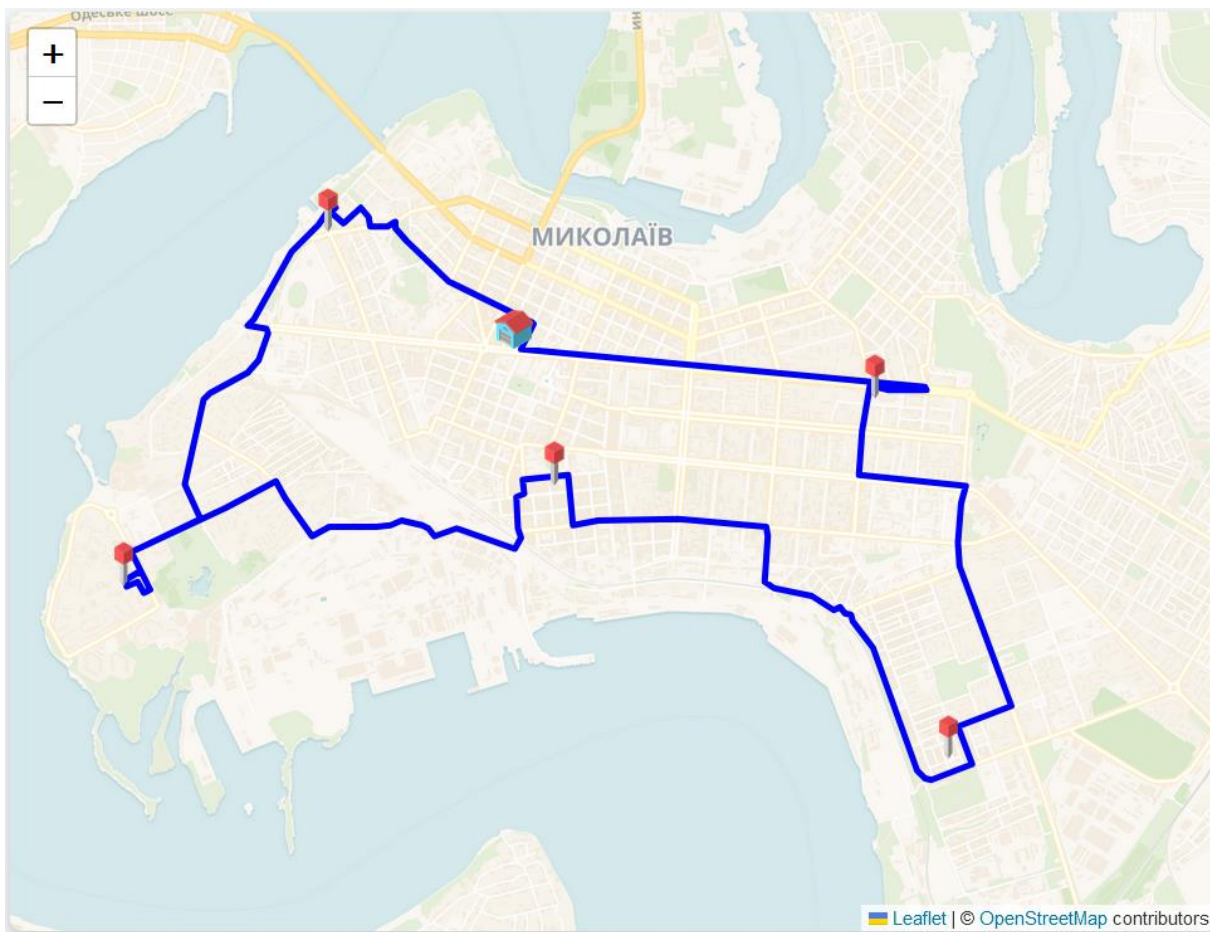


Рисунок 4.22 – Побудований маршрут на карті з алгоритмом TSP

Route Summary
<b>Order of Travel:</b> Warehouse - Spaskyi Descent - Ozerna Street - Dalia Street - Petra Saranchuka Lane - Centralny Avenue - Warehouse
<b>Total Distance:</b> 25.95 km
<b>Fuel Needed:</b> 2.076 liters
<b>Driving Time:</b> 0.8948 hours
<b>Unloading Time:</b> 6 minutes
<b>Total Time:</b> 0.9948 hours

Рисунок 4.23 – Результати розрахунків



Алгоритм TSP створює більш ефективний маршрут, мінімізуючи загальну відстань і час у дорозі. Він оцінює всі можливі маршрути, щоб знайти оптимальну послідовність точок доставки, що призводить до економії як споживання палива, так і загального часу доставки. Для зручності аналізу, відобразимо результати в порівняльній таблиці.

Таблиця 4.1 – Результати роботи двох алгоритмів та їх порівняння

Метрика	NN алгоритм	TSP алгоритм	Різниця алгоритмів (NN - TSP)
Загальна відстань (км)	29,5	25,95	+3.55
Витрата пального (л)	2,36	2,07	+0.28
Час у дорозі (год)	1,01 (~61 хв)	0,89 (~54 хв)	+0.12 (~7.3 хв)
Час розвантаження (хв)	7	7	+0
Загальний час (години)	1,13 (~68 хв)	0,99 (~60 хв)	+0.13 (~8.3 хв)

Підбиваючи підсумки, можна констатувати, що алгоритм TSP значно перевершує алгоритм NN за показниками загальної відстані, споживання пального та загального часу доставки. Він дозволяє скоротити відстань на **12%**, заощадити паливо та скоротити час у дорозі. Хоч алгоритм NN простіший і швидший в обчисленні, але жертвує ефективністю. Він більше підходить для швидких оцінок або невеликих наборів даних, але не є ідеальним для великих або складних мереж доставки.

В нашому прикладі для оптимального планування логістики з мінімальними витратами часу і коштів кращим вибором є алгоритм TSP, особливо якщо обчислювальні ресурси дозволяють його використовувати.

Далі розглянемо роботу алгоритму економії Кларка-Райта. У прикладі розглядається більша кількість точок доставки (рис. 4.24), адрес (рис. 4.25) та два транспортні засоби. Для кожної точки вказано попит (кількість автозапчастин), а для

кожного транспортного засобу – місткість (максимальна кількість автозапчастин). Крім того, створено нову матрицю відстаней (рис. 4.27).

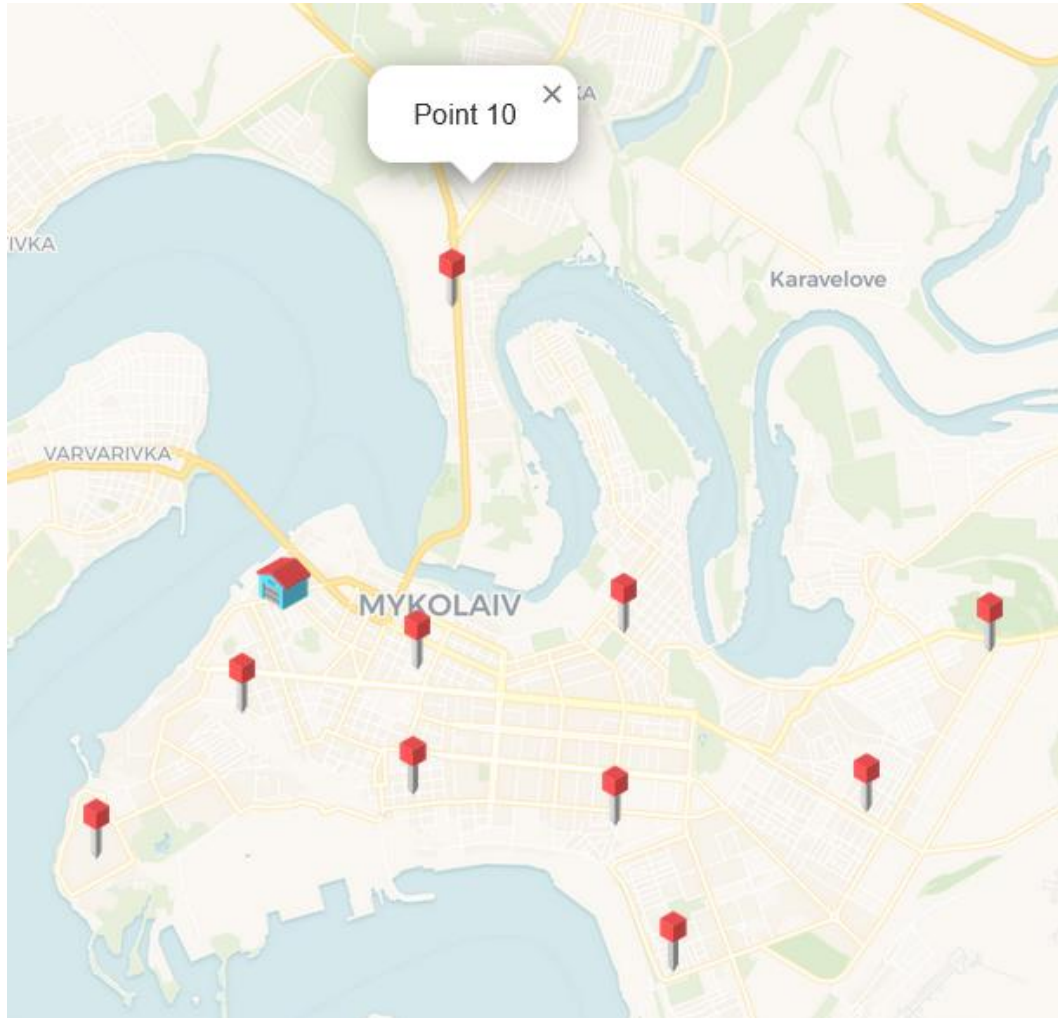


Рисунок 4.24 – Позначені точки доставки та складу автозапчастин

Locations
Warehouse: Pavla skoropadskoho Street, Mykolaiv (46.97577, 31.97245) - 0.00 km
Point 1: Marka Kropyvnytskoho Street, 74, Mykolaiv (46.96979, 31.99373) - 2.30 km
Point 2: Stoliarna Street, 35, Mykolaiv (46.97366, 32.02652) - 5.00 km
Point 3: Koval's'ka Street, 213, Mykolaiv (46.95280, 32.02515) - 6.02 km
Point 4: провулок Космонавтів, 2, Mykolaiv (46.95420, 32.06480) - 8.15 km
Point 5: Ozerna Street, 15б, Mykolaiv (46.94928, 31.94309) - 4.97 km
Point 6: Bohorodychna Street, 18, Mykolaiv (46.95608, 31.99322) - 3.54 km
Point 7: 6-a Poperechna Street, 1, Mykolaiv (46.96499, 31.96592) - 1.85 km
Point 8: Petra Saranchuka Lane, 24, Mykolaiv (46.93720, 32.03424) - 8.89 km
Point 9: Troitska Street, Mykolaiv (46.97155, 32.08454) - 9.87 km
Point 10: Flotska Street, 155, Mykolaiv (47.00880, 31.99923) - 6.32 km

Рисунок 4.25 – Перелік адрес

Number of Delivery Vehicles:

**Vehicle 1 Information**

Fuel Capacity (liters):

Truck Capacity (kg):

Fuel Consumption (L/100 km):

Average Truck Speed (km/h):

**Vehicle 2 Information**

Fuel Capacity (liters):

Truck Capacity (kg):

Fuel Consumption (L/100 km):

Average Truck Speed (km/h):

Рисунок 4.26 – Кількість транспортних засобів та їх змінні

Distance Matrix									
	Warehouse	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	Point 7	Point 8
Warehouse	0.00 km	2.18 km	4.95 km	5.92 km	8.06 km	4.94 km	3.47 km	2.12 km	8.48 km
Point 1	2.18 km	0.00 km	2.97 km	4.10 km	6.25 km	6.02 km	1.96 km	2.70 km	6.68 km
Point 2	4.95 km	2.97 km	0.00 km	3.03 km	4.54 km	8.68 km	4.31 km	5.36 km	5.31 km
Point 3	5.92 km	4.10 km	3.03 km	0.00 km	4.26 km	7.40 km	2.65 km	5.60 km	2.93 km

Рисунок 4.27 – Матриця відстаней

Після роботи алгоритму маємо наступні результати (рис. 4.28).

Route Summary
<b>Route:</b> Warehouse -> Marka Kropyvnytskoho Street, 74, Mykolaiv -> Stoliarna Street, 35, Mykolaiv -> Warehouse <b>Total Distance:</b> 10.1 km <b>Fuel Needed:</b> 0.84 liters <b>Driving Time:</b> 0.31 hours <b>Total Demand:</b> 210 <b>Assigned Vehicle:</b> Vehicle 1
Route Summary
<b>Route:</b> Warehouse -> Koval's'ka Street, 213, Mykolaiv -> провулок Космонавтів, 2, Mykolaiv -> Troitska Street, Mykolaiv -> Petra Saranchuka Lane, 24, Mykolaiv -> Warehouse <b>Total Distance:</b> 27.45 km <b>Fuel Needed:</b> 2.28 liters <b>Driving Time:</b> 0.84 hours <b>Total Demand:</b> 220 <b>Assigned Vehicle:</b> Vehicle 2
Route Summary
<b>Route:</b> Warehouse -> Ozerna Street, 156, Mykolaiv -> 6-a Poperechna Street, 1, Mykolaiv -> Bohorodychna Street, 18, Mykolaiv -> Warehouse <b>Total Distance:</b> 15.01 km <b>Fuel Needed:</b> 1.25 liters <b>Driving Time:</b> 0.46 hours <b>Total Demand:</b> 200 <b>Assigned Vehicle:</b> Vehicle 1
Route Summary
<b>Route:</b> Warehouse -> Flotska Street, 155, Mykolaiv -> Warehouse <b>Total Distance:</b> 12.28 km <b>Fuel Needed:</b> 1.02 liters <b>Driving Time:</b> 0.38 hours <b>Total Demand:</b> 85 <b>Assigned Vehicle:</b> Vehicle 2

Рисунок 4.28 – Побудовані маршрути доставки

Також ми бачимо візуалізацію зв'язків між складом та всіма точками доставки (рис. 4.29).

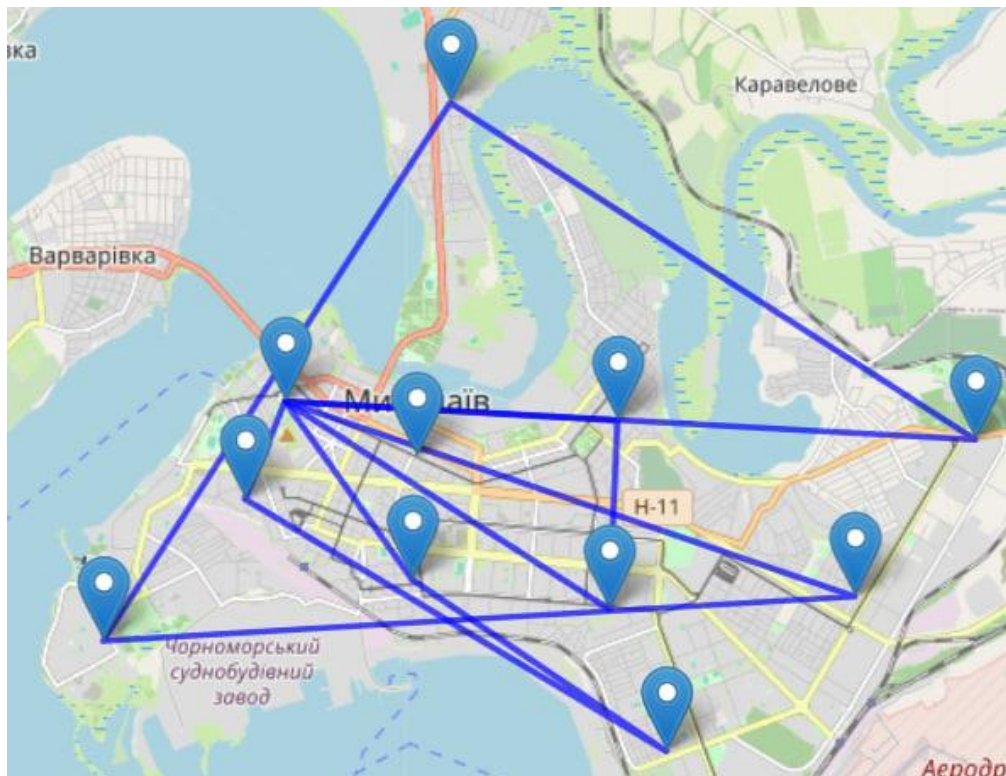


Рисунок 4.29 – Візуалізація зв’язків локацій на карті

Результати роботи алгоритму при оптимізації маршруту з 10 точками доставки зображені в табл. 4.2.

Таблиця 4.2 – Результати роботи алгоритму Кларка-Райта

Метрика	Маршрут 1	Маршрут 2	Маршрут 3	Маршрут 4	Всього
Відстань (км)	10,1	27,45	15,01	12,28	64,84
Час (години)	0,31	0,84	0,46	0,38	1,99
Паливо (літри)	0,84	2,28	1,25	1,02	5,39
Попит (одиниць)	210	220	200	85	715

За підсумком маршрути добре оптимізовані, місткість кожного транспортного засобу використовується ефективно. Маршрут 2 є найдовшим, на нього припадає **42%** загальної відстані. Він об’єднує кілька віддалених пунктів доставки, зменшуючи перекриття з іншими маршрутами. Маршрут 4, найкоротший, має одну зупинку, що

спрощує його виконання. Обидва транспортні засоби ефективно використовуються для рівномірного розподілу загального попиту, не допускаючи перевантаження жодного транспортного засобу. Середня витрата палива на кілометр становить приблизно **0,083 л/км**, що свідчить про економне планування маршруту. Всі маршрути не перевищують 1 годину, що забезпечує своєчасну доставку.

Цей план демонструє ефективний розподіл ресурсів для мінімізації загальної відстані та часу, задовольняючи при цьому всі вимоги до доставки.

#### **Висновки до розділу 4**

У цьому розділі було впроваджено комплексну систему планування та оптимізації ланцюга постачання автозапчастин. Система включає кілька модулів, таких як сторінка управління запасами, яка дозволяє користувачам ефективно відстежувати автозапчастини та керувати ними, а також модуль оптимізації доставки для планування маршрутів доставки. Модуль управління запасами надає функції для додавання, редагування та видалення продуктів, управління рівнями запасів та категоризації автозапчастин для кращої організації. Користувачі можуть вводити дані про товар, включаючи опис, кількість запасів, зображення та вагу. Система також включає інтелектуальний огляд запасів, який висвітлює дефіцитні та відсутні товари, забезпечуючи постійний моніторинг запасів на предмет необхідності їх поповнення.

На сторінці інвентаризації інтегровані опції пошуку та фільтрації в режимі реального часу, що дозволяє користувачам швидко знаходити товари на основі різних критеріїв, таких як назва товару, категорія та рівень запасів. Функція візуалізації історії запасів за допомогою графіків покращує процес прийняття рішень, прогнозуючи майбутні потреби в запасах. Крім того, система дозволяє безперешкодно імпортувати та експортувати дані про запаси, покращуючи загальний робочий процес управління. Модуль доставки інтегрує JavaScript для фронтенду, Plumber API для внутрішньої комунікації та скрипти на R для алгоритмів оптимізації маршрутів.

Користувачі можуть вводити точки доставки, місця розташування складів і специфікації транспортних засобів, вибирати алгоритм і переглядати оптимізовані маршрути та показники.

Алгоритми оптимізації включають задачу комівояжера (TSP), найближчого сусіда та економію Кларка-Райта, кожен з яких підходить для конкретних сценаріїв маршрутизації. Результати системи включають оптимальні послідовності маршрутів, загальну відстань, витрати палива та час доставки, які візуалізуються в інтерактивному режимі за допомогою Leaflet.js. Зручний інтерфейс дозволяє імпортувати/експортувати дані та візуалізувати точки доставки на карті в режимі реального часу. Ця реалізація підвищує ефективність логістики коштом автоматизації управління запасами та оптимізації маршрутів, що в кінцевому підсумку зменшує операційні витрати та час доставки, одночасно підвищуючи загальну ефективність ланцюга постачання.



## ВИСНОВКИ

В ході цієї кваліфікаційної роботи була розроблена комплексна система планування та оптимізації ланцюжка постачання автозапчастин. Система містить кілька ключових модулів, призначених для підвищення ефективності управління запасами та планування маршруту доставки. Модуль керування запасами містить функції для додавання, редагування та видалення продуктів, керування рівнями запасів і класифікації автозапчастин для кращої організації. Користувачі можуть вводити детальні дані про продукт, включаючи описи, кількість, зображення та вагу. Інтелектуальний огляд запасів виділяє дефіцитні товари та товари, яких немає в наявності, забезпечуючи своєчасне поповнення. Пошук і фільтрація в реальному часі разом із візуалізацією історії запасів полегшують ефективне прийняття рішень і прогнозування майбутніх потреб у запасах. Система також підтримує плавний імпорт і експорт даних інвентаризації, підвищуючи загальну ефективність робочого процесу.

Модуль оптимізації доставки інтегрує JavaScript для інтерфейсу, Plumber API для внутрішнього зв'язку та сценарії R для реалізації розширених алгоритмів оптимізації маршруту. Користувачі можуть визначати точки доставки, розташування складів і характеристики транспортного засобу, а потім вибрати алгоритм для розрахунку оптимальних маршрутів і переглядати ключові показники, такі як загальна відстань, споживання палива та час доставки. Реалізовані алгоритми включають проблему комівояжера, найближчого сусіда та заощаджень Кларка-Райта, які задовольняють різні логістичні сценарії. Результати інтерактивно візуалізуються за допомогою Leaflet.js, надаючи користувачам інтуїтивно зрозумілий інтерфейс на основі карти. Система також дозволяє імпортувати та експортувати дані, пов'язані з доставками, що ще більше покращує її використання.

Впровадження успішно вирішує ключові проблеми ланцюжка постачання шляхом автоматизації управління запасами та оптимізації маршрутів доставки. Це

зменшує експлуатаційні витрати, скорочує час доставки та покращує загальну ефективність ланцюжка постачання.

У майбутньому систему можна вдосконалити, додавши більш просунуті алгоритми, щоб запропонувати користувачам ширший діапазон оптимізації. Крім того, скорочення часу, необхідного для розрахунку матриці відстані, є пріоритетом для подальшого розвитку. Наразі обробка понад 10 точок доставки може тривати до 30 секунд, а для понад 25 точок може тривати до 1 хвилини. Оптимізація цих обчислень значно покращить масштабованість системи та покращить роботу користувача.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. ОСНОВНІ ФАКТОРИ ЗОВНІШНЬОГО МАКРОСЕРЕДОВИЩА ФУНКЦІОНУВАННЯ СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВ | Л А Євчук | Ефективна економіка №1 2009. Журнал «Ефективна економіка» - наукове фахове видання з питань економіки. URL: <http://www.economy.nauka.com.ua/?op=1&z=6> (дата звернення: 28.11.2024).
2. Miller, Ch., & Sirgy, M. (2011). Impact of globalization of the automotive industry on the quality of life of the US Southeast. In *The Economic Geography of Globalization* (pp. 146-162). DOI: 10.5772/17674 (дата звернення: 28.11.2024).
3. Lam V. V. SAP supply chain management (SCM): features & benefits. *Shipping Optimization Blog*. URL: <https://blog.shiperp.com/sap-101-supply-chain-management-scm> (дата звернення: 28.11.2024).
4. What is Oracle SCM Cloud. *Blog | ennVee*. URL: <https://blog.ennvee.com/blog/oracle-scm-cloud> (дата звернення: 28.11.2024).
5. Contributors to Wikimedia projects. Blue yonder - wikipedia. Wikipedia, the free encyclopedia. URL: [https://en.wikipedia.org/wiki/Blue\\_Yonder](https://en.wikipedia.org/wiki/Blue_Yonder) (дата звернення: 28.11.2024).
6. Modernize your supply chain with SCM software. URL: <https://www.infor.com/solutions/scm> (дата звернення: 28.11.2024).
7. Lusiani A., Sartika E., Binarto A., Habinuddin E., Azis I. (2021). Determination of the Fastest Path on Logistics Distribution by Using Dijkstra Algorithm. *Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021)*, 246-250. DOI: 10.2991/aer.k.211106.039 (дата звернення: 28.11.2024).
8. Tarek Sadraoui, and Nejib Mchirgui, “Supply Chain Management Optimization within Information System Development.” *International Journal of*

Econometrics and Financial Management, vol. 2, no. 2 (2014): 59-71. doi: 10.12691/ijefm-2-2-2 (дата звернення: 28.11.2024).

9. Masoud, Sherif A., "Integrated Models and Algorithms for Automotive Supply Chain Optimization" (2014). All Dissertations. 1860. [https://open.clemson.edu/all\\_dissertations/1860](https://open.clemson.edu/all_dissertations/1860) (дата звернення: 28.11.2024).

10. A multi-objective meta-heuristic approach for the design and planning of green supply chains - MBSA / N. Chibeles-Martins та ін. Expert Systems with Applications. 2016. Т. 47. С. 71–84. URL: <https://doi.org/10.1016/j.eswa.2015.10.036> (дата звернення: 28.11.2024).

11. Filip F. G., Duta L. Decision Support Systems in Reverse Supply Chain Management. Procedia Economics and Finance. 2015. Т. 22. С. 154–159. URL: [https://doi.org/10.1016/s2212-5671\(15\)00249-x](https://doi.org/10.1016/s2212-5671(15)00249-x) (дата звернення: 28.11.2024).

12. A Multiobjective Optimization Model in Automotive Supply Chain Networks / A. Sadrnia та ін. Mathematical Problems in Engineering. 2013. Т. 2013. С. 1–10. URL: <https://doi.org/10.1155/2013/823876> (дата звернення: 28.11.2024).

13. Shin J., Kim S., Lee J.-M. Production and inventory control of auto parts based on predicted probabilistic distribution of inventory. Digital Communications and Networks. 2015. Т. 1, № 4. С. 292–301. URL: <https://doi.org/10.1016/j.dcan.2015.10.002> (дата звернення: 28.11.2024).

14. Alimohammadi Ardekani M., Kabiri Naeini M. An integrated dynamic model to locate a competitive closed-loop supply chain facility under conditions of uncertainty: a case study of the auto parts industry. RAIRO - operations research. 2022. URL: <https://doi.org/10.1051/ro/2022091> (дата звернення: 29.11.2024).

15. Automobile chain maintenance parts delivery problem using an improved ant colony algorithm / J. Gao та ін. Advances in Mechanical Engineering. 2016. Т. 8, № 9. С. 168781401666529. URL: <https://doi.org/10.1177/1687814016665297> (дата звернення: 28.11.2024).

16. alavikia, seyed ali, TAGHAVIFARD, MOHAMMAD TAGHI, AMIRI, MAGHSOUD, & AZIMI, PARHAM. (2019). A quantitative model for optimization and disruption mitigation in a supply chain (Auto parts case study). *JOURNAL OF INDUSTRIAL ENGINEERING AND MANAGEMENT STUDIES (JIEMS)*, 6(1), 119-129. SID. <https://sid.ir/paper/345071/en> (дата звернення: 28.11.2024).
17. Enderson Luiz Pereira, Miguel Ângelo Lellis Morreira, Carlos Francisco Simões Gomes, Marcos dos Santos, Arthur Pinheiro de Araújo Costa, Stephanye dos Santos Sbano Chagas, Igor Pinheiro de Araújo Costa, Emerson Hissao Kojima, Supply Chain Management (SCM): an Analysis based on the CRITIC-GRA-3N Method in the selection of auto parts suppliers for an auto parts dealer in the city of Guaratinguetá, *Procedia Computer Science*, Volume 221, 2023, Pages 402-409, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2023.07.055> (дата звернення: 28.11.2024).
18. A review on reinforcement learning algorithms and applications in supply chain management / B. Rolf та ін. *International Journal of Production Research*. 2022. С. 1–29. URL: <https://doi.org/10.1080/00207543.2022.2140221> (дата звернення: 28.11.2024).
19. Lv A., Sun B. Multi-Objective Robust Optimization for the Sustainable Location-Inventory-Routing Problem of Auto Parts Supply Logistics. *Mathematics*. 2022. Т. 10, № 16. С. 2942. URL: <https://doi.org/10.3390/math10162942> (дата звернення: 28.11.2024).
20. Nikolopoulos K. I., Babai M. Z., Vozos K. Forecasting supply chain sporadic demand with nearest neighbor approaches. *International Journal of Production Economics*. 2016. Т. 177. С. 139–148. URL: <https://doi.org/10.1016/j.ijpe.2016.04.013> (дата звернення: 28.11.2024).
21. Pourhejazy P., Kwon O. The New Generation of Operations Research Methods in Supply Chain Optimization: A Review. *Sustainability*. 2016. Т. 8, № 10. С. 1033. URL: <https://doi.org/10.3390/su8101033> (дата звернення: 28.11.2024).

22. Giannakis M., Spanaki K., Dubey R. A cloud-based supply chain management system: effects on supply chain responsiveness. *Journal of Enterprise Information Management*. 2019. Т. 32, № 4. С. 585–607. URL: <https://doi.org/10.1108/jeim-05-2018-0106> (дата звернення: 10.12.2024).
23. Dumitrascu O., Dumitrascu M., Dobrotă D. Performance Evaluation for a Sustainable Supply Chain Management System in the Automotive Industry Using Artificial Intelligence. *Processes*. 2020. Т. 8, № 11. С. 1384. URL: <https://doi.org/10.3390/pr8111384> (дата звернення: 10.12.2024).
24. Shafiei Nikabadi M. A framework for technology-based factors for knowledge management in supply chain of auto industry. *VINE*. 2014. Т. 44, № 3. С. 375–393. URL: <https://doi.org/10.1108/vine-09-2013-0057> (дата звернення: 10.12.2024).
25. Seth M., Goyal D. P., Kiran R. Development of a Model for Successful Implementation of Supply Chain Management Information System in Indian Automotive Industry. *Vision: The Journal of Business Perspective*. 2015. Т. 19, № 3. С. 248–262. URL: <https://doi.org/10.1177/0972262915599465> (дата звернення: 10.12.2024).
26. Botha A., Grobler J., Yadavalli V. S. S. System dynamics comparison of three inventory management models in an automotive parts supply chain. *Journal of Transport and Supply Chain Management*. 2017. Т. 11. URL: <https://doi.org/10.4102/jtscm.v11i0.281> (дата звернення: 10.12.2024).
27. Yao J., Guo W. Research on the Multi-Level Inventory Optimization of Automotive Parts Supply Chain Systems. *Research Journal of Applied Sciences, Engineering and Technology*. 2013. Т. 5, № 19. С. 4701–4704. URL: <https://doi.org/10.19026/rjaset.5.4305> (дата звернення: 10.12.2024).
28. Supply Chain Management Market Size, Trend & Outlook by 2033. Market Research and Consulting | Future Market Insights, Inc. URL: <https://www.futuremarketinsights.com/reports/supply-chain-management-market> (дата звернення: 28.11.2024).

29. Routing & Input. An API Generator for R • plumber. URL: <https://www.rplumber.io/articles/routing-and-input.html> (дата звернення: 28.11.2024).
30. D3 or Chart.js for Data Visualisation?. Create With Data. URL: <https://www.createwithdata.com/d3js-or-chartjs/> (дата звернення: 28.11.2024).
31. Що таке AJAX та як вона працює. Дзудзило 😊. URL: <https://dzudzylo.com/javascript/shcho-take-ajax-ta-iak-tsia-tekhnohohiia-pratsiuie.html> (дата звернення: 28.11.2024).
32. Chandra A. ClarkeWright Saving Heuristics | PDF | Computer Programming | Discrete Mathematics. Scribd. URL: <https://www.scribd.com/document/401684845/ClarkeWright-Saving-Heuristics> (дата звернення: 28.11.2024).
33. 16 Types of sensors used in IoT - Freeway. Freeway. URL: <https://freeway.com/16-types-of-sensors-used-in-iot/> (дата звернення: 28.11.2024).
34. What are RFID tags, what are they used for? - camcode. Camcode. URL: <https://www.camcode.com/blog/what-are-rfid-tags/> (дата звернення: 28.11.2024).
35. Winters P. R. Forecasting sales by exponentially weighted moving averages. Management science. 1960. Т. 6, № 3. С. 324–342. URL: <https://doi.org/10.1287/mnsc.6.3.324> (дата звернення: 28.11.2024).
36. Open source routing machine - openstreetmap wiki. OpenStreetMap Wiki. URL: [https://wiki.openstreetmap.org/wiki/Open\\_Source\\_Routing\\_Machine](https://wiki.openstreetmap.org/wiki/Open_Source_Routing_Machine) (дата звернення: 28.11.2024).
37. Uk:nominatim - openstreetmap wiki. OpenStreetMap Wiki. URL: <https://wiki.openstreetmap.org/wiki/Uk:Nominatim> (дата звернення: 28.11.2024).

## ДОДАТОК А

### Апробація роботи

Міністерство освіти і науки України  
Чорноморський національний  
університет ім. Петра Могили  
Факультет комп'ютерних наук  
Кафедра інтелектуальних інформаційних  
систем

УДК 004.896

*Мілев В. В.*  
*магістрант,*  
*Козлов О. В.*  
*д-р техн. наук, професор,*  
*ЧНУ імені Петра Могили*



#### ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ПЛАНУВАННЯ ТА ОПТИМІЗАЦІЇ ЛАНЦЮГІВ ПОСТАЧАННЯ АВТОЗАПЧАСТИН

У цій доповіді досліджується питання підвищення ефективності процесів доставки в ланцюгах постачання автозапчастин за рахунок розробки та впровадження спеціалізованої інформаційної системи. Метою даного дослідження є розробка системи оптимізації логістики та транспортування автозапчастин зі складів у роздрібні точки. Ця система дозволяє користувачам вести облік автозапчастин, оцінювати різні алгоритми оптимізації постачань та розраховувати експлуатаційні витрати, враховуючи низку факторів, такі як відстань, час доставки та використання ресурсів. Структура запропонованої інформаційної системи зображена на рис. 1.

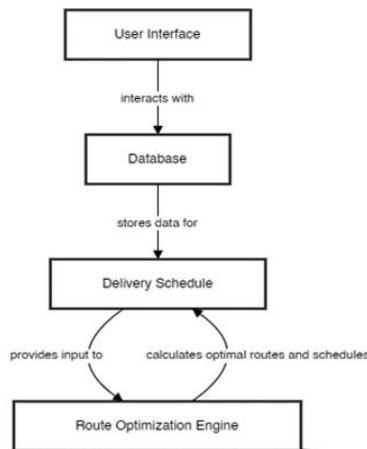
#### Інформаційний лист

*Всеукраїнська науково-  
практична конференція  
молодих вчених, аспірантів і  
студентів*

#### Інтелектуальні інформаційні системи

2 – 4 грудня 2024 року

Миколаїв





## ДОДАТОК Б

### Лістинг коду інформаційної системи

```
document.addEventListener("DOMContentLoaded", function () {
  const addProductBtn = document.getElementById("add-product-btn");
  const addProductModal = document.getElementById("add-product-modal");
  const cancelAddProductBtn = document.getElementById("cancel-add-product");
  const productForm = document.getElementById("product-form");
  const submitButton = productForm.querySelector('button[type="submit"]');
  const imagePreview = document.getElementById("image-preview");
  const searchBar = document.getElementById("search-bar");
  const productCategoryDropdown = document.getElementById("product-category");

  let isEditing = false;
  let editingProductId = null;
  let currentSort = { column: "name", direction: "asc" };
  let searchQuery = "";

  const categories = [
    "Everything for maintenance",
    "Engine and exhaust system",
    "Body and components",
    "Suspension and steering",
    "Brake system",
    "Gearbox",
    "Cooling and heating",
    "Electrics and lighting",
    "Auto chemicals and auto goods",
  ];

  function populateCategoryDropdown() {
    productCategoryDropdown.innerHTML = categories
      .map((category) => `<option value="${category}">${category}</option>`)
      .join("");
  }

  populateCategoryDropdown();

  addProductBtn.addEventListener("click", function () {
    openProductModal();
  });

  cancelAddProductBtn.addEventListener("click", function () {
    closeProductModal();
  });

  document
    .getElementById("product-image")
    .addEventListener("change", function (e) {
      const file = e.target.files[0];
      if (file) {
        const reader = new FileReader();
        reader.onload = function (e) {
```

```
        imagePreview.style.display = "block";
        imagePreview.src = e.target.result;
    });
    reader.readAsDataURL(file);
} else {
    imagePreview.style.display = "none";
}
});

function openProductModal(product = null) {
    if (product) {
        isEditing = true;
        editingProductId = product._id;
        submitButton.textContent = "Save Changes";
        document.getElementById("product-name").value = product.name;
        document.getElementById("product-description").value =
            product.description;
        document.getElementById("product-category").value = product.category;
        document.getElementById("product-stock").value = product.stock;
        document.getElementById("product-weight").value = product.weight;
        imagePreview.src = product.image;
        imagePreview.style.display = product.image ? "block" : "none";
    } else {
        isEditing = false;
        editingProductId = null;
        submitButton.textContent = "Add New Product";
        productForm.reset();
        imagePreview.style.display = "none";
    }
    addProductModal.style.display = "block";
}

function closeProductModal() {
    addProductModal.style.display = "none";
}

productForm.addEventListener("submit", function (e) {
    e.preventDefault();

    const stock = document.getElementById("product-stock").value;
    if (stock < 0) {
        alert("Stock quantity cannot be negative.");
        return;
    }

    const formData = new FormData(productForm);
    const token = localStorage.getItem("token");
    if (!token) {
        alert("You must be logged in to add a product.");
        window.location.href = "login.html";
        return;
    }

    let url = "http://localhost:3000/api/products";
    let method = "POST";
```

```
if (isEditing) {
  url += `/${editingProductId}`;
  method = "PUT";
}

fetch(url, {
  method: method,
  headers: {
    Authorization: `Bearer ${token}`,
  },
  body: formData,
})
.then((response) => response.json())
.then(() => {
  alert(
    isEditing
      ? "Product updated successfully!"
      : "Product added successfully!"
  );
  closeProductModal();
  loadProducts();
})
.catch((error) => {
  console.error("Error saving product:", error);
  alert("There was an error saving the product.");
});
});

function loadProducts() {
  const token = localStorage.getItem("token");
  if (!token) {
    alert("You must be logged in to view the products.");
    window.location.href = "login.html";
    return;
  }

  fetch("http://localhost:3000/api/products", {
    method: "GET",
    headers: { Authorization: `Bearer ${token}` },
  })
  .then((response) => response.json())
  .then(async (data) => {
    updateStockOverview(data);

    data = filterProducts(data, searchQuery);
    data = sortProducts(data, currentSort.column, currentSort.direction);

    const productList = document.getElementById("product-list");
    productList.innerHTML = "";
    for (const product of data) {
      const history = JSON.stringify(product.history || []);

      const smoothedStock = await getSmoothedStock(product._id);
```

```

const row = document.createElement("tr");
row.innerHTML = `
    <td class="hidden-id">${product._id}</td>
    <td>${product.name}</td>
    <td>${product.description}</td>
    <td>${product.category}</td>
    <td class="stock-cell" data-history='${history}'>${
        product.stock
    }</td>
    <td>${
        smoothedStock !== null ? smoothedStock.toFixed(2) : "N/A"
    }</td>
    <td>${product.weight}</td>
    <td></td>
    <td>
        ${
            product.stock < 5
            ? '<span class="low-stock">Low Stock</span>'
            : ""
        }
        <button class="actions-btn edit" onclick="editProduct('${
            product._id
        }')">
            <i class="fas fa-edit"></i> Edit
        </button>
        <button class="actions-btn delete" onclick="deleteProduct('${
            product._id
        }')">
            <i class="fas fa-trash-alt"></i> Delete
        </button>
    </td>
`;
productList.appendChild(row);
}

initializeStockChartHover();
})
.catch((error) => {
    console.error("Error loading products:", error);
    alert("There was an error loading the products.");
});
}

function initializeStockChartHover() {
    const tooltip = document.getElementById("chart-tooltip");
    const canvas = document.getElementById("stock-chart");
    let chartInstance = null;

    document.querySelectorAll(".stock-cell").forEach((cell) => {
        cell.addEventListener("mouseenter", (event) => {
            const history = JSON.parse(event.target.getAttribute("data-history"));
            if (!history.length) return;

```

```
const labels = history.map((entry) =>
  new Date(entry.timestamp).toLocaleDateString()
);
const data = history.map((entry) => entry.stock);

tooltip.style.left = `${event.pageX + 10}px`;
tooltip.style.top = `${event.pageY + 10}px`;
tooltip.style.display = "block";

if (chartInstance) {
  chartInstance.destroy();
}

chartInstance = new Chart(canvas.getContext("2d"), {
  type: "line",
  data: {
    labels: labels,
    datasets: [
      {
        label: "Stock History",
        data: data,
        borderColor: "rgba(75, 192, 192, 1)",
        backgroundColor: "rgba(75, 192, 192, 0.2)",
        fill: true,
      },
    ],
  },
  options: {
    responsive: false,
    plugins: {
      legend: {
        display: false,
      },
    },
  },
  scales: {
    x: {
      display: true,
      title: {
        display: true,
        text: "Date",
      },
    },
    y: {
      display: true,
      title: {
        display: true,
        text: "Stock History",
      },
    },
  },
});
});

cell.addEventListener("mouseleave", () => {
```

```
        tooltip.style.display = "none";
    });
});
}

async function getSmoothedStock(productId) {
    const token = localStorage.getItem("token");
    if (!token) {
        alert("You must be logged in to view stock analysis.");
        window.location.href = "login.html";
        return null;
    }

    try {
        const response = await fetch(
            `http://localhost:3000/api/products/${productId}/smoothed`,
            {
                headers: {
                    Authorization: `Bearer ${token}`,
                },
            }
        );
        if (!response.ok) {
            throw new Error("Failed to fetch smoothed stock.");
        }
        const data = await response.json();
        return data.smoothedStock;
    } catch (error) {
        console.error("Error fetching smoothed stock:", error);
        return null;
    }
}

function updateStockOverview(products) {
    let totalStock = 0;
    let lowStock = 0;
    let outOfStock = 0;

    products.forEach((product) => {
        totalStock += product.stock;
        if (product.stock === 0) {
            outOfStock++;
        } else if (product.stock < 5) {
            lowStock++;
        }
    });
    document.getElementById(
        "total-stock"
    ).innerText = `Total Stock: ${totalStock}`;
    document.getElementById("low-stock").innerText = `Low Stock: ${lowStock}`;
    document.getElementById(
        "out-of-stock"
    ).innerText = `Out of Stock: ${outOfStock}`;
}
```

```
function sortProducts(products, column, direction) {
  return products.sort((a, b) => {
    const valA = a[column];
    const valB = b[column];

    if (direction === "asc") {
      return valA < valB ? -1 : valA > valB ? 1 : 0;
    } else {
      return valA > valB ? -1 : valA < valB ? 1 : 0;
    }
  });
}

function filterProducts(products, query) {
  return products.filter(
    (product) =>
      product.name.toLowerCase().includes(query.toLowerCase()) ||
      product.description.toLowerCase().includes(query.toLowerCase()) ||
      product.category.toLowerCase().includes(query.toLowerCase())
  );
}

window.searchProducts = function () {
  searchQuery = searchBar.value.trim();
  loadProducts();
};

window.editProduct = function (id) {
  const token = localStorage.getItem("token");
  if (!token) {
    alert("You must be logged in to edit a product.");
    window.location.href = "login.html";
    return;
  }

  fetch(`http://localhost:3000/api/products/${id}`, {
    headers: {
      Authorization: `Bearer ${token}`,
    },
  })
  .then((response) => {
    if (!response.ok) {
      throw new Error("Unauthorized or product not found.");
    }
    return response.json();
  })
  .then((product) => {
    openProductModal(product);
  })
  .catch((error) => {
    console.error("Error loading product:", error);
    alert("Could not load product details.");
  });
};
```

```
window.deleteProduct = function (id) {
  const token = localStorage.getItem("token");
  if (!token) {
    alert("You must be logged in to delete a product.");
    window.location.href = "login.html";
    return;
  }

  fetch(`http://localhost:3000/api/products/${id}`, {
    method: "DELETE",
    headers: { Authorization: `Bearer ${token}` },
  })
  .then(() => {
    alert("Product deleted successfully!");
    loadProducts();
  })
  .catch((error) => {
    console.error("Error deleting product:", error);
    alert("There was an error deleting the product.");
  });
};

window.changeSort = function (column) {
  const direction =
    currentSort.column === column && currentSort.direction === "asc"
      ? "desc"
      : "asc";
  currentSort = { column, direction };
  loadProducts();
};

loadProducts();
});
```