

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Додаток для планування логістики підприємства
на основі аналізу споживчого попиту на продукцію»**

Виконав студент групи ІПЗм-24-1
БАСИСТИЙ Іван Олександрович

Керівник роботи:
ЧИЖМОТРЯ Олена Геннадіївна

Рецензент:
ВЛАСЕНКО Олег Васильович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«18» вересня 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **БАСИСТИЙ Іван Олександрович**

Керівник роботи: **ЧИЖМОТРЯ Олена Геннадіївна**

Тема роботи: **«Додаток для планування логістики підприємства на основі аналізу споживчого попиту на продукцію»,**

затверджена Наказом закладу вищої освіти від **«18» вересня 2025 р. № 491/с**

Вихідні дані для роботи: **методи та інформаційні технології автоматизації логістичного планування з використанням аналізу даних споживчого попиту; процеси управління матеріальними потоками та планування запасів на підприємстві.**

Консультанти з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря О.Г.	22.09.2025р.	22.09.2025р.
2	Чижмотря О.Г.	22.09.2025р.	22.09.2025р.
3	Чижмотря О.Г.	22.09.2025р.	22.09.2025р.

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «магістр» присвячена розробці програмного забезпечення для оптимізації логістичних процесів підприємства та прогнозування потреб ринку.

Пояснювальна записка складається з вступу, трьох розділів та списку використаної літератури. Пояснювальна записка до кваліфікаційної магістерської роботи містить 59 сторінок, 19 ілюстрацій, 7 таблиць, 2 додатки та 40 посилань на зовнішні джерела.

Метою роботи є створення додатку для автоматизації планування поставок та управління складськими запасами на основі аналітичних даних про споживчий попит.

В роботі описано процес проєктування архітектури додатку, вибір технологічного стеку та реалізацію алгоритмів обробки даних продажів. Досліджено методи прогнозування попиту (наприклад, методи часових рядів або регресійний аналіз) для оптимізації логістичних ланцюгів. Розроблено зручний графічний інтерфейс для взаємодії користувача з системою. Проведено тестування коректності роботи алгоритмів та стабільності додатку.

КЛЮЧОВІ СЛОВА: ЛОГІСТИКА, ПРОГНОЗУВАННЯ ПОПИТУ, УПРАВЛІННЯ ЗАПАСАМИ, ВЕБ-ДОДАТОК, АНАЛІЗ ДАНИХ, ОПТИМІЗАЦІЯ ПОСТАВОК.

ABSTRACT

The qualification work for the educational degree of ‘Master’ is devoted to the development of software for optimising the logistics processes of an enterprise and forecasting market demand.

The explanatory note consists of an introduction, three chapters and a list of references. The explanatory note to the Master's thesis contains 59 pages, 19 illustrations, 7 tables, 2 appendices and 40 references to external sources.

The aim of the thesis is to create an application for automating supply planning and warehouse inventory management based on analytical data on consumer demand.

The thesis describes the process of designing the application architecture, selecting the technology stack, and implementing sales data processing algorithms. Methods of demand forecasting (e.g., time series methods or regression analysis) for optimising logistics chains are investigated. A user-friendly graphical interface for user interaction with the system has been developed. The correctness of the algorithms and the stability of the application have been tested.

KEYWORDS: LOGISTICS, DEMAND FORECASTING, INVENTORY MANAGEMENT, WEB APPLICATION, DATA ANALYSIS, SUPPLY OPTIMISATION.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1. Аналіз проблем і особливостей процесу планування логістики та прогнозування попиту	9
1.2. Постановка задачі та перелік задач, що підлягають автоматизації.....	11
1.3. Аналіз аналогів програмного продукту.....	13
1.4. Вибір засобів реалізації та вимоги до апаратних засобів.	15
Висновки до першого розділу	17
РОЗДІЛ 2. РОЗРОБКА ТА ПРОЄКТУВАННЯ СИСТЕМИ АНАЛІЗУ ЛОГІСТИЧНИХ ПРОЦЕСІВ	18
2.1. Підготовка та попередній аналіз даних	18
2.2. Проєктування та побудова моделей машинного навчання	22
2.3. Проєктування архітектури програмного забезпечення.....	26
Висновки до другого розділу.....	34
РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ	36
3.1. Графічний інтерфейс	36
3.2. Налаштування історичних даних для додатку планування логістики	42
3.3. Відображення результатів аналізу та прогнозів	44
Висновки до третього розділу	48
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	55

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SCM - (англ. Supply Chain Management) - управління ланцюгами поставок.

ERP - (англ. Enterprise Resource Planning) - планування ресурсів підприємства.

SKU - (англ. Stock Keeping Unit) - ідентифікатор товарної позиції (одиниця складського обліку).

CRM - (англ. Customer Relationship Management) - система управління взаємовідносинами з клієнтами.

API - (англ. Application Programming Interface) - прикладний програмний інтерфейс.

UI - (англ. User Interface) - інтерфейс користувача.

UX - (англ. User Experience) - досвід користувача.

JSON - (англ. JavaScript Object Notation) - текстовий формат обміну даними.

MAPE - (англ. Mean Absolute Percentage Error) - середня абсолютна відсоткова похибка (метрика точності прогнозу).

RMSE - (англ. Root Mean Square Error) - середньоквадратична похибка.

ВСТУП

Актуальність теми. Сучасний ринок характеризується високою динамікою змін потреб споживачів, що ставить перед підприємствами нові виклики у сфері управління товарними запасами та логістикою. Ефективність логістичної системи є одним із ключових факторів конкурентоспроможності бізнесу. Традиційні методи планування поставок, що базуються на інтуїтивних рішеннях або статичних звітах, часто призводять або до дефіциту популярних товарів, або до затоварювання складів неліквідною продукцією.

Сучасні інформаційні технології дозволяють вирішити цю проблему шляхом впровадження автоматизованих систем, здатних аналізувати великі масиви даних про продажі та прогнозувати попит. Розробка спеціалізованого програмного забезпечення для планування логістики на основі аналізу споживчого попиту дозволяє мінімізувати витрати на зберігання, оптимізувати маршрути постачання та підвищити рівень обслуговування клієнтів. Тому тема дослідження є актуальною та має практичне значення для автоматизації бізнес-процесів підприємства.

Метою роботи є дослідження методів прогнозування попиту, проєктування архітектури та розробка програмного додатку для автоматизації планування логістичних процесів підприємства на основі аналітичних даних.

Поставлена мета обумовлює необхідність виконання наступних завдань:

- провести аналіз предметної області та існуючих рішень у сфері логістичного планування;
- визначити основні фактори, що впливають на споживчий попит, та методи їх врахування;
- сформулювати функціональні та нефункціональні вимоги до системи;
- обґрунтувати вибір засобів реалізації (стеку технологій);
- спроектувати мікросервісну архітектуру програмного забезпечення;
- розробити клієнтську та серверну частини додатку, модуль аналітики;
- провести тестування розробленої системи.

Об'єктом дослідження є процеси управління матеріальними потоками та планування запасів на підприємстві.

Предметом дослідження є методи та інформаційні технології автоматизації логістичного планування з використанням аналізу даних споживчого попиту.

Наукова новизна роботи полягає у вдосконаленні підходу до автоматизації логістичного планування шляхом поєднання веб-технологій та спеціалізованих аналітичних мікросервісів, що дозволяє, на відміну від існуючих монолітних рішень, гнучко масштабувати систему та інтегрувати складні алгоритми прогнозування без зміни основного ядра програми.

Практичне значення одержаних результатів. Розроблено повнофункціональний веб-додаток, який дозволяє автоматизувати розрахунок потреби в товарах на основі історії продажів. Впровадження системи дозволяє зменшити час на обробку замовлень та підвищити точність планування складських запасів.

Апробація. За темою кваліфікаційної магістерської роботи було опубліковано 2 тези конференції [1, 2].

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Аналіз проблем і особливостей процесу планування логістики та прогнозування попиту

В умовах глобалізації економіки та переходу до концепції Індустрії 4.0, ефективність управління ланцюгами поставок (англ. Supply Chain Management) трансформується з операційної необхідності у критичний фактор стратегічної конкурентоспроможності підприємства. Сучасні логістичні системи функціонують у середовищі високої турбулентності, де головним викликом для менеджменту стає стохастична природа та невизначеність майбутнього споживчого попиту.

Традиційні підходи до планування, що базуються на евристичних методах, експертній інтуїції менеджерів або спрощених лінійних розрахунках у електронних таблицях, демонструють свою неспроможність в умовах динамічного ринку. Вони не здатні оперативно обробляти багатofакторні залежності, що призводить до зниження рентабельності бізнесу [3].

Глибокий аналіз предметної області дозволив структурувати та деталізувати основні проблеми процесу логістичного планування, що потребують вирішення:

Проблема оптимізації балансу запасів (англ. Inventory Optimization Trade-off). Управління запасами - це постійний пошук компромісу між двома конфліктуючими фінансовими цілями: мінімізацією витрат на утримання запасів та максимізацією рівня обслуговування клієнтів (англ. Service Level).

Помилка у бік завищення прогнозу призводить до виникнення «overstock» - накопичення надлишкових запасів. Це спричиняє «заморожування» обігових коштів у неліквідних активах, зростання витрат на складську логістику та ризик фізичного або морального старіння продукції [4].

Помилка у бік заниження прогнозу викликає ситуацію «out-of-stock». Це призводить не лише до прямих фінансових збитків у вигляді втраченої вигоди, але й до довгострокових репутаційних втрат: зниження лояльності клієнтів та їх переходу до конкурентів.

Ефект хлиста (англ. Bullwhip Effect). Це явище спотворення інформації у ланцюгу поставок, коли незначні відхилення попиту з боку кінцевого споживача призводять до значних амплітудних коливань замовлень на рівнях дистриб'юторів, оптовиків та виробників. Причиною цього є відсутність інформаційної прозорості, запізнення в отриманні даних та реактивне планування. Дестабілізація ланцюга призводить до неефективного використання виробничих потужностей та хаотичних логістичних витрат.

Вплив сезонності, трендів та маркетингових факторів. Попит на переважну більшість товарних груп (SKU) не є лінійним стаціонарним процесом. Він піддається впливу складної комбінації факторів:

- коливання в межах тижня, місяця або року;
- вплив державних свят, вихідних днів;
- проведення промо-акцій, знижок, зміна цінової політики конкурентів;
- інфляційні очікування, зміна купівельної спроможності. Ручне врахування мультиплікативного впливу всіх цих змінних є надзвичайно трудомістким процесом, що неминуче призводить до помилок [5].

Проблема обробки великих даних (англ. Big Data Processing). Сучасні торговельні підприємства оперують номенклатурою, що може налічувати тисячі або десятки тисяч товарних позицій. Кожна транзакція генерує нові дані. Людина фізично не здатна проаналізувати багатовимірні часові ряди історії продажів для кожної позиції за кілька років, щоб виявити приховані (латентні) закономірності та кореляції. Обробка такого масиву інформації вимагає автоматизованих засобів [6].

Отже, ключовою особливістю предметної області є об'єктивна необхідність переходу від експертних оцінок до концепції Data-Driven Decision Making (прийняття рішень на основі даних). Це вимагає автоматизації процесів збору та очищення історичних даних, а також застосування інтелектуальних алгоритмів машинного навчання для підвищення точності прогнозів та адаптивності логістичної системи.

1.2. Постановка задачі та перелік задач, що підлягають автоматизації

В умовах нестабільності сучасної економіки та високої конкуренції на ринку рітейлу, ефективність логістичних процесів підприємства перетворюється з допоміжної функції на ключовий фактор прибутковості бізнесу. Критичного значення набуває точність прогнозування споживчого попиту, оскільки саме від неї залежить раціональне використання обігових коштів та якість обслуговування клієнтів.

Традиційні підходи до управління запасами, що базуються на інтуїтивних рішеннях менеджерів або ручному плануванні закупівель у табличних редакторах (наприклад, Excel), демонструють свою неефективність при роботі з великими номенклатурними групами. Такий підхід є надзвичайно трудомістким, не дозволяє оперативно реагувати на зміни ринкової кон'юнктури та супроводжується високим ризиком людських помилок. Наслідком цього стають прямі фінансові втрати через списання простроченої продукції або втрату прибутку через відсутність популярних товарів [7].

На основі детального аналізу господарської діяльності підприємства та огляду сучасних літературних джерел у сфері Supply Chain Management (SCM), було виявлено нагальну необхідність цифрової трансформації логістичних процесів. Оптимальним рішенням є впровадження спеціалізованої інформаційної системи, яка б об'єднала в єдиному контурі процеси збору «сирих» даних про транзакції, їх інтелектуальну обробку та формування управлінських рішень.

Основними функціональними задачами, що підлягають автоматизації в межах розробки проєктованої системи, визначено:

Збір, валідація та консолідація даних про продажі. Забезпечення автоматизованого імпорту історії транзакцій з облікових систем підприємства (ERP, CRM) через API-інтерфейси або шляхом пакетного завантаження файлів форматів CSV/XLSX. Реалізація етапу ETL (Extract, Transform, Load), що включає перевірку цілісності даних, приведення їх до єдиного формату, обробку дублікатів та заповнення пропущених значень для створення якісної вибірки.

Глибинний аналіз історичних даних (англ. Data Mining). Використання статистичних методів для дослідження динаміки продажів у минулих періодах. Задача включає автоматичне виявлення коефіцієнтів сезонності (тижневої, місячної, річної), визначення життєвого циклу товарів, ідентифікацію глобальних трендів (зростання/спаду популярності) та фільтрацію аномальних викидів, спричинених разовими акціями або форс-мажорними обставинами [8].

Інтелектуальне прогнозування попиту. Розрахунок очікуваного обсягу продажів на майбутні планові періоди (тиждень, місяць, квартал) з використанням сучасних математичних моделей. Передбачається застосування методів машинного навчання, зокрема рекурентних нейронних мереж (LSTM), для виявлення нелінійних залежностей, які неможливо врахувати класичними методами екстраполяції [9].

Розрахунок потреби у закупівлях та оптимізація запасів. Автоматичне формування рекомендацій щодо обсягів замовлення (англ. Order Quantity) для кожної товарної позиції. Алгоритм повинен враховувати поточні залишки на складах, товари в дорозі, розраховані прогнози попиту, а також логістичні параметри: час доставки (англ. Lead Time), точку перезамовлення (Reorder Point) та необхідний рівень страхового запасу (англ. Safety Stock) [10].

Візуалізація аналітики та звітність. Розробка інтерактивного дашборду для відображення ключових показників ефективності (KPI). Система повинна забезпечувати візуалізацію графіків план-фактного аналізу продажів, динаміки складських залишків, теплових карт дефіциту та надлишків у зручному для сприйняття вигляді, що дозволить логісту швидко оцінювати ситуацію та приймати рішення.

Управління нормативно-довідковою інформацією (англ. Master Data Management). Створення та ведення централізованої бази даних, що містить структуровану інформацію про номенклатуру товарів, ієрархію категорій, перелік складів зберігання та реєстр постачальників з умовами співпраці [11].

Реалізація зазначених функцій дозволить підприємству перейти від реактивної до проактивної моделі управління запасами. Це забезпечить мінімізацію рівня затовареності складів неліквідною продукцією ("overstock"),

вивільнення обігових коштів, а також суттєве зниження ризиків виникнення дефіциту товарів ("out-of-stock"), що в сукупності підвищить рівень сервісу та лояльність клієнтів.

1.3. Аналіз аналогів програмного продукту.

На ринку систем автоматизації логістики та планування ресурсів представлено низку потужних продуктів, кожен з яких має свої особливості архітектури та функціонального наповнення. Одним із лідерів корпоративного сегмента є SAP Integrated Business Planning (SAP IBP). Це рішення базується на технології in-memory бази даних SAP HANA, що забезпечує можливість обробки великих масивів даних у реальному часі. Функціональні можливості системи дозволяють моделювати різноманітні сценарії типу «що-якщо», враховувати фінансові показники при плануванні ланцюгів постачання, а також використовувати алгоритми «demand sensing» для високоточного короткострокового прогнозування. До безперечних переваг SAP IBP належать глибока інтеграція з екосистемою SAP, висока точність розрахунків та масштабованість. Однак впровадження цієї системи пов'язане з надзвичайно високими фінансовими витратами на ліцензування та консалтинг, складністю інтерфейсу, що вимагає тривалого навчання персоналу, а також надлишковістю функціоналу для потреб малого та середнього бізнесу. Вигляд рішення зображено на рисунку 1.1.

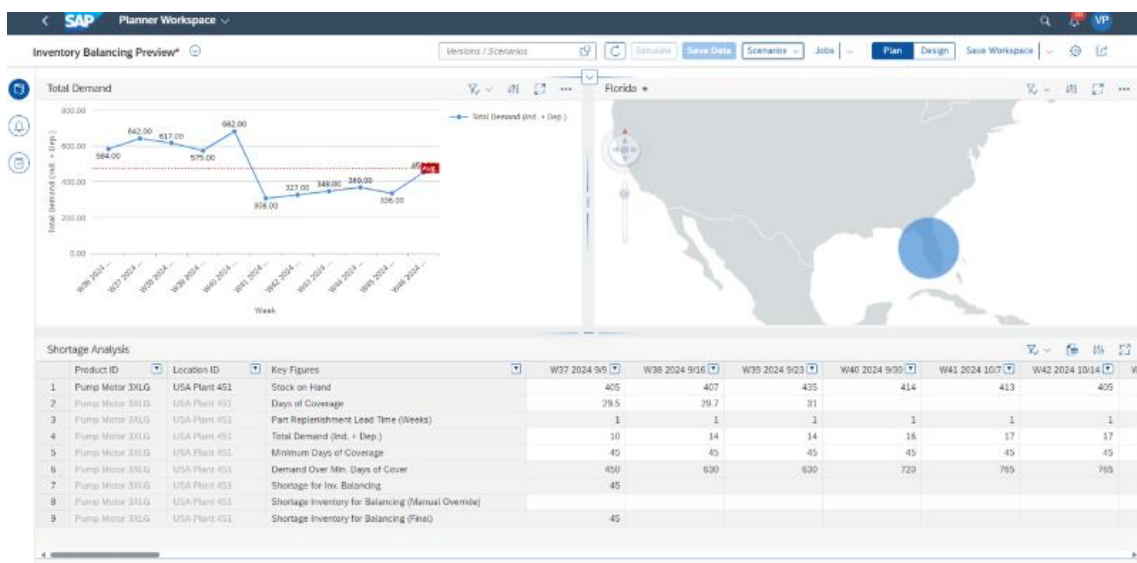


Рис. 1.1. Вигляд платформи SAP Integrated Business Planning

Альтернативним підходом вирізняється Oracle SCM Cloud - хмарна платформа, яка активно інтегрує технології Інтернету речей (IoT) та штучного інтелекту. Модуль Oracle Demand Management застосовує методи машинного навчання для виявлення сезонності, трендів та впливу промо-акцій, автоматично підбираючи оптимальну статистичну модель для кожної товарної позиції. Використання моделі SaaS позбавляє замовника необхідності утримувати власні сервери та забезпечує доступ до сучасних дашбордів і алгоритмів. Попри це, система має суттєві недоліки, серед яких висока вартість підписки, критична залежність від стабільності інтернет-з'єднання та складність кастомізації під специфічні бізнес-процеси («vendor lock-in»), що обмежує гнучкість системи для невеликих підприємств. Вигляд рішення зображено на рисунку 1.2.

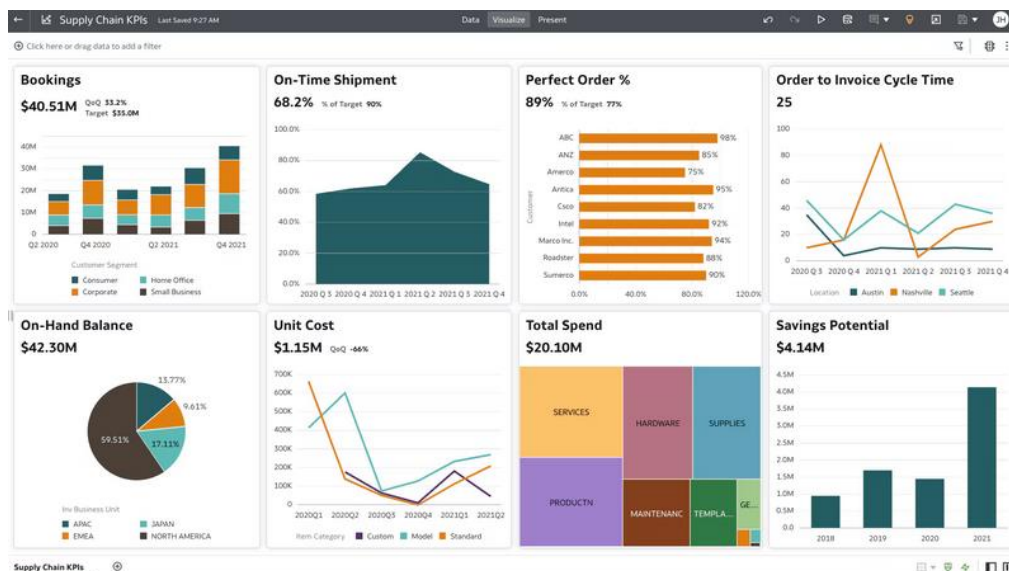


Рис. 1.2. Вигляд Oracle SCM Cloud

У сегменті рішень з відкритим вихідним кодом популярністю користується модульна ERP-система Odoo (зокрема модулі Inventory та Purchase). Базова версія продукту дозволяє ефективно керувати складськими запасами, відстежувати переміщення товарів та формувати замовлення постачальникам, використовуючи для прогнозування прості правила поповнення (Min/Max rules) або методи ковзного середнього. Головними перевагами Odoo є доступність, наявність безкоштовної Community-версії, сучасний веб-інтерфейс та використання мови Python, що спрощує доопрацювання системи. Водночас, базовий математичний апарат системи є слабким: відсутні вбудовані нейронні

мережі або складні методи прогнозування (такі як ARIMA чи LSTM). Реалізація інтелектуального аналізу попиту в середовищі Odoo вимагає або встановлення дорогих сторонніх плагінів, або значних витрат часу на розробку власного коду.

Для наочності порівняння наведено у таблиці 1.1.

Таблиця 1.1

Порівняльний аналіз систем автоматизації логістики

Критерій порівняння	SAP IBP	Oracle SCM Cloud	Odoo (Base)	Проектована система
Цільова аудиторія	Великі корпорації	Великий бізнес	Малий та середній бізнес	Середній бізнес
Методи прогнозування	Статистика + AI (Demand Sensing)	Машинне навчання (ML)	Прості статистичні (Min/Max)	Нейромережі (LSTM)
Вартість впровадження	Дуже висока	Висока (підписка)	Низька / Середня	Низька
Гнучкість налаштування	Низька (стандартизовано)	Середня	Висока	Максимальна
Складність інтерфейсу	Висока	Середня	Низька	Низька

1.4. Вибір засобів реалізації та вимоги до апаратних засобів.

Для розробки програмного комплексу обрано сучасну мікросервісну архітектуру, що дозволяє відокремити бізнес-логіку веб-додатку від ресурсномістких обчислень прогнозування. Вибір інструментальних засобів базується на критеріях швидкодії, масштабованості та наявності розвинених бібліотек для аналізу даних.

Серверна платформа. В якості серверного середовища обрано Node.js. Це кросплатформне середовище виконання JavaScript, що базується на рушії V8.

Асинхронна модель введення-виведення робить Node.js ідеальним для обробки великої кількості одночасних запитів від користувачів та швидкої передачі даних між клієнтом і мікросервісом аналітики. Використання єдиної мови JavaScript на фронтенді та бекенді спрощує розробку та підтримку коду.

Клієнтська частина. Для реалізації інтерфейсу користувача обрано фреймворк Vue.js 3 у поєднанні з CSS-фреймворком Tailwind CSS.

Vue.js це прогресивний JavaScript-фреймворк для створення односторінкових додатків (SPA). Він забезпечує реактивність даних: при зміні

параметрів прогнозу графіки та таблиці оновлюються миттєво без перезавантаження сторінки. Компонентний підхід дозволяє створювати перевикористовувані елементи (наприклад, віджет графіка попиту) [12].

Tailwind CSS це фреймворк, який дозволяє швидко створювати сучасний, адаптивний дизайн. Це критично важливо для логістичних дашбордів, які можуть переглядатися як на настільних моніторах, так і на планшетах на складі.

Для розробки модуля Data Science Microservice обрано мову програмування Python, яка є загальновизнаним стандартом у сфері аналізу даних. Технологічний стек базується на комплексному використанні спеціалізованих бібліотек, що забезпечують повний цикл обробки інформації: від підготовки даних до візуалізації результатів.

Первинна обробка та підготовка даних здійснюються за допомогою бібліотек Pandas та NumPy. Pandas виступає основним інструментом для маніпуляцій з табличними даними (англ. DataFrames), забезпечуючи завантаження CSV-файлів, очищення від пропусків та агрегацію продажів по періодах [13]. У синергії з нею працює NumPy, який відповідає за ефективні математичні операції з багатовимірними масивами, що є критично важливим етапом підготовки вхідних векторів для нейромережі.

Архітектура машинного та глибокого навчання реалізована через бібліотеки TensorFlow, Keras та Scikit-learn. Екосистема TensorFlow та Keras використана для побудови моделі LSTM, де Keras надає високорівневий API для конфігурації шарів (Sequential, LSTM, Dense) та налаштування оптимізаторів [14]. Паралельно використовується Scikit-learn для задач класичного машинного навчання та препроцесингу: зокрема, MinMaxScaler виконує нормалізацію даних, LogisticRegression застосовується для класифікації ризиків затримки доставки, а train_test_split забезпечує валідацію моделей.

Фінальний етап аналізу - візуалізація - реалізована засобами Matplotlib, Seaborn та Plotly. Ці інструменти дозволяють генерувати детальні аналітичні графіки, такі як динаміка попиту, теплові карти ризиків та географічний розподіл продажів, для їх подальшої інтеграції у звіти.

Система управління базами даних. Для збереження структурованої інформації (користувачі, каталог товарів, історія транзакцій) обрано Mongo. Вона забезпечує надійність даних, підтримку складних запитів для аналітики та можливість роботи з JSON-форматом, що спрощує інтеграцію з Node.js.

Висновки до першого розділу

У першому розділі магістерської роботи здійснено комплексний аналіз предметної області та обґрунтовано необхідність автоматизації процесів управління товарними запасами підприємства. Встановлено, що критичними факторами, які знижують ефективність логістики, є неточність прогнозування попиту, суттєвий вплив сезонності та людський фактор при ручному плануванні, що часто призводить до проблем затоварення складів або дефіциту продукції.

Проведений порівняльний аналіз існуючих на ринку рішень, таких як SAP IBP, Oracle SCM та Odoo, засвідчив, що більшість комерційних продуктів є або фінансово недоступними для середнього бізнесу, або не володіють достатньо гнучким інструментарієм для точного прогнозування специфічних трендів. Це підтвердило актуальність та доцільність розробки власного спеціалізованого веб-додатку.

На основі визначених вимог сформовано архітектурне бачення майбутньої системи та обґрунтовано вибір мікросервісного підходу, який дозволяє розділити логіку веб-додатку та обчислювального модуля. В якості технологічного стеку для реалізації клієнтської та серверної частин обрано зв'язку Node.js та Vue.js, що забезпечує високу швидкодію та зручність інтерфейсу, а для розробки аналітичного ядра обрано мову Python та документо-орієнтовану базу даних MongoDB.

Також у розділі визначено математичне забезпечення системи: для прогнозування часових рядів продажів обрано рекурентну нейронну мережу архітектури LSTM, яка демонструє високу ефективність у виявленні довгострокових залежностей, а для оцінки ризиків зриву поставок запропоновано застосувати метод логістичної регресії.

РОЗДІЛ 2. РОЗРОБКА ТА ПРОЄКТУВАННЯ СИСТЕМИ АНАЛІЗУ ЛОГІСТИЧНИХ ПРОЦЕСІВ

2.1. Підготовка та попередній аналіз даних

Ефективність роботи алгоритмів машинного навчання та нейронних мереж критично залежить від якості вхідних даних. Етап підготовки даних у даній роботі включає збір, аналіз структури, очищення від шуму, збагачення географічною інформацією та трансформацію у формат, придатний для навчання моделей.

Характеристика набору даних

У якості основи для дослідження та навчання системи було обрано набір даних DataCo Smart Supply Chain for Big Data Analysis. Цей датасет містить деперсоніфіковані записи про логістичні транзакції міжнародної компанії, що охоплюють повний цикл ланцюга постачання: від оформлення замовлення до доставки кінцевому споживачу.

Початковий обсяг вибірки складає 180 519 записів, кожен з яких характеризується 53 атрибутами (ознаками). Структура даних представлена у табличному форматі CSV (Comma-Separated Values). Для детального аналізу дані приведено до табличного вигляду та наведені в таблиці 2.1.

Таблиця. 2.1.

Опис полів набору даних

№ з/п	Назва поля	Назва поля (англійською)	Опис
1.	2.	3.	4.
1	Тип транзакції	Type	Вид здійсненої оплати або транзакції (DEBIT, TRANSFER, PAYMENT, CASH)
2	Фактичні дні доставки	Days for shipping (real)	Реальна кількість днів, фактично витрачена на доставку товару
3	Планові дні доставки	Days for shipment (scheduled)	Запланована кількість днів на доставку товару (норматив)
4	Прибуток за замовлення	Benefit per order	Чистий прибуток компанії з конкретного замовлення
5	Продажі на клієнта	Sales per customer	Загальна сума продажів для конкретного клієнта (з урахуванням знижок)
6	Статус доставки	Delivery Status	Поточний стан (Advance shipping, Late delivery, Shipping canceled, Shipping on time)

Продовження табл. 2.1.

1.	2.	3.	4.
7	Ризик затримки	Late_delivery_risk	Категоріальна цільова змінна (1 - доставка затримана, 0 - вчасно)
8	Ідентифікатор категорії	Category Id	Унікальний числовий код категорії товару
9	Назва категорії	Category Name	Текстова назва категорії продукту
10	Місто клієнта	Customer City	Місто, де клієнт здійснив покупку (або адреса реєстрації)
11	Країна клієнта	Customer Country	Країна проживання клієнта
12	Email клієнта	Customer Email	Електронна адреса замовника
13	Ім'я клієнта	Customer Fname	Ім'я замовника
14	Ідентифікатор клієнта	Customer Id	Унікальний ідентифікатор клієнта в базі даних
15	Прізвище клієнта	Customer Lname	Прізвище замовника
16	Пароль	Customer Password	Захешований ключ доступу клієнта (не використовується в аналізі)
17	Сегмент клієнта	Customer Segment	Тип клієнта (Consumer, Corporate, Home Office)
18	Штат клієнта	Customer State	Регіон або штат проживання клієнта
19	Вулиця клієнта	Customer Street	Адреса проживання (вулиця)
20	Індекс клієнта	Customer Zipcode	Поштовий індекс замовника
21	Ідентифікатор магазину	Department Id	Унікальний код відділу магазину
22	Назва магазину	Department Name	Текстова назва відділу магазину
23	Широта	Latitude	Географічна широта магазину або складу
24	Довгота	Longitude	Географічна довгота магазину або складу
25	Ринок	Market	Ринок, на який здійснюється доставка (Africa, Europe, LATAM, Pacific Asia, USCA)
26	Місто призначення	Order City	Місто, куди фактично доставляється замовлення
27	Країна призначення	Order Country	Країна, куди фактично доставляється замовлення
28	Код замовника	Order Customer Id	Ідентифікатор клієнта в рамках конкретного замовлення
29	Дата замовлення	Order date (DateOrders)	Точна дата та час оформлення замовлення в системі
30	Ідентифікатор замовлення	Order Id	Унікальний код замовлення
31	Ідентифікатор товару	Order Item Cardprod Id	Код продукту, згенерований RFID-зчитувачем
32	Знижка на товар	Order Item Discount	Грошова сума знижки на конкретну позицію в замовленні
33	Відсоток знижки	Order Item Discount Rate	Відсоток наданої знижки (значення від 0.0 до 1.0)
34	Ідентифікатор позиції	Order Item Id	Унікальний код конкретної позиції (рядка) в замовленні
35	Ціна товару	Order Item Product Price	Базова ціна одиниці продукту без урахування знижки

1.	2.	3.	4.
36	Коефіцієнт прибутку	Order Item Profit Ratio	Відношення прибутку до вартості товару
37	Кількість	Order Item Quantity	Кількість одиниць товару в замовленні
38	Сума продажу	Sales	Загальна вартість продажу (розраховується як ціна * кількість)
39	Разом за позицію	Order Item Total	Підсумкова сума за позицію з урахуванням знижок
40	Прибуток за замовлення	Order Profit Per Order	Прибуток, отриманий з конкретного замовлення
41	Регіон замовлення	Order Region	Географічний регіон доставки (наприклад, Western Europe, South Asia)
42	Штат доставки	Order State	Штат або область регіону доставки
43	Статус замовлення	Order Status	Стан обробки (COMPLETE, PENDING, CANCELED, SUSPECTED_FRAUD тощо)
44	Код продукту	Product Card Id	Внутрішній код товару в системі
45	Ідентифікатор категорії продукту	Product Category Id	Код категорії, до якої належить товар
46	Опис продукту	Product Description	Текстовий опис товару (часто пустий)
47	Зображення	Product Image	URL-посилання на зображення товару
48	Назва продукту	Product Name	Повне найменування товару
49	Ціна продукту	Product Price	Роздрібна ціна одиниці товару
50	Статус наявності	Product Status	Стан складу (0 - товар доступний, 1 - недоступний/архівний)
51	Дата відправлення	Shipping date (DateOrders)	Точна дата та час фактичного відвантаження товару зі складу
52	Тип доставки	Shipping Mode	Спосіб транспортування (Standard Class, First Class, Second Class, Same Day)

Ключовим атрибутом для задачі виявлення аномалій є бінарна змінна `Late_delivery_risk` (0 - доставка вчасно, 1 - затримка), яка виступає цільовою міткою для навчання класифікатора. Для прогнозування попиту цільовою змінною виступає атрибут `Order Item Quantity` у розрізі часового ряду.

Попередня обробка

У процесі розвідувального аналізу даних (EDA) було виявлено низку проблем, таких як некоректне кодування символів, невідповідність географічних координат реальним адресам клієнтів та наявність записів, що не відповідають бізнес-правилам системи [15].

Для вирішення цих проблем було розроблено автоматизований ETL-конвеєр, реалізований мовою Python з використанням бібліотеки Pandas.

Вхідний файл використовує кодування Latin-1, що вимагало явного вказання параметрів зчитування для коректної обробки спеціальних символів у назвах міст та товарів. Назви всіх стовпців було приведено до єдиного стандарту snake_case (наприклад, Order City - order_city) для зручності програмної обробки.

Відповідно до вимог сучасної логістичної системи, було впроваджено модуль фільтрації геополітичних ризиків. Розроблений скрипт clean_dataset.py автоматично вилучає транзакції, які не відповідають вимогам. Це забезпечує відповідність даних актуальним бізнес-реаліям та запобігає некоректному навчанню моделі на недоступних ринках.

У вихідному датасеті виявлено системну похибку: атрибути Latitude та Longitude вказують на координати централізованих складів (переважно у Пуерто-Рико), а не на фактичне місцезнаходження клієнтів. Це унеможлиблювало коректну кластеризацію попиту [16].

Для вирішення цієї задачі було розроблено модуль геокодування fix_geodata.py, який використовує бібліотеку geopy та API сервісу Nominatim (OpenStreetMap).

Алгоритм роботи модуля передбачає виділення унікальних пар «Країна-Місто» з атрибутів замовлення та виконання запитів до API для отримання реальних географічних координат центроїдів цих локацій. Окрім цього, модуль створює локальний кеш координат у файлі country_lookup.json, що дозволяє пришвидшити обробку даних при повторних запусках системи та зменшити навантаження на зовнішній API.

В рамках етапу обробки часових рядів та інженерії ознак атрибути дати та часу (order_date, shipping_date) були конвертовані у формат datetime. На їх основі було згенеровано ряд похідних ознак: order_year та order_month для аналізу сезонності попиту, показник shipping_delay, який розраховується як різниця між фактичним та плановим часом доставки, а також late_delivery_actual - верифікована мітка затримки, необхідна для перехресної перевірки якості даних. Результатом цього етапу є очищений, структурований та збагачений набір даних, який завантажується у базу даних MongoDB та використовується для тренування нейронних мереж у наступних розділах.

2.2. Проектування та побудова моделей машинного навчання

У цьому підрозділі детально розглядається математичний апарат, принципи роботи та архітектурні особливості моделей, обраних для вирішення трьох ключових задач розроблюваної системи. Кожна з цих задач має свої специфічні вимоги до якості прогнозування, масштабованості та стійкості до шуму в даних, тому вибір відповідних алгоритмів та структур моделей є критично важливим.

Першою задачею є класифікація ризиків доставки, яка передбачає аналіз історичних логістичних записів, параметрів маршруту, характеристик товарів та поведінкових патернів постачальників. Для її вирішення використовуються моделі машинного навчання, здатні ефективно працювати з гетерогенними даними та виявляти нелінійні залежності. Зокрема, розглядаються методи дерев рішень, ансамблеві моделі на кшталт Random Forest та Gradient Boosting, а також нейронні мережі з багат шаровою структурою. У підрозділі уточнюються вибрані функції втрат, критерії оптимізації та механізми регуляризації для забезпечення узагальнювальної здатності моделі [17].

Другою ключовою задачею є прогнозування часових рядів попиту, що вимагає моделювання сезонності, трендів та короткотермінових флуктуацій. Для цього аналізуються як класичні статистичні методи (ARIMA, SARIMA, експоненціальне згладжування), так і сучасні підходи глибинного навчання, зокрема рекурентні нейронні мережі (LSTM, GRU), а також трансформерні архітектури, які продемонстрували високу точність при роботі з довгими часовими залежностями. У підрозділі описуються принципи вибору гіперпараметрів, стратегії крос-валідації для часових даних та особливості підготовки тренувальних вибірок [18].

Третьою задачею є геопросторова кластеризація складських потужностей, спрямована на оптимізацію логістичної мережі та визначення найбільш ефективного розподілу ресурсів. Для цього застосовуються алгоритми просторової кластеризації, зокрема k-means, DBSCAN, HDBSCAN та їх модифікації, здатні враховувати нерівномірність щільності об'єктів та

географічні обмеження. У підрозділі пояснюється математичний апарат метрик відстані (евклідова, гейверсинова, мангеттенська), способи масштабування координатних даних та методи оцінювання якості кластеризації.

Модель прогнозування ризиків

Для задачі бінарної класифікації стану доставки (“затримка” / “вчасно”) було обрано алгоритм Random Forest (Випадковий Ліс) - один із найефективніших ансамблевих методів навчання. Основною перевагою Random Forest є його здатність поєднувати велику кількість слабких моделей (дерев рішень), формуючи стійкий та узагальнюючий класифікатор. На відміну від поодинокого дерева, яке часто схильне до перенавчання, ансамбль дерев суттєво зменшує дисперсію моделі завдяки випадковості як у виборі даних, так і у виборі ознак [19].

Нехай навчальна вибірка X складається з N векторів ознак, де кожен вектор x_i містить параметри замовлення (тип доставки, регіон, плановий час, категорія товару тощо):

$$x_i = \{shipping_mode, scheduled_days, category, region, \dots\}$$

Цільова змінна $x_i \in \{0,1\}$, де 1 - факт затримки (Late_delivery_risk).

Кожне дерево будується на основі бутстреп-вибірки - випадкової підмножини даних, відібраної з поверненням. Це створює природну варіативність між деревами, дозволяючи моделі краще узагальнювати інформацію та зберігати стійкість до аномальних чи шумних спостережень.

На кожному вузлі дерево виконує вибір ознаки та порога розбиття, що мінімізує невизначеність (імпурність) дочірніх вузлів. У цій роботі використовують популярний критерій неоднорідності - індекс Джині (Gini Impurity) [20].

Для вузла t індекс Джині обчислюється за формулою:

$$G(t) = 1 - \sum_{k=0}^1 (p_k)^2$$

де p_k - частка об'єктів класу k у вузлі t .

Алгоритм обирає таку ознаку та поріг розбиття, які максимізують приріст інформації (Information Gain), або, еквівалентно, мінімізують зважену неоднорідність після розбиття:

$$\Delta G = G(parent) - \left(\frac{N_{left}}{N} G(left) + \frac{N_{right}}{N} G(right) \right)$$

де:

N - кількість об'єктів у батьківському вузлі,

N_{left}, N_{right} - кількість об'єктів у дочірніх вузлах,

$G(left), G(right)$ - індекс Джині для кожного із дочірніх вузлів.

Остаточний прогноз моделі для нового екземпляра x формується шляхом мажоритарного голосування всіх B дерев ансамблю:

$$\hat{y} = mode\{T_1(x), T_2(x), \dots, T_B(x)\}$$

Також модель дозволяє оцінити ймовірність класу (probabilistic prediction), що використовується в системі для відображення відсотка ризику [21]:

$$P(y = 1|x) = \frac{1}{B} \sum_{b=1}^B I(T_b(x) = 1)$$

де $I(\cdot)$ - індикаторна функція.

Нейронна мережа для прогнозування попиту

Прогнозування обсягу продажів у логістичній системі є задачею регресії часових рядів, оскільки майбутній попит залежить від історичних значень, сезонних факторів, трендів та короткострокових коливань. Для моделювання таких залежностей у роботі використовується рекурентна нейронна мережа типу LSTM (Long Short-Term Memory), яка здатна ефективно зберігати інформацію на тривалих інтервалах та долати проблему "зникання градієнта", властиву класичним RNN [22].

Архітектура комірки LSTM в момент часу t описується системою рівнянь.

Гейт забування. Вирішує, яку частину інформації з попереднього стану комірки C_{t-1} слід відкинути.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Вхідний гейт. Визначає, яку нову інформацію записати в стан комірки.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Оновлення стану

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Вихідний гейт. Формує вихідний сигнал h_t (який також є прихованим станом для наступного кроку).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Де:

x_t - вхідний вектор (значення попиту за попередній місяць).

h_{t-1} - вихід попереднього кроку.

W та b - вагові матриці та вектори зміщення, що навчаються.

σ - сигмоїдальна функція активації (діапазон $[0, 1]$).

$\tanh$ - гіперболічний тангенс (діапазон $[-1, 1]$).

Оскільки навчання окремої LSTM для кожного товару є ресурсомістким, у системі реалізовано підхід "зверху-вниз" [23]. Спочатку прогнозується глобальний тренд продажів Y_{total} , а потім він декомпозиується для конкретної групи товарів і на основі коефіцієнта частки ринку k_i :

$$k_i = \frac{\sum_{t=T-3}^T S_{i,t}}{\sum_{t=T-3}^T S_{total,t}}$$

$$Y_{forecast,i} = Y_{total_forecast,i} \cdot k_i$$

де S - історичні продажі за останні 3 місяці.

Кластеризація логістичних хабів

Для визначення оптимальних географічних локацій складів у системі використовується метод кластеризації без учителя K-Means, який належить до найбільш поширених і ефективних алгоритмів групування просторових даних. Основою методу є припущення, що клієнти, розташовані географічно близько один до одного, доцільно обслуговувати з одного логістичного хабу або складу.

Вхідними даними є множина географічних координат клієнтів $X = \{(\phi_1, \lambda_1), \dots, (\phi_m, \lambda_m)\}$, де ϕ - широта, λ – довгота [24].

Мета алгоритму - знайти такі координати центроїдів (складів) $\mu_1, \dots, \mu_k$, які мінімізують сумарне квадратичне відхилення:

$$J = \sum_{j=1}^k \sum_{i \in c_j} \|x_i - \mu_j\|^2 \rightarrow \min$$

де:

k - кількість кластерів (у нашій системі $k=6$).

x_i - координати i -го клієнта.

μ_j - координати центру j -го складу.

$\|\dots\|^2$ - квадрат евклідової відстані між клієнтом і складом.

Алгоритм працює ітеративно за схемою Ллойда:

Ініціалізація. Випадковий вибір k початкових центроїдів.

Призначення. Кожна точка даних відноситься до найближчого центроїда.

Оновлення. Перерахунок координат центроїдів як середнього арифметичного координат усіх точок кластера:

$$\mu_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$$

Процес повторюється до збіжності (коли координати μ_j перестають змінюватися) [25].

Отримані центроїди μ_j інтерпретуються системою як рекомендовані географічні точки для розміщення логістичних хабів.

2.3. Проектування архітектури програмного забезпечення

Для реалізації додатку було обрано мікросервісну архітектуру, яка є потужним підходом для побудови розподілених систем. Цей підхід дозволяє розділити функціонал на незалежні модулі або сервіси, кожен з яких відповідає за окрему задачу або бізнес-процес. Завдяки такій архітектурі кожен сервіс можна розвивати, тестувати, масштабувати та оновлювати незалежно від інших

компонентів системи, що значно підвищує гнучкість і знижує ризики при змінах або оновленнях. Крім того, мікросервіси дають змогу швидко адаптувати систему до змінюваних умов, таких як збільшення навантаження або зміни у вимогах бізнесу [26].

Архітектура системи складається з трьох основних компонентів: клієнтської частини, серверної частини та сервісу штучного інтелекту, кожен з яких виконує певну роль у забезпеченні функціонування всього додатку. Клієнтська частина відповідає за взаємодію з кінцевими користувачами через інтерфейс, надаючи доступ до різних функцій системи через веб-додаток. Вона має бути максимально зручною, зрозумілою та реагувати на запити в реальному часі, що дозволяє користувачам працювати з даними та прогнозами [27].

Вибір технологій розробки

Для створення ефективного додатку було обрано сучасний технологічний стек, який дозволяє забезпечити високу продуктивність, масштабованість та зручність супроводу програмного забезпечення. Клієнтська складова системи реалізована на базі фреймворку Vue.js 3 із використанням архітектурного підходу Composition API [28]. Такий вибір обумовлений необхідністю забезпечення високої реактивності інтерфейсу, що є критичним фактором для відображення динамічних графіків та моніторингу процесів у реальному часі. Компонентна структура Vue.js дозволила створити модульну систему з перевикористовуваними елементами інтерфейсу, що значно спрощує подальшу підтримку та розширення функціоналу. Для побудови адаптивного дизайну, який гарантує коректне відображення системи на пристроях із різною роздільною здатністю, було використано CSS-фреймворк Tailwind CSS. Завдання візуалізації аналітичних даних вирішено шляхом інтеграції бібліотек Chart.js та Leaflet: перша забезпечує побудову інтерактивних графіків попиту та сегментації, а друга відповідає за роботу з геопросторовими даними, дозволяючи візуалізувати логістичні маршрути на карті [29].

Серверна частина системи, яка виконує роль єдиної точки входу, побудована на платформі Node.js з використанням фреймворку Express.js. Вибір даної платформи аргументовано її здатністю ефективно обробляти велику

кількість одночасних запитів завдяки асинхронній моделі введення-виведення. На цьому рівні реалізовано логіку маршрутизації, валідацію вхідних даних, а також механізми безпеки, зокрема автентифікацію та авторизацію користувачів за стандартом JWT (JSON Web Token), що забезпечує захищений доступ до ресурсів без необхідності зберігання сесій на сервері.

Для реалізації сервісу інтелектуального аналізу даних та прогнозування обрано мову програмування Python у поєднанні з високопродуктивним веб-фреймворком FastAPI. Python є загальновизнаним стандартом у галузі Data Science, що дозволило залучити потужний інструментарій для обробки інформації. Зокрема, бібліотека Pandas використовується для завантаження, очищення та нормалізації масивів історичних даних, Scikit-learn застосовується для тренування базових моделей машинного навчання, а бібліотека TensorFlow забезпечує реалізацію складних архітектур глибокого навчання, таких як рекурентні нейронні мережі. Використання LSTM дозволяє враховувати часові ряди та сезонність даних, що значно підвищує точність прогнозування попиту. FastAPI, забезпечує швидку обробку запитів від основного сервера та автоматичну генерацію документації інтерфейсів. Поєднання зазначених технологій дозволило створити збалансовану систему, яка інтегрує зручний користувацький інтерфейс, надійну серверну логіку та передові методи машинного навчання для оптимізації логістичних процесів [30].

Проектування бази даних

У якості системи керування базами даних для реалізації програмного комплексу було обрано MongoDB. Це документо-орієнтована NoSQL база даних, яка зберігає дані у форматі BSON (бінарний JSON). Вибір саме цієї технології зумовлений специфікою предметної області, а саме необхідністю зберігати великі масиви логістичних даних зі складною ієрархічною структурою без жорсткої фіксації схеми, що є характерним обмеженням традиційних реляційних баз даних. Використання MongoDB дозволяє зберігати інформацію про замовлення разом із вкладеним масивом товарів у межах одного документа, що

значно підвищує швидкість операцій читання та запису, оскільки виключає необхідність виконання ресурсномістких операцій об'єднання таблиць (JOIN).

Архітектура бази даних побудована навколо чотирьох основних колекцій, кожна з яких відповідає за окремий аспект функціонування системи [31].

Колекція Users (Користувачі) розроблена для забезпечення функціонування підсистеми безпеки та розмежування прав доступу. Кожен документ у цій колекції містить унікальний ідентифікатор користувача, його логін для входу в систему та хешований пароль, оброблений криптографічним алгоритмом для захисту від витоку даних. Крім того, структура документа включає поле ролі, яке визначає рівень повноважень користувача в системі (адміністратор, менеджер або звичайний користувач), а також поле прив'язки до ринку. Останнє дозволяє реалізувати принцип мультиорендності, за якого менеджери отримують доступ лише до аналітики та замовлень свого регіону, що є критично важливим для великих логістичних мереж. Детальний опис наведено в таблиці 2.2.

Таблиця 2.2.

Опис колекції “Користувачі (Users)”

Поле	Тип даних	Опис поля
id	ObjectId	Унікальний ідентифікатор користувача.
username	string	Унікальне ім'я користувача (логін).
password_hash	string	Захешований пароль (з міркувань безпеки не зберігається у відкритому вигляді).
role	string	Рівень доступу користувача. Можливі значення: "admin", "manager", "user".
market	string	Регіон або ринок, за який відповідає менеджер

Колекція Products (Товари) виконує роль централізованого довідника номенклатури. У ній зберігається статична інформація про асортимент, включаючи назву товару, посилання на зображення для відображення в інтерфейсі, унікальні ідентифікатори категорії та картки товару. Важливим динамічним атрибутом є статус наявності, який дозволяє системі аналізу ризиків перевіряти доступність товару на складі перед формуванням рекомендацій. Окрім цього, кожен документ містить інформацію про базову ціну товару, що

використовується для фінансових розрахунків у модулях прогнозування. Детальний опис наведено в таблиці 2.3.

Таблиця 2.3.

Опис колекції “Товари (Products)”

Поле	Тип даних	Опис поля
id	ObjectId	Унікальний ідентифікатор запису товару в базі.
name	string	Назва товару.
category	string	Категорія, до якої належить товар (наприклад, "Електроніка", "Одяг").
price	number	Базова ціна товару.
is_avaible	boolean	Прапорець наявності (true - є в наявності, false - немає).
image_url	string	Посилання на зображення товару.
product_card_id	int	Зовнішній або внутрішній ID картки товару (SKU або ID з ERP-системи).

Колекція Locations (Локації) слугує геопросторовим фундаментом системи. Вона містить записи про унікальні пари «Країна – Місто», які зустрічаються в історії замовлень. Ключовими атрибутами цієї колекції є географічні координати - широта та довгота, отримані в результаті процесу геокодування. Ці дані є вхідними параметрами для алгоритму кластеризації K-Means, що дозволяє візуалізувати покриття логістичної мережі на карті та розраховувати оптимальні точки для розташування нових розподільчих хабів. Детальний опис наведено в таблиці 2.4.

Таблиця 2.4.

Опис колекції “Локації (Locations)”

Поле	Тип даних	Опис поля
id	ObjectId	Унікальний ідентифікатор локації.
city	string	Назва міста.
country	string	Назва країни.
latitude	float	Географічна широта.
longitude	float	Географічна довгота.

Центральним елементом бази даних є колекція Orders (Замовлення), яка агрегує в собі всю інформацію про логістичні транзакції. Структура документа замовлення є найбільш складною і включає ідентифікатори замовлення та

клієнта, часові мітки оформлення та фактичної відправки, а також розрахункові показники ризику затримки. Детальний опис наведено в таблиці 2.5.

Таблиця 2.5.

Опис колекції “Замовлення (Orders)”

Поле	Тип даних	Опис поля
_id	ObjectId	Технічний ідентифікатор документа замовлення.
order_id	string	Унікальний, читабельний код замовлення (для клієнтів/менеджерів).
order_date	date	Дата та час створення замовлення.
shipping_date	date	Дата та час відвантаження.
status	string	Поточний стан замовлення.
total_sales	float	Загальна сума продажу по чеку.
profit	float	Чистий прибуток з цього замовлення.
shipping mode	string	Спосіб доставки.
late_delivery_risk	float	Оцінка ризику затримки доставки.
order region	string	Регіон замовлення.
order city	string	Місто доставки.
order country	string	Країна доставки.
items	array	Масив об'єктів. Опис наведено в таблиці 2.6.

Таблиця 2.6.

Опис об'єкта поля items в колекції “Замовлення (Orders)”

Поле	Тип даних	Опис поля
product_id	ObjectId	Посилання на документ у колекції PRODUCTS.
product_name	string	Назва товару (дублюється для історії, якщо назва зміниться в каталозі).
price_per_unit	number	Ціна за одиницю на момент покупки.
quantity	int	Кількість придбаних одиниць товару.
discount_rate	float	Знижка на товар.
total_price	float	Підсумок по позиції (price_per_unit * quantity - знижка).
profit_ratio	float	Коефіцієнт прибутковості для конкретної позиції.

Фінансовий блок полів містить дані про загальну суму продажу, прибуток та статус оплати. Головною особливістю цієї колекції є використання вкладеного масиву об'єктів для зберігання переліку придбаних товарів. Кожен об'єкт у масиві містить деталі про конкретну позицію: ціну за одиницю, кількість, застосовану знижку та розрахований прибуток. Такий підхід дозволяє

отримувати повну картину замовлення за один запит до бази даних. Для візуалізації створено ER діаграму, вона зображена на рисунку 2.1.



Рис. 2.1. ER-діаграма бази даних

Система контролю доступу

Для забезпечення конфіденційності та цілісності даних у розробленій системі реалізовано механізм керування доступом на основі ролей (англ. RBAC - Role-Based Access Control). Фундаментом підсистеми автентифікації виступає відкритий стандарт JWT (JSON Web Token), який забезпечує безпечну передачу інформації між клієнтом та сервером у форматі JSON-об'єктів. Процес автентифікації побудований таким чином, що при успішному введенні облікових даних сервер генерує криптографічно підписаний токен [32]. Клієнтська частина програмного забезпечення зберігає цей токен локально та автоматично додає його до заголовка Authorization кожного наступного HTTP-запиту, що дозволяє серверу ідентифікувати користувача без необхідності повторної передачі пароля.

Архітектура безпеки системи передбачає ієрархічну рольову модель, яка чітко розмежовує функціональні можливості користувачів. Базовий рівень

доступу надається ролі User, яка дозволяє переглядати каталог продукції та історію замовлень виключно в межах закріпленого за користувачем регіону, реалізуючи таким чином архітектурний принцип мультиорендності (Multi-tenancy). Розширені повноваження має роль Manager, яка відкриває доступ до аналітичних модулів системи, включаючи інструменти прогнозування попиту, сегментації клієнтської бази та інтерактивну карту складів, а також надає право змінювати статуси виконання замовлень. Найвищий рівень доступу закріплено за роллю Admin, яка володіє повним контролем над системою, включаючи адміністративні функції керування обліковими записами, зміни ролей та прив'язки до регіонів, а також можливість ініціювати процес перенавчання нейронних мереж шляхом завантаження нових наборів даних [33].

На рівні програмного інтерфейсу захист реалізовано за допомогою спеціалізованих функцій проміжної обробки. Функція `authenticate` відповідає за валідацію підпису та терміну дії JWT-токена, тоді як функція `authorize` перевіряє наявність у користувача відповідних прав для доступу до конкретного ресурсу чи виконання операції, гарантуючи захист чутливої інформації від несанкціонованого доступу.

Для графічного відображення логіки процесу входу в систему та розгалуження сценаріїв роботи залежно від прав доступу доцільно використати діаграму активностей. На такій діаграмі початковим вузлом є запит на авторизацію. Далі відбувається процес валідації даних: у разі помилки потік керування повертається до форми входу, а у разі успіху переходить до вузла перевірки ролі. Від цього вузла розходяться три альтернативні потоки, що ведуть до різних інтерфейсів: панелі адміністратора, аналітичного дашборда менеджера або каталогу замовлень користувача [34]. Це дозволяє наочно продемонструвати логіку роботи підсистеми безпеки. Діаграма зображена на рисунку 2.2.

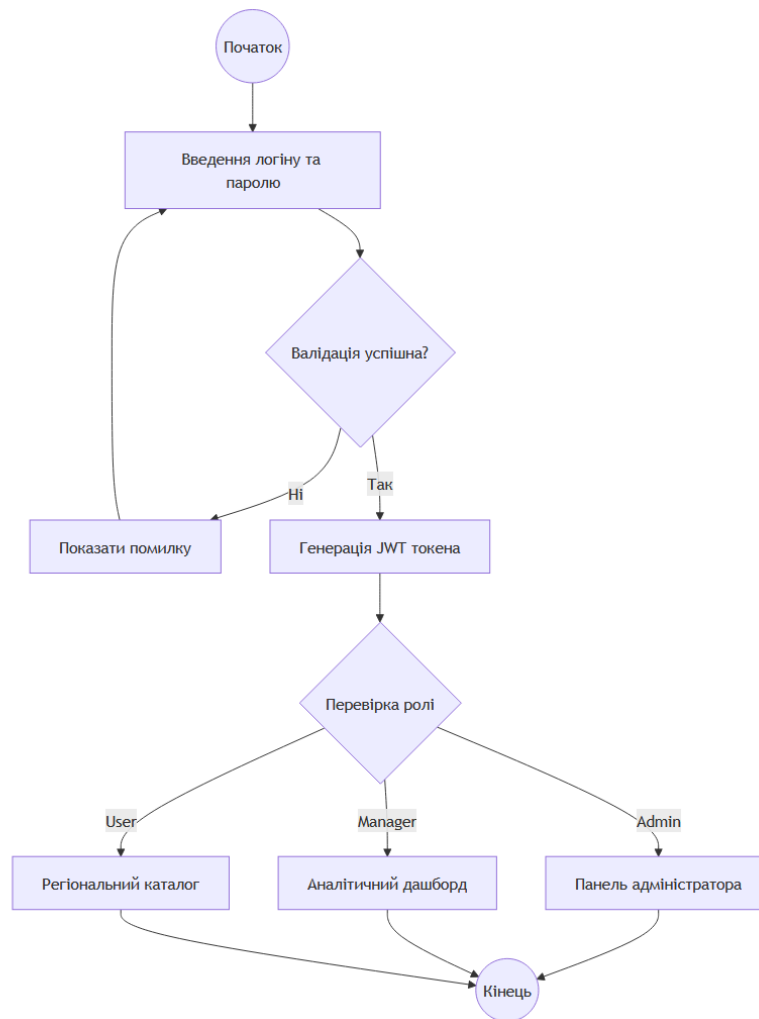


Рис.2.2. Діаграма активностей

Висновки до другого розділу

У другому розділі було проведено комплексне проєктування та програмну реалізацію інтелектуальної інформаційної системи, призначеної для аналізу та оптимізації логістичних процесів підприємства. На етапі підготовки даних було створено автоматизований конвеєр обробки даних, що забезпечує зчитування, очищення та нормалізацію вхідного масиву DataCo Supply Chain. Особливу увагу приділено розробці модулів, які відповідають за фільтрацію ризиків, та збагачення географічних даних. Це дозволило усунути наявні системні неточності у вихідному датасеті, що суттєво підвищило точність подальшого аналітичного опрацювання.

Для вирішення завдань прогнозування та оптимізації було створено та навчено кілька моделей машинного навчання. Алгоритм випадкового лісу використано для оцінювання ризиків доставки, зокрема для прогнозування

ймовірності затримок на основі аналізу взаємозв'язків між категорією товару, регіоном та способом транспортування. Для прогнозування споживчого попиту було розроблено рекурентну нейронну мережу зі структурою LSTM, яка враховує часові коливання, сезонні ефекти та глобальні тренди продажів. Додатково застосовано метод масштабування верхнього рівня, що дозволяє адаптувати загальний тренд до окремих груп товарів. Для оптимізації географічного розміщення майбутніх логістичних центрів було використано алгоритм кластеризації, який на основі координат клієнтів визначає найбільш доцільні локації для потенційних хабів.

Програмна архітектура побудована відповідно до мікросервісного підходу, що забезпечує гнучкість, масштабованість та високу стійкість до відмов. Розподіл системи на незалежні компоненти, серед яких сервіс штучного інтелекту на Python з FastAPI для виконання обчислень, серверна частина на Node.js для обробки бізнес-логіки та клієнтська частина на Vue.js для відображення даних, дозволяє опрацьовувати значні обсяги інформації та водночас забезпечувати користувачу зручний і зрозумілий інтерфейс.

Система контролю доступу, побудована на основі ролей, разом із принципами розмежування доступу для різних організаційних підрозділів, гарантує безпечну роботу з даними та належну ізоляцію інформаційних потоків для адміністраторів, менеджерів та операторів.

РОЗДІЛ 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ

3.1. Графічний інтерфейс

Для початку роботи з системою користувач повинен пройти процес реєстрації, якщо у нього ще немає облікового запису. Під час реєстрації необхідно вказати основні дані, такі як логін, та створити надійний пароль. Після цього, система проведе валідацію даних, та створить новий обліковий запис. Сторінка реєстрації зображена на рисунку 3.1.

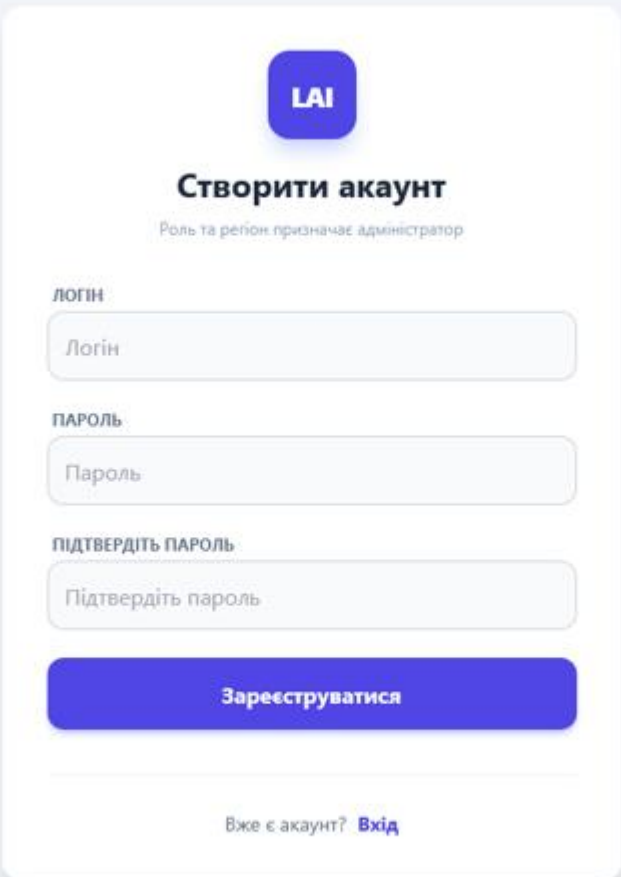


Рис. 3.1. Сторінка реєстрації

Для користувачів, які вже мають створений обліковий запис, передбачено механізм автентифікації, що дозволяє отримати доступ до системи без необхідності повторної реєстрації. Цей процес передбачає заповнення

спеціальної форми, де необхідно вказати коректні облікові дані, зокрема електронну пошту (або логін) та особистий пароль [35]. Після відправки запиту серверна частина перевіряє відповідність введеної інформації збереженим даним і, у разі успішної валідації, перенаправляє користувача до його робочого простору. Графічний інтерфейс та компонування елементів сторінки авторизації продемонстровано на рисунку 3.2.

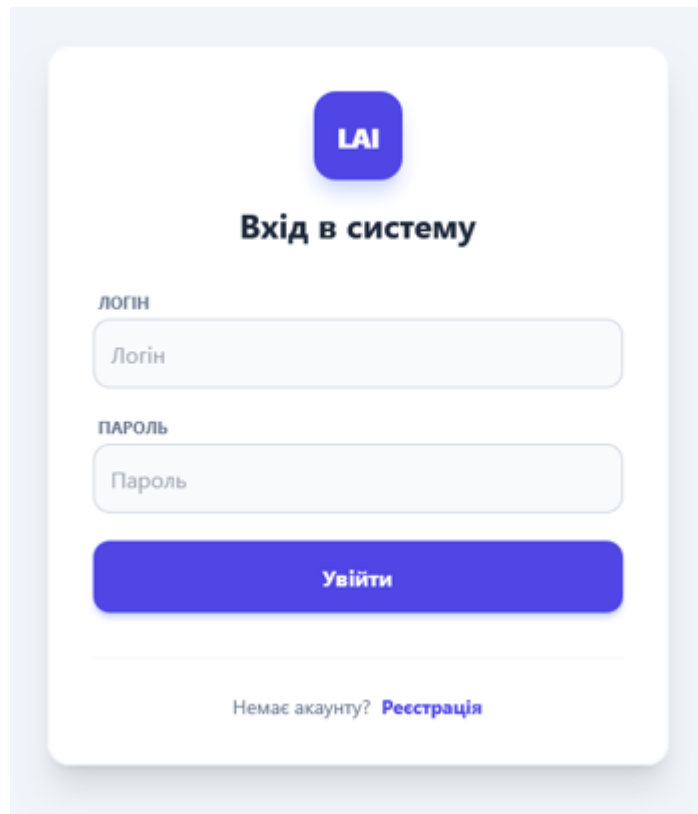


Рис. 3.2. Сторінка авторизації

Головна сторінка системи має зручний та інтуїтивно зрозумілий інтерфейс, основним елементом якого є адаптивна верхня панель навігації. Вона змінюється в залежності від ролі авторизованого користувача - чи це звичайний користувач (User), менеджер (Manager) чи адміністратор (Admin).

Неподалік розташовані посилання на основні модулі, доступні користувачу, такі як Каталог, Замовлення, Ризики та Аналітика. Ці розділи дозволяють швидко переходити до необхідних функцій та здійснювати ефективне управління.

Для зручності користувачів є перемикач мови інтерфейсу, що дає можливість обирати між українською та англійською мовою. Ця

функціональність реалізована за допомогою бібліотеки `vue-i18n` і дозволяє легко адаптувати платформу до різних мовних потреб.

Також на панелі знаходиться перемикач теми оформлення, що дає змогу користувачам вибирати між світлою та темною темою. Вигляд головної сторінки зображено на рисунках 3.3-3.4.

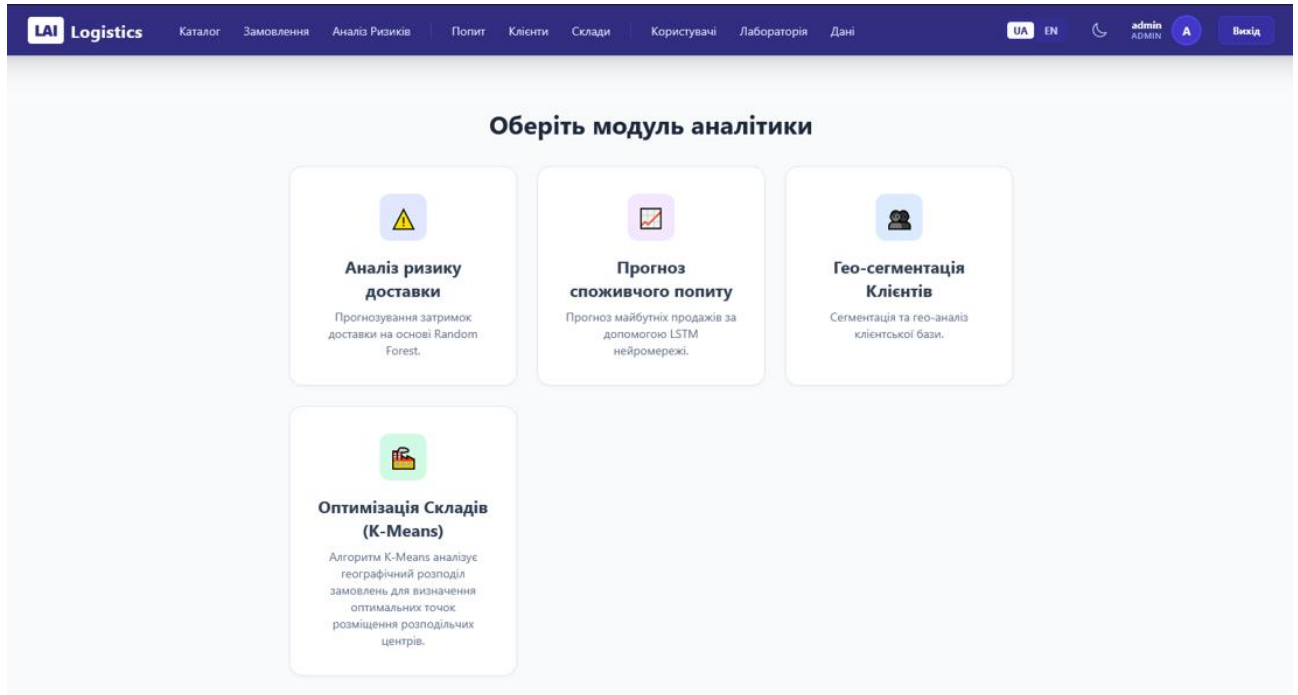


Рис. 3.3. Головна сторінка додатку

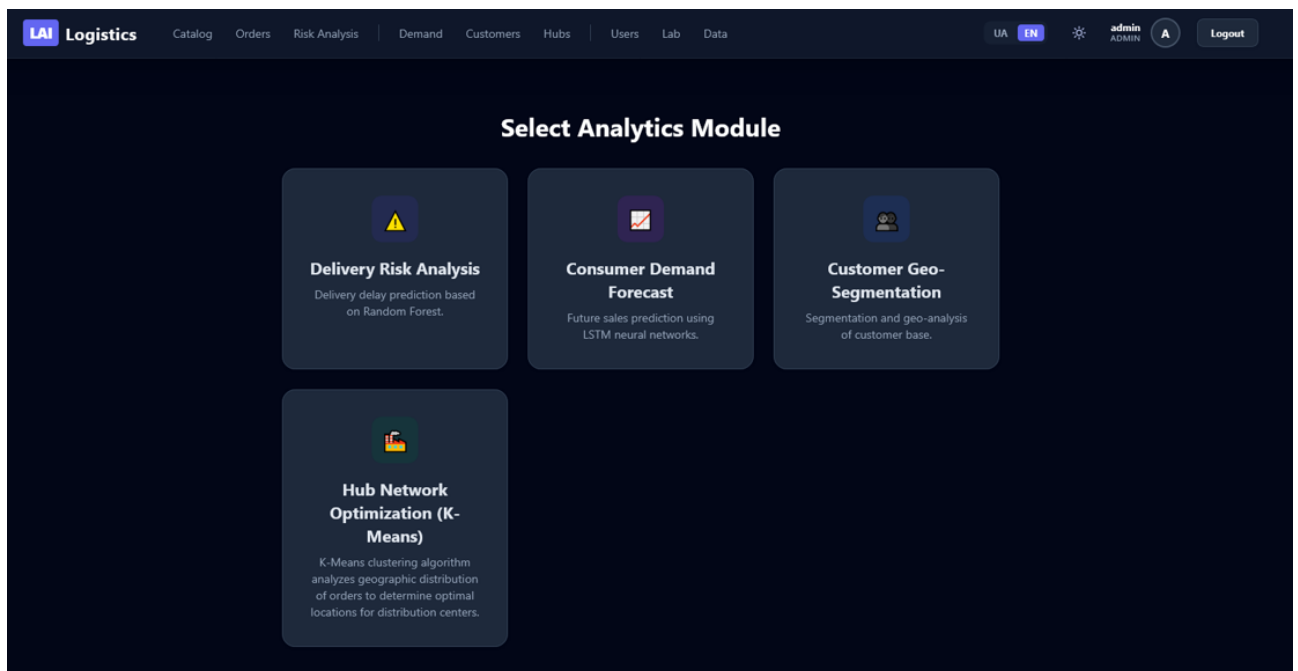


Рис. 3.4. Головна сторінка додатку з англійською локалізацією та темною темою

Далі ми детально розглянемо сторінки, які відповідають за основні функціональні блоки системи, забезпечуючи зручне управління та ефективну взаємодію користувачів [36].

Першою буде сторінка Каталог товарів, де користувачі зможуть переглядати асортимент товарів, здійснювати пошук за назвою або категорією, також можна перейти до аналізу ризиків доставки для обраного товару. Вигляд сторінки каталогу товарів зображено на рисунку 3.5.

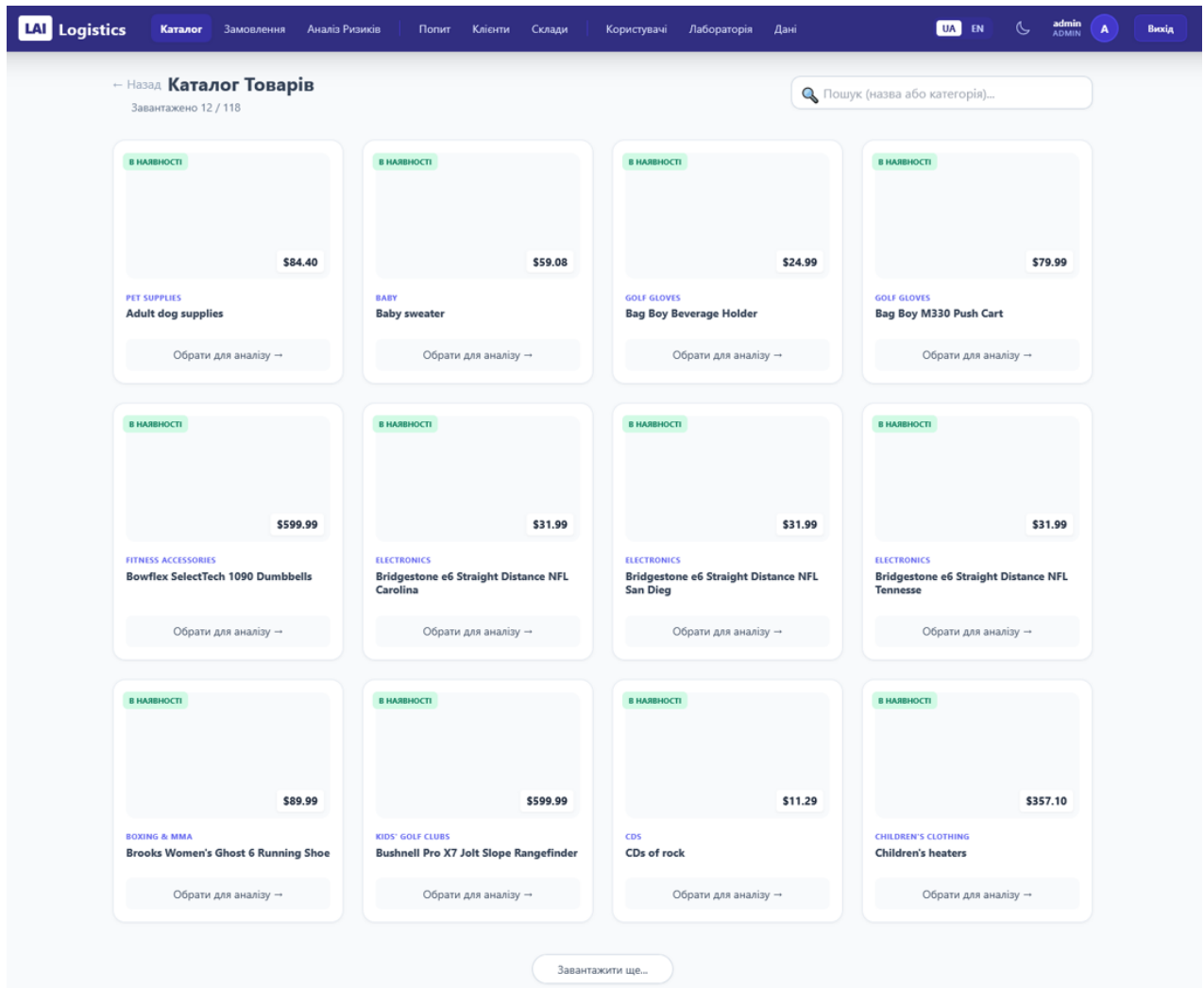


Рис. 3.5. Сторінка каталогу товарів

Сторінка «Замовлення» слугує основним інструментом операційного контролю, забезпечуючи користувачам зручний доступ до повної історії транзакцій та детальної інформації щодо кожного відвантаження. Інтерфейс системи дозволяє глибоко аналізувати специфіку заявок, відображаючи склад кошика, дані клієнта та актуальні часові мітки проходження логістичного ланцюга [37]. Для менеджерів та адміністраторів реалізовано розширений

функціонал управління, який надає можливість оперативно змінювати статус обробки замовлення відповідно до реальної ситуації. Приклади візуальної реалізації загального списку замовлень та форми редагування їх параметрів продемонстровано на рисунках 3.6-3.7.

ID	ДАТА	РЕГІОН	СТАТУС	ПУНКТ ПРИЗНАЧЕННЯ	СУМА	ДІЯ
#77204	01 лютого 2018 р.	Pacific Asia	CLOSED	Gold Coast Australia	\$34.98	Деталі
#77203	01 лютого 2018 р.	Pacific Asia	PENDING_PAYMENT	Bekasi Indonesia	\$10.91	Деталі
#77202	01 лютого 2018 р.	Pacific Asia	COMPLETE	Bekasi Indonesia	\$314.64	Деталі
#77201	01 лютого 2018 р.	Pacific Asia	COMPLETE	Bekasi Indonesia	\$172.66	Деталі
#77200	01 лютого 2018 р.	Pacific Asia	PROCESSING	Bekasi Indonesia	\$161.87	Деталі
#77199	31 січня 2018 р.	Pacific Asia	PENDING_PAYMENT	Orange Australia	\$215.82	Деталі
#77198	31 січня 2018 р.	Pacific Asia	CLOSED	Delhi India	\$213.66	Деталі
#77197	31 січня 2018 р.	Pacific Asia	COMPLETE	Yakarta Indonesia	\$211.50	Деталі

Рис. 3.6. Сторінка історії замовлень

ID	ДАТА	РЕГІОН	СТАТУС	ПУНКТ ПРИЗНАЧЕННЯ	СУМА	ДІЯ
#77204	01 лютого 2018 р.	Pacific Asia	CLOSED	Gold Coast Australia	\$34.98	Деталі
#77203	01 лютого 2018 р.	Pacific Asia	PENDING_PAYMENT	Bekasi Indonesia	\$10.91	Деталі
#77202	01 лютого 2018 р.	Pacific Asia	COMPLETE	Bekasi Indonesia	\$314.64	Деталі
#77201	01 лютого 2018 р.	Pacific Asia	COMPLETE	Bekasi Indonesia	\$172.66	Деталі
#77200	01 лютого 2018 р.	Pacific Asia	PROCESSING	Bekasi Indonesia	\$161.87	Деталі
#77199	31 січня 2018 р.	Pacific Asia	PENDING_PAYMENT	Orange Australia	\$215.82	Деталі
#77198	31 січня 2018 р.	Pacific Asia	CLOSED	Delhi India	\$213.66	Деталі
#77197	31 січня 2018 р.	Pacific Asia	COMPLETE	Yakarta Indonesia	\$211.50	Деталі
#77196	31 січня 2018 р.	Pacific Asia	COMPLETE	Demopart	\$208.35	Деталі
#77195	31 січня 2018 р.	Pacific Asia	PROCESSING	Demopart	\$207.18	Деталі

ЗАМОВЛЕННЯ #77203

Standard Class

01 лютого 2018 р. • PAYMENT

СПИСОК ТОВАРІВ (1)

	Toys	\$1153998996 / шт.	\$10.91	x1
--	-------------	--------------------	----------------	----

PENDING_PAYMENT **Зберегти** **Total: \$10.91**

Рис. 3.7. Сторінка історії замовлень та вікно деталей про замовлення в темній темі

Сторінка «Користувачі» є ключовим елементом адміністративної панелі, що надає повний доступ до реєстру облікових записів та інструментів моніторингу активності персоналу в системі. Завдяки цьому інтерфейсу адміністратори можуть оперативно керувати правами доступу, змінюючи системні ролі співробітників для забезпечення належного рівня інформаційної безпеки [38]. Критично важливою функцією є можливість налаштування доступних географічних зон для кожного користувача, що визначає обсяг даних для подальшого аналізу та прогнозування попиту. Детальний вигляд таблиці користувачів та елементів керування їхніми профілями продемонстровано на рисунку 3.8.

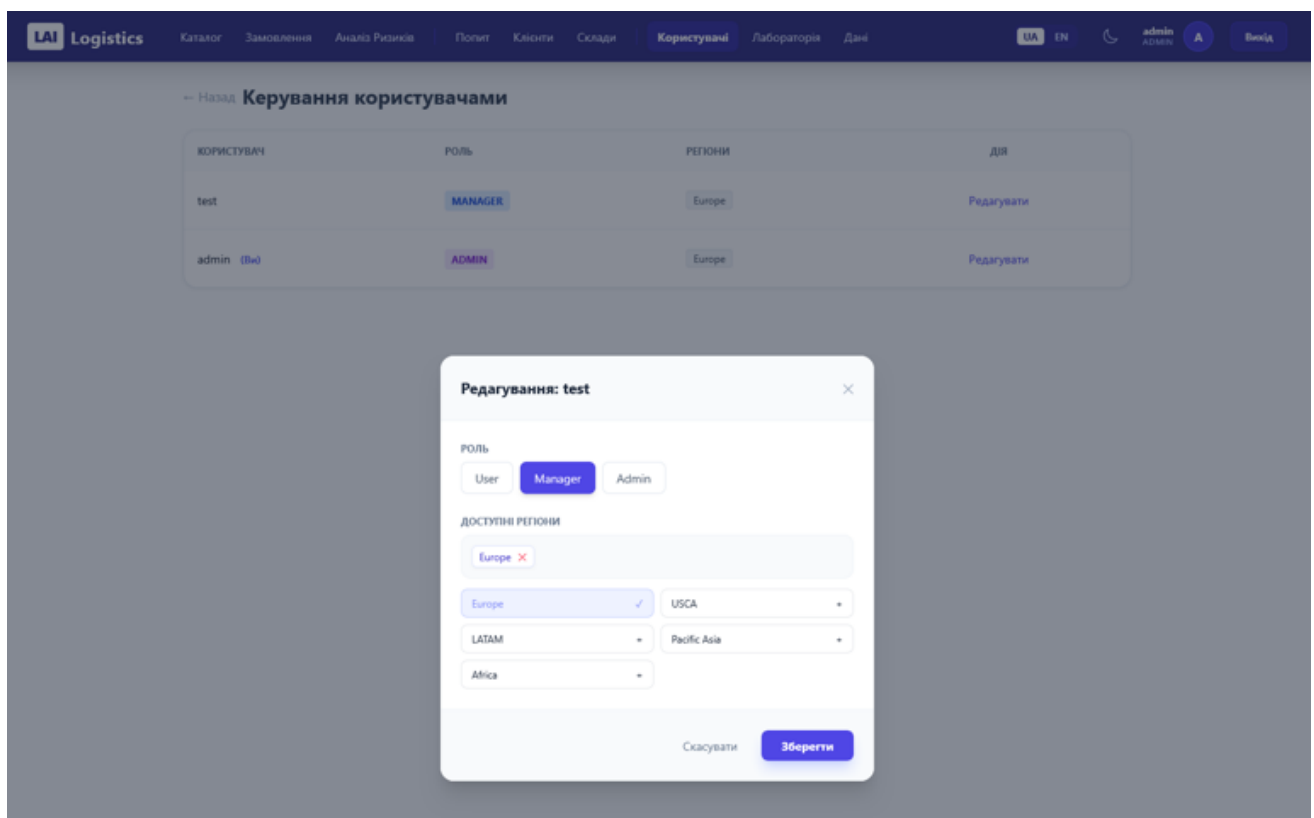


Рис. 3.8. Сторінка налаштування доступу користувачів

Сторінка «Профіль» виступає централізованим вузлом для перегляду персональних даних та налаштувань облікового запису користувача. В інтерфейсі чітко відображаються призначені системні ролі, які визначають рівень повноважень та доступ до специфічних функцій платформи. Окрему увагу приділено секції доступних регіонів, що дозволяє менеджеру швидко ідентифікувати закріплені за ним географічні зони для подальшого планування

логістичних операцій. Графічну реалізацію та компоновання елементів цієї сторінки наведено на рисунку 3.9.

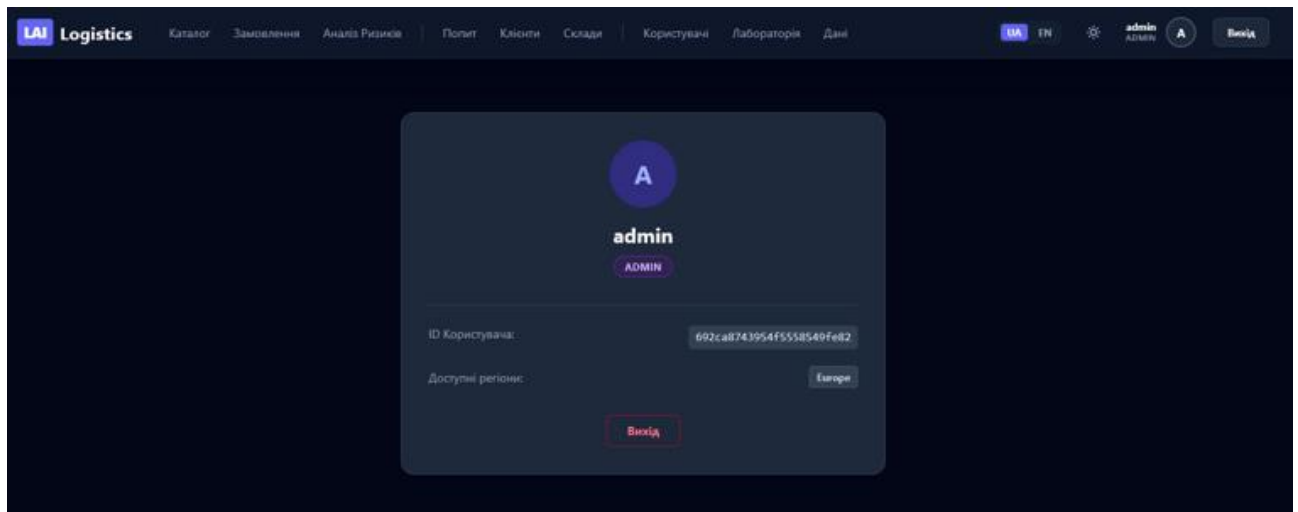


Рис. 3.9. Вигляд сторінки “Профіль”

3.2. Налаштування історичних даних для додатку планування логістики

При запуску сервісу, у рамках процесу ETL, система завантажує набори даних у форматі CSV, що є основою для подальшого аналізу та роботи з інформацією. Після того, як файл завантажено, система автоматично виконує кілька важливих етапів обробки даних.

Першим кроком, здійснюється валідація структури файлу, що дозволяє переконатися, що дані відповідають необхідному формату і не містять помилок у структурі. Наступним кроком є попереднє очищення даних, яке включає видалення дублікатів і фільтрацію, що гарантує високу якість та точність завантажених даних. Після очищення даних система оновлює базу даних MongoDB, зберігаючи нову інформацію для подальшого використання.

Крім того, для підвищення точності аналітичних прогнозів система ініціює процес перетренування нейронної мережі LSTM та моделі Random Forest на нових даних. Це дозволяє забезпечити актуальність моделей і підтримувати їх на високому рівні ефективності.

Щоб протестувати нові масиви даних без впливу на основну базу даних, система надає можливість використовувати модуль “Лабораторія”. В цьому режимі користувач може завантажити тестовий файл, і система проводить аналіз на льоту, не змінюючи основну базу даних. Система розраховує загальні

показники ефективності, такі як продажі та прибуток, будує графіки тренду продажів та здійснює експрес-сегментацію клієнтів. Це дозволяє аналітикам тестувати нові дані і перевіряти гіпотези перед тим, як застосувати їх до реальних операцій. Вигляд сторінки “Лабораторія” зображено на рисунку 3.10.

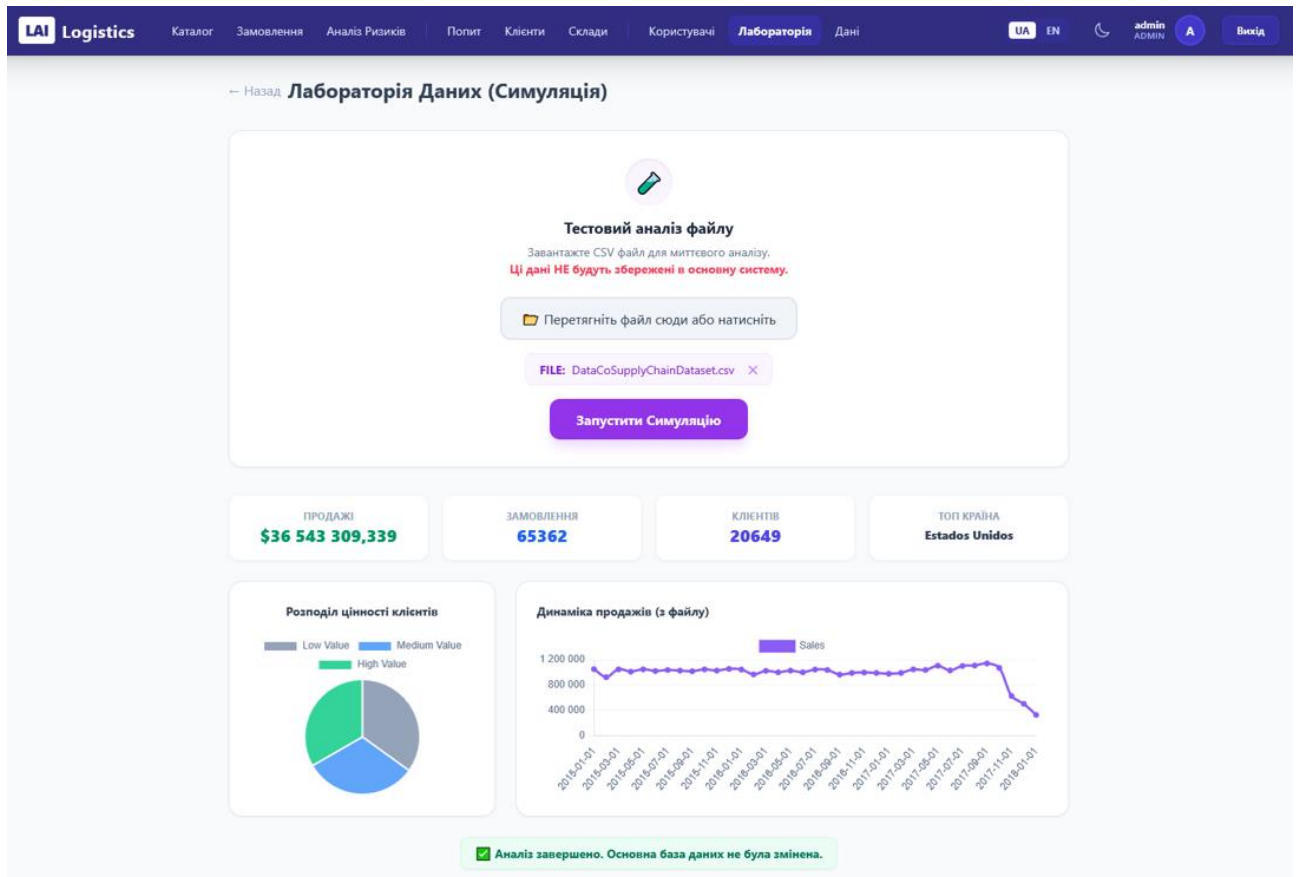


Рис. 3.10. Вигляд сторінки “Лабораторія” з побудованими діаграмами

Додатково до основних функцій, адміністратори системи мають можливість завантажувати нові датасети, що використовуються для подальшої роботи користувачів із даними. Це здійснюється через спеціально розроблену сторінку "Дані".

На цій сторінці адміністратори можуть завантажувати нові набори даних у форматі CSV, що дозволяє зручно додавати необхідну інформацію для обробки та аналізу. Система підтримує автоматичне виконання всіх етапів ETL-процесу, включаючи перевірку структури файлів, очищення даних від зайвих або дубльованих записів, а також фільтрацію згідно з правил. Після завантаження даних система автоматично оновлює базу даних MongoDB, що забезпечує своєчасне оновлення інформації, доступної для користувачів. Вигляд сторінки зображено на рисунку 3.11.

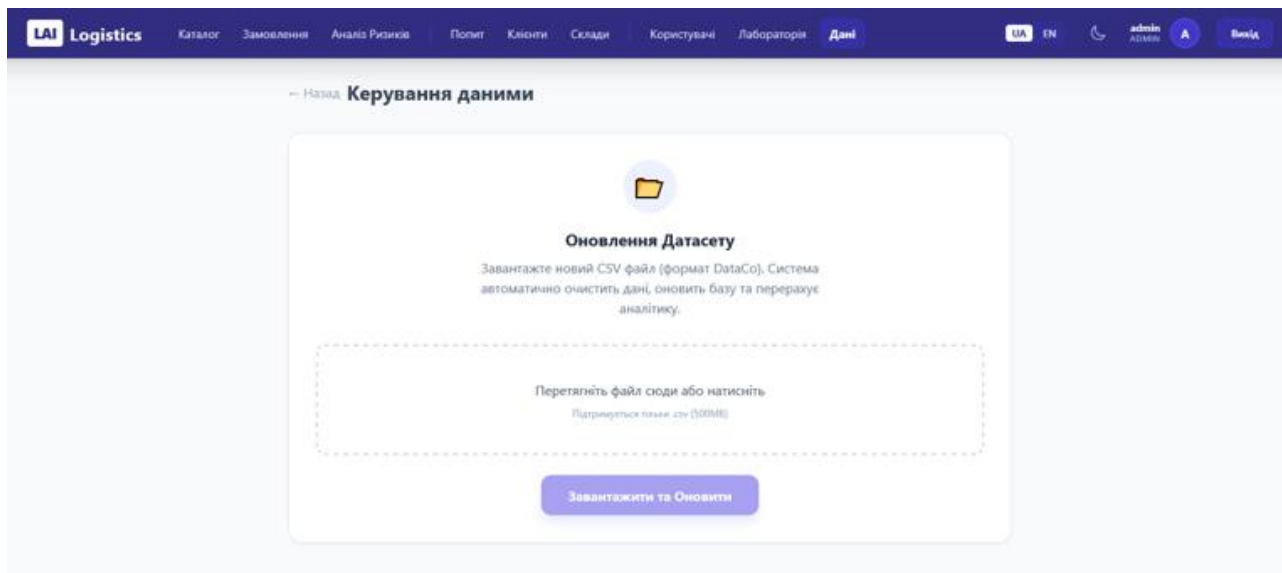


Рис. 3.11. Вигляд сторінки “Дані”

3.3. Відображення результатів аналізу та прогнозів

Відображення результатів роботи алгоритмів у додатку реалізовано через інтерактивні дашборди, які створені з використанням популярних бібліотек для візуалізації даних, таких як Chart.js для побудови графіків та Leaflet для інтерактивної роботи з картографічними даними. В рамках модуля аналізу ризиків доставки інтерфейс надає користувачам - зокрема менеджерам - інструменти для налаштування параметрів замовлення, таких як тип товару, місце доставки, вибір способу доставки та кількість одиниць товару.

Після того як користувач ініціює розрахунок, система звертається до моделі машинного навчання на основі алгоритму Random Forest, яка проводить прогнозування ймовірності затримки доставки у відсотках. На екрані автоматично відображається результат у вигляді графічного індикатора, що відображає рівень ризику затримки, з відповідним кольоровим маркером для візуалізації: зелений - низький рівень ризику, жовтий - середній, червоний - високий рівень ризику.

Це є перший прогноз, який система генерує для користувача. У випадку, якщо ймовірність затримки виявиться високою, система автоматично пропонує альтернативний спосіб доставки, який має менший ризик затримки, та додатково відображає картку товару з актуальними даними про його наявність на складі,

що дозволяє менеджеру оперативно здійснити корекцію або оптимізацію плану доставки.

Вигляд сторінки з прогнозом затримки доставки зображено на рисунку 3.12. Тут користувач зможе побачити, як працює система прогнозування та отримати конкретні рекомендації щодо подальших кроків, що забезпечує максимальну ефективність управління ланцюгом постачання та мінімізацію можливих ризиків [39].

← Назад **Аналіз ризику доставки**

ТОВАР
Garmin Forerunner 910XT GPS Watch

ПУНКТ ПРИЗНАЧЕННЯ
Paris

ТИП ДОСТАВКИ
Standard Class

КІЛЬКІСТЬ
100

ПЛАН (ДНІВ)
1

Розрахувати ризики

ЙМОВІРНІСТЬ ЗАТРИМКИ
74%
High Risk

РЕКОМЕНДАЦІЯ AI
Switch to 'Same Day' to reduce risk to 46.1%.

ДЕТАЛІ

Garmin Forerunner 910XT GPS Watch
KIDS' GOLF CLUBS

ЦІНА
\$399.99

НАЯВНІСТЬ
В наявності

МОДЕЛЬ ОЦІНКИ РИЗИКІВ (RANDOM FOREST)
Використовує ансамбль дерев рішень для прогнозування ймовірності затримки.

КЛЮЧОВІ ФАКТОРИ ВПЛИВУ:

- Тип доставки: Стандартна доставка має вищий ризик, ніж експрес.
- Регіон: Деякі регіони мають складнішу логістику.
- План (днів): Стислі терміни збільшують ризик зриву доставки.
- Кількість: Великі партії можуть довше оброблятися на митниці.

Рис. 3.12. Вигляд сторінки аналізу ризиків доставки

Наступним кроком у візуалізації даних після модуля прогнозування затримки доставки є функціонал прогнозування попиту, який забезпечує розширені аналітичні можливості для стратегічного планування. Цей модуль візуалізує результати у вигляді комбінованого графіка, що поєднує два ключові компоненти. Перший - це історичні показники продажів, відображені суцільною лінією, що дає змогу оцінити реальну динаміку попиту в минулому. Другий - це прогноз, згенерований нейромережею LSTM, позначений пунктирною лінією, який демонструє очікувану тенденцію розвитку продажів у майбутньому.

Інструментарій модуля дозволяє користувачеві налаштовувати горизонт прогнозування на 3, 6, 12 або 24 місяці, залежно від потреб бізнес-аналітики. Додатково реалізовано підхід Top-Down, що дає можливість спершу оцінити загальний ринок, а потім деталізувати аналіз до конкретних груп товарів. Для кожної обраної групи система автоматично відображає її розраховану частку ринку, що підсилює аналітичну глибину та забезпечує коректність порівняння між сегментами.

Таким чином, цей модуль є логічним продовженням попереднього етапу та дозволяє не лише розуміти ризики в логістичних процесах, а й здійснювати прогнозування попиту, необхідне для оптимального планування запасів, виробництва та стратегічних рішень компанії. Вигляд сторінки з прогнозуванням попиту зображено на рисунку 3.13.

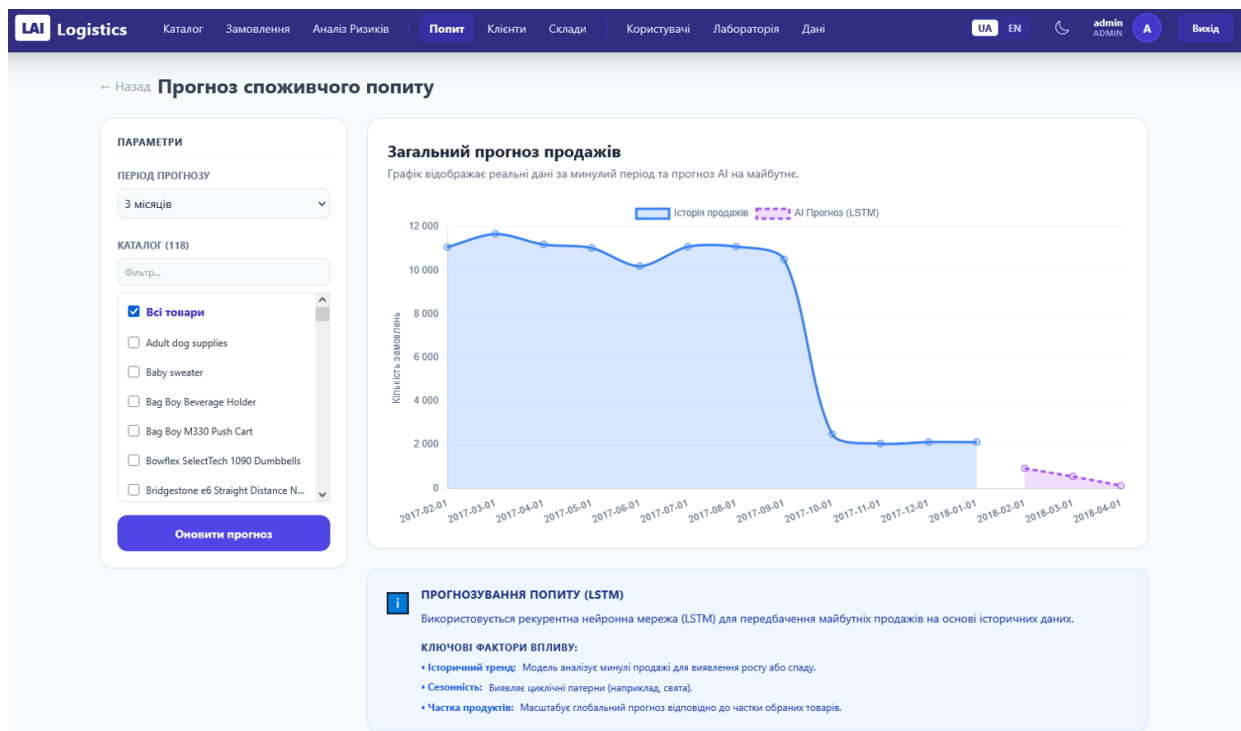


Рис. 3.13. Сторінка прогнозу споживчого попиту

Третій ключовий модуль є інструментом, який поєднує маркетинговий та логістичний аналіз, забезпечуючи глибоке розуміння не лише поведінки клієнтів, а й ефективності територіального охоплення компанії. Всі ці дані інтегровані в інтерактивні візуалізації з картою, що дозволяє користувачам отримати комплексне уявлення про різні аспекти бізнес-процесів.

На першій сторінці відображається RFM-аналіз, що дає змогу сегментувати клієнтів за трьома критеріями: Recency (нещодавність покупки), Frequency (частота покупок) та Monetary (сума витрат). Система автоматично класифікує клієнтів на три основні сегменти: Champions (найбільш цінні клієнти), Loyal (лояльні клієнти) та Lost (втрачені клієнти), кожен з яких відображається на різних кольорах. Це дозволяє легко ідентифікувати найбільш вигідні сегменти для маркетингових кампаній, а також фокусувати зусилля на відновленні стосунків з втраченими клієнтами [40].

Геопросторовий розподіл доповнено картою попиту, де кольорові маркери або зони сигналізують про прибутковість чи збитковість різних регіонів присутності компанії. Таке візуальне представлення дає можливість швидко оцінити ефективність існуючих регіональних стратегій та виявити зони з низьким попитом, які потребують додаткової уваги або зміни маркетингової стратегії. Вигляд сторінки зображено на рисунку 3.14.

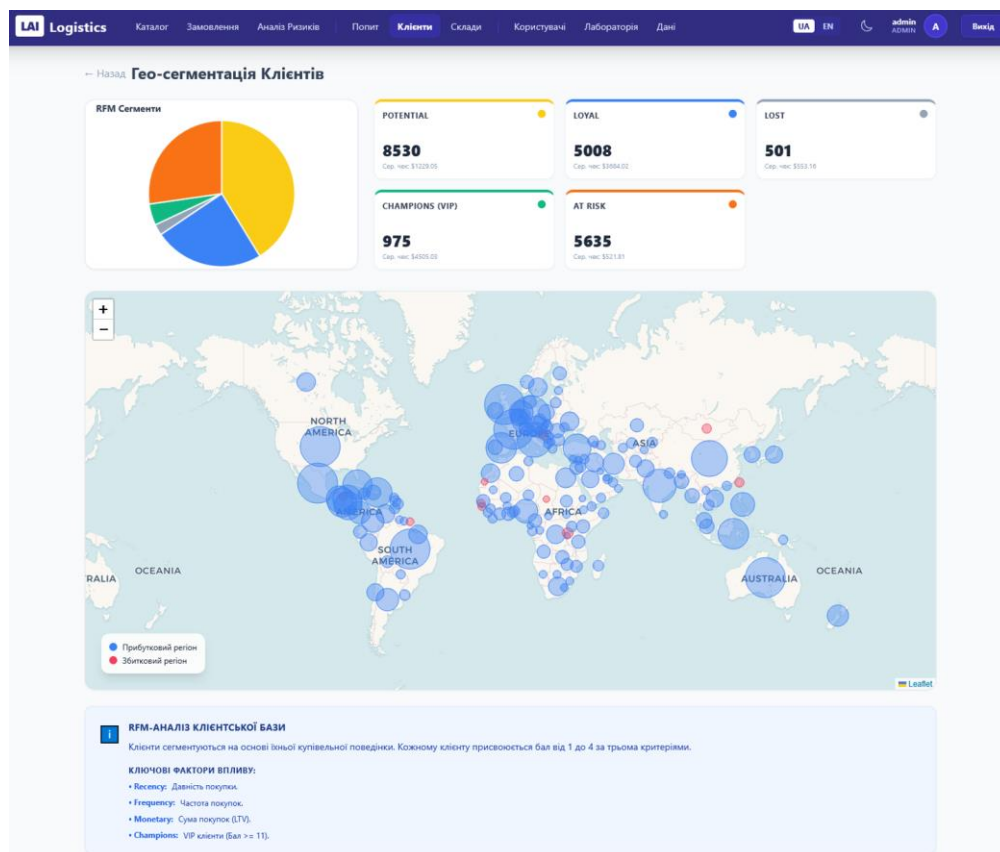


Рис. 3.14. Сторінка сегментації клієнтів

Ще одним важливим елементом цього модуля є кластеризація K-Means, яка автоматично аналізує географічні дані і на основі результатів класифікації

визначає оптимальні локації для розміщення складів або хабів. Алгоритм обчислює зони покриття для кожного складу, що дозволяє візуалізувати на карті, які саме території охоплюватимуться новими складськими об'єктами. Це важливо для планування логістики, оскільки дозволяє мінімізувати час доставки та знизити витрати на транспортування товарів. Користувачі можуть взаємодіяти з маркерами, щоб отримати детальну інформацію про кожну рекомендовану локацію, включаючи обґрунтування, що включає аналіз попиту, витрат на доставку, а також прогнозовану прибутковість. Вигляд сторінки зображено на рисунку 3.15.

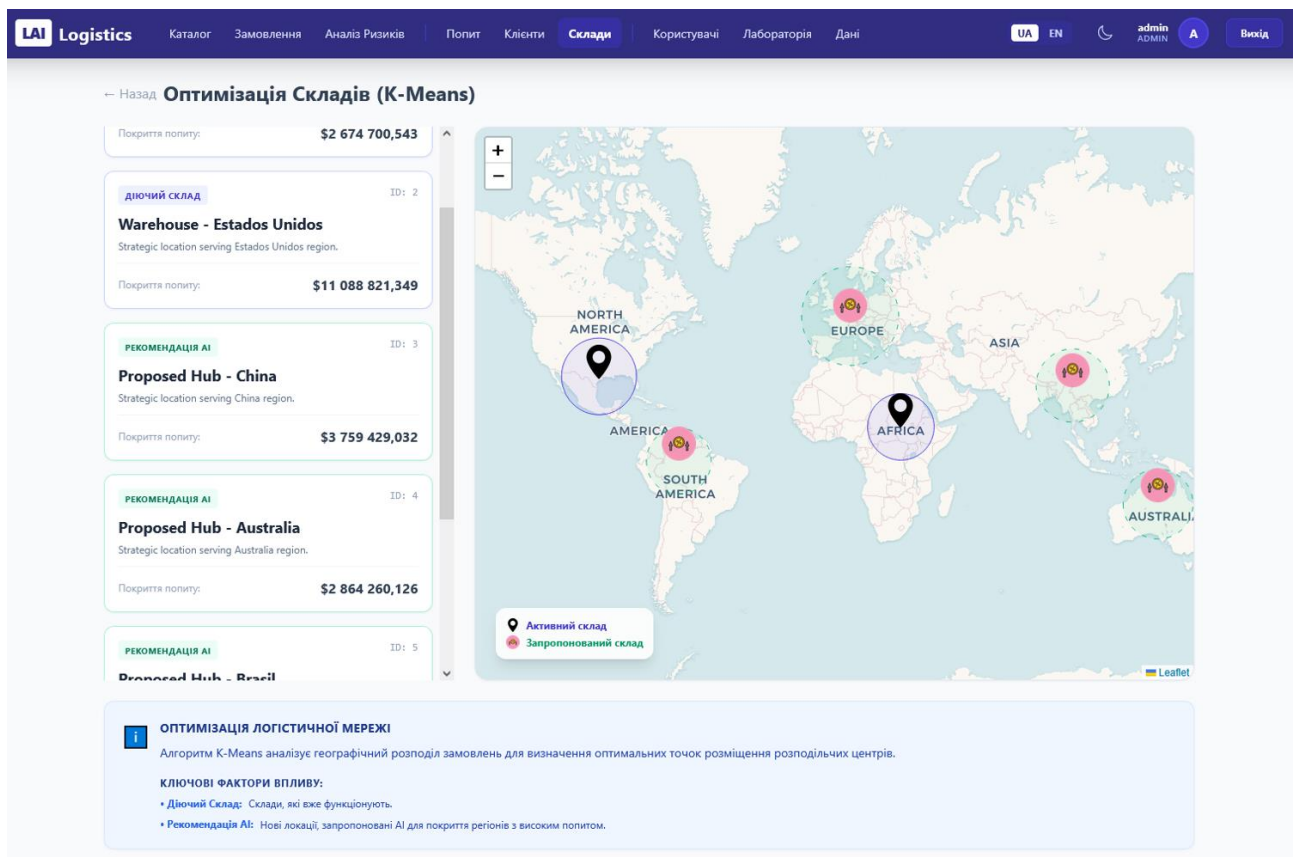


Рис. 3.15. Сторінка розрахунків зони покриття

Висновки до третього розділу

У третьому розділі описано реалізований програмний продукт, який відповідає сучасним стандартам веб-розробки. Інтерфейс є адаптивним і інтуїтивно зрозумілим, що забезпечує зручність використання на різних пристроях. Система надає гнучкі інструменти для налаштування параметрів аналізу, включаючи можливість завантаження власних даних, керування

правами доступу та використання режиму симуляції для тестування різних сценаріїв.

Візуалізація результатів аналізу реалізована через інтерактивні графіки та карти, що дозволяє швидко оцінювати ризики, прогнозувати попит та оптимізувати логістичні процеси. Завдяки інтеграції AI-моделей (Random Forest, LSTM, K-Means) з інтерфейсом користувача, система забезпечує реальну цінність для бізнес-задач, дозволяючи аналізувати попит, ризики та оптимізувати ланцюги постачання. Це робить систему потужним інструментом для стратегічного планування та ухвалення обґрунтованих рішень у реальному часі.

ВИСНОВКИ

У магістерській роботі вирішено задачу розробки додатку для аналізу прогнозування логістики. В ході дослідження було проаналізовано предметну область, сформовано технічні вимоги та спроектовано архітектуру програмного комплексу. Основним результатом роботи є реалізований програмний продукт, що поєднує сучасні веб-технології та алгоритми машинного навчання.

В рамках виконання роботи розроблено та реалізовано автоматизований ETL-конвеєр для обробки великих масивів даних, який вирішує проблему якості вхідної інформації шляхом її очищення, нормалізації та збагачення географічними відомостями, а також забезпечує фільтрацію даних відповідно до актуальних бізнес-правил та геополітичних обмежень. На основі підготовлених даних створено комплекс моделей штучного інтелекту: для прогнозування попиту з урахуванням сезонності та історичних трендів побудовано рекурентну нейронну мережу архітектури LSTM. Для виявлення ризиків та оцінки ймовірності затримок доставки застосовано алгоритм Random Forest Classifier. Для оптимізації логістичної мережі та автоматичного визначення локацій нових розподільчих центрів використано метод кластеризації K-Means.

Програмна реалізація системи виконана у вигляді повнофункціонального веб-додатку на основі мікросервісної архітектури з використанням стеку технологій Vue.js, Node.js та Python, що забезпечило високу швидкодію, зручність інтерфейсу та надійний захист даних завдяки системі розмежування прав доступу. Для візуалізації та аналітики розроблено інтерактивні дашборди, які надають користувачам можливість у зручному форматі переглядати результати моделювання, аналізувати географію продажів та отримувати рекомендації. Проведене тестування підтвердило коректність роботи системи, високу точність прогнозів та відповідність поставленим вимогам, а впровадження даного продукту дозволяє автоматизувати процеси прийняття рішень, мінімізувати ризики та підвищити ефективність планування ресурсів у логістиці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Basystyi I., Vakaliuk T. Use of machine learning and artificial intelligence in logistics. Тези VII Всеукраїнської науково-технічної конференції «Комп'ютерні технології: інновації, проблеми, рішення», м. Житомир, 02-03 грудня 2024 р. – Житомир: Житомирська політехніка, 2024. С. 21-22.
2. Basystyi I., Vakaliuk T., Chyzhmotria O., Use of the internet of things in logistics. Тези XV Міжнародної науково-технічної конференції «Інформаційно - комп'ютерні технології», м. Житомир, 28-29 березня 2025 р. – Житомир: Житомирська політехніка, 2025. С. 5-6.
3. Data Mining: Practical Machine Learning Tools and Techniques / I. H. Witten, E. Frank, M. A. Hall, C. J. Pal. 4th ed. Amsterdam : Morgan Kaufmann, 2016. 654 p. DOI: <https://doi.org/10.1016/C2015-0-02071-8>.
4. Van der Aalst W. M. P. Process Mining: Data Science in Action. 2nd ed. Berlin : Springer, 2016. 486 p. DOI: https://doi.org/10.1007/978-3-662-49851-4_2.
5. An Introduction to Statistical Learning: with Applications in R / G. James, D. Witten, T. Hastie, R. Tibshirani. 2nd ed. New York : Springer, 2021. 607 p. DOI: <https://doi.org/10.1007/978-1-0716-1418-1>.
6. Campbell J. O., Shin M. Essentials of Geographic Information Systems. Saylor Foundation, 2011. URL: <https://open.umn.edu/opentextbooks/textbooks/essentials-of-geographic-information-systems> (дата звернення: 11.12.2025).
7. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. 3rd ed. Melbourne: OTexts, 2021. URL: <https://otexts.com/fpp3/> (дата звернення: 11.12.2025).
8. Downey A. B. Think Bayes: Bayesian Statistics in Python. 2nd ed. Needham: Green Tea Press, 2021. 260 p. URL: <https://greenteapress.com/wp/think-bayes/> (дата звернення: 11.12.2025).
9. Array programming with NumPy / C. R. Harris et al. // Nature. 2020. Vol. 585. P. 357–362. DOI: <https://doi.org/10.1038/s41586-020-2649-2>.
10. VanderPlas J. Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media, 2016. 548 p. URL:

- <https://jakevdp.github.io/PythonDataScienceHandbook/> (дата звернення: 11.12.2025).
11. De Raedt L., Kersting K., Natarajan S., Poole D. Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Cham : Morgan & Claypool Publishers, 2016. 192 p. DOI: <https://doi.org/10.2200/S00692ED1V01Y201601AIM032>.
 12. Wilke C. O. Fundamentals of Data Visualization: A Primer on Making Informative and Compelling Figures. O'Reilly Media, 2019. URL: <https://clauswilke.com/dataviz/> (дата звернення: 11.12.2025).
 13. Zhang A., Lipton Z. C., Li M., Smola A. J. Dive into Deep Learning. arXiv preprint. 2021. DOI: <https://doi.org/10.48550/arXiv.2106.11342>. URL: <https://d2l.ai/> (дата звернення: 11.12.2025).
 14. Goodfellow I., Bengio Y., Courville A. Deep Learning. Cambridge : MIT Press, 2016. 800 p. URL: <https://www.deeplearningbook.org/> (дата звернення 11.12.2025)
 15. Reis J., Housley M. Fundamentals of Data Engineering: Plan and Build Robust Data Systems. O'Reilly Media, 2022. 446 p. URL: <https://www.oreilly.com/library/view/fundamentals-of-data/9781098108298/> (дата звернення 11.12.2025)
 16. Leskovec J., Rajaraman A., Ullman J. D. Mining of Massive Datasets. 3rd ed. Cambridge: Cambridge University Press, 2020. 569 p. DOI: <https://doi.org/10.1017/9781108684163>.
 17. James G., Witten D., Hastie T., Tibshirani R., Taylor J. An Introduction to Statistical Learning: with Applications in Python. Springer, 2023. 607 p. DOI: <https://doi.org/10.1007/978-3-031-38747-0>.
 18. Murphy K. P. Probabilistic Machine Learning: An Introduction. Cambridge : MIT Press, 2022. 864 p. URL: <https://probml.github.io/pml-book/book1.html> (дата звернення 11.12.2025)
 19. Hanck C., Arnold M., Gerber A., Schmelzer M. Introduction to Econometrics with R. Essen: University of Duisburg-Essen, 2020. URL: <https://www.econometrics-with-r.org/> (дата звернення: 11.12.2025).

20. Lara-Benítez P., Carranza-García M., Riquelme J. C. Deep Learning for Time Series Forecasting: A Survey. Knowledge-Based Systems. 2021. Vol. 227. Art. 107119. DOI: <https://doi.org/10.1016/j.knosys.2021.107119>.
21. Lim B., Zohren S. Time-series forecasting with deep learning: a survey. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2021. Vol. 379, no. 2194. Art. 20200209. DOI: <https://doi.org/10.1098/rsta.2020.0209>.
22. Lazzeri F. Machine Learning for Time Series Forecasting with Python. Hoboken : Wiley, 2020. 208 p. DOI: <https://doi.org/10.1002/9781119682394>.
23. Freitag M., Kotzab H., Pannek J. Dynamics in Logistics. Cham: Springer, 2022. 450 c. DOI: <https://doi.org/10.1007/978-3-031-05359-7>.
24. Pebesma E., Bivand R. Spatial Data Science: With Applications in R. Boca Raton: Chapman & Hall/CRC, 2023. 314 p. DOI: <https://doi.org/10.1201/9780429459016>.
25. Hardt M., Recht B. Patterns, Predictions, and Actions: Foundations of Machine Learning. Princeton: Princeton University Press, 2022. 296 p. URL: <https://mlstory.org/> (дата звернення: 11.12.2025).
26. van Steen M., Tanenbaum A. S. Distributed Systems. 3rd ed. Leiden: Maarten van Steen, 2017. URL: <https://www.distributed-systems.net/> (дата звернення: 11.12.2025).
27. Winters T., Manshreck T., Wright H. Software Engineering at Google: Lessons Learned from Programming Over Time. O'Reilly Media, 2020. URL: <https://abseil.io/resources/swe-book> (дата звернення: 11.12.2025).
28. Osmani A. Learning JavaScript Design Patterns. 2nd ed. O'Reilly Media, 2023. URL: <https://www.patterns.dev/vanilla/> (дата звернення: 11.12.2025).
29. Grigorik I. High Performance Browser Networking: What every web developer must know about networking and web performance. O'Reilly Media, 2013. URL: <https://hpbn.co/> (дата звернення: 11.12.2025).
30. McKinney W. Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter. 3rd ed. O'Reilly Media, 2022. URL: <https://wesmckinney.com/book/> (дата звернення: 11.12.2025).

- 31.Davoudian A., Chen L., Liu M. A survey on NoSQL stores. ACM Computing Surveys. 2018. Vol. 51, no. 2. P. 1–43. DOI: <https://doi.org/10.1145/3158661>.
- 32.Anderson R. Security Engineering: A Guide to Building Dependable Distributed Systems. 3rd ed. Indianapolis: Wiley, 2020. URL: <https://www.cl.cam.ac.uk/~rja14/book.html> (дата звернення: 11.12.2025).
- 33.Beyer B., Jones C., Petoff J., Murphy N. R. Site Reliability Engineering: How Google Runs Production Systems. O'Reilly Media, 2016. URL: <https://sre.google/books/>
- 34.Copes F. The Node.js Handbook. 2020. URL: <https://www.freecodecamp.org/news/the-node-js-handbook/> (дата звернення: 11.12.2025).
- 35.Vue.js Team. Vue.js 3 Documentation. 2023. URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 11.12.2025).
- 36.Haverbeke M. Eloquent JavaScript: A Modern Introduction to Programming. 4th ed. No Starch Press, 2024. URL: <https://eloquentjavascript.net/> (дата звернення: 11.12.2025).
- 37.Ecma International. ECMAScript® 2023 Language Specification. 14th ed. Ecma International, 2023. Standard ECMA-262. URL: <https://www.ecma-international.org/publications-and-standards/standards/ecma-262/> (дата звернення: 11.12.2025).
- 38.Jackson C. Micro Frontends. Martin Fowler Blog. 2019. URL: <https://martinfowler.com/articles/micro-frontends.html> (дата звернення: 11.12.2025).
- 39.Morote E. S. M. Pinia: The Intuitive Store for Vue.js. 2023. URL: <https://pinia.vuejs.org/> (дата звернення: 11.12.2025).
- 40.Abelson H., Sussman G. J., Sussman J., Hekmat M. Structure and Interpretation of Computer Programs: JavaScript Edition. Cambridge : MIT Press, 2022. 640 p. URL: <https://mitpress.mit.edu/9780262543231/structure-and-interpretation-of-computer-programs/> (дата звернення: 11.12.2025).

ДОДАТКИ

Лістинг алгоритму обробки датасету

```

import pandas as pd
import numpy as np
import os
def load_and_clean_data(file_path):
    if not os.path.exists(file_path):
        raise FileNotFoundError(f"File not found: {file_path}")
    print(f>Loading data from {file_path}...")
    try:
        df = pd.read_csv(file_path, encoding='latin-1')
    except UnicodeDecodeError:
        df = pd.read_csv(file_path, encoding='utf-8')
    rename_map = {
        'order date (DateOrders)': 'order_date',
        'Shipping date (DateOrders)': 'shipping_date',
        'Days for shipping (real)': 'shipping_real',
        'Days for shipment (scheduled)': 'shipping_scheduled',
        'Late_delivery_risk': 'late_delivery_risk',
        'Order Item Quantity': 'order_item_quantity',
        'Order Item Discount Rate': 'order_item_discount_rate',
        'Shipping Mode': 'shipping_mode',
        'Market': 'market',
        'Order Region': 'order_region',
        'Latitude': 'latitude',
        'Longitude': 'longitude',
        'Order Status': 'order_status',
        'Type': 'type',
        'Order City': 'order_city',
        'Order Country': 'order_country',
        'Sales': 'sales',
        'Product Name': 'product_name',
        'Category Name': 'category_name',
        'Product Price': 'product_price',
        'Product Image': 'product_image',
        'Product Status': 'product_status',
        'Customer City': 'customer_city',
        'Customer Country': 'customer_country',
        'Order Id': 'order_id',
        'Customer Id': 'customer_id',
        'Order Profit Per Order': 'order_profit_per_order',
        'Order Item Total': 'order_item_total'
    }
    df = df.rename(columns=rename_map)
    df.columns = df.columns.str.lower().str.replace(' ', '_')
    if 'order_country' in df.columns:
        df = df[~df['order_country'].isin(blacklist)]

```

```

if 'order_date' in df.columns:
    df['order_date'] = pd.to_datetime(df['order_date'], errors='coerce')
    df['order_year'] = df['order_date'].dt.year
    df['order_month'] = df['order_date'].dt.month
if 'shipping_date' in df.columns:
    df['shipping_date'] = pd.to_datetime(df['shipping_date'], errors='coerce')
if 'shipping_real' in df.columns and 'shipping_scheduled' in df.columns:
    df['shipping_delay'] = df['shipping_real'] - df['shipping_scheduled']
    df['late_delivery_actual'] = np.where(df['shipping_delay'] > 0, 1, 0)
critical_cols = ['order_date', 'shipping_mode', 'latitude', 'longitude']
existing_critical = [col for col in critical_cols if col in df.columns]
if existing_critical:
    df.dropna(subset=existing_critical, inplace=True)
print(f'Data loaded successfully. Shape: {df.shape}')
return df

```

Лістинг тренування моделей для прогнозування

```

import pandas as pd
import numpy as np
import os
import joblib
import warnings
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from dotenv import load_dotenv
from data_loader import load_and_clean_data
load_dotenv()
warnings.filterwarnings('ignore')

MODELS_DIR = os.getenv("MODELS_DIR")
DATA_PATH = os.getenv("DATA_PATH")
if not os.path.exists(MODELS_DIR):
    os.makedirs(MODELS_DIR)
try:
    df = load_and_clean_data(DATA_PATH)
except Exception as e:
    print(e)
    exit()
monthly_demand = df.groupby(['order_year',
                              'order_month'])['order_item_quantity'].sum().reset_index()
data_series = monthly_demand['order_item_quantity'].values.reshape(-1, 1)
scaler_lstm = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler_lstm.fit_transform(data_series)
def create_sequences(data, seq_length):
    X, y = [], []
    for i in range(len(data) - seq_length):
        X.append(data[i:(i + seq_length), 0])
        y.append(data[i + seq_length, 0])
    return np.array(X), np.array(y)
SEQUENCE_LENGTH = 3
if len(scaled_data) > SEQUENCE_LENGTH:
    X_lstm, y_lstm = create_sequences(scaled_data, SEQUENCE_LENGTH)
    X_lstm = X_lstm.reshape(X_lstm.shape[0], X_lstm.shape[1], 1)
    model_lstm = Sequential()
    model_lstm.add(LSTM(50, activation='relu', input_shape=(SEQUENCE_LENGTH, 1)))
    model_lstm.add(Dense(1))
    model_lstm.compile(optimizer='adam', loss='mse')
    model_lstm.fit(X_lstm, y_lstm, epochs=50, batch_size=2, verbose=0)
    model_lstm.save(os.path.join(MODELS_DIR, 'lstm_demand_model.h5'))

```

```

joblib.dump(scaler_lstm, os.path.join(MODELS_DIR, 'lstm_scaler.joblib'))

cat_features = ['shipping_mode', 'market', 'order_region', 'category_name', 'order_country']
num_features = ['shipping_scheduled', 'order_item_discount_rate', 'order_item_quantity']
X = df[cat_features + num_features]
y = df['late_delivery_actual']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
preprocessor = ColumnTransformer(transformers=[
    ('num', StandardScaler(), num_features),
    ('cat', OneHotEncoder(handle_unknown='ignore'), cat_features)
])
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(n_estimators=50))
])
pipeline.fit(X_train, y_train)
joblib.dump(pipeline, os.path.join(MODELS_DIR, 'delivery_risk_pipeline.joblib'))
if 'latitude' in df.columns:
    kmeans = KMeans(n_clusters=6, random_state=42, n_init=10)
    kmeans.fit(df[['latitude', 'longitude']])
    joblib.dump(kmeans, os.path.join(MODELS_DIR, 'location_kmeans.joblib'))

```