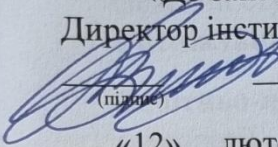


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

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

«До захисту в ЕК»

Директор інституту(декан факультету)



Андрій Форсюк

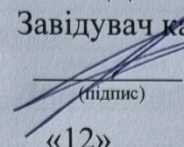
(підпис)

(ім'я та прізвище)

«12» лютого 2024р.

«До захисту допущено»

Завідувач кафедри



Сергій Грибков

(підпис)

(ім'я та прізвище)

«12» лютого 2024р.

КВАЛІФІКАЦІЙНА РОБОТА  
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Інформаційні управляючі системи та технології

на тему: Дослідження та розроблення інформаційної системи для відділу логістики ТОВ «Чернівецький хлібокомбінат»

Виконав: здобувач 2 курсу, групи ІС-2-3М

Цветков Олексій Олегович

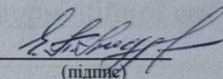
(прізвище, ім'я, по батькові повністю)



(підпис)

Керівник Андріюк Олена Петрівна

(прізвище, ім'я та по батькові повністю)



(підпис)

Консультанти

(ім'я та прізвище)

(підпис)

(ім'я та прізвище)

(підпис)

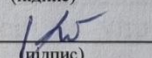
(ім'я та прізвище)

(підпис)

Рецензент

Александр Пунев

(ім'я та прізвище)



(підпис)

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

Здобувач



(підпис)

Київ - 2024р.

## НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем  
 Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки  
 Освітній ступінь магістр  
 Спеціальність 122 «Комп'ютерні науки»  
(код і назва)  
 Освітньо-професійна програма Інформаційні управляючі системи та технології  
(назва)

### ЗАТВЕРДЖУЮ

Завідувач  
 кафедри Інформаційних технологій,  
 штучного інтелекту і кібербезпеки

  
Грибков С.В.  
 “ 19 ” грудня 2023 року

### ЗАВДАННЯ

#### НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Цветков Олексій Олегович

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розроблення інформаційної системи для відділу логістики ТОВ «Чернівецький хлібокомбінат»

керівник роботи Андріюк Олена Петрівна доцент, кандидат фізико-математичних наук

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 19 грудня 2023 року №1006-кс

2. Строк подання здобувачем роботи 23.01.2024

3. Вихідні дані до роботи специфікації додатку, серверні сценарії, специфікації захисту даних, система контролю версій та управління кодом, документація відділу логістики,

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) розділ 1. Системний аналіз ТОВ «чернівецький хлібокомбінат» та постановка задачі на проектування, розділ 2 дослідження логістичних інформаційних систем 3 Розділ Розробка інформаційної системи для відділу логістики ТОВ «Чернівецький Хлібокомбінат»

5. Перелік графічного матеріалу

Організаційна структура підприємства, інтерфейс користувача, схема бази даних

6. Консультанти розділів роботи



## АНОТАЦІЯ

Дослідження та розроблення інформаційної системи для відділу логістики ТОВ «Чернівецький хлібокомбінат»

Магістерська робота містить: 90 сторінок, 3 розділи, 42 рисунків та 15 літературних джерел.

Спеціальність: 122 «Комп'ютерні науки»

Метою магістерської роботи є дослідження, проектування та розробка інформаційної системи для відділу логістики ТОВ «Чернівецький хлібокомбінат». Система має на меті оптимізацію логістичних процесів, підвищення ефективності управління запасами та покращення моніторингу логістичних маршрутів.

У магістерській роботі: проведено дослідження інформаційної системи логістичного відділу ТОВ «Чернівецький хлібокомбінат». Проведено дослідження логістичних інформаційних систем. Розроблено інформаційну систему для логістичного відділу.

Результатом виконання магістерської роботи є створена інформаційна система для відділу логістики.

**КЛЮЧОВІ СЛОВА :** ТОВ «Чернівецький хлібокомбінат», додаток, веб-платформа, сервер, API, REST, MVVM, Visual Studio, ООП, С#, фреймворк, паттерн, аналіз, відгуки, база даних, запит

## ABSTRACT

Research and Development of an Information System for the Logistics Department of TOV "Chernivetskyi Khlibokombinat"

The master's thesis consists of 90 pages, 3 chapters, 42 figures, and 15 literary sources.

Specialty: 122 "Computer Science"

The aim of the master's work is the investigation, design, and development of an information system for the logistics department of TOV "Chernivetskyi Khlibokombinat." The system aims to optimize logistical processes, enhance inventory management efficiency, and improve monitoring of logistics routes.

In the master's work, the following was accomplished: research on the information system of the logistics department of TOV "Chernivetskyi Khlibokombinat" was conducted, as well as an investigation of logistics information systems. An information system for the logistics department was developed.

The result of the master's work is the creation of an information system for the logistics department.

**KEYWORDS:** TOV "Chernivetskyi Khlibokombinat," application, web platform, server, API, REST, MVVM, Visual Studio, OOP, C#, framework, pattern, analysis, feedback, database, query.

## ЗМІСТ

Вступ.....	10
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТОВ «ЧЕРНІВЕЦЬКИЙ ХЛІБОКОМБІНАТ» ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ.....	14
1.1 Опис та особливості функціонування компанії ТОВ «Чернівецький хлібокомбінат».....	14
1.1.1 Виробничі можливості підприємства.....	15
1.1.2 Споживчий ринок.....	16
1.1.3 Корпоративна філософія.....	17
1.2 Організаційна структура підприємства, роль та взаємодія підрозділів ...	18
1.3 Логістичний відділ .....	19
1.3.1 Основні задачі Логістичного відділу .....	19
1.3.2 Взаємодія з іншими підрозділами .....	21
1.4 Дослідження та аналіз існуючої на підприємстві інформаційної системи	23
1.4.1 Апаратне забезпечення .....	23
1.4.2 Програмне забезпечення .....	23
1.5 Функції відділу логістики.....	25
1.5.1 Управління складською діяльністю: .....	25
1.5.2 Оптимізація виробничих процесів: .....	25
1.5.3 Управління транспортними ланками: .....	26
1.5.4 Інвентаризація та управління запасами: .....	27
1.5.5 Оптимізація ланцюга постачання: .....	27
1.6 Аналіз існуючих аналогів розробки: .....	28
1.6.1 SAP Integrated Business Planning:.....	28
1.6.2 Oracle SCM Cloud: .....	29

1.6.3 Microsoft Dynamics 365 Supply Chain Management:.....	29
1.6.4 Аналіз альтернативних рішень: .....	30
1.7 Обґрунтування доцільності проектування й розроблення інформаційної системи для відділу логістики: .....	30
1.8 Постановка задачі.....	32
1.8.1 Призначення та цілі системи: .....	32
1.8.2 Функції, які повинна виконувати система:.....	34
1.8.3 Вимоги до інтерфейсу користувача: .....	34
1.9 Концептуальна модель системи.....	35
1.9.1 Користувачі системи.....	35
1.9.2 Функціональні модулі.....	36
1.9.3 Взаємодія функціональних модулів .....	38
1.10 Висновок до першого розділу.....	39
<b>2 РОЗДІЛ ДОСЛІДЖЕННЯ ЛОГІСТИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ..</b>	<b>41</b>
2.1 Загальні відомості та завдання логістичних інформаційних систем .....	41
2.1.1 Визначення логістичних інформаційних систем .....	41
2.1.2 Завдання логістичних інформаційних систем.....	42
2.1.3 Роль Логістичних Інформаційних Систем у Сучасному Бізнесі.....	43
2.1.4 Вплив Логістичних Інформаційних Систем на Зниження Витрат та Підвищення Ефективності .....	44
2.1.6 Критичний Аналіз Логістичних Інформаційних Систем.....	45
2.2 Сучасні технології в управлінні транспортною логістикою .....	46
2.2.1 Системи машинного навчання в управлінні транспортною логістикою .....	47
2.2.2 Нейронні мережі в управлінні транспортною логістикою .....	49

2.2.3 Системи геолокації в управлінні транспортною логістикою .....	50
2.2.4 Системи прогнозування трафіку в управлінні транспортною логістикою.....	51
2.2.5 Висновки до підрозділу .....	52
2.3 Використання цифрових двійників у логістиці .....	53
2.3.1 Визначення цифрового двійника .....	53
2.3.2 Зв'язки між цифровими двійниками в системах.....	54
2.3.3 Цифровий двійник в життєвому циклі об'єкту .....	56
2.4 Висновки до другого розділу .....	58
3 РОЗДІЛ РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВІДДІЛУ ЛОГІСТИКИ ТОВ «ЧЕРНІВЕЦЬКИЙ ХЛІБОКОМБІНАТ» .....	59
3.1 Вибір та опис засобів розробки .....	59
Розділ 3.1.1 Мова програмування C# .....	59
Розділ 3.1.2 Фреймворк ASP.NET Core .....	61
Розділ 3.1.3 Microsoft SQL Server .....	63
Розділ 3.1.4 Visual Studio .....	64
Розділ 3.1.5: .NET MAUI .....	65
Розділ 3.1.6 JavaScript .....	66
Розділ 3.1.7 ReactJS .....	67
Розділ 3.1.8 Bootstrap .....	68
3.2 Архітектура системи .....	69
3.2.1 Rest API .....	69
3.2.2 MVVM та .Net Maui .....	71
3.3 Розробка системи .....	72
3.3.1 Створення серверної частини .....	72

3.3.2 Створення веб платформи .....	82
Висновок .....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	89

## ВСТУП

### Актуальність теми

Сучасний бізнес, зокрема у сфері логістики, невіддільно пов'язаний із стрімким розвитком інформаційних технологій. Зростання обсягів даних, логістичні виклики та потреба в ефективному управлінні ресурсами ставлять перед підприємствами завдання створення інтегрованих інформаційних систем. Таким чином, розробка інформаційної системи для відділу логістики ТОВ «Чернівецький хлібокомбінат» стає актуальною та перспективною задачею.

Зв'язок роботи з науковими програмами, планами, темами кафедри, університету

Дослідження здійснюється в контексті сучасних тенденцій розвитку інформаційних технологій та логістики. Виконана робота відповідає високим стандартам університетської та науково-дослідної діяльності.

### Мета дослідження

Метою дипломної роботи є дослідження, проектування та розробка інформаційної системи для відділу логістики ТОВ «Чернівецький хлібокомбінат». Система має на меті оптимізацію логістичних процесів, підвищення ефективності управління запасами та покращення моніторингу логістичних маршрутів.

### Завдання дослідження

Дослідження та вибір технічних засобів для розробки мобільного застосунку.

Розробка інформаційної системи яка спрямована на оптимізацію робочих процесів.

Дослідження логістичних інформаційних систем.

### Об'єкт дослідження

Об'єктом даного дослідження є розробка інформаційної системи для ТОВ "Чернівецький хлібокомбінат".

### Предмет дослідження

Предметом дослідження є процес розробки та впровадження інформаційної системи для відділу логістики.

### Методи дослідження

Для досягнення поставлених цілей використовуються методи системного аналізу, дослідження логістичних інформаційних систем, а також методи проектування та програмування інформаційних систем.

### Наукова новизна одержаних результатів

Розробка комплексу завдань автоматизації для відділу логістики включає в себе використання сучасних технологій та інструментів, що робить результати дослідження новаторськими у контексті використання цифрових рішень у логістичній сфері.

### Практичне значення одержаних результатів

Розроблена інформаційна система стане ефективним інструментом для оптимізації логістичних процесів ТОВ «Чернівецький хлібокомбінат». Вона сприятиме підвищенню продуктивності роботи відділу логістики, зменшенню витрат та підвищенню конкурентоспроможності підприємства.

### Особистий внесок здобувача

Автор роботи особисто здійснив вибір та опис засобів розробки, розробив архітектуру системи, провів аналіз логістичних інформаційних систем, та взяв участь у всіх етапах розробки інформаційної системи.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Скорочення:

ТОВ - Товариство з обмеженою відповідальністю

IT - Інформаційні технології (Information technologies)

API - Інтерфейс програмування застосунків (Application Programming Interface).

MVVM - Модель-Вид-Модель Виду (Model-View-ViewModel).

ООР - Об'єктно-орієнтоване програмування (Object-Oriented Programming).

REST - Представлення репрезентативного стану (Representational State Transfer).

IDE - Інтегроване середовище розробки (Integrated Development Environment).

HTML - Гіпертекстова мова розмітки (Hypertext Markup Language)

CSS - Каскадні таблиці стилів (Cascading Style Sheets)

SQL - мова структурованих запитів (Structured Query Language)

JSON - JavaScript Object Notation

XML - eXtensible Markup Language

Цифровий двійник (Digital Twin):

Терміни

Цифровий двійник - це віртуальне представлення реального об'єкта чи системи в цифровому вигляді, яке включає в себе дані про його стан та поведінку.

Машинне навчання (Machine Learning):

Машинне навчання - це галузь штучного інтелекту, яка дозволяє комп'ютерам вчитися з даних та вдосконалювати свої алгоритми без явного програмування.

Транспортна логістика:

Транспортна логістика - це галузь управління ланцюгами постачання, що вивчає та оптимізує ефективність переміщення товарів від постачальника до споживача.

Системи геолокації:

Системи геолокації - це технології, які визначають місцезнаходження об'єктів за допомогою супутників, сенсорів та інших засобів.

Оптимізація маршрутів:

Оптимізація маршрутів - це процес вибору найефективнішого шляху для переміщення об'єкта від одного пункту до іншого з урахуванням різних факторів, таких як вартість та час.

## **РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТОВ «ЧЕРНІВЕЦЬКИЙ ХЛІБОКОМБІНАТ» ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ**

### **1.1 Опис та особливості функціонування компанії ТОВ «Чернівецький хлібокомбінат»**

ТОВ "Чернівецький хлібокомбінат" - сучасний виробник високоякісної хлібобулочної продукції, який знаходиться в місті Чернівці. Комбінат визначається своєю дбайливою увагою до якості, інноваційним підходом до виробництва та відданістю задоволенню потреб споживачів.

Основні особливості "Чернівецького хлібокомбінату":

- Широкий асортимент продукції: Комбінат виготовляє різноманітні сорти хліба, булок, багетів, пирогів та інших хлібобулочних виробів. Різноманітність виробів робить його популярним серед різних смакових уподобань споживачів.
- Використання натуральних інгредієнтів: "Чернівецький хлібокомбінат" покладає особливий акцент на використання високоякісних та натуральних інгредієнтів у виробництві. Це гарантує не лише чудовий смак, але й корисність хлібобулочної продукції.
- Технологічне обладнання: Комбінат оснащений сучасним технологічним обладнанням, яке дозволяє автоматизувати багато етапів виробництва, забезпечуючи стабільну якість продукції.
- Система контролю якості: "Чернівецький хлібокомбінат" приділяє особливу увагу контролю якості на кожному етапі виробництва. Це включає в себе випробування сировини, виробничі процеси та готову продукцію.
- Екологічна відповідальність: Комбінат дбає про екологічну стійкість свого виробництва, використовуючи енергоефективні технології та приділяючи увагу утилізації відходів.

- **Спільнота та соціальна відповідальність:** Комбінат активно взаємодіє з місцевою спільнотою та підтримує різноманітні благодійні та соціальні ініціативи.

Такий підхід дозволяє "Чернівецькому хлібокомбінату" займати достойне місце на ринку хлібобулочних виробів, завойовуючи довіру споживачів своєю якістю та відданістю принципам сучасного виробництва.

### **1.1.1 Виробничі можливості підприємства**

ТОВ "Чернівецький хлібокомбінат" володіє вражаючими виробничими можливостями, що дозволяють підприємству забезпечувати високий рівень якості та ефективності у виробництві хлібобулочних виробів.

- **Автоматизовані технологічні процеси:** Всі етапи виробництва, від замішування тіста до упаковки готової продукції, ретельно автоматизовані. Використання сучасного обладнання та систем автоматизації дозволяє значно підвищити продуктивність та забезпечити стабільну якість виробів.

- **Гнучкість виробництва:** Виробництво адаптоване до змінних умов ринку. Підприємство може швидко реагувати на зміни сезонності або зростання попиту на конкретні види продукції. Гнучкість виробництва забезпечує швидкий відгук на потреби споживачів та мінливі умови ринку.

- **Технологічна інноваційність:** Підприємство постійно вдосконалює технологічні процеси та впроваджує новітні розробки у галузі хлібопекарської промисловості. Використання інноваційних технологій дозволяє не лише підтримувати високий стандарт якості, але й залишатися конкурентоспроможними на ринку.

- **Контроль якості:** Система контролю якості охоплює всі етапи виробництва, включаючи вхідні матеріали, технологічні процеси та готову продукцію. Використання сучасних методів контролю та аналізу дозволяє виявляти та усувати потенційні проблеми на ранніх етапах виробництва.

Ці виробничі можливості створюють сильну основу для розвитку підприємства та відповідають високим стандартам якості та ефективності в галузі хлібопекарства.

### **1.1.2 Споживчий ринок**

ТОВ "Чернівецький хлібокомбінат" має чітко визначений споживчий ринок, що включає різноманітний асортимент хлібобулочних виробів та пристосований до змін у попиті споживачів. Опис споживчого ринку подано нижче:

- **Діапазон продукції:** "Чернівецький хлібокомбінат" забезпечує широкий вибір хлібобулочної продукції, що охоплює різні види хлібу, булочок, плячків, тістечок та інших виробів. Асортимент продукції постійно оновлюється, враховуючи сучасні тенденції та смакові уподобання споживачів.

- **Дистрибуція:** Мережа розподільних центрів та транспортна інфраструктура ретельно організовані для забезпечення ефективного розподілу продукції. Завдяки цьому, товари швидко та надійно доставляються до магазинів, супермаркетів, ресторанів та інших торгових точок.

- **Якість продукції:** Репутація "Чернівецького хлібокомбінату" ґрунтується на високій якості виробів. Система контролю якості охоплює кожен етап виробництва, починаючи від вибору сировини та закінчуючи упаковкою готової продукції.

- **Підтримка лояльності споживачів:** Завдяки постійному вивченню смакових та культурних уподобань споживачів, підприємство створює продукцію, яка відповідає актуальним тенденціям ринку. Результатом є висока лояльність споживачів та стабільна популярність продукції на ринку.

Ці аспекти гарантують успішне позиціонування "Чернівецького хлібокомбінату" на ринку хлібобулочних виробів, підтримуючи стабільний попит та задоволення потреб різних сегментів споживачів.

### 1.1.3 Корпоративна філософія

Корпоративна філософія ТОВ "Чернівецький хлібокомбінат" є фундаментальним принципом, який визначає цінності та стратегічний курс підприємства. Основні принципи корпоративної філософії підприємства:

- **Якість:** Як один з основних принципів корпоративної філософії, підприємство прагне до найвищих стандартів якості в усіх аспектах виробництва. Використання тільки високоякісної сировини та технологічних інновацій дозволяє підтримувати стабільний рівень якості хлібобулочних виробів.
- **Інновації:** Корпоративна філософія активно підтримує інновації та дослідження в галузі технологій виробництва та логістики. Здійснення постійних вкладень у дослідження та впровадження новітніх розробок дозволяє підприємству залишатися на передових позиціях у галузі.
- **Соціальна відповідальність:** "Чернівецький хлібокомбінат" дотримується принципу соціальної відповідальності у всій своїй діяльності. Підприємство активно сприяє розвитку місцевих спільнот, вживає заходів для збереження навколишнього середовища та визначає чіткі стандарти корпоративної поведінки.
- **Прозорість та етика:** Взаємодія зі співробітниками, партнерами та клієнтами ґрунтується на принципах прозорості та етичності. Підприємство дотримується чесних та відкритих стосунків в усіх сферах своєї діяльності.
- **Розвиток персоналу:** Компанія вкладає значні зусилля у розвиток свого персоналу через систему навчання та підвищення кваліфікації. Забезпечення співробітників необхідними знаннями та навичками є ключовим елементом стратегії управління персоналом.

Ці принципи корпоративної філософії сприяють створенню стійкої основи для розвитку підприємства, сприяючи його стабільності, конкурентоспроможності та сприятливій репутації на ринку.



Фінансовий відділ займається фінансовим плануванням, обліком витрат та доходів, управлінням бюджетом та забезпеченням фінансової стабільності підприємства. Взаємодіє з усіма підрозділами для надання фінансової підтримки та забезпечення оптимального використання ресурсів.

Маркетинговий та збутовий відділ відповідає за рекламу, планування продажів, вивчення ринку та забезпечення високого рівня збуту продукції. Взаємодіє з виробничим та логістичним відділами для планування обсягів виробництва та логістичних можливостей.

Відділ стратегічного планування відповідає за розробку та впровадження стратегічних планів, аналіз ринкових умов та визначення довгострокових цілей підприємства. Взаємодіє з усіма підрозділами для врахування стратегічних цілей та спрямування діяльності підприємства на досягнення успіху в довгостроковій перспективі.

Ці підрозділи утворюють взаємопов'язану систему, що сприяє ефективному функціонуванню та досягненню стратегічних цілей "Чернівецького хлібокомбінату".

### **1.3 Логістичний відділ**

Логістичний відділ на ТОВ "Чернівецький хлібокомбінат" є невід'ємною частиною організаційної структури, спеціалізованою на ефективному управлінні логістичними процесами. Відділ відзначається високим рівнем кваліфікації персоналу та використанням передових методів для досягнення стратегічних цілей підприємства.

#### **1.3.1 Основні задачі Логістичного відділу**

Постачання сировини та матеріалів:

- Забезпечення надійного та своєчасного постачання необхідних сировинних матеріалів для виробництва.

- Оптимізація ланцюга постачання для зниження витрат та мінімізації ризиків.

Управління запасами:

- Моніторинг рівня запасів та розробка стратегій для їх ефективного управління.

- Оптимізація запасів для підтримання необхідних обсягів без надмірності.

Оптимізація поставок та транспортування:

- Визначення оптимальних маршрутів та транспортних засобів для ефективності та зменшення витрат.

- Контроль за доставкою та вчасне реагування на можливі затримки чи непередбачені обставини.

Контроль якості логістичних процесів:

- Розробка та впровадження стандартів контролю за якістю логістичних послуг та процесів.

- Співпраця з іншими відділами для забезпечення високої якості продукції та послуг.

Управління ланцюгом постачання:

- Координація всіх етапів ланцюга постачання для забезпечення гармонійної роботи всього виробничого процесу.

- Взаємодія з виробничим та маркетинговим відділами для врахування стратегічних цілей підприємства.

Планування та координація:

- Розробка логістичних стратегій та планів дій для досягнення цілей компанії.

- Координація діяльності з іншими відділами для оптимального використання ресурсів.

Інтеграція технологій:

- Впровадження сучасних інформаційних технологій для автоматизації логістичних процесів та отримання актуальної інформації.

Аналіз та вдосконалення:

- Постійний моніторинг логістичних процесів з метою виявлення можливостей для оптимізації та підвищення ефективності.

Ризик-менеджмент:

- Виявлення та оцінка ризиків, пов'язаних з логістичними процесами, та розробка стратегій їхнього управління.

### **1.3.2 Взаємодія з іншими підрозділами**

Взаємодія Логістичного відділу з іншими підрозділами підприємства є важливим чинником для забезпечення високої ефективності та виконання стратегічних цілей. Такий підхід дозволяє досягати синергії між різними функціональними групами та забезпечує інтегрований підхід до управління логістикою на підприємстві.

Виробничий підрозділ:

- Планування поставок: Логістичний відділ співпрацює з виробничим підрозділом для визначення оптимального графіку поставок сировини та компонентів, забезпечуючи безперебійний виробничий процес.
- Управління запасами: Взаємодія з виробничим відділом у сфері управління запасами, щоб визначити оптимальні рівні і запобігти дефіцитам чи перевищенням запасів.

Відділ кадрів:

- Кваліфікації та навчання: Логістичний відділ співпрацює з відділом кадрів для розвитку та підтримки кваліфікаційного потенціалу працівників, особливо тих, які працюють у сфері логістики.

- Процеси найму: Взаємодія в процесі найму та оцінки персоналу для забезпечення належної кількості та якості робочої сили.

Фінансовий відділ:

- Бюджетування: Обговорення логістичних витрат та планів з фінансовим відділом для визначення бюджету та ефективного використання ресурсів.

- Аналіз витрат: Спільний аналіз витрат логістичних операцій для забезпечення фінансової ефективності.

Маркетинговий та збутовий відділ:

- Прогнозування попиту: Логістичний відділ обмінюється інформацією про прогнози попиту з маркетинговим відділом для належного планування логістичних операцій.

- Спільна рекламна та просувальна кампанія: Взаємодія в організації спільних заходів для просування продукції та забезпечення логістичної підтримки.

Відділ стратегічного планування:

- Реалізація стратегічних ініціатив: Логістичний відділ співпрацює у впровадженні стратегій підприємства, забезпечуючи логістичну інфраструктуру для їх успішної реалізації.

- Забезпечення виробничої гнучкості: Взаємодія для забезпечення логістичної гнучкості у відповідь на стратегічні виклики.

## **1.4 Дослідження та аналіз існуючої на підприємстві інформаційної системи**

### **1.4.1 Апаратне забезпечення**

Проведене дослідження апаратного забезпечення на підприємстві "Чернівецький хлібокомбінат" виявило, що стан існуючого апаратного парку є гарним та відповідає сучасним вимогам ефективності та надійності.

Мережеве обладнання:

На підприємстві "Чернівецький хлібокомбінат" мережеве обладнання представлено сучасними засобами забезпечення стабільної та швидкої мережевої інфраструктури:

Комп'ютерне обладнання:

Комп'ютерне обладнання на робочих станціях є сучасним та відповідає вимогам завдань співробітників.

Серверне обладнання:

Серверне обладнання використовується відомих виробників і забезпечує стабільну роботу корпоративних систем.

Засоби безпеки

- Встановлені та регулярно оновлюються брандмауери та антивірусне ПЗ для забезпечення високого рівня безпеки мережі та інформації.
- Використовуються сучасні системи автоматичного резервного копіювання для забезпечення можливості швидкого відновлення даних у випадку аварій або втрати.

Цей аналіз підтверджує, що апаратне забезпечення на підприємстві "Чернівецький хлібокомбінат" є сучасним, функціональним та високоефективним, що сприяє стабільності та ефективності бізнес-процесів.

### **1.4.2 Програмне забезпечення**

Програмне забезпечення визначає сучасність та ефективність бізнес-процесів, надаючи комплексні рішення для автоматизації різноманітних завдань. Детальний розгляд різних категорій програмного забезпечення відкриває високий рівень технологічної готовності підприємства та його готовність до викликів сучасного бізнес-середовища.

- Серверні операційні системи:

Використовується операційна система Windows Server, яка є стандартом для серверних середовищ. Вона забезпечує стабільну роботу серверів та підтримує ефективність мережевих операцій.

- Операційні системи на робочих станціях:

Встановлені сучасні версії операційної системи Windows 10 для робочих станцій користувачів. Це забезпечує високий рівень сумісності з програмним забезпеченням та зручний інтерфейс для користувачів.

- Офісні програми:

Користуються офісним пакетом Microsoft Office, який включає в себе програми для роботи з текстовими документами (Word), електронною таблицею (Excel), презентаціями (PowerPoint) та іншими офісними інструментами. Це сприяє ефективному веденню документації та комунікації.

- Бухгалтерські програми:

Застосовується спеціалізоване бухгалтерське програмне забезпечення, яке дозволяє автоматизувати облік фінансових операцій, ведення податкового обліку та формування звітності.

- Засоби безпеки:

Встановлені системи антивірусного захисту та засоби безпеки для запобігання кібератак та забезпечення безпеки інформації. Використовуються відомі рішення від провідних виробників

- Електронні торгові платформи:

Для оптимізації закупівель та співпраці з постачальниками використовуються сучасні електронні торгові платформи, які спрощують процеси управління закупівлями та взаємодії з партнерами.

Цей аналіз програмного забезпечення на підприємстві "Чернівецький хлібокомбінат" виявив, що компанія ефективно використовує сучасні інструменти для автоматизації та оптимізації різноманітних бізнес-процесів. Різні категорії програмного забезпечення використовуються для різних аспектів діяльності підприємства, забезпечуючи його ефективність та конкурентоспроможність на ринку.

## **1.5 Функції відділу логістики**

В данному розділі проводиться детальний розгляд функціональних аспектів відділу логістики ТОВ "Чернівецький хлібокомбінат". Кожна функція відділу розглядається окремо для визначення її ролі в загальній логістичній структурі підприємства

### **1.5.1 Управління складською діяльністю:**

- **Приймання товарів на склад:**

Приймання товарів включає їхню реєстрацію та перевірку якості. Мета полягає в точному обліку та уникненні помилок у подальшій експлуатації.

- **Розміщення товарів на складі:**

Розміщення передбачає правильне розташування товарів враховуючи їхні особливості та оптимізацію просторових ресурсів.

- **Видача товарів зі складу:**

Видача товарів передбачає організований процес з урахуванням підготовки замовлень та документування. Мета полягає в ефективному управлінні та оптимізації всіх етапів складської діяльності для швидкого виконання замовлень та максимізації використання просторових ресурсів.

### **1.5.2 Оптимізація виробничих процесів:**

Управління виробничими процесами є ключовою функцією відділу логістики ТОВ "Чернівецький хлібокомбінат". Ця функція включає в себе планування та координацію операцій виробництва для досягнення оптимальної ефективності та раціонального використання ресурсів.

- **Планування виробничих операцій:**

Планування включає в себе розробку оптимального графіка виробничих операцій, визначення необхідної кількості робочих годин, матеріалів та обладнання для забезпечення плавного виробництва. Мета полягає в забезпеченні ефективного використання виробничих ресурсів, зниженні часу виробництва та підвищенні продуктивності.

- **Координація технологічних процесів:**

Координація технологічних процесів включає визначення оптимальних технологічних рішень для виробництва продукції. Основна мета полягає в підтримці безперебійності технологічного процесу, максимізації використання обладнання та досягненні високої якості продукції.

### **1.5.3 Управління транспортними ланками:**

**Маршрутизація та планування транспорту:**

Маршрутизація та планування транспорту включає в себе розробку оптимальних маршрутів для перевезення сировини та готової продукції. Основна мета цієї функції - мінімізація часу та витрат на транспортування, зниження ризиків затримок і витрат.

**Моніторинг та відстеження вантажів:**

Моніторинг та відстеження вантажів передбачає систематичне спостереження за рухом та розташуванням вантажів під час транспортування. Мета полягає в забезпеченні точності та своєчасності доставки, а також уникненні втрат та зниженні ризиків.

### **1.5.4 Інвентаризація та управління запасами:**

- **Планування запасів:**

Планування запасів включає в себе аналіз попиту на продукцію, визначення оптимальних рівнів запасів та розробку стратегій управління запасами. Основна мета цієї функції - забезпечення належного рівня запасів для забезпечення неперервного виробництва та задоволення потреб споживачів.

- **Ведення обліку запасів:**

Ведення обліку запасів передбачає систематичний облік руху та кількості наявних матеріалів і готової продукції. Мета - забезпечення точного контролю за станом запасів, уникнення дефіцитів та зайвих витрат.

- **Оптимізація обігу запасів:**

Оптимізація обігу запасів включає в себе розробку стратегій для зменшення часу, необхідного для обертання запасів. Мета - максимізація ефективності управління запасами та зниження витрат на їх утримання.

### **1.5.5 Оптимізація ланцюга постачання:**

- **Співпраця з постачальниками:**

Співпраця з постачальниками передбачає активну взаємодію та обмін інформацією зі сторонніми постачальниками сировини та матеріалів. Мета цієї функції - встановлення стійких та ефективних відносин з постачальниками для забезпечення надійного постачання необхідних ресурсів.

- **Управління взаємовідносинами з клієнтами:**

Управління взаємовідносинами з клієнтами включає в себе аналіз їхніх потреб, взаємодію та вирішення можливих проблем. Мета - підтримання високого рівня задоволеності клієнтів та забезпечення їхньої лояльності.

- **Впровадження технологій для оптимізації ланцюга постачання:**

Впровадження технологій передбачає використання сучасних ІТ-рішень для автоматизації та оптимізації управління ланцюгом постачання. Мета - покращення ефективності та прозорості процесів у всьому ланцюгу постачання.

### **1.6 Аналіз існуючих аналогів розробки:**

Перед початком розробки інформаційної системи для відділу логістики ТОВ "Чернівецький хлібокомбінат" важливо провести аналіз існуючих аналогів та альтернативних рішень на ринку. Це дозволить отримати уявлення про те, які можливості та функціонал вже присутні в схожих продуктах, а також визначити їхні переваги та недоліки.

Актуальний ринок програмних продуктів пропонує різноманітні інформаційні системи, призначені для управління логістичними процесами на підприємствах різних розмірів. До аналогів розроблюваної системи входять такі продукти, як SAP Integrated Business Planning, Oracle SCM Cloud, Microsoft Dynamics 365 Supply Chain Management.

#### **1.6.1 SAP Integrated Business Planning:**



«Рисунок 1.2» - SAP IBP

SAP IBP є інтегрованою системою для планування бізнес-процесів, включаючи логістику. Вона володіє широким функціоналом для управління

запасами, плануванням виробництва та оптимізацією логістичних маршрутів. Однак вона може бути складною у впровадженні та вимагати значних витрат.

### **1.6.2 Oracle SCM Cloud:**

Oracle SCM Cloud пропонує обширний спектр інструментів для управління ланцюгом постачання та логістичними процесами. Вона відома своєю інтеграцією з іншими платформами Oracle та великою кількістю модулів. Проте, вартість реалізації та підтримки може бути високою.



«Рисунок 1.3» - Oracle SCM Cloud

### **1.6.3 Microsoft Dynamics 365 Supply Chain Management:**



«Рисунок 1.4» - Microsoft Dynamics 365 Supply Chain Management

Microsoft Dynamics 365 SCM пропонує рішення для управління всім ланцюгом постачання та логістичними виробничими процесами. Вона вигідна для підприємств, які вже використовують інші продукти Microsoft. Проте, вона може вимагати додаткових налаштувань для повної відповідності потребам конкретного підприємства

#### **1.6.4 Аналіз альтернативних рішень:**

У додаток до великих корпоративних систем існують менші, але гнучкі альтернативи, такі як Open Source ERP (Enterprise Resource Planning), які можуть бути придатні для середніх та невеликих підприємств.

ERPNext та Odoo – це відкриті системи управління ресурсами підприємства, які надають можливість відслідковування логістичних процесів, управління запасами та планування виробництва. Вони є безкоштовними та можуть бути пристосовані до конкретних потреб підприємства, але можуть вимагати більше ресурсів для розробки та налаштування.

Аналіз існуючих аналогів дозволяє визначити сильні та слабкі сторони конкурентів, а також врахувати найкращі практики при розробці інформаційної системи для логістики на Чернівецькому хлібокомбінаті.

#### **1.7 Обґрунтування доцільності проектування й розроблення інформаційної системи для відділу логістики:**

Розробка та впровадження інформаційної системи для відділу логістики ТОВ "Чернівецький хлібокомбінат" має ключове значення для підвищення ефективності та конкурентоспроможності підприємства. Обґрунтування цієї ініціативи базується на ряді факторів:

### 1. Вдосконалення оперативності:

Розробка інформаційної системи дозволить оптимізувати та автоматизувати рутинні логістичні процеси, що призведе до прискорення обробки замовлень, скорочення термінів доставки та збільшення оперативності взагалі.

### 2. Підвищення точності та надійності інформації:

Впровадження інформаційної системи сприятиме створенню єдиної цифрової платформи для обліку та обробки даних, що позбавить підприємство від ручного введення та обробки інформації. Це забезпечить високу точність та надійність даних.

### 3. Оптимізація управлінських рішень:

Інформаційна система надасть керівництву підприємства доступ до детальної аналітики та звітності, що полегшить процес ухвалення управлінських рішень та допоможе в ідентифікації та вирішенні можливих проблем.

### 4. Забезпечення конкурентних переваг:

Впровадження інформаційної системи дозволить підприємству підтримувати високий стандарт обслуговування клієнтів, а також ефективно конкурувати на ринку шляхом прискорення реакції на зміни та вдосконалення логістичних процесів.

### 5. Легка адаптація:

Однією з ключових переваг розробленої системи є можливість легкої адаптації до специфічних потреб і особливостей підприємства. Це означає, що система може ефективно враховувати унікальні процеси логістики, що істотно полегшує її впровадження та використання.

### 6. Оптимізація вартості власності (ТСО):

Розроблена система вигідно виділяється своєю оптимізацією вартості власності (Total Cost of Ownership). Порівняно з великими корпоративними системами, які можуть вимагати значних інвестицій у ліцензії та обслуговування, розроблена система має низькі витрати на впровадження та підтримку.

### 7. Гнучкість та масштабованість:

Розроблена система пропонує гнучкість та масштабованість, які дозволяють змінювати та розширювати функціонал відповідно до зростання бізнесу. Це робить систему ефективною як для поточних, так і для майбутніх потреб підприємства.

### 8. Забезпечення конфіденційності:

Однією з переваг вибору розробленої системи є забезпечення конфіденційності даних. Оскільки підприємство має справу з чутливою інформацією про логістичні маршрути та запаси, безпека даних є пріоритетом.

### 9. Простота використання:

Розроблена система відрізняється простотою використання, що робить її доступною для всіх користувачів, включаючи менеджерів з логістики та водіїв. Це сприяє швидкому впровадженню та зменшує час, необхідний для навчання персоналу.

Обґрунтування проектування та розроблення інформаційної системи для відділу логістики є обов'язковим етапом перед початком реалізації проекту, і вищезазначені аспекти вказують на важливість та необхідність цього кроку для успішного функціонування підприємства.

## **1.8 Постановка задачі**

### **1.8.1 Призначення та цілі системи:**

Розроблена інформаційна система для відділу логістики ТОВ "Чернівецький хлібокомбінат" має наступні детально розглянуті призначення та цілі:

#### 1. Оптимізація Управління Запасами:

Система призначена для автоматизації та ефективного управління запасами сировини та готової продукції. Це включає в себе точне відстеження

кількості та місця розташування товарів на складах, автоматизацію замовлень та поповнення запасів для підтримки безперебійного виробництва.

## 2. Вдосконалення Маршрутизації та Відслідковування Транспорту:

Ціль системи - забезпечити точне відслідковування та оптимізацію маршрутів транспортних засобів. Система повинна надавати можливість менеджерам логістики в реальному часі відстежувати рух транспорту, визначати оптимальні маршрути та забезпечувати точну інформацію щодо часу доставки.

## 3. Забезпечення Надійності та Безпеки Даних:

Система ставить перед собою завдання забезпечити високий рівень безпеки даних, особливо конфіденційної інформації про логістичні маршрути та обіг товарів. Вона має використовувати сучасні технології шифрування та механізми аутентифікації для запобігання несанкціонованому доступу та збереження конфіденційності даних.

## 4. Високий Рівень Гнучкості та Адаптивності:

Система повинна бути гнучкою та легко адаптуватись до змін у бізнес-процесах та вимогах підприємства. Це означає, що вона повинна мати інструменти для швидкої модифікації та налаштування функціоналу, щоб відповідати новим вимогам та викликам.

## 5. Забезпечення Інтеграції з Існуючими Системами:

Система повинна бути проєктована так, щоб легко інтегруватись з існуючими інформаційними системами підприємства. Це дозволить плавно обмінювати даними між різними відділами та підрозділами підприємства, уникнути дублювання інформації та підвищити загальну ефективність управління.

Ці призначення та цілі системи є ключовими визначеннями, які визначають напрямок та функціонал розробленої інформаційної системи для відділу логістики ТОВ "Чернівецький хлібокомбінат".

### **1.8.2 Функції, які повинна виконувати система:**

1. Автоматизація обліку та контролю за запасами сировини та готової продукції.

Система повинна забезпечити автоматизацію процесів обліку запасів, включаючи приймання, відвантаження, переміщення та замовлення товарів. Це допомагає уникнути помилок в обліку та забезпечити точні та актуальні дані щодо наявності продукції на складі.

2. Відслідковування руху транспортних засобів та оптимізація маршрутів доставки.

Система повинна включати засоби відслідковування геолокації транспортних засобів, що використовуються для доставки. Це дозволяє в реальному часі визначати місцезнаходження транспорту, оптимізувати маршрути та планувати ефективну доставку.

3. Забезпечення ефективного управління інформацією про стан складів та транспорту.

Система повинна забезпечувати широкий доступ до інформації про запаси та транспорт, щоб різні відділи та працівники могли ефективно планувати та координувати свою роботу. Це допомагає уникнути зайвого часу та збільшити оперативність управлінських рішень.

Ці функції є ключовими для забезпечення ефективного управління логістичними процесами, відслідковування та контролю за запасами та оптимізації маршрутів доставки.

### **1.8.3 Вимоги до інтерфейсу користувача:**

Інтуїтивний та простий інтерфейс:

Інтерфейс користувача повинен бути інтуїтивно зрозумілим та легким для використання. Менеджерам логістики та іншим користувачам системи має бути

легко орієнтуватися в функціях системи, швидко здійснювати необхідні операції та отримувати доступ до важливої інформації.

Можливість персоналізації інтерфейсу:

Система повинна надавати можливість налаштування інтерфейсу відповідно до індивідуальних потреб користувачів. Це включає в себе можливість вибору та розташування важливих елементів, відображення необхідної інформації та зміну кольорової схеми.

Мобільний інтерфейс:

Враховуючи сучасні тенденції, система повинна мати адаптивний мобільний інтерфейс. Це дозволяє користувачам зручно користуватися системою на різних пристроях, таких як смартфони та планшети, що полегшує доступ до системи в будь-який час та в будь-якому місці.

Мультимовність:

Система повинна підтримувати мультимовність, щоб забезпечити зручність роботи для користувачів з різних регіонів та мовних груп. Можливість вибору мови інтерфейсу допомагає підвищити доступність та ефективність користувачів.

Ці вимоги до інтерфейсу користувача спрямовані на створення зручного та ефективного середовища для взаємодії з системою, полегшуючи використання та забезпечуючи високий рівень задоволення користувачів.

## **1.9 Концептуальна модель системи**

У даному розділі розглядається концептуальна модель інформаційної системи для відділу логістики ТОВ «Чернівецький хлібокомбінат». Ця модель визначає структурні компоненти та взаємодії між ними.

### **1.9.1 Користувачі системи**

Менеджери логістики:

Основні користувачі системи, які володіють правами на ведення обліку запасів, планування маршрутів та контролю за логістичними операціями через веб-клієнт.

Водії транспортних засобів:

Користувачі мобільного додатка, які мають можливість передавати геолокацію та отримувати завдання під час транспортування товарів.

Адміністратори системи:

Користувачі, які відповідають за адміністрування системи, керування правами користувачів та налаштування системи.

## 1.9.2 Функціональні модулі

### 1. Серверна частина:

Серверна частина інформаційної системи є ключовим компонентом, забезпечуючи надійне зберігання та обробку даних. Основною функцією є управління базою даних, що забезпечує ефективне зберігання, оновлення та вибірку інформації. Здатність оптимізувати запити та забезпечити консистентність даних є ключовою для високої продуктивності системи.

Особлива увага приділяється заходам безпеки даних, включаючи шифрування, автентифікацію користувачів та контроль доступу. Система забезпечує захист конфіденційності та цілісності інформації.

До функціональних можливостей серверної частини також входить моніторинг та журналювання. Засоби моніторингу дозволяють виявляти та вирішувати можливі проблеми, а журналювання подій допомагає в аналізі роботи системи для подальшого її вдосконалення.

Окремою ключовою характеристикою є реалізація REST API для взаємодії з іншими модулями системи, такими як веб-клієнт та мобільний додаток. Це створює стандартизований та ефективний механізм обміну даними між різними компонентами системи, що сприяє її надійній та спрощеній роботі.

### 2. Веб-клієнт:

Веб-клієнт є інтерфейсом для менеджера логістики та використовується для ведення обліку запасів та відстеження логістичних маршрутів. Основною метою веб-клієнта є створення інтуїтивного інтерфейсу користувача, забезпечуючи легкість навігації та консистентність дизайну.

Функціонал обліку запасів дозволяє менеджеру проводити детальний облік товарів, включаючи їх додавання, редагування та видалення. Інформація може бути представлена за допомогою графіків та звітів для зручності аналізу.

Відстеження логістичних маршрутів в режимі реального часу стає можливим завдяки функціональності веб-клієнта. Це включає визначення місцезнаходження вантажів, прогнозування часу прибуття та оптимальне маршрутування з урахуванням поточного трафіку.

Забезпечення безпеки даних включає в себе використання шифрування та автентифікацію користувачів для збереження конфіденційності та цілісності інформації. Взаємодія з іншими компонентами системи реалізується через REST API, забезпечуючи ефективний обмін даними між веб-клієнтом та серверною частиною.

### 3. Мобільний додаток:

Мобільний додаток є важливою складовою інформаційної системи, спрямованою на оптимізацію роботи водіїв транспортних засобів. Однією з ключових можливостей додатку є інтерактивне відстеження геолокації, що дозволяє в реальному часі відслідковувати рух транспортних засобів та ефективно управляти маршрутами.

Додаток також забезпечує водіїв інформацією про завдання, включаючи призначені точки та деталі вантажів. Це сприяє ефективній організації та виконанню завдань, покращуючи загальну логістичну діяльність.

Окремим аспектом є система сповіщень та комунікації, яка забезпечує взаємодію між водіями та менеджерами через зручний обмін повідомленнями. Це важливий елемент для швидкого реагування на зміни у маршрутах чи інші важливі події.

Додаток приділяє особливу увагу питанням безпеки та конфіденційності. Використання сучасних технологій шифрування та автентифікації гарантує захист особистих даних та інформації про завдання від несанкціонованого доступу.

### **1.9.3 Взаємодія функціональних модулів**

#### **1. Обмін даними між серверною частиною та веб-клієнтом:**

Забезпечення ефективного обміну даними між серверною частиною та веб-клієнтом є ключовою складовою інформаційної системи. Для забезпечення цього обміну використовується REST API (Representational State Transfer Application Programming Interface).

REST API дозволяє веб-клієнту звертатися до серверної частини для отримання або надсилання даних за допомогою HTTP-запитів. Цей підхід гарантує легкість взаємодії між різними компонентами системи та їх швидку інтеграцію.

В процесі обміну даними веб-клієнт може отримувати інформацію про стан запасів, логістичні маршрути та інші важливі дані. Крім того, відбувається передача даних, які стосуються змін у вантажах чи інших аспектів логістичних операцій.

REST API є стандартом для реалізації взаємодії між серверною частиною та веб-клієнтом, що забезпечує простоту, гнучкість та масштабованість системи.

#### **Передача геолокації від мобільного додатка до серверної частини:**

Цей функціональний модуль визначається передачею геолокаційних даних від мобільного додатка, використовуючи вбудовані механізми навігації мобільних пристроїв. Метою даної операції є надання можливості в реальному часі відслідковувати місцезнаходження транспортних засобів, які обладнані мобільним додатком.

Коли водій розпочинає роботу та активує мобільний додаток, він автоматично розпочинає передачу геолокаційних даних на серверну частину системи. Цей процес забезпечує постійне оновлення інформації про розташування транспортних засобів у режимі реального часу.

Передача геолокації використовується для ведення точного обліку маршрутів, моніторингу транспортних засобів та планування оптимальних логістичних маршрутів. За допомогою цього функціоналу система надає користувачам можливість ефективно взаємодіяти та відслідковувати рух транспортних засобів у реальному часі.

### **1.10 Висновок до першого розділу**

У цьому розділі проведений глибокий аналіз ТОВ «Чернівецький хлібокомбінат», включаючи опис його виробничих можливостей, споживчого ринку та корпоративної філософії. Виявлено, що підприємство є великим гравцем на ринку хлібопродуктів та має розвинену інфраструктуру.

Організаційна структура підприємства була ретельно розглянута, з визначенням ролі та взаємодії основних підрозділів, таких як виробничий підрозділ, логістичний відділ, відділ кадрів, фінансовий відділ та інші.

Логістичний відділ, як ключовий елемент дослідження, отримав розширений опис, включаючи його завдання, функції та взаємодію з іншими підрозділами підприємства. Виявлено, що ефективна робота логістичного відділу є критичною для оптимізації ланцюга постачання та забезпечення ефективності всіх виробничих процесів.

Дослідження та аналіз існуючої інформаційної системи на підприємстві дозволило визначити ключові аспекти, такі як стан апаратного та програмного забезпечення, а також питання безпеки даних.

На основі здобутих знань та виявлених потреб розроблено постановку задачі для проектування та розроблення інформаційної системи для логістичного відділу. Система має бути адаптованою до особливостей підприємства,

забезпечуючи ефективне управління логістичними процесами, відслідковування транспортних засобів та оптимізацію маршрутів.

В цілому, системний аналіз та постановка задачі надають необхідну основу для подальшого проектування та розроблення інформаційної системи, спрямованої на оптимізацію логістичних процесів та підвищення ефективності діяльності ТОВ «Чернівецький хлібокомбінат».

## **2 РОЗДІЛ ДОСЛІДЖЕННЯ ЛОГІСТИЧНИХ ІНФОРМАЦІЙНИХ СИСТЕМ**

### **2.1 Загальні відомості та завдання логістичних інформаційних систем**

З розвитком логістики з'явилась необхідність в ефективному управлінні всіма аспектами постачання, виробництва та розподілу товарів чи послуг. У цьому контексті виникає ключова роль логістичних інформаційних систем (ЛІС), які є необхідним інструментом для автоматизації та оптимізації логістичних процесів на підприємствах.

#### **2.1.1 Визначення логістичних інформаційних систем**

Логістичні інформаційні системи (ЛІС) представляють собою комплексні програмні засоби та технічні рішення, спрямовані на автоматизацію та оптимізацію логістичних процесів у сфері постачання, виробництва та розподілу товарів чи послуг. Вони інтегрують інформацію з усіх ланцюгів постачання, надаючи цілісний погляд на всі етапи товарообігу.

Логістичні інформаційні системи включають в себе різноманітні функції, які сприяють ефективному управлінню логістичними процесами. Серед найважливіших характеристик ЛІС можна виділити:

- **Інтеграція і автоматизація.** ЛІС забезпечують автоматизацію великої кількості операцій, починаючи від замовлення матеріалів та завершуючи доставкою готової продукції, що сприяє збільшенню ефективності та уникненню помилок.
- **Моніторинг та контроль.** ЛІС дозволяють в режимі реального часу відстежувати рух товарів, рівень запасів, терміни постачання та інші параметри, що дозволяє оперативно реагувати на зміни та уникати затримок.

- Оптимізація ланцюга постачання. ЛІС допомагають визначити оптимальні шляхи постачання, враховуючи різноманітні фактори, такі як вартість, терміни та якість.
- Аналіз даних та звітність. Системи надають можливість аналізу великої кількості даних, що допомагає в прийнятті обґрунтованих стратегічних рішень.
- Взаємодія з партнерами. ЛІС сприяють зміцненню взаємодії з постачальниками та клієнтами шляхом обміну інформацією та автоматизації спільних процесів.

Враховуючи ці аспекти, логістичні інформаційні системи стають ключовим інструментом для підприємств, що прагнуть оптимізувати свою логістичну діяльність та підвищити конкурентоспроможність на ринку.

### **2.1.2 Завдання логістичних інформаційних систем**

Логістичні інформаційні системи виконують ряд стратегічних та операційних завдань, спрямованих на підтримку та поліпшення логістичних процесів на підприємстві. Серед ключових завдань ЛІС варто визначити:

Оптимізація ланцюга постачання: ЛІС дозволяють ефективно управляти постачальниками, автоматизувати процеси замовлення та отримання матеріалів, що веде до зменшення часу на постачання та оптимізації витрат.

Інтеграція та автоматизація операцій: Ці системи спрощують рутинні операції, такі як складський облік, перевезення та відправлення товарів, шляхом автоматичного виконання багатьох завдань, що покращує ефективність робочих процесів.

Моніторинг та контроль запасів: ЛІС надають можливість в режимі реального часу відслідковувати рух товарів, визначати рівень запасів та прогнозувати потреби у матеріалах чи продукції.

Аналіз даних та оптимізація прийняття рішень: Системи допомагають управлінцям отримувати доступ до розширеного аналізу даних, що полегшує прийняття обґрунтованих стратегічних та тактичних рішень.

Забезпечення взаємодії з партнерами: ЛІС дозволяють покращити співпрацю з постачальниками та клієнтами через ефективний обмін інформацією та автоматизацію спільних процесів.

Забезпечення надійності та безпеки даних: ЛІС забезпечують високий рівень безпеки для конфіденційної інформації, уникаючи ризиків втрати даних чи несанкціонованого доступу.

### **2.1.3 Роль Логістичних Інформаційних Систем у Сучасному Бізнесі**

З розвитком технологій та зростанням конкуренції логістичні інформаційні системи (ЛІС) стають стратегічно важливим інструментом для сучасних підприємств. Їхня роль в бізнес-середовищі полягає в забезпеченні ефективного управління логістичними процесами та наданні конкурентної переваги.

#### **Стратегічне Розгортання:**

Логістичні інформаційні системи допомагають підприємствам стратегічно розгорнути свої логістичні зусилля. Вони надають можливість аналізувати ринкові тенденції, прогнозувати попит та оптимізувати ланцюжки постачання, що є ключовим для розвитку підприємства в умовах змінного середовища.

#### **Підвищення Клієнтської Задоволеності:**

ЛІС сприяють поліпшенню обслуговування клієнтів шляхом оптимізації доставки та вирішення логістичних завдань. Інтеграція засобів відстеження дозволяє забезпечити точні та своєчасні поставки, що призводить до позитивного впливу на задоволеність клієнтів.

#### **Оптимізація Ланцюжка Постачання:**

ЛІС дозволяють підприємствам ефективно керувати ланцюжком постачання, автоматизуючи процеси замовлення, постачання та виробництва. Це

призводить до зменшення запасів, оптимізації витрат та підвищення загальної ефективності.

Адаптація до Змін:

Логістичні інформаційні системи надають підприємствам гнучкість та швидкість в адаптації до змін в ринкових умовах. Вони дозволяють оперативно реагувати на зміни попиту, управляти ризиками та швидко впроваджувати нові стратегії.

Загальною метою використання ЛІС є створення стратегічної переваги для підприємства, що дозволяє ефективно конкурувати та забезпечувати сталість в умовах сучасного бізнес-середовища.

#### **2.1.4 Вплив Логістичних Інформаційних Систем на Зниження Витрат та Підвищення Ефективності**

Логістичні інформаційні системи (ЛІС) в сучасному бізнес-середовищі виявляються важливим інструментом для оптимізації логістичних процесів, зниження витрат та забезпечення стабільності підприємства. Їх вплив на ключові аспекти логістичної діяльності дозволяє значно поліпшити різні аспекти управління та забезпечити конкурентоспроможність на ринку.

Автоматизація та Оптимізація Процесів:

ЛІС надають можливість автоматизації багатьох логістичних процесів, що раніше вимагали значних ручних зусиль. Завдяки цьому підприємство може ефективно впоратися з рутинною роботою, що дозволяє використовувати робочий час більш ефективно.

Ефективне Управління Запасами:

ЛІС дозволяють в режимі реального часу відстежувати та управляти рівнем запасів. Забезпечуючи точну інформацію про наявні запаси, система допомагає уникнути перевитрат на утримання надмірних запасів та забезпечує надійне забезпечення виробництва та вчасну поставку товарів.

### Оптимізація Транспортних Витрат:

ЛІС дозволяють оптимізувати транспортні витрати шляхом ефективного маршрутизації та відстеження транспортних засобів. Це призводить до зменшення витрат на паливо, скорочення часу доставки та підвищення загальної продуктивності транспортного процесу.

### Інтеграція з Постачальниками та Партнерами:

ЛІС надають можливість підприємству ефективно взаємодіяти з постачальниками та партнерами через автоматизацію процесів замовлення, відстеження поставок та обмін інформацією. Це створює потужні логістичні мережі та сприяє швидкому реагуванню на зміни на ринку.

### Підвищення Якості Обслуговування:

ЛІС сприяють забезпеченню точності та швидкості обробки замовлень, що веде до підвищення якості обслуговування клієнтів. Можливість оперативного реагування на зміни сприяє задоволенню клієнтських потреб та утриманню високого рівня лояльності.

Таким чином, впровадження та ефективне використання ЛІС стає стратегічним кроком для підприємства, що прагне досягти оптимального управління логістичними процесами та досягти стабільності у конкурентному бізнес-середовищі.

## **2.1.6 Критичний Аналіз Логістичних Інформаційних Систем**

Критичний аналіз логістичних інформаційних систем (ЛІС) є важливою складовою для розуміння їхньої ефективності та визначення можливих вдосконалень. У рамках цього аналізу розглядаються ключові аспекти, які визначають якість та придатність системи до вирішення завдань логістичного управління.

### Функціональність та Надійність:

Першочерговим критерієм є функціональність ЛІС та її надійність. Система повинна безперебійно виконувати всі завдання, передбачені для

оптимізації логістичних процесів. Надійність гарантує стабільну та ефективну роботу системи навіть при великому обсязі операцій.

**Інтеграція та Сумісність:**

Ефективність ЛІС також залежить від її можливості інтеграції з іншими інформаційними системами, що використовуються на підприємстві. Сумісність з існуючими технологічними рішеннями гарантує плавний обмін даними та єдину інформаційну екосистему.

**Витрати та Оптимізація Ресурсів:**

Інвестиції у ЛІС повинні виправдовувати себе шляхом економії ресурсів та оптимізації логістичних витрат. Аналіз витрат на впровадження та утримання системи порівнюється із збільшенням ефективності та скороченням витрат.

**Складність та Ергономіка:**

Зручність використання системи впливає на її ефективність. Мінімізація складності управління та інтуїтивно зрозумілий інтерфейс забезпечують швидке навчання персоналу та уникнення помилок.

**Системи Захисту та Кібербезпека:**

Оскільки ЛІС містять велику кількість конфіденційної інформації, система захисту та кібербезпеки грають критичну роль. Вони повинні гарантувати конфіденційність, цілісність та доступність даних.

**Реалізація Інноваційних Рішень:**

Кращі ЛІС впроваджують інноваційні технології, такі як штучний інтелект, аналіз даних, та інші. Їхня ефективність визначається здатністю впроваджувати сучасні рішення та відповідати вимогам ринку.

Загальний критичний аналіз ЛІС визначає ступінь її відповідності вимогам логістичного управління та вказує на можливості для подальшого удосконалення.

## **2.2 Сучасні технології в управлінні транспортною логістикою**

Управління транспортною логістикою визнається ключовою складовою сучасної організації логістичних процесів, охоплюючи планування, організацію, контроль та управління транспортними ресурсами. Забезпечення координації всіх етапів логістичного ланцюжка, від закупівлі та виробництва до доставки та складування, є основним завданням управління транспортною логістикою.

Мета цього управління полягає в зниженні витрат на логістику та підвищенні задоволеності клієнтів. Досягнення цієї мети передбачає оптимізацію використання ресурсів, таких як транспортні засоби, трудові ресурси, паливо та інші матеріальні ресурси. Ефективне управління транспортною логістикою включає своєчасну та надійну доставку товарів чи послуг, оптимізацію маршрутів доставки, зменшення часу на перевантаження та вивантаження вантажу, раціональне використання транспортних засобів та ресурсів.

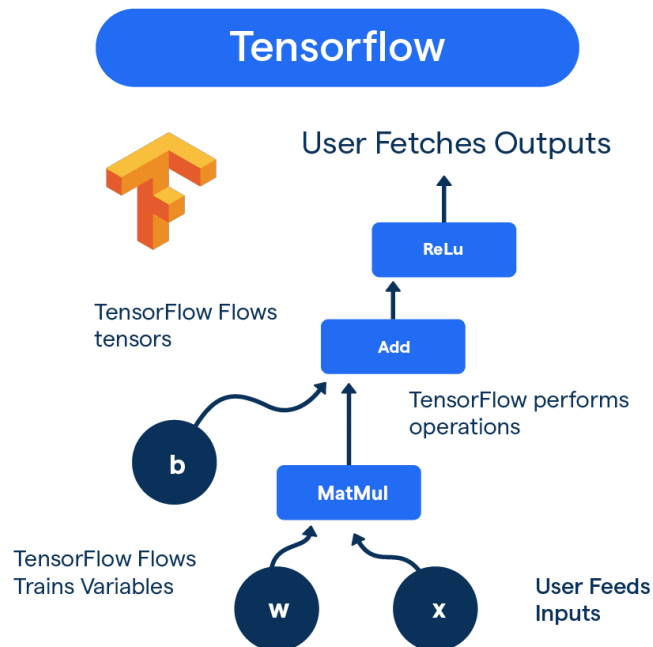
Одним із ключових аспектів ефективного управління транспортною логістикою є автоматизація операцій, ідентифікація потенційних ризиків та прийняття заходів для їх управління та мінімізації. Це допомагає підвищити продуктивність та ефективність роботи, покращити задоволеність клієнтів та знизити витрати на транспортування, складське утримання, операційні витрати та інші пов'язані з логістикою витрати.

Узагальнюючи, управління транспортною логістикою є важливим інструментом для оптимізації логістичних процесів, що сприяє підвищенню продуктивності та ефективності підприємства, забезпечує конкурентоспроможність та репутацію на ринку.

### **2.2.1 Системи машинного навчання в управлінні транспортною логістикою**

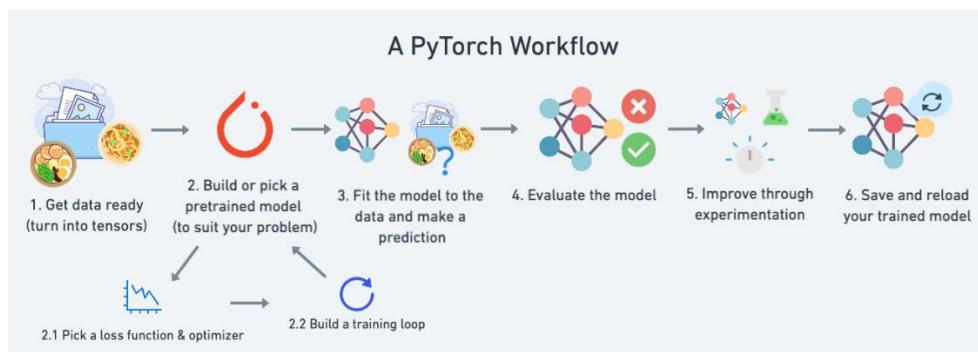
У сучасному управлінні транспортною логістикою системи машинного навчання виявляються невід'ємною складовою для досягнення оптимальності та ефективності логістичних процесів. Декілька ключових систем машинного навчання відзначаються своїми основними властивостями та призначенням:

- TensorFlow: Це відкрите програмне забезпечення для розробки та виконання алгоритмів машинного навчання. Використовується для прогнозування попиту на транспортні послуги, оптимізації маршрутів та аналізу вартості та прибутковості.



«Рисунок 2.1» - TensorFlow

- PyTorch: Це фреймворк глибокого навчання, який активно використовується для аналізу поведінки пасажирів, обробки зображень та відео, а також для управління транспортними потоками.



«Рисунок 2.2» - PyTorch

- Scikit-learn: Ця бібліотека машинного навчання має широкий спектр застосувань, включаючи управління ризиками та аналіз даних у логістичних системах.
- Keras: Використовується для розробки моделей глибокого навчання, зокрема для прогнозування та оптимізації логістичних процесів.

Ці системи машинного навчання забезпечують прогресивні технологічні рішення для вирішення конкретних завдань у сфері транспортної логістики, таких як прогнозування попиту, оптимізація маршрутів, управління ризиками та багато інших. Застосування цих систем дозволяє автоматизувати та покращити рішення у сфері транспортної логістики, забезпечуючи оптимальні та інтелектуальні підходи до управління логістичними викликами.

### 2.2.2 Нейронні мережі в управлінні транспортною логістикою

У сучасному управлінні транспортною логістикою нейронні мережі знаходять широке застосування, допомагаючи вирішувати різноманітні завдання. Деякі з ключових типів нейронних мереж, які використовуються в цьому контексті, включають:

- (RNN): Ці мережі призначені для роботи з послідовними даними та можуть використовуватися для прогнозування попиту на транспортні послуги, аналізу поведінки пасажирів та визначення оптимальних маршрутів.
- Згорткові нейронні мережі (CNN): CNN використовуються для обробки зображень та відео. В контексті транспортної логістики, вони можуть застосовуватися для аналізу даних датчиків та камер, визначення порушень та оптимізації системи обслуговування.
- Мережі з довготривалою пам'яттю (LSTM): LSTM забезпечують обробку послідовних даних з врахуванням довготривалої залежності. Вони можуть використовуватися для прогнозування затримок, аналізу кліматичних умов та визначення оптимальних місць для зупинок.

- Рекурентні згорткові нейронні мережі (RCNN): Цей тип мережі комбінує переваги RNN та CNN і може застосовуватися для прогнозування часу прибуття транспорту та аналізу даних з сенсорів.
- Мережі прямого поширення (Feedforward Neural Networks, FNN): Вони використовуються для широкого спектру завдань, таких як визначення оптимальних маршрутів та прогнозування аварійних ситуацій.
- Мережі з асоціативною пам'яттю: Цей тип мережі може бути застосований для оптимізації системи обслуговування та прогнозування часу прибуття транспорту.

Ці нейронні мережі дозволяють вдосконалити управління транспортною логістикою, забезпечуючи точні прогнози та оптимальні рішення в різних аспектах логістичних операцій.

### **2.2.3 Системи геолокації в управлінні транспортною логістикою**

В сучасному управлінні транспортною логістикою системи геолокації, такі як GPS (Global Positioning System), Beidou Navigation Satellite System та Galileo, грають важливу та невід'ємну роль у забезпеченні ефективності та точності логістичних процесів. Ці системи забезпечують надійне відстеження місцезнаходження транспортних засобів, відкриваючи безліч можливостей для оптимізації управління автопарком та розвитку логістичних стратегій.

Перш за все, відстеження місцезнаходження транспортних засобів надає можливість в реальному часі контролювати рух автопарку. Це особливо важливо для уникнення можливих затримок, планування оптимальних маршрутів та забезпечення швидкої доставки. Використання GPS дозволяє ефективно враховувати умови руху та пробки, що в свою чергу сприяє розробці стратегій доставки з урахуванням реальних дорожніх умов.

Додатково, системи геолокації визначають оптимальні маршрути для транспортних засобів, що сприяє підвищенню ефективності та скороченню часу

доставки. Це особливо важливо в умовах високої конкуренції та зростаючих вимог споживачів до швидкості обслуговування.

Такі системи виявляються корисними не лише для визначення місцезнаходження транспортних засобів, але й для вирішення питань, пов'язаних з аналізом кліматичних умов, визначенням оптимальних місць для зупинок та прогнозуванням часу прибуття транспорту. Це надає операторам логістичних систем інструменти для більш точного та гнучкого управління транспортними потоками.

Отже, використання систем геолокації в сучасному управлінні транспортною логістикою є ключовим фактором для досягнення високої ефективності, точності та конкурентоспроможності в логістичній галузі.

#### **2.2.4 Системи прогнозування трафіку в управлінні транспортною логістикою**

У сучасному управлінні транспортною логістикою системи прогнозування трафіку виявляються не тільки невід'ємною частиною оптимізації транспортних процесів, але й ключовим інструментом для підвищення загальної продуктивності та конкурентоспроможності логістичних підприємств.

Зазначені системи, такі як Google Maps, Waze, INRIX, TomTom Traffic та HERE Traffic, пропонують комплексний підхід до управління транспортною логістикою. Вони не лише надають користувачам інформацію про поточний стан дорожнього руху, а й враховують різноманітні фактори, які впливають на транспортні процеси.

Однією з ключових переваг є здатність систем прогнозування трафіку аналізувати великі обсяги даних в реальному часі. Це дозволяє операторам логістичних систем оперативно реагувати на зміни в умовах руху, швидко перепланувати маршрути та уникати можливих заторів.

Крім того, вони ефективно планують маршрути, враховуючи різні параметри, такі як інтенсивність трафіку, погодні умови та робота доріг. Це допомагає зменшити час доставки та підвищити точність прибуття вантажу.

Також важливою функцією є можливість систем передбачати та уникати транспортних пригод, ремонтів чи інших факторів, що можуть вплинути на рух транспорту. Це сприяє підтриманню стабільності та надійності логістичних процесів.

У величезному обсязі даних, які надають ці системи, також закладена можливість аналізу та прогнозування тенденцій у дорожньому русі, що дозволяє логістичним підприємствам адаптувати свої стратегії та процеси до змін на дорогах.

Отже, системи прогнозування трафіку відіграють критичну роль у досягненні логістичної ефективності, забезпечуючи швидку, надійну та оптимальну доставку вантажів у сучасних умовах навантаженого дорожнього руху.

### **2.2.5 Висновки до підрозділу**

У розділі 2.2 "Сучасні технології в управлінні транспортною логістикою" детально розглянуті ключові аспекти використання передових технологій у логістиці та транспортному управлінні. Аналізуючи системи машинного навчання, нейронні мережі, системи геолокації та системи прогнозування трафіку, можна визначити їхню важливу роль у оптимізації транспортних процесів та підвищенні ефективності логістичних систем.

Системи машинного навчання, такі як TensorFlow, PyTorch та інші, виявляються ефективним інструментом для прогнозування попиту на транспортні послуги, оптимізації маршрутів, управління транспортними потоками та аналізу вартості та прибутковості. Їхні алгоритми можуть

адаптуватися до різноманітних умов, забезпечуючи гнучкість та точність в прийнятті рішень.

Нейронні мережі, такі як рекурентні, згорткові та глибокі мережі, знаходять широке застосування в управлінні транспортом та логістикою. Вони допомагають вирішувати завдання визначення оптимальних маршрутів, прогнозування попиту, аналізу поведінки пасажирів та багато інших, сприяючи підвищенню точності та ефективності.

Системи геолокації, зокрема GPS, Beidou та Galileo, є невід'ємною частиною управління транспортними засобами. Вони дозволяють відстежувати місцезнаходження транспортних засобів, визначати оптимальні маршрути та уникати заторів, що є важливим для швидкої та надійної доставки.

Системи прогнозування трафіку, представлені такими платформами, як Google Maps, Waze, INRIX та інші, забезпечують інформацію про трафік, планують маршрути та оптимізують проїзд, сприяючи зменшенню затримок та підвищенню швидкості перевезень.

Узагальнюючи, використання сучасних технологій в управлінні транспортною логістикою дозволяє підприємствам оптимізувати процеси, підвищувати ефективність та надійність транспортних послуг, що є важливим чинником у конкурентній сфері логістики.

## **2.3 Використання цифрових двійників у логістиці**

### **2.3.1 Визначення цифрового двійника**

Цифровий двійник у контексті логістики представляє собою віртуальну репліку фізичних об'єктів та процесів, яка здатна відображати їх стан, характеристики та взаємодію в реальному часі. Це поняття виникло на хвилі цифрової трансформації та інновацій в інформаційних технологіях.

Однією з ключових особливостей цифрового двійника є його здатність автоматично оновлюватися відповідно до змін в реальному світі, реагуючи на нові дані та події. Він враховує реальні умови та параметри, що дозволяє операторам систем логістики приймати обґрунтовані рішення та вживати заходів для оптимізації.

Цифрові двійники в логістиці можуть стосуватися різних об'єктів, таких як транспортні засоби, товари, складські приміщення чи навіть вантажні місця. Вони використовуються для створення повноцінної моделі логістичних ланцюгів, а також для визначення ефективності та точності проходження різних етапів логістичних процесів.

Цифровий двійник виявляється особливо корисним у сфері прогнозування та планування, оскільки він дозволяє аналізувати імовірні сценарії та реагувати на зміни в режимі реального часу. Його використання також розширює можливості моніторингу за станом транспортних засобів, виявлення можливих проблем та автоматизації процесів управління логістичним ланцюгом.



«Рисунок 2.3» - Позначення цифрового двійника

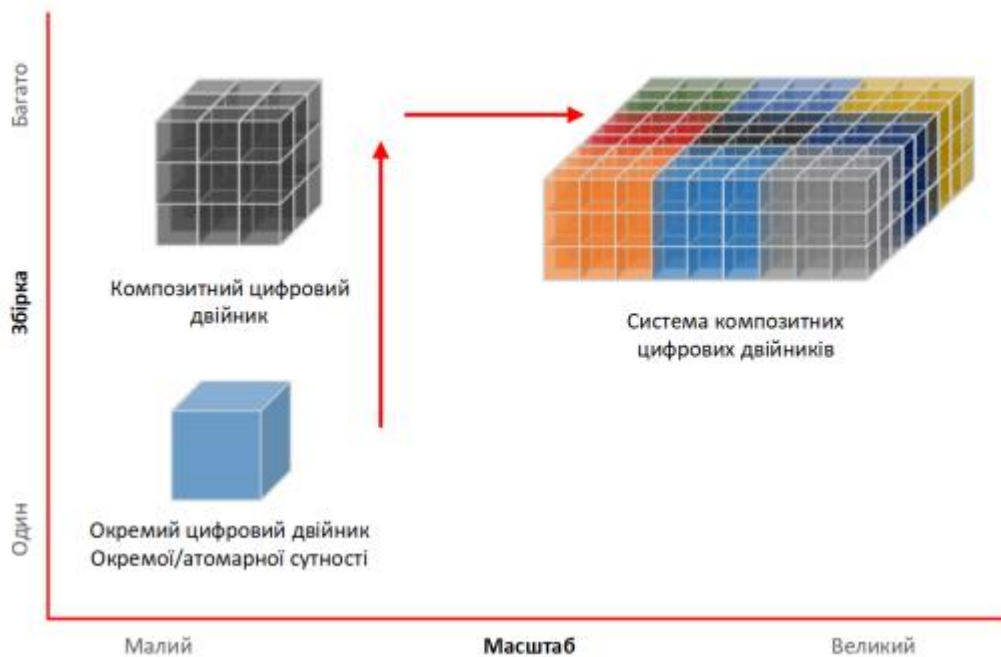
### **2.3.2 Зв'язки між цифровими двійниками в системах**

Цифрові двійники в системах логістики взаємодіють між собою, використовуючи технології збору та обробки даних. Вони створюють велику мережу взаємозв'язаних цифрових двійників, які сприяють уніфікації та оптимізації логістичних процесів.

У праці [15] виділяють такі рівні абстракції цифрового двійника залежить від умов використання, для яких він розробляється.

- Окремий цифровий двійник – це єдине ціле, що створює цінність без потреби подальшого розбиття. Наприклад, коробка передач чи двигун шарового млина в шахті на рівні моніторингу може розглядатися як єдиний об’єкт. Збірка дискретних цифрових двійників для утворення композитного цифрового двійника (рис. 2) показана у вигляді вертикального розширення, що описує збільшення складу від одиниці до множини сутностей.

- Складений (композитний) цифровий двійник – це комбінація окремих цифрових двійників, що представляє собою об’єкт, який складається з кількох окремих компонентів чи частин. Композиція може проходити на різних рівнях. Для прикладу, виробнича комірка – це складне утворення, чий цифровий двійник складається з цифрових двійників пристроїв у цій комірці.



«Рисунок 2.4» - Цифрові двійники

Згідно праці [15] у композитному цифровому двійнику зв’язки між елементами (рис. 3) можуть бути:

- ієрархічними: так само, як і їх реальні аналоги, набір цифрових двійників-компонентів об’єднується в цифрового двійника устаткування, кілька

цифрових двійників устаткування становлять цифрового двійника виробничої лінії, а набір цифрових двійників виробничих ліній може бути зібраний в цифрового двійника заводу і так далі.

- асоціативними: між цифровими двійниками, як і між їх реальними прототипами, є асоціації. Цифровий двійник газопроводу пов'язаний цифровими двійниками процесів видобування та споживання газу.
- одноранговими (peer-to-peer): зв'язки між рівноправними вузлами можна спостерігати в групі обладнання однакового чи схожого типів, що виконує такі ж чи подібні функції. Загальний ефект всього обладнання – це просто сума ефектів, що продукуються кожним елементом групи. Наприклад, вітрова електростанція, де вітрогенератори формують композитний цифровий двійник вітрової турбіни.



«Рисунок 2.5» - композиції цифрових двійників

### 2.3.3 Цифровий двійник в життєвому циклі об'єкту

У сучасних умовах інформація про об'єкт логістичного управління часто розподілена по різних інформаційних ресурсах, які належать і управляються різними організаціями. Ця децентралізація інформації призводить до розриву інформаційного потоку протягом усього життєвого циклу об'єкта. Зазначений

розрив виникає внаслідок того, що різні інформаційні ресурси можуть некоректно обмінюватися даними. Внаслідок цього деяка інформація може дублюватися або містити протиріччя, а інша може бути відсутньою.

Під час життєвого циклу об'єкта виникає необхідність у пошуку, взаємодії та аналізі різноманітної інформації. Зазначена проблема управління даними може витратити значні ресурси на пошук та агрегацію необхідних даних, а також створення семантичних зв'язків між ними. Потрібний час і зусилля можуть бути витрачені на приведення інформації до єдиного вигляду та забезпечення консистентності.

Додатково, подібні проблеми можуть призводити до конфліктів в оперативній інформації та сприяти ухваленню неефективних рішень. Важливо враховувати, що окремі інформаційні банки можуть ускладнювати впровадження передових методів, таких як розширена аналітика та штучний інтелект. Ці методи часто вимагають доступу до великих об'ємів інформації, а децентралізованість може ускладнювати їхню ефективну реалізацію.

Цифрові двійники виявляються дієвим інструментом для вирішення проблем децентралізованого розташування інформації в логістичному управлінні. Як проксі-системи, цифрові двійники централізовано збирають та об'єднують дані про кожний об'єкт, що сприяє подоланню розкидання інформації по різних інформаційних ресурсах. Це важливо для створення єдиного, консолідованого джерела даних, яке стає доступним різним сферам бізнесу через інтеграційні інтерфейси, такі як API.

Централізоване збирання та представлення інформації цифровим двійником впливає на процес прийняття рішень в логістичному управлінні. Завдяки загальному розумінню робочого стану об'єктів, бізнес-сфери можуть узгоджено пристосовувати свої стратегії та вирішувати завдання, пов'язані з логістикою. Це сприяє підвищенню ефективності та точності прийняття рішень, оскільки сторони мають доступ до об'єктивних та актуальних даних.

Окрім цього, цифрові двійники відіграють важливу роль у зниженні загальної вартості життєвого циклу експлуатації та обслуговування об'єктів. Це

досягається завдяки оптимізації процесів, що ґрунтується на доступі до консолідованих та точних даних. Цифрові двійники можуть служити фундаментом для впровадження передових технологій, зокрема розширеної аналітики та штучного інтелекту, сприяючи їхньому успішному впровадженню в управління транспортною логістикою.

## **2.4 Висновки до другого розділу**

Даний розділ висвітлив різноманітні аспекти сучасних технологій в управлінні транспортною логістикою, включаючи використання нейронних мереж, систем геолокації, систем прогнозування трафіку та цифрових двійників. Застосування цих технологій визначається необхідністю оптимізації та підвищення ефективності логістичних процесів.

Нейронні мережі використовуються для прогнозування попиту, оптимізації маршрутів, аналізу поведінки пасажирів та інших завдань, що полегшують управління транспортною логістикою. Системи геолокації надають точну інформацію про місцезнаходження транспортних засобів, що важливо для визначення оптимальних маршрутів та відстеження ситуації на дорогах. Системи прогнозування трафіку, у свою чергу, допомагають планувати маршрути, оптимізувати проїзд та зменшувати затримки. Цифрові двійники вирішують проблему розкидання інформації та сприяють уніфікації даних, що спрощує прийняття рішень та знижує загальні витрати.

Створення додатку, розробленого під час магістерської роботи виявляється доречним у зв'язку з реальною потребою в покращенні логістичних процесів. Такий додаток може забезпечити збалансоване та оптимізоване управління транспортною логістикою, забезпечуючи точні дані, швидкі рішення та підвищену ефективність логістичного ланцюга.

## **3 РОЗДІЛ РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВІДДІЛУ ЛОГІСТИКИ ТОВ «ЧЕРНІВЕЦЬКИЙ ХЛБОКОМБІНАТ»**

### **3.1 Вибір та опис засобів розробки**

Вибір засобів розробки є важливим етапом створення інформаційної системи для відділу логістики. У рамках цього процесу було уважно вибрано та обґрунтовано ряд інструментів, що оптимально відповідали вимогам та завданням проекту. Мова програмування C# була обрана за її високою продуктивністю та широким спектром можливостей. Фреймворк ASP.Net Core використовувався для створення серверної частини системи, забезпечуючи гнучкість у відповіді на зміни у вимогах експлуатації. REST API використовується для забезпечення ефективного обміну даними між різними компонентами системи, а MS SQL Server гарантує надійність та стабільність роботи бази даних. Використання Visual Studio, інтегрованого середовища розробки, сприяє зручному та ефективному процесу написання коду. Для розробки мобільного додатку обрано .Net Maui, що дозволяє забезпечити універсальність та легкість підтримки різних платформ. Використання JavaScript, ReactJS та Bootstrap було використано для створення веб платформи, та сприяло створенню сучасного та інтуїтивно зрозумілого інтерфейсу користувача. Загалом, цей набір засобів забезпечив ефективну та надійну розробку інформаційної системи для відділу логістики, відповідно до поставлених завдань.

#### **Розділ 3.1.1 Мова програмування C#**

Вибір мови програмування – стратегічний аспект розробки системи, і в даному випадку, обрана мова програмування C# виявилася оптимальним

рішенням. C# (C-Sharp) є потужним та сучасним об'єктно-орієнтованим мовою програмування, розробленою Microsoft. Її висока продуктивність, ефективність та широкий функціонал дозволяють ефективно реалізовувати різноманітні завдання, включаючи створення складних інформаційних систем.



«Рисунок 3.1» - C#

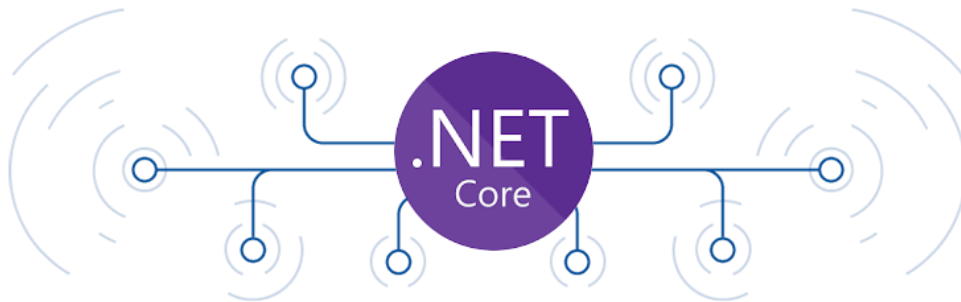
Мова програмування C# була розроблена компанією Microsoft та представлена в 2000 році як частина ініціативи .NET Framework. Її створення велося під керівництвом Андерса Гейлсберга (Anders Hejlsberg) – видатного програміста, який вже мав досвід розробки Turbo Pascal і Delphi.

C# була створена з метою об'єднати в собі простоту та ефективність мов програмування, таких як C++ та Java, а також для покращення розробки програмних засобів для платформи Windows.

Однією з ключових переваг C# є його інтеграція з .NET-екосистемою, що сприяє легкому з'єднанню з іншими технологіями та бібліотеками. Ця мова підтримує розробку великих проєктів завдяки чіткому синтаксису та сучасним можливостям, таким як асинхронне програмування.

.NET Core – це високопродуктивний та відкритий фреймворк для розробки крос-платформених додатків, який визначає новий етап в еволюції екосистеми .NET від компанії Microsoft. Започаткований у 2016 році, .NET Core привніс значні покращення у порівнянні з попередніми версіями .NET Framework. Однією з ключових переваг .NET Core є його крос-платформеність, що дозволяє

розробникам створювати додатки, які можуть запускатися на різних операційних системах, включаючи Windows, Linux та macOS.



«Рисунок 3.2» - .Net Core

Фреймворк має відкритий вихідний код, що сприяє активній участі спільноти та розвитку екосистеми. Він також відзначається високою продуктивністю, ефективним використанням ресурсів та оптимізацією швидкості виконання додатків. .NET Core підтримує різноманітні сценарії розробки, включаючи веб-застосунки, мікросервіси, хмарні додатки та багато інших.

Використання C# також забезпечує високу безпеку коду, включаючи перевірку типів та автоматичне управління пам'яттю. Це особливо важливо при розробці критичних застосунків, де надійність та стабільність грають вирішальну роль.

Узагальнюючи, обрання мови програмування C# дозволяє ефективно втілити функціонал інформаційної системи, забезпечуючи високу продуктивність та легку підтримку у майбутньому.

### **Розділ 3.1.2 Фреймворк ASP.NET Core**

ASP.NET Core – це сучасний та потужний фреймворк для розробки веб-додатків, що є частиною екосистеми .NET Core. Цей фреймворк визначається високою продуктивністю, відкритістю та широким спектром можливостей для розробників.



«Рисунок 3.3» - ASP .Net Core

ASP.NET Core, заснований у 2016 році, є сучасним фреймворком для веб-розробки, що виник у контексті стрімкого розвитку та потреби в більш гнучкому та ефективному інструменті. Забезпечуючи крос-платформеність та акцентуючи відкритість, він еволюціонував з попередніх версій ASP.NET, ставши ключовою технологією для швидкозмінного світу розробки веб-застосунків. ASP.NET Core визначається історією відкритості, інновацій та стрімкого розвитку, роблячи його важливим інструментом для створення сучасних та надійних веб-додатків.

Однією з ключових переваг ASP.NET Core є його кросс-платформенність, що дозволяє створювати веб-застосунки, які можуть бути запуснені на різних операційних системах, включаючи Windows, Linux та macOS. Це робить фреймворк універсальним і відкриває можливості для розгортання додатків в різних хмарних середовищах.

ASP.NET Core надає високий рівень абстракції для роботи з HTTP-протоколом та дозволяє створювати ефективні та масштабовані веб-додатки. Він підтримує модель роботи "Middleware", яка дозволяє додавати та налаштовувати компоненти в обробці HTTP-запитів, що робить його гнучким та розширюваним.

ASP.NET Core володіє вбудованим механізмом впровадження залежностей, шаблонами для розробки, аутентифікацією та авторизацією, а також підтримкою технологій SignalR для реального часу взаємодії. Цей фреймворк є невід'ємною частиною нашої інформаційної системи, забезпечуючи надійну та ефективну основу для веб-клієнта нашого додатку.

### Розділ 3.1.3 Microsoft SQL Server

Microsoft SQL Server (MS SQL Server) є потужною реляційною системою управління базами даних (RDBMS), розробленою компанією Microsoft, і має багатолітню історію, починаючи з 1990-х років. Започаткована як продукт для операційної системи OS/2, MS SQL Server швидко став ключовим елементом екосистеми Microsoft, еволюціонуючи в потужний та надійний інструмент для управління даними.



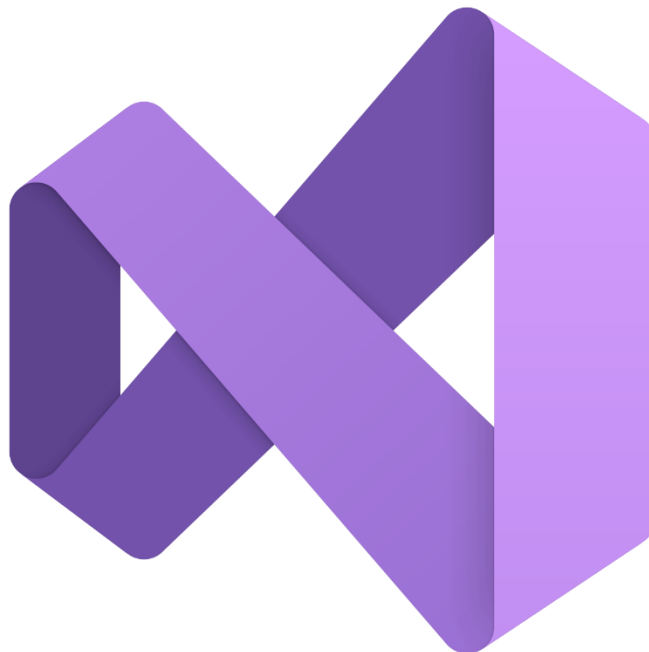
«Рисунок 3.4» - Microsoft SQL Server

MS SQL Server пропонує широкий спектр функціоналу для роботи з базами даних, включаючи розширену підтримку транзакцій, процедурного мовлення (T-SQL), можливості резервного копіювання даних та інші. Ця система використовує концепції реляційної моделі та забезпечує розширені оптимізаційні можливості для ефективної роботи з великими обсягами даних.

Інтеграція MS SQL Server з .NET Framework та .NET Core робить його оптимальним вибором для нашої інформаційної системи. Використання MS SQL Server гарантує стабільність, безпеку та ефективність у роботі з базою даних, надаючи критичний фундамент для надійності та продуктивності розроблюваної системи.

### Розділ 3.1.4 Visual Studio

Visual Studio, створене Microsoft, є вдосконаленим інтегрованим середовищем розробки (IDE), що визначається своєю потужністю та зручністю використання для розробки різноманітних програмних проектів. Заснована в 1997 році, платформа Visual Studio стала необхідним інструментом для програмістів та розробників у всьому світі.



«Рисунок 3.5» - Visual Studio

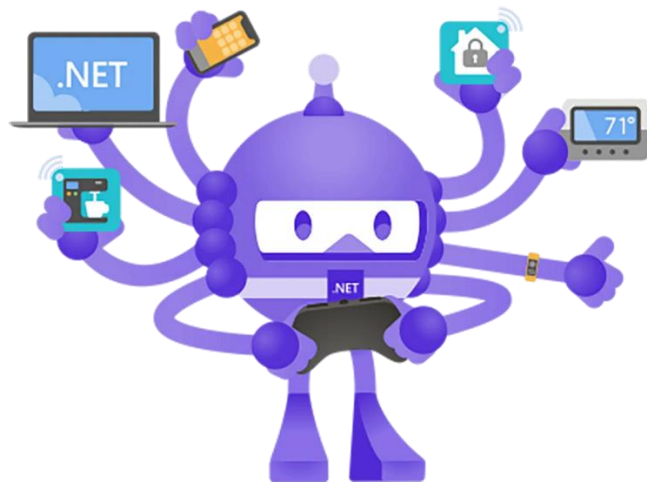
Visual Studio надає розширені засоби для написання, відлагодження та відстеження коду. Його інтеграція з мовами програмування, такими як C#, підтримує створення різноманітних застосунків, включаючи веб-додатки, мобільні додатки та програмне забезпечення для різних платформ.

Основні характеристики Visual Studio включають вбудований відлагоджувач, систему керування версіями, потужний дизайнер інтерфейсу користувача та інші інструменти, що сприяють швидкому та ефективному процесу розробки. Платформа також підтримується розширюваними

компонентами та додатковими розширеннями, що робить її ідеальним інструментом для створення розроблюваної системи. Використання Visual Studio вдосконалює процес розробки та сприяє ефективній реалізації наших програмних рішень.

### Розділ 3.1.5: .NET MAUI

.NET MAUI (Multi-platform App UI) представляє собою інноваційний фреймворк для розробки крос-платформених мобільних додатків в межах екосистеми .NET. Розроблений корпорацією Microsoft, цей фреймворк наслідує успіх фреймворка Xamarin і визначає новий стандарт у сфері мобільної розробки, спрощуючи створення додатків для операційних систем, таких як Android, iOS та Windows.



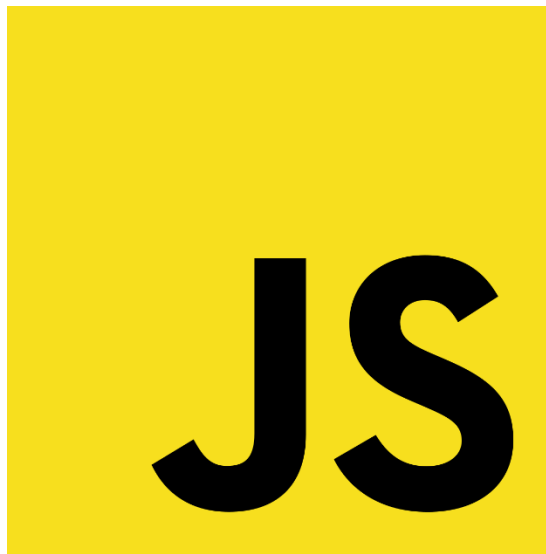
«Рисунок 3.6» - .NET MAUI

.NET MAUI відрізняється високою продуктивністю та зручністю використання. Він забезпечує одночасний доступ до нативних API та інструментів для створення сучасних інтерфейсів користувача, надаючи розробникам потужні засоби для творчості та ефективної реалізації ідей.

Важливою перевагою .NET MAUI є можливість спільного використання коду, що дозволяє писати бізнес-логіку та частину інтерфейсу лише один раз для всіх платформ. Це робить розробку та підтримку мобільних додатків ефективнішою та менш трудомісткою.

### Розділ 3.1.6 JavaScript

JavaScript визначається як критична складова для розробки веб-додатків і відіграє ключову роль у втіленні фронтенду нашої розроблюваної системи. Ця універсальна мова програмування була створена для додавання динамічного функціоналу до веб-сторінок та взаємодії з користувачем без перезавантаження сторінки.



«Рисунок 3.7» - JavaScript

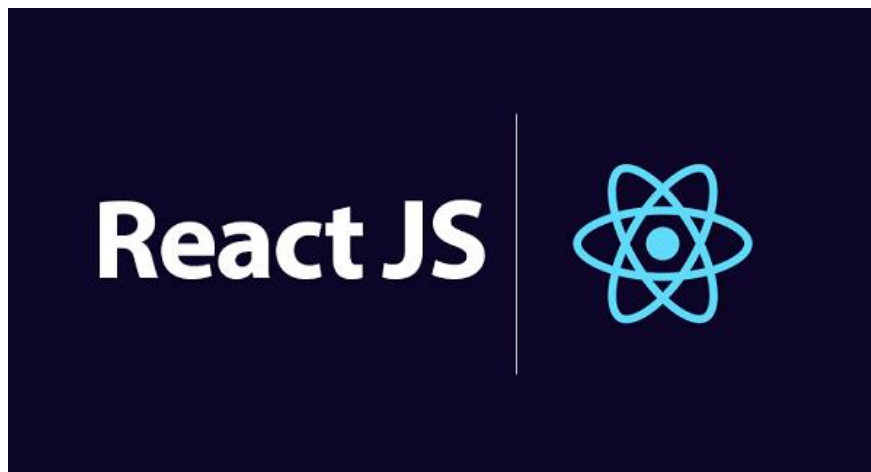
JavaScript, вперше представлений компанією Netscape Communications у 1995 році, швидко став ключовою мовою для реалізації динамічних елементів на веб-сторінках. Створений Бренданом Ейхом, мова була задумана як інструмент для надання можливості веб-розробникам створювати взаємодію та динаміку на сторінках, які до того часу були статичними.

JavaScript є стандартом для взаємодії з об'єктами на стороні клієнта веб-додатків. Забезпечуючи можливість створювати динамічні та інтерактивні

інтерфейси користувача, він стає невід'ємною частиною зручної та продуктивної роботи з системою.

JavaScript визнається за великий розрив та спрощення розробки завдяки багатьом фреймворкам та бібліотекам, таким як React, Angular та Vue.js. Цей вибір забезпечує високий стандарт та оптимальний користувацький досвід в розроблюваній системі, дозволяючи використовувати сучасні технології та забезпечити відмінну взаємодію з користувачем.

### Розділ 3.1.7 ReactJS



«Рисунок 3.8» - ReactJS

ReactJS, розроблений компанією Facebook, виступає невід'ємною складовою фронтенд-розробки, пропонуючи потужний та гнучкий інструментарій для створення інтерфейсів користувача веб-додатків. Запущений у 2013 році, React став однією з найпопулярніших бібліотек, завдяки своєму інноваційному підходу та високій продуктивності.

Унікальність React полягає в його компонентному підході, що дозволяє створювати невеликі та повторно використовувані компоненти, що спрощує розробку та підтримку коду. Використання Virtual DOM дозволяє оптимізувати

маніпуляції з реальним DOM, підвищуючи продуктивність додатків та забезпечуючи відмінний користувацький досвід.

### Розділ 3.1.8 Bootstrap

Bootstrap, розроблений Twitter, є потужним інструментарієм для швидкої та ефективної розробки веб-додатків та інтерфейсів. Запущений у 2011 році, цей фреймворк визначається своєю простотою використання та гнучкістю, надаючи розробникам широкий спектр засобів для створення стильних та адаптивних веб-сторінок.



«Рисунок 3.9» - Bootstrap

Однією з ключових переваг Bootstrap є його система сітки, яка дозволяє легко організувати контент на сторінці та робити її адаптивною до різних розмірів екранів. Крім того, він включає широкий набір готових до використання компонентів, таких як кнопки, форми, навігаційні панелі тощо, що полегшує створення структурованих та привабливих інтерфейсів.

Bootstrap також пропонує можливості кастомізації та темізації, що дозволяє адаптувати зовнішній вигляд додатка до корпоративного стилю або

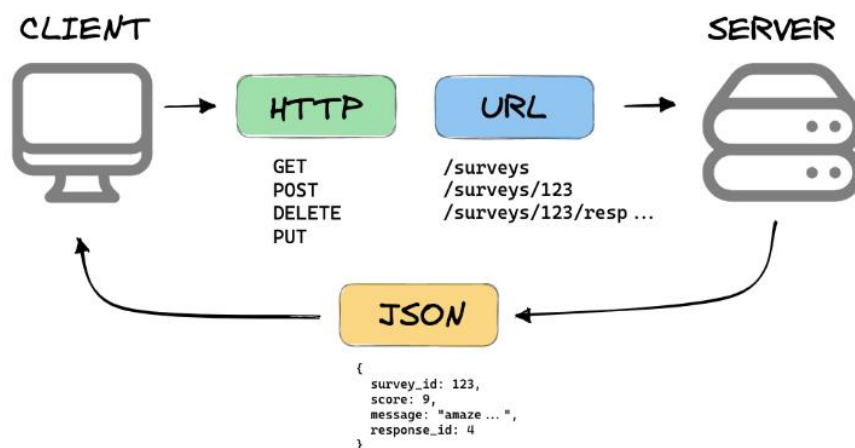
індивідуальних вимог. Із своєю широкою популярністю та активною спільнотою, Bootstrap стає ефективним інструментом розробки, роблячи його невід'ємною частиною розроблюваної системи.

## 3.2 Архітектура системи

### 3.2.1 Rest API

Для обміну інформацією між клієнтами і сервером було обрано архітектурний стиль REST API.

REST API (Representational State Transfer Application Programming Interface) – це архітектурний стиль для розробки веб-служб, який забезпечує взаємодію між різними компонентами програмного забезпечення через мережу.



«Рисунок 3.10» - Rest API

Основні принципи REST:

- Простота: REST прагне до максимальної простоти у взаємодії між системами, що полегшує розуміння та впровадження.
- Стандартизація: Використання стандартних HTTP-методів і URI спрощує комунікацію та робить її зрозумілою для розробників.
- Незалежність від стану: REST не зберігає стан на сервері між запитами від клієнта, що сприяє простоті та масштабованості.

У REST API кожен ресурс, такий як об'єкти бази даних чи послуги, ідентифікується унікальним URI (Uniform Resource Identifier), що надає чіткість та однозначність взаємодії.

Взаємодія з ресурсами відбувається за допомогою стандартних HTTP-методів:

- GET: Отримання інформації.
- POST: Створення нових ресурсів.
- PUT: Оновлення існуючих ресурсів.
- DELETE: Видалення ресурсів.

Відповіді від сервера, як правило, представлені у форматах JSON або XML, що спрощує їх обробку клієнтами та забезпечує їхню структурованість.

REST API використовується для побудови високопродуктивних та розподілених систем з великою кількістю клієнтів та ресурсів. Його ефективність полягає в розподіленому та легко масштабованому характері, що дозволяє системам зростати із збільшенням завдань та обсягів даних.

REST API широко використовується у сферах веб-розробки, мобільних додатках, хмарних сервісах та інших різноманітних інформаційних технологіях. Він дозволяє легко інтегрувати різні компоненти систем та забезпечує стандартизований підхід до взаємодії між ними.

Застосування REST API розповсюджене у сучасних сервісах, таких як соціальні мережі, онлайн-магазини, фінансові системи та багато інших. Його гнучкість та простота забезпечують швидку інтеграцію нових функціональностей та розвиток додатків без зайвих труднощів.

Зокрема, у світі мікросервісної архітектури REST API є ключовим компонентом, дозволяючи незалежно розвивати та масштабувати окремі сервіси великого застосунку. Отже, REST API визначається не лише своєю простотою та стандартами, але і високою ефективністю у різноманітних областях сучасних інформаційних технологій.

### 3.2.2 MVVM та .Net Maui

Оскільки у платформі .Net maui реалізовано механізм привязки, було обрано шаблон проектування MVVM.

Архітектурний паттерн MVVM (Model-View-ViewModel) є ефективним інструментом для розробки програмного забезпечення, особливо в графічних інтерфейсах користувача. Цей паттерн розділяє структуру програми на три основні компоненти: Model, View і ViewModel.

- Model

Model відповідає за управління даними і виконання бізнес-логіки. Це місце, де зберігаються дані та виконуються операції, що стосуються обробки цих даних.

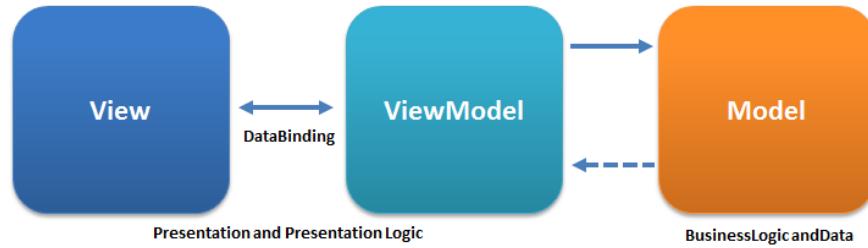
- View

View відображає графічний інтерфейс користувача і реагує на події, сприймаючи введення користувача. Вона відображає інформацію, яку отримує від ViewModel, і надсилає події, що виникають від користувача, назад до ViewModel.

- ViewModel

ViewModel виступає як посередник між Model і View. Вона обробляє дані з Model та підготовлює їх для відображення в View. ViewModel також реагує на введення користувача і взаємодіє з Model для забезпечення оновлення даних.

Однією з ключових особливостей MVVM є можливість двостороннього зв'язування даних між View і ViewModel. Це дозволяє автоматично оновлювати інтерфейс при зміні даних і навпаки, спрощуючи роботу з відображенням і зберіганням даних.



«Рисунок 3.11» - MVVM

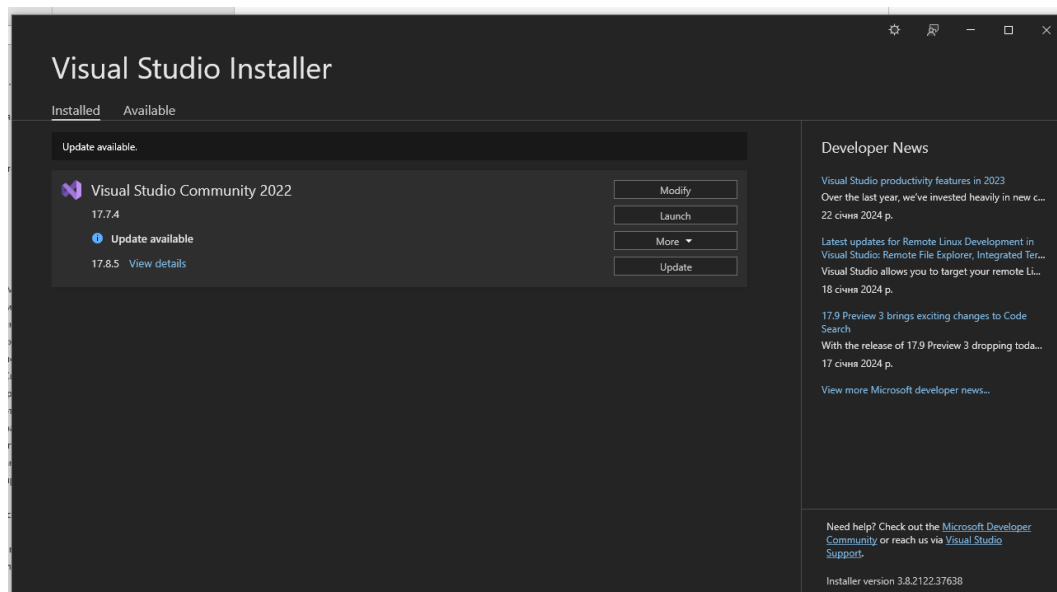
MVVM сприяє тестуванню, оскільки бізнес-логіка розміщена в ViewModel, що може бути протестована незалежно від інтерфейсу користувача. Це полегшує створення автоматизованих тестів для перевірки правильності роботи програми.

Завдяки зручності роботи з великими інтерфейсами та складною бізнес-логікою, MVVM став популярним шаблон для розробників програмного забезпечення, що працюють з графічними інтерфейсами користувача.

### 3.3 Розробка системи

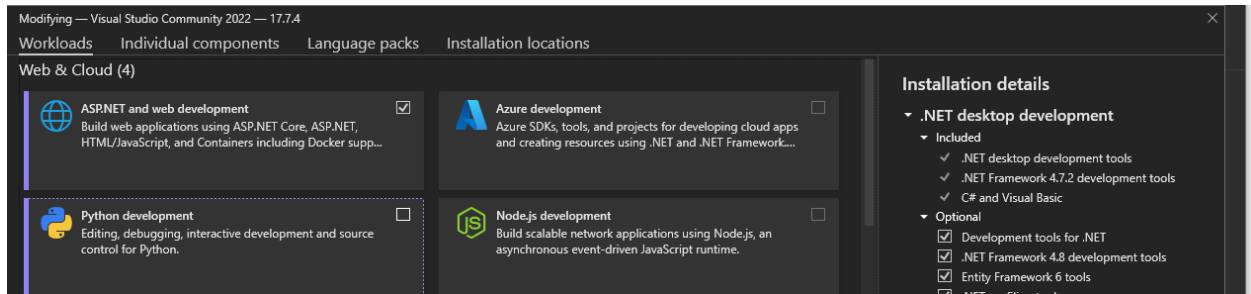
#### 3.3.1 Створення серверної частини

Для створення серверної частини системи потрібно встановити необхідні компоненти розробки. Для цього потрібно відкрити Visual Studio Installer



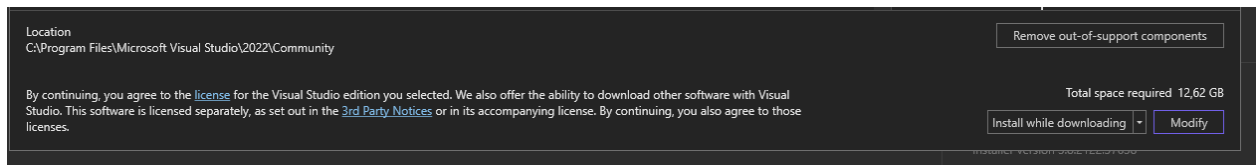
«Рисунок 3.12» - Visual Studio Installer

Натиснути «Modify» та обрати необхідні компоненти, тобто ASP .Net



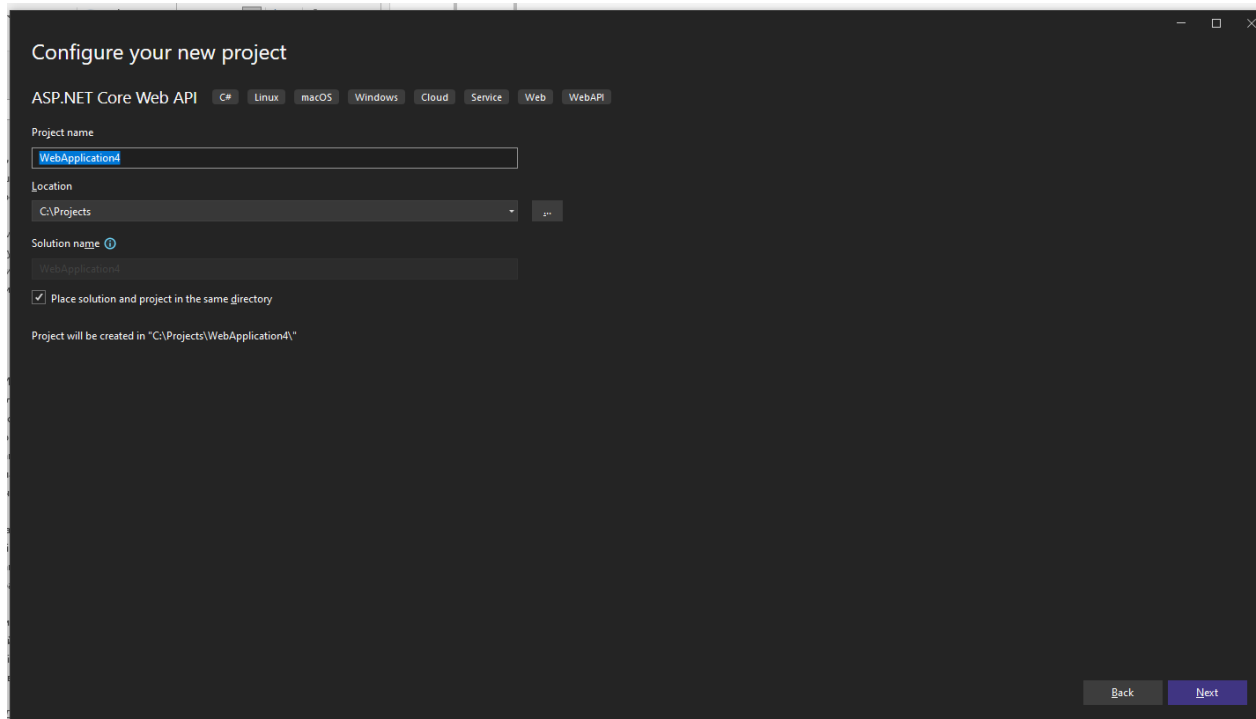
«Рисунок 3.13» - Visual Studio Installer

Та натиснути «Modify»



«Рисунок 3.14» - Visual Studio Installer

Далі потрібно створити ASP .Net додаток



«Рисунок 3.15» - Створення проекту

Робота додатку починається з файлу Program.cs.

```

public class Program
{
    0 references
    public static void Main(string[] args)
    {
        var builder = WebApplication.CreateBuilder(args);

        builder.Services.AddControllers();

        builder.Services.AddEndpointsApiExplorer();
        builder.Services.AddSwaggerGen();

        var app = builder.Build();

        if (app.Environment.IsDevelopment())
        {
            app.UseSwagger();
            app.UseSwaggerUI();
        }

        app.UseHttpsRedirection();

        app.UseAuthorization();

        app.MapControllers();

        app.Run();
    }
}

```

«Рисунок 3.16» - Файл Program.cs

У ньому створюється об'єкт класу `WebApplicationBuilder`.

Далі для реалізації механізму `Dependency injection` використано контейнер `Services`.

Впровадження залежностей (`Dependency Injection` або `DI`) - це шаблон проектування, широко використовуваний у розробці програмного забезпечення для покращення модульності, збереження, та легкого тестування коду. У основі `DI` лежить зовнішнє надання залежностей, які необхідні класу, замість того, щоб клас створював чи керував цими залежностями внутрішньо. Це вивертає контроль та дозволяє розділити компоненти, зменшуючи їх взаємозалежність та роблячи систему більш гнучкою. Існують три основні типи Впровадження залежностей: впровадження через конструктор, впровадження через сеттер та внедрення через метод. Впровадження через конструктор передбачає передачу залежностей через конструктор класу, що гарантує наявність необхідних залежностей при створенні екземпляру класу. Впровадження через сеттер використовує методи сеттера для внесення залежностей після створення об'єкта,

що надає більш гнучкий підхід. Впровадження через метод передбачає передачу залежностей параметрами до конкретних методів, коли це потрібно. DI сприяє принципу "програмування до інтерфейсів, а не до реалізацій", дозволяючи легше заміщення компонентів та полегшуючи інтеграцію різних модулів у системі. Завдяки впровадженню залежностей розробники можуть досягти більшої гнучкості та обслуговуваності свого коду.

У контейнер *Sevrvies* додаються сервіси які у подальшому будуть використані у додатку.

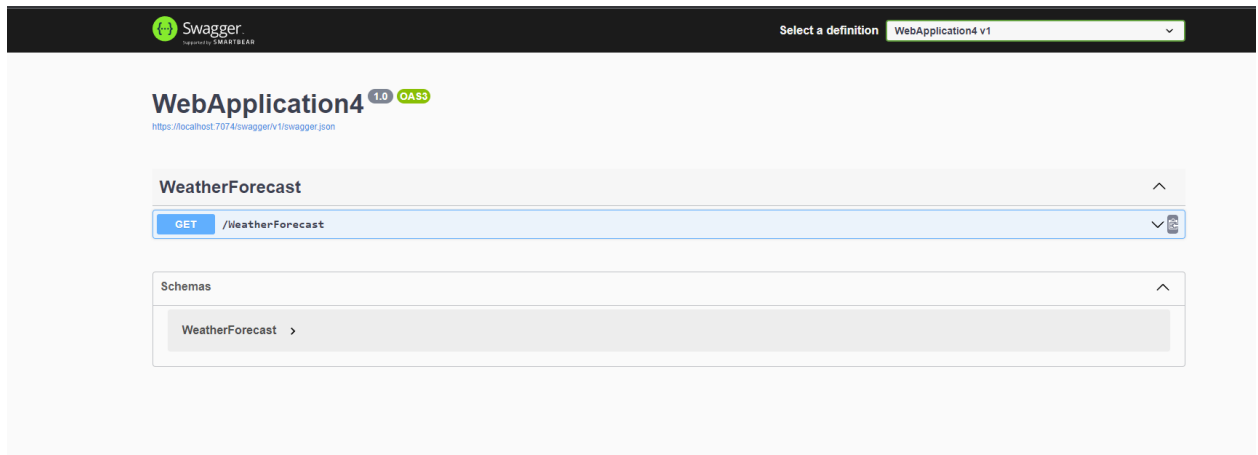
Далі оголошуються компоненти проміжного програмного забезпечення.

У контексті ASP.NET, проміжне програмне забезпечення (Middleware) відіграє ключову роль у реалізації обробки запитів та відправлення відповідей. ASP.NET Middleware представляє собою ланцюжок компонентів, які обробляють HTTP-запити і відповіді під час їх проходження через веб-додаток. Кожен компонент Middleware виконує конкретну функцію, таку як маршрутизація, аутентифікація, авторизація, логування чи кешировання.

```
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
```

«Рисунок 3.17» - роміжне програмне забезпечення для розробки

У даній частині коду оголошується проміжне програмне забезпечення яке застосовується тільки при розробці. Swagger це набір інструментів для розробки програмного забезпечення, який спрощує розробку, документацію та спілкування з веб-сервісами, які підтримують архітектуру RESTful. Основною метою Swagger є створення інтерактивної та легко зрозумілої документації для API, щоб розробники могли швидко розуміти, як взаємодіяти з конкретним веб-сервісом.



«Рисунок 3.18» - Swagger

Далі потрібно створити контролер. Розглянемо контролер авторизації користувача.

```
[Route("api/[controller]")]
[ApiController]
1 reference
public class AuthController : ControllerBase
{
    public static User user = new User(); //test
    private readonly IConfiguration _configuration;
    private readonly IUserService _userService;
```

«Рисунок 3.19» - Контролер авторизації

Для того зоб створений клас вважався контролером потрібно щоб його назва закінчувалась словом Controller, він повинен наслідувати клас Controller або ControllerBase, також він має мати атрибут [ApiController].

```
[HttpPost("register")]
0 references
public ActionResult<User> Register(UserDto request)
{
    string passwordHash
        = BCrypt.Net.BCrypt.HashPassword(request.Password);

    user.Username = request.Username;
    user.PasswordHash = passwordHash;

    return Ok(user);
}
```

«Рисунок 3.20» - Запит реєстрації

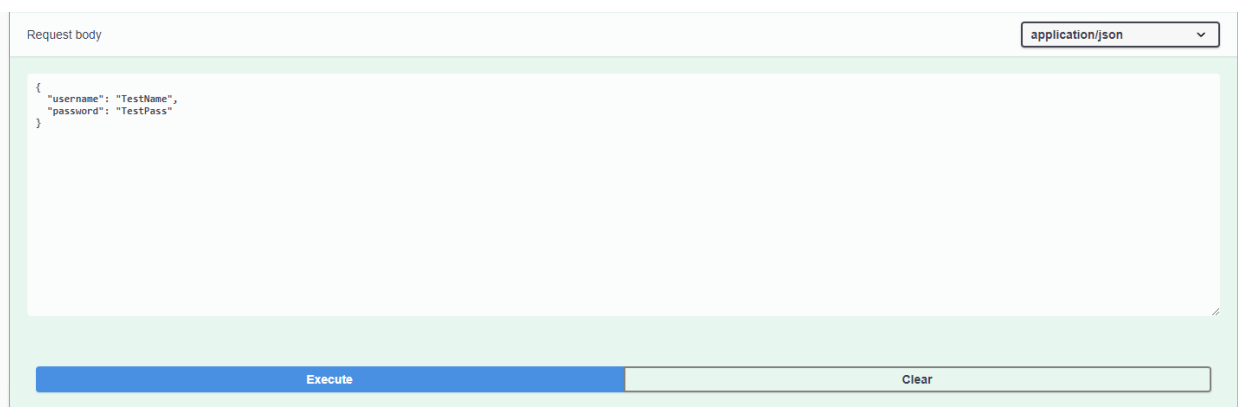
Далі створено запит з реєстрацією користувача. Це Post запит, тобто він може приймати аргументи. Запит приймає аргумент з типом UserDto. UserDto це DTO (Data Transfer object) класу User.

DTO (Data Transfer Object) - це об'єкт, який використовується для передачі даних між об'єктами в програмі. DTO призначений для зручної і ефективної передачі даних між різними компонентами програми, такими як клієнт-серверні взаємодії, обробники запитів веб-служб, або інші частини системи. Основна ідея DTO полягає в тому, щоб об'єднати дані, які потрібно передати, в окремий об'єкт, а потім передати цей об'єкт між компонентами програми. Це може допомогти уникнути передачі зайвих даних і забезпечити структурований спосіб обміну інформацією.

Клас User має атрибути Name тобто ім'я, та hashPassword тобто хеш паролю. Об'єкти даного класу зберігаються в базі даних.

Клас UserDto має поля Name тобто ім'я, та Password тобто пароль. Об'єкти даного класу використовується лише для передачі інформації між клієнтом та сервером.

У ньому з даних відправлених від клієнта створюється об'єкт User, для цього з паролю отриманого від клієнта створюється хеш паролю, та відправляється клієнту.



«Рисунок 3.21» - введення даних тестового запиту реєстрації



«Рисунок 3.22» - Результат запиту реєстрації

```
[HttpPost("login")]
0 references
public ActionResult<User> Login(UserDto request)
{
    if (user.Username != request.Username)
    {
        return BadRequest("User not found.");
    }

    if (!BCrypt.Net.BCrypt.Verify(request.Password, user.PasswordHash))
    {
        return BadRequest("Wrong password.");
    }

    string token = CreateToken(user);

    return Ok(token);
}
```

«Рисунок 3.23» - Створення запити авторизації

Далі створено запит з авторизацією користувача. Це Post запит, тобто він може приймати аргументи. Запит приймає аргумент з типом UserDto. UserDto це DTO (Data Transfer object) класу User. У ньому за допомогою методу Bcrypt.Verify порівнюється пароль та хеш паролю, якщо пароль не верифікується то запит надає відповідь «невірний пароль». Далі створюється токен та передається користувачу.

```

private string CreateToken(User user)
{
    List<Claim> claims = new List<Claim>();

    var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(
        _configuration.GetSection("AppSettings:Token").Value!));

    var creds = new SigningCredentials(key, SecurityAlgorithms.HmacSha512Signature);

    var token = new JwtSecurityToken(
        claims: claims,
        expires: DateTime.Now.AddDays(1),
        signingCredentials: creds
    );

    var jwt = new JwtSecurityTokenHandler().WriteToken(token);

    return jwt;
}

```

«Рисунок 3.24» - Метод створення JWT токена

У методі CreateToken створюється JWT токен користувача.

JWT (JSON Web Token) - це відкритий стандарт (RFC 7519), який визначає компактний і самостійний спосіб представлення інформації між двома сторонами у вигляді об'єкта JSON. JWT може бути використаний для безпечної передачі інформації між сторонами, такими як клієнт і сервер.

Перевірка роботи запиту, якщо введено неправильний пароль

The screenshot shows a REST client interface for a POST request to the endpoint `/api/Auth/login`. The request body is set to `application/json` and contains the following JSON data:

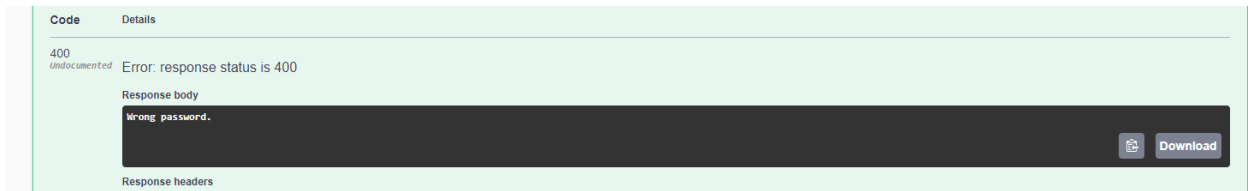
```

{
  "username": "TestName",
  "password": "WrongPass"
}

```

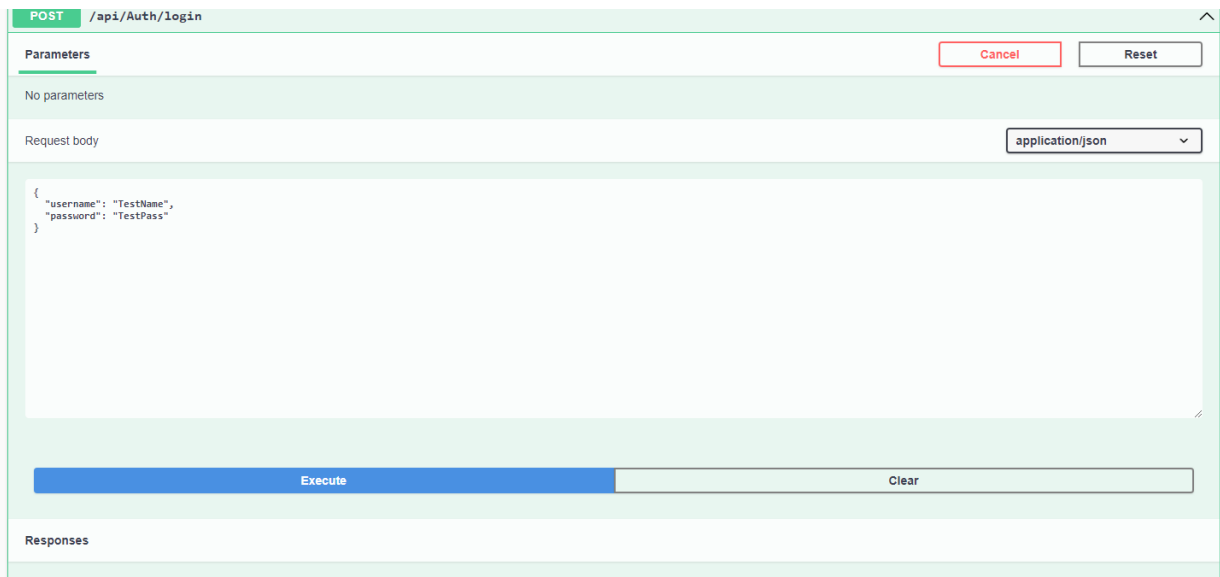
At the bottom of the interface, there are buttons for "Execute" and "Clear". The "Responses" section is currently empty.

«Рисунок 3.25» - Введення даних для запиту авторизації з невірними даними

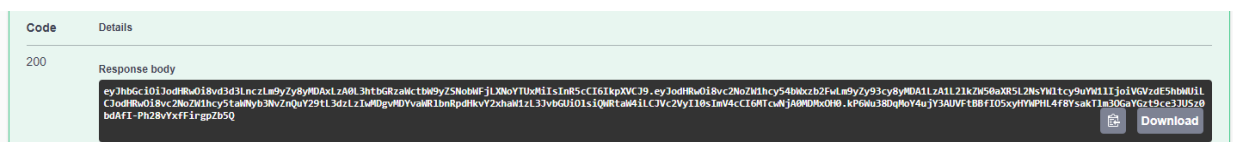


«Рисунок 3.26» - Результат запиту авторизації з невірними даними

### Перевірка роботи запиту, якщо введено правильний пароль

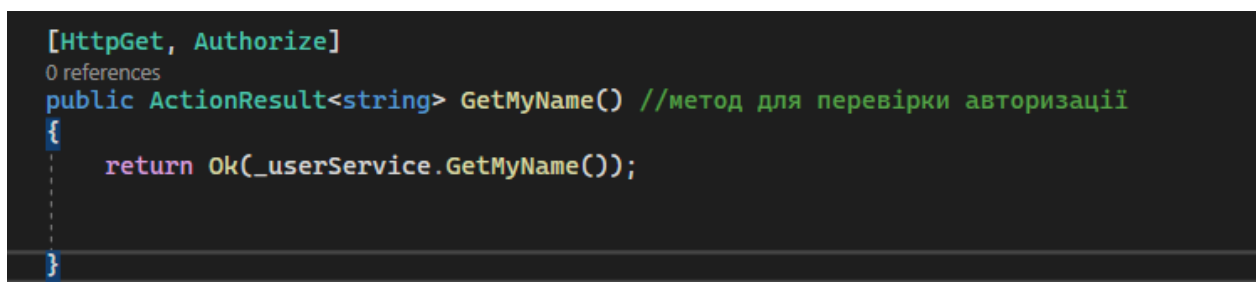


«Рисунок 3.27» - Введення даних для запиту авторизації з вірними даними



«Рисунок 3.28» - - Результат запиту авторизації з вірними даними

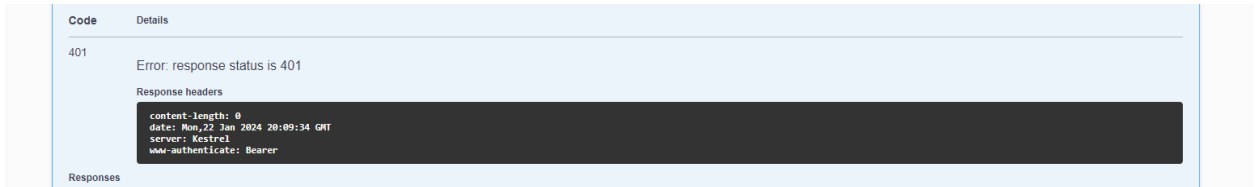
Далі перевірено роботу механізму авторизації. Для цього створено запит який як результат надає ім'я користувача.



«Рисунок 3.29» - Тестовий метод GetMyName

У ньому задано атрибут [Authorize] який означає що запит може бути виконаний лише авторизованими користувачами.

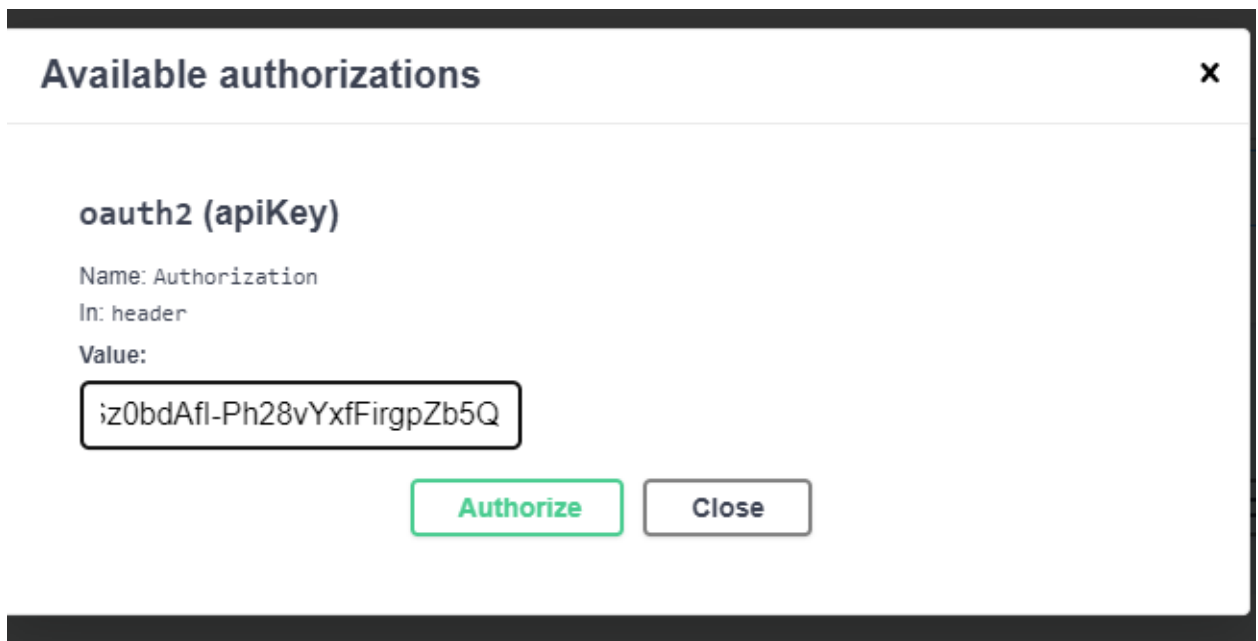
Перевірка роботи запиту за умови не авторизованого користувача.



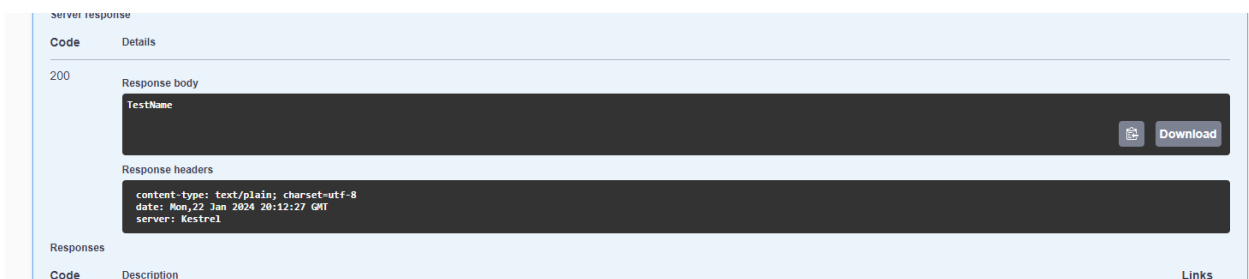
«Рисунок 3.30» - Перевірка роботи запиту

Отримано помилку з кодом 401 тобто користувач не авторизований.

Авторизовано користувача, для цього потрібен токен отриманий про запиті Login.



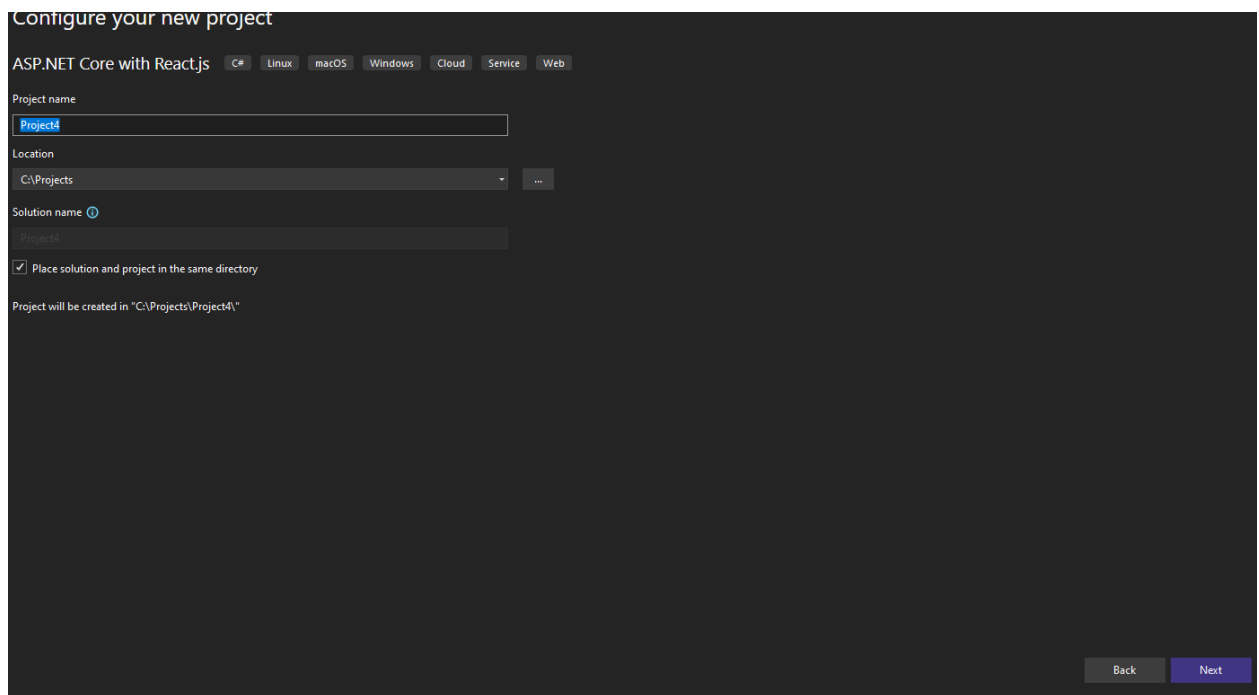
«Рисунок 3.31» - Введення токена авторизації



«Рисунок 3.32» - Результат запиту авторизованого користувача.

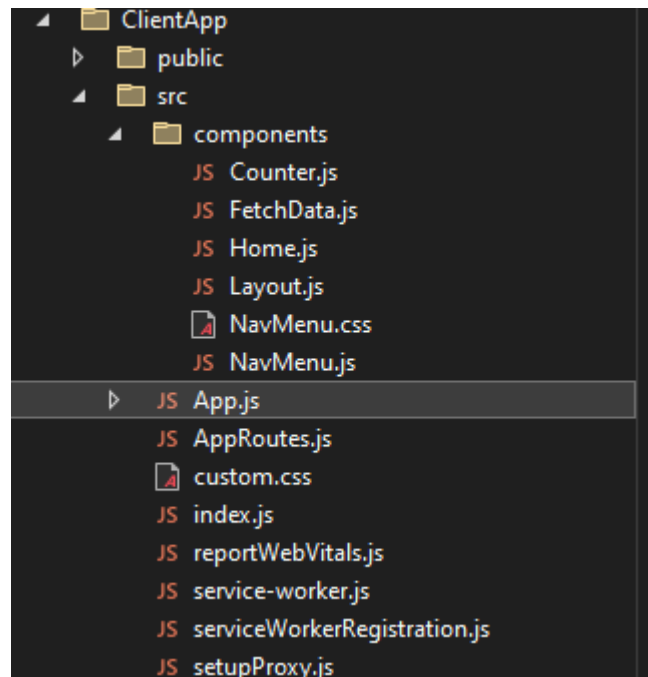
### 3.3.2 Створення веб платформи

Для створення веб додатку використано ReactJS. Було створено новий проект за допомогою Visual Studio.



«Рисунок 3.33» - Створення проекту.

Створений проект ReactJs з корневим компонентом, та деякими базовими компонентами.



«Рисунок 3.34» - Файлова система

Компонент Layout це компонент який є контейнером для вмісту сайту. Даний компонент потрібен щоб зменшити обсяг кореневого компоненту та додати реюзабельність компоненту.

```

import React, { Component } from 'react'
import { Container } from 'reactstrap';
import { NavMenu } from './NavMenu';

4 references
export class Layout extends Component {
  static displayName = Layout.name;

  render() {
    return (
      <div>
        <NavMenu />
        <Container>
          {this.props.children}
        </Container>
      </div>
    );
  }
}

```

«Рисунок 3.35» - Компонент «Layout»

З початку використовується ключове слово `import` для підключення бібліотек. Підключається компонент `Container` з бібліотеки `reactstrap`.

`Reactstrap` - це бібліотека компонентів для `React`, яка надає готові до використання компоненти, стилі та функціональність інтерфейсу користувача (UI) на основі бібліотеки дизайну `Bootstrap`.

`Bootstrap` - це популярна бібліотека фронтенд-компонентів та стилів, яка дозволяє розробникам швидко створювати сучасні та ефективні інтерфейси користувача. `Reactstrap` використовує можливості `Bootstrap`, але адаптована для використання в середовищі `React`.

Також підключається компонент `NavMenu`. Він потрібен для реалізації навігаційного меню додатку.

```
render() {
  return (
    <header>
      <Navbar className="navbar-expand-sm navbar-toggleable-sm ng-white border-bottom box-shadow mb-3" container light>
        <NavbarBrand tag={Link} to="/">Project3</NavbarBrand>
        <NavbarToggler onClick={this.toggleNavbar} className="mr-2" />
        <Collapse className="d-sm-inline-flex flex-sm-row-reverse" isOpen={!this.state.collapsed} navbar>
          <ul className="navbar-nav flex-grow">
            <NavItem>...</NavItem>
            <NavItem>...</NavItem>
            <NavItem>...</NavItem>
          </ul>
        </Collapse>
      </Navbar>
    </header>
  );
}
```

«Рисунок 3.36» - Розмітка компоненту «NavMenu»

У файлі `App.js` оголошені головні змінні стану додатку, та рендер елементів. Якщо користувач не авторизований то користувач бачить сторінку авторизації, якщо авторизований то виводить основний зміст сайту.

```

render() {
  if (!this.state.loggedIn)
  {
    { console.log(typeof this.CheckLogin) }
    return <Login CheckAuth= {this.CheckAuth}/>
  }
  else {
    return (
      <Layout>
        <Routes>
          {AppRoutes.map((route, index) => {
            const { element, ...rest } = route;
            return <Route key={index} {...rest} element={element} />;
          })}
        </Routes>
      </Layout>
    );
  }
}

```

«Рисунок 3.37» - Розмітка головної сторінки

У компонент передається функція CheckAuth. Дана функція отримує 2 аргументи, ім'я та пароль. У requestOptions задається тип запиту, заголовки, та вміст запиту на сервер. Далі за допомогою функції fetch виконується запит на сервер. За умови якщо запит виконується з кодом 200 (тобто успішно), встановлюється змінна стану LoggedIn. Завдяки механізмам платформи ReactJs сторінка оновиться автоматично.

```

CheckAuth = async (name, pass) => {
  const requestOptions = {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ Username: name, Password: pass })
  };

  const response = await fetch('/login/userlogin', requestOptions);
  console.log(response);

  if (response.ok) {
    const data = await response.json();
    console.log(data.body);
    console.log("logged in");
    this.setState({ loggedIn: true });
    this.setState({ JwtToken: data.body });
    return;
  }
  console.log(response);
  return;
}

```

«Рисунок 3.38» - Функція CheckAuth

Створено сторінку входу у систему.

```

<Container>
  <Row className="justify-content-center">
    <Col md={6}>
      <h2 className="mt-5 mb-4">Вхід в систему</h2>
      <Form onSubmit={handleSubmit}>
        <FormGroup>
          <Label for="username">Імя</Label>
          <Input
            type="text"
            name="username"
            id="username"
            placeholder="Введіть ваше імя"
            value={loginData.username}
            onChange={handleInputChange}
          />
        </FormGroup>
        <FormGroup>
          <Label for="password">Пароль</Label>
          <Input
            type="password"
            name="password"
            id="password"
            placeholder="Введіть ваш пароль"
            value={loginData.password}
            onChange={handleInputChange}
          />
        </FormGroup>
        <Button color="primary" type="submit">
          Вхід
        </Button>
      </Form>
    </Col>
  </Row>
</Container>
);

```

«Рисунок 3.39» - Сторінка входу

Далі створена функція `handleInputChange` яка оброблює введення тексту.

```

const handleInputChange = (e) => {
  const { name, value } = e.target;
  setLoginData((prevData) => ({
    ...prevData,
    [name]: value,
  }));
};

```

«Рисунок 3.40» - Обробка введення даних

Далі створена функція `handleSubmit` яка оброблює введення тексту.

```

const handleSubmit = (e) => {
  e.preventDefault();
  console.log(typeof props.CheckAuth);
  props.CheckAuth(loginData.username, loginData.password);
  console.log('Login data submitted:', loginData);
};

```

«Рисунок 3.41» - Обробка натискання кнопки

## Вхід в систему

Імя

Пароль

«Рисунок 3.42» - Сторінка входу

## ВИСНОВОК

В результаті виконаної роботи було створено інформаційну систему для управління транспортною логістикою. Менеджери логістики можуть авторизуватись в системі, отримати доступ до інформації про перевезення, вантажі і маршрути. Також працівники мають можливість керувати маршрутами та редагувати їх.

Дослідження сучасних технологій управління транспортною логістикою, представлений у роботі, підкреслює важливість впровадження інновацій для оптимізації великих обсягів даних та покращення прийняття стратегічних рішень. Використання систем машинного навчання та цифрових двійників дозволяє не тільки ефективно вирішувати різноманітні завдання, але й створює підґрунтя для подальшого розвитку та вдосконалення системи у майбутньому.

Особистий внесок дослідника полягає у розробці інформаційної системи згідно всіх вимог, а також потреб потенціальних користувачів.

Створення додатку, розробленого під час магістерської роботи виявляється доречним у зв'язку з реальною потребою в покращенні логістичних процесів. Такий додаток може забезпечити збалансоване та оптимізоване управління транспортною логістикою, забезпечуючи точні дані, швидкі рішення та підвищену ефективність логістичного ланцюга.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 3008-95. Документація. Звіти в галузі науки і техніки. Структура і правила оформлення. - К.: Держстандарт України, 1995
2. Інтернет-ресурс <http://microsoft.com>
3. Інтернет-ресурс <https://learn.microsoft.com/en-us/dotnet/>
5. Інтернет ресурс <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui>
6. Інтернет ресурс <https://github.com/dotnet/maui>
7. Єдина система конструкторської документації. Основні написи (ГОСТ 2.104-2006, ІДТ). ДСТУ ГОСТ 2.104:2006 / Вид. Офіційне. - Київ: Держспоживстандарт України, 2007.-16 с.
8. Текстові документи. Загальні вимоги СОУ 207.01:2004. - 3-тє вид., із змінами / В. Каплун, В. Олександренко, Л. Першина, Л. Безсмертна, О. Снозик. - Хмельницький: ХНУ, 2008. - 40 с.
9. ДСТУ 3008-95 Документація. Звіти у сфері науки і техніки. Структура і правила оформлення [Електронний ресурс]. - Чинний 01.01.1996. - К.: Держстандарт України, 1996. - 29 с. // Портал «Безкоштовна бібліотека ДСТУ, ДБН». - Режим доступу: <http://dstu-biblio.3dn.ru/1-1-0-11>
10. Коноваленко І.В. Платформа .NET та мова програмування С# 8.0: навчальний посібник / І. В. Коноваленко, П. О. Марущак. – Тернопіль: ФОП Паляниця В. А., 2020. – 320 с. [Електронний ресурс] – Режим доступу до ресурсу: <http://elartu.tntu.edu.ua/handle/lib/32825> .
11. Коноваленко І.В. Програмування мовою С#: курс лекцій з дисципліни Об'єктно-орієнтоване програмування/І.В. Коноваленко. - Тернопіль: ТНТУ імені Івана Пулюя, 2019. – 229 с. [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/7078156/page:6/>
12. Інтернет-ресурс <https://www.w3schools.com/>
13. Інтернет-ресурс <https://learn.microsoft.com/en-us/sql/sql-server>

14. Інтернет-ресурс <https://learn.microsoft.com/en-gb/aspnet/core/?view=aspnetcore-8.0>
15. Somayeh Malakut, Pieter van Schalkwyk, Shi-Wan Lin, Kilian Baechle  
Digital Twins Industrial Apps White Paper 2020 02 18
16. Головіна Олена. Сучасні технології в управлінні транспортною логістикою. International Science Journal of Management, Economics & Finance. Vol. 2, No. 3, 2023, pp. 35-42. doi: 10.46299/j.isjmef.20230203.04
17. Інтернет-ресурс <https://react.dev/reference/react>