

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

## КВАЛІФІКАЦІЙНА РОБОТА

бакалавр  
Освітній рівень

Інформаційна система управління логістичними перевезеннями  
Назва теми

КвРІСТ 220168.22.01.05 ПЗ  
Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва


Виконав: студент III курсу, група ICT-22-1

  
Підпис

Валентин КЛИНЧУК

Ініціали, прізвище

Керівник


  
Підпис, дата

14.06.2025

Дмитро ДЕНИСЮК

Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:  
зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«16» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

" 10 " 01 2025 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

Валентину КЛИНЧУКУ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система управління логістичними перевезеннями

Керівник проекту (роботи) Дмитро Денисюк, старший викладач.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Інформаційна система управління логістичними перевезеннями та постановка задачі щодо її удосконалення

Проектування інформаційної системи для керування та обробки запитів

Розробка апаратної частини проекту Інформаційної системи та тестування прототипу

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Структурна схема роботи інформаційної системи для логістики

Структурна схема взаємодії системи з користувачем

Результат роботи інформаційної системи

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагіат	Андрій НІЧЕПОРУК, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів для проектування інформаційної системи управління логістичними перевезеннями.	01.04.2025	виконано
5	Робота над розділом 3 – проектування інформаційної системи управління логістичними перевезеннями	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Керівник роботи

Підпис

Валентин КЛИНЧУК  
Ініціали, прізвище

Підпис

Дмитро ДЕНИСЮК  
Ініціали, прізвище

№ р я д к а	ф о р м а т	Позначення	Найменування	К і л - л и с т і в	№ с к з	П р и м і т к в
1		КвРІСТ 220168.22.01.05 ПЗ	<u>Текстові документи</u> Пояснювальна записка	56		
2		КвРІСТ 220168.22.01.05 Е8	<u>Графічні матеріали</u> Структурна схема роботи інформаційної системи для логістики	1		
3		КвРІСТ 220168.22.01.05 Е8	Структурна схема взаємодії системи з користувачем	1		
4		КвРІСТ 220168.22.01.05 Е8	Результат роботи інформаційної системи	1		
КвРІСТ 220168.22.01.05 ВП						
Зм	Арк	№ докум	Підпис	Дата	Відомість проекту  ХНУ, ІСТс-22-1	
Розробив		Клиничук	<i>[Signature]</i>	17.06.25		
Перевір.		Денисюк	<i>[Signature]</i>	17.06.25		
Н. контр.		Кисіль	<i>[Signature]</i>	17.06.25		
Затв.		Павлова	<i>[Signature]</i>	17.06.25		

Літера	Аркуш	Аркушів
У	1	1

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система управління логістичними перевезеннями».

Автор роботи: Валентин Кличук Александрович.

Керівник роботи: Денисюк Дмитро Александрович.

Пояснювальна записка: 55 с., 23 рис., 9 табл., 3 дод., 40 джерел.

Графічна частина: 3 креслення.

ЛОГІСТИКА, ІНФОРМАЦІЙНА СИСТЕМА, АРХІТЕКТУРА,  
МОНІТОРИНГ, БАЗА ДАНИХ.

Метою даної дипломної роботи є аналіз умов впровадження та особливостей використання технічного забезпечення інформаційної системи, а також дослідження процесів обробки інформації в межах логістичної системи з метою підвищення ефективності транспортних операцій.

Об'єкт дослідження становить процес функціонування інформаційної системи, спрямованої на оптимізацію управління транспортними перевезеннями.

Предметом дослідження є ефективність застосування інформаційної системи для координації логістичних перевезень.

У процесі виконання дослідження було застосовано метод систематичного огляду інформаційних джерел, що дозволило провести ґрунтовне вивчення предметної області на основі аналізу як текстових, так і електронних матеріалів.







Підпис студента

30.05.2025

Дата

## ЗМІСТ

<b>ВСТУП</b> .....	4
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОШУК ОПТИМАЛЬНО РІШЕННЯ ДЛЯ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПЕРЕВЕЗЕННЯМИ</b> .....	5
1.1 Аналіз предметної області і виявлення наявних проблем і завдань.....	14
1.2 Порівняльний аналіз переваг та недоліків існуючих рішень.....	23
1.3 Підходи до вирішення задачі за темою дослідження.....	25
1.4 Висновки до першого розділу.....	27
<b>2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПЕРЕВЕЗЕННЯМИ</b> .....	27
2.1 Визначення вимог і технічних характеристик.....	28
2.2 Опис середовища розробки Visual Studio Code.....	33
2.3 Використання Tailwind CSS у проєкті ІСУЛП.....	36
2.4 JavaScript та React: основа сучасної веб-розробки.....	40
2.5 Висновки до другого розділу.....	41
<b>3 ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПЕРЕВЕЗЕННЯМИ</b> .....	41
3.1 Опис розробки інформаційної системи управління логістичними перевезеннями.....	44
3.2 Модулі та їх функціональність у системі.....	48
3.3 Взаємодії компонентів у системі.....	50
3.4 Результати впровадження системи.....	54
3.5 Висновки до третього розділу.....	56
<b>ВИСНОВКИ</b> .....	56

КвРІСТ 220168.22.01.05 ПЗ								
Зм.	Арк.	Надокум.	Підпис	Дата		Літера	Аркуш	Аркушів
Виконав		Валентин Клинич		17.06.25	Інформаційна система управління логістичними перевезеннями. Пояснювальна записка	у	2	56
Перевір.		Дмитро Денисюк		17.06.25				
Н.контр.		Тетяна КИСІЛЬ		17.06.25				
Затвер.		Ольга ПАВЛОВА		17.06.25				
						ХНУ, ІСТс-22-1		

## ВСТУП

В умовах сучасної ринкової економіки ефективне управління логістичними перевезеннями є однією з ключових передумов конкурентоспроможності та стабільного розвитку підприємств різних галузей. Логістичні процеси, які охоплюють транспортування, зберігання, обробку вантажів та управління інформаційними потоками, вимагають чіткої координації, оперативності та високої точності даних. Водночас збільшення обсягів перевезень та зростання складності логістичних ланцюгів зумовлює необхідність автоматизації та вдосконалення існуючих систем управління.

У зв'язку з цим, актуальною стає розробка і впровадження сучасних інформаційних систем управління логістичними перевезеннями, які здатні оптимізувати та інтегрувати різноманітні процеси, скоротити витрати часу та ресурсів, підвищити прозорість і надійність перевезень, а також покращити рівень сервісного обслуговування клієнтів. Такі інформаційні системи стають ефективними інструментами стратегічного управління, що забезпечують прийняття своєчасних та обґрунтованих управлінських рішень на всіх рівнях логістичного менеджменту.

					КвРІСТ 220168.22.01.05 ПЗ	Арк.
						3
Зм.	Арк.	№ докум.	Підпис	Дата		

# АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОШУК ОПТИМАЛЬНО РІШЕННЯ ДЛЯ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПЕРЕВЕЗЕННЯМИ

## 1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Управління логістичними перевезеннями сьогодні займає центральне місце у діяльності підприємств, визначаючи швидкість реакції компаній на зміни ринкового середовища, підвищення ефективності управління ресурсами та забезпечення конкурентоспроможності. В умовах стрімкого розвитку цифрових технологій інформаційні системи відіграють критично важливу роль, тому дослідження та удосконалення їх функціональності є вкрай актуальним.

Перший етап, який охоплює період 1960-1970-х років, став початком автоматизації логістичних операцій. У цей час світова економіка переживала стрімку індустріалізацію та зростання міжнародної торгівлі, що супроводжувалося збільшенням обсягів виробництва та переміщення товарів. Ускладнення логістичних процесів, таких як облік запасів, планування постачань та управління складами, вимагало нових рішень для підвищення ефективності та зниження витрат. Саме тоді з'явилася потреба в автоматизованих підходах до управління логістикою.

Одним із ключових технологічних досягнень цього етапу стало впровадження електронно-обчислювальних машин (ЕОМ) у бізнес-практику. Перші комп'ютери почали використовуватись для обробки великих обсягів даних, пов'язаних із закупівлями, обліком товарів та транспортуванням. Це дало змогу значно зменшити обсяг ручної праці, знизити кількість помилок та підвищити точність управлінських рішень.

У цей період розпочався розвиток систем планування потреб у матеріалах (MRP – Material Requirements Planning), які дозволяли оптимізувати постачання й виробничі процеси. Паралельно почали з'являтися перші системи автоматичного зчитування даних, зокрема штрих-кодування, що згодом стало стандартом в обліку

					КвРІСТ 220168.22.01.05 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

товарів. У великих компаніях впроваджувались термінали для обробки замовлень, що пришвидшило обмін інформацією між відділами та підрозділами.

Унаслідок цих змін логістика почала трансформуватись із другорядної функції у стратегічно важливу сферу бізнесу. Автоматизація сприяла підвищенню швидкості логістичних операцій, покращенню точності планування та зменшенню витрат. Цей перший етап заклав фундамент для подальшої цифрової трансформації логістичних систем, яка продовжилася в наступні десятиліття.

У 1980-1990-х роках автоматизація логістики перейшла на якісно новий рівень. Цей період характеризується активним розвитком комп'ютерних технологій, широким впровадженням персональних комп'ютерів та розвитком телекомунікацій, що значно розширило можливості управління логістичними процесами.

Одним із ключових досягнень цього етапу стало поширення інтегрованих інформаційних систем, зокрема ERP-систем (Enterprise Resource Planning), які дозволяли об'єднувати в єдину платформу управління закупівлями, виробництвом, запасами, транспортом і фінансами. Це зробило логістику прозорішою, керованішою та орієнтованою на планування в реальному часі.

Водночас продовжувався розвиток MRP-систем, які еволюціонували у MRP II - системи, що охоплювали ширший спектр функцій, включаючи планування виробничих потужностей, бюджетування та управління персоналом.

У цей період також зростає значення складської автоматизації. Впроваджуються автоматизовані системи управління складами (WMS - Warehouse Management Systems), які забезпечують точний облік товарів, оптимізацію розміщення продукції, контроль за переміщенням та відвантаженням. У великих логістичних центрах починають використовуватися конвеєрні системи, роботизовані комплекси та автоматичні засоби навантаження/розвантаження.

Ще важливим аспектом стало впровадження електронного обміну даними (EDI – Electronic Data Interchange). Це дозволило підприємствам обмінюватися документацією (замовленнями, рахунками, накладними) у стандартизованому

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

електронному форматі, значно скорочуючи час і мінімізуючи помилки, пов'язані з людським фактором.

Другий етап автоматизації логістичних операцій був спрямований на інтеграцію логістики з іншими бізнес-процесами, підвищення ефективності управлінських рішень та скорочення витрат завдяки використанню новітніх ІТ-рішень. Це створило передумови для переходу до ще глибшої цифрової трансформації логістики у XXI столітті.

У період з 2000 по 2010 роки почалася глобальна цифровізація, яка супроводжувалася активним використанням інтернет-технологій, мобільних додатків, GPS-навігації та RFID-технологій. У цьому контексті логістика перетворилася на один із ключових чинників забезпечення ефективності діяльності компаній. Саме в цей період відбувся перехід від традиційної автоматизації до цифрової інтеграції логістичних процесів.

Однією з головних ознак третього етапу стало впровадження інтернет-технологій у логістику. Широке використання веб-інтерфейсів, віддаленого доступу до систем управління, онлайн-моніторингу маршрутів транспорту та GPS-навігації дозволило компаніям здійснювати контроль логістичних потоків у режимі реального часу.

Активно розвиваються TMS (Transportation Management Systems) - системи управління транспортом, які оптимізують маршрути, розраховують витрати на перевезення, аналізують продуктивність логістичних операцій. Удосконалення WMS (Warehouse Management Systems) також продовжується: вони починають підтримувати технології штрих-кодів, RFID (Radio-Frequency Identification) та мобільні сканери, що значно підвищує точність і швидкість обробки вантажів.

Ще однією знаковою рисою стало поширення хмарних сервісів (cloud computing). Завдяки їм компанії отримали можливість зберігати великі обсяги даних, здійснювати оперативний обмін інформацією між партнерами по ланцюгу постачань, а також масштабувати свої логістичні рішення без необхідності значних інвестицій у локальну інфраструктуру.

					КвРІСТ 220168.22.01.05 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

У цей період особливу популярність набувають SCM-системи (Supply Chain Management) - комплексні платформи управління ланцюгами постачань, які об'єднують логістику, виробництво, продажі, закупівлі та планування в єдине інформаційне середовище. Вони дозволяють будувати гнучкі, адаптивні та стійкі ланцюги постачання.

Третій етап позначений зростанням значення аналітики даних (Business Intelligence). Компанії починають використовувати інструменти для аналізу ефективності логістики, прогнозування попиту, оптимізації запасів та прийняття стратегічних рішень на основі даних.

Також цей етап автоматизації логістичних операцій був спрямований на інформаційну інтеграцію, мобільність і прозорість. Саме в цей період логістика остаточно переходить у цифрову площину, створюючи підґрунтя для інтелектуальних систем майбутнього.

Починаючи з 2010-х років, автоматизація логістики увійшла в нову фазу, яку умовно можна назвати етапом розумної (інтелектуальної) логістики. У центрі цього етапу - використання новітніх цифрових технологій, штучного інтелекту, великих даних та автоматизованих систем, здатних до самонавчання та адаптації. Основна мета - створення гнучких, прозорих, сталих і максимально ефективних ланцюгів постачань, здатних оперативно реагувати на зміни середовища.

Важливу роль у цьому процесі відіграє штучний інтелект (AI). Завдяки йому логістичні системи отримали можливість аналізувати великі масиви даних у режимі реального часу, прогнозувати попит, оптимізувати маршрути та графіки доставок, виявляти аномалії та автоматично ухвалювати рішення. Наприклад, алгоритми машинного навчання використовуються для прогнозування заторів на дорогах, оцінки часу прибуття транспорту або визначення найефективнішого складу для відвантаження товару.

Широко застосовуються Big Data - великі дані з різних джерел: сенсорів IoT (інтернету речей), GPS-трекерів, даних клієнтів і постачальників. Завдяки цьому забезпечується точна аналітика і глибоке розуміння усіх процесів у ланцюгу

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

постачання. Технології IoT дозволяють відстежувати стан вантажів (температуру, вологість, рух), що особливо важливо для фармацевтики, харчової промисловості та інших галузей з високими вимогами до умов транспортування.

У логістичних центрах активно впроваджуються роботизовані системи: автономні візки (AGV), роботизовані руки для сортування та пакування товарів, безпілотники (дрони) для інвентаризації та доставки. У поєднанні з автоматизованими складами це забезпечує швидку обробку замовлень із мінімальним втручанням людини.

Сучасна логістика також активно використовує хмарні технології та блокчейн. Хмарні платформи забезпечують миттєвий обмін даними між усіма учасниками ланцюга постачання, а блокчейн гарантує прозорість і безпеку транзакцій, що особливо актуально для міжнародної торгівлі, де важливо мати підтвердження кожного етапу доставки.

Особливу увагу у цей період приділяють сталому розвитку. Зростає попит на "зелену логістику": оптимізацію маршрутів для зменшення викидів CO<sub>2</sub>, впровадження електротранспорту, використання енергоефективного обладнання та екологічної упаковки. Логістика більше не обмежується економічними показниками - вона все частіше враховує соціальну відповідальність і вплив на довкілля.

У результаті четвертий етап автоматизації - це перехід до глибоко цифрової, екологічної та клієнтоорієнтованої логістики, де швидкість, адаптивність, прозорість і сталість є визначальними факторами конкурентоспроможності.

Таблиця 1.1 - Ключові аспекти, які охоплює інформаційними системами управління логістичними перевезеннями ІСУЛП

Ключовий аспект	Опис
Планування маршрутів	Оптимізація маршрутів доставки з урахуванням відстані, часу, обмежень,

Кінець таблиці 1.1

	дорожніх умов і доступності транспорту. Це зменшує витрати на паливо і час.
Управління транспортом	Відстеження транспорту в реальному часі через GPS, контроль руху, оперативне реагування на непередбачені ситуації.
Управління складами	Облік і контроль запасів, руху товарів, оптимізація розміщення, керування прийманням і відвантаженням.
Управління замовленнями	Автоматизація обробки замовлень, відстеження статусу, інформування клієнтів про стан доставки.
Звітність та аналітика	Генерація звітів для аналізу логістичних процесів, виявлення проблемних зон, прийняття обґрунтованих рішень.
Інтеграція з іншими системами	Сумісність з ERP, CRM, e-commerce платформами для забезпечення безперервного потоку інформації.

Але існують і певні проблеми, що перешкоджають максимальному використанню потенціалу інформаційних систем. До них належать недостатній рівень автоматизації, що призводить до використання ручної праці у багатьох процесах, та відсутність централізованого контролю, що ускладнює управління логістичними потоками. Також гостро стоїть питання інтеграції логістичних інформаційних систем з іншими корпоративними платформами, такими як CRM та ERP, а використання застарілого програмного забезпечення спричиняє високі витрати на підтримку.

Таблиця 1.2 - Проблеми, пов'язані з інформаційними системами управління логістичними перевезеннями (ІСУЛП)

Категорія	Проблема	Опис
Технічні проблеми	Інтеграція з існуючими системами	Складність інтеграції з ERP, CRM, GPS тощо; несумісність форматів даних і протоколів.
	Надійність та масштабованість	Потреба у стабільній роботі з великими обсягами даних у реальному часі; важливість масштабованості.
	Безпека даних	Захист конфіденційної інформації від несанкціонованого доступу та кібератак.
	Підтримка та обслуговування	Необхідність кваліфікованого персоналу; ризики через відсутність технічної підтримки.
	Застарівання технологій	Потреба в регулярному оновленні та модернізації для підтримання актуальності системи.
Організаційні проблеми	Впровадження та навчання	Потреба в часі та ресурсах для впровадження та навчання персоналу.

Кінець таблиці 1.2

	Опір змінам	Небажання персоналу приймати нові технології через страх або труднощі адаптації.
	Недостатня підтримка керівництва	Відсутність лідерської підтримки, фінансування та мотивації.
	Невизначеність цілей та показників	Неможливість оцінити успіх впровадження без чітких цілей і КРІ.
Проблеми, пов'язані з даними	Якість даних	Неповні або помилкові дані знижують ефективність системи.
	Обробка великих обсягів даних	Потреба у потужному обладнанні та ефективних алгоритмах для аналізу даних.
Проблеми, пов'язані з користувачами	Складність інтерфейсу	Незручний інтерфейс ускладнює роботу та збільшує кількість помилок.
	Недостатня підтримка користувачів	Відсутність якісної допомоги знижує ефективність використання системи.

Основні завдання спрямованих на оптимізацію та автоматизацію всіх етапів логістичного процесу можна умовно розділити на такі категорії.

#### Планування та оптимізація:

- планування маршрутів: визначення найефективніших маршрутів доставки з урахуванням різних факторів (відстань, час, дорожні умови, обмеження, доступність транспорту). Це включає в себе використання алгоритмів оптимізації для мінімізації витрат та часу доставки;
- оптимізація використання транспортних засобів: планування завантаження транспортних засобів, вибір оптимального типу транспорту залежно від вантажу та маршруту, мінімізація простоїв;
- прогнозування попиту: аналіз історичних даних для прогнозування майбутнього попиту на перевезення, що дозволяє краще планувати ресурси;
- управління запасами: моніторинг рівня запасів на складах, прогнозування потреби в товарах та оптимізація процесів зберігання.

#### Моніторинг та контроль:

- відстеження транспортних засобів: моніторинг місцезнаходження транспортних засобів у реальному часі за допомогою GPS-трекерів;
- контроль замовлень: відстеження статусу замовлень, інформування клієнтів про хід доставки;
- моніторинг виконання маршрутів: контроль дотримання запланованих маршрутів та своєчасного виконання доставки;
- виявлення та усунення проблем: швидке виявлення та усунення непередбачених ситуацій (затримки, аварії, зміни маршрутів).

#### Управління та адміністрування:

- управління транспортними засобами: облік та управління транспортним парком, планування технічного обслуговування;
- управління персоналом: контроль роботи водіїв та інших співробітників, що беруть участь у перевезеннях;

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

- управління документацією: автоматизація процесів створення та обробки транспортних документів (накладні, акти, тощо) ;
- звітність та аналітика: генерація звітів про ефективність логістичних операцій, аналіз витрат та показників продуктивності.

З огляду на це, актуальними завданнями стають досягнення повної автоматизації логістичних операцій, розробка та впровадження єдиної інтегрованої платформи, використання сучасних технологій, таких як штучний інтелект і великі дані, для аналізу та прогнозування, а також підвищення сумісності з корпоративними інформаційними системами і зниження витрат на підтримку сучасних рішень.

## 1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

У процесі вибору програмного забезпечення для автоматизації логістичних процесів підприємства можуть використовувати різні підходи. Найпоширенішими серед них є: власна розробка (custom development), використання готових програмних рішень (off-the-shelf software), хмарні рішення (cloud-based solutions) та застосування програмного забезпечення з відкритим кодом (open source).

**Власні розробки (Custom Development):** цей підхід передбачає створення програмного забезпечення з нуля відповідно до унікальних потреб підприємства.

**Готові програмні рішення (Off-the-Shelf Software):** це програмні продукти, які вже розроблені та продаються на комерційній основі. Їх можна швидко впровадити та почати використовувати одразу після покупки.

**Хмарні рішення (Cloud-Based Solutions):** хмарні технології дозволяють використовувати програмне забезпечення через інтернет без необхідності встановлення на локальні сервери. Такі рішення дедалі частіше застосовуються в логістиці завдяки своїй зручності та мобільності.

**Open Source рішення:** такий підхід передбачає використання ПЗ з відкритим програмним кодом, який можна змінювати, адаптувати та поширювати.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 1.3 – Порівняння підходів до впровадження програмного забезпечення

Критерій / Підхід	Власна розробка (Custom)	Готове ПЗ (Off-the-Shelf)	Хмарні рішення (Cloud-Based)	Open Source
Індивідуалізація	Висока	Низька	Середня	Висока
Гнучкість	Висока	Обмежена	Висока	Висока
Швидкість впровадження	Низька	Висока	Висока	Середня
Вартість впровадження	Висока	Помірна	Низька	Низька
Контроль над ПЗ	Повний	Обмежений	Обмежений	Високий
Підтримка	Залежить від розробника	Надається постачальником	Надається постачальником	Відсутня офіційна
Оновлення	Потрібно реалізовувати самостійно	Регулярні	Автоматичні	Самостійно або спільнотою
Інтеграція з іншими системами	Легка	Може бути складною	Може бути складною	Може бути складною
Безпека даних	Високий контроль	Середній контроль	Залежить від постачальника	Повністю на відповідальності користувача
Залежність від постачальника	Низька	Висока	Висока	Низька

Кінець таблиці 1.3

Потреба в ІТ-команді	Висока	Мінімальна	Мінімальна	Висока
----------------------	--------	------------	------------	--------

Приклади інформаційних систем для логістичних перевезень:



Рисунок 1.1 – Логотип компанії

SAP Transportation Management (SAP TM) - це модуль у складі SAP Supply Chain Management, призначений для планування, виконання, моніторингу та оптимізації транспортних процесів у логістиці. Він особливо корисний для великих компаній, що мають складні транспортні мережі та потребують інтеграції з іншими бізнес-процесами.

Основні можливості SAP TM:

- планування перевезень: автоматичне створення маршрутів з урахуванням вартості, часу, типу транспорту та інших параметрів;
- виконання транспортування: управління замовленнями на транспортування, документообігом, трекінгом;
- розрахунок витрат: автоматичний розрахунок транспортних витрат, тарифів та податків;
- інтеграція: працює у зв'язці з іншими модулями SAP (наприклад, SAP ERP, SAP EWM, SAP GTS) ;
- моніторинг у реальному часі: можливість відстежувати статус відправлень, отримувати сповіщення про відхилення.

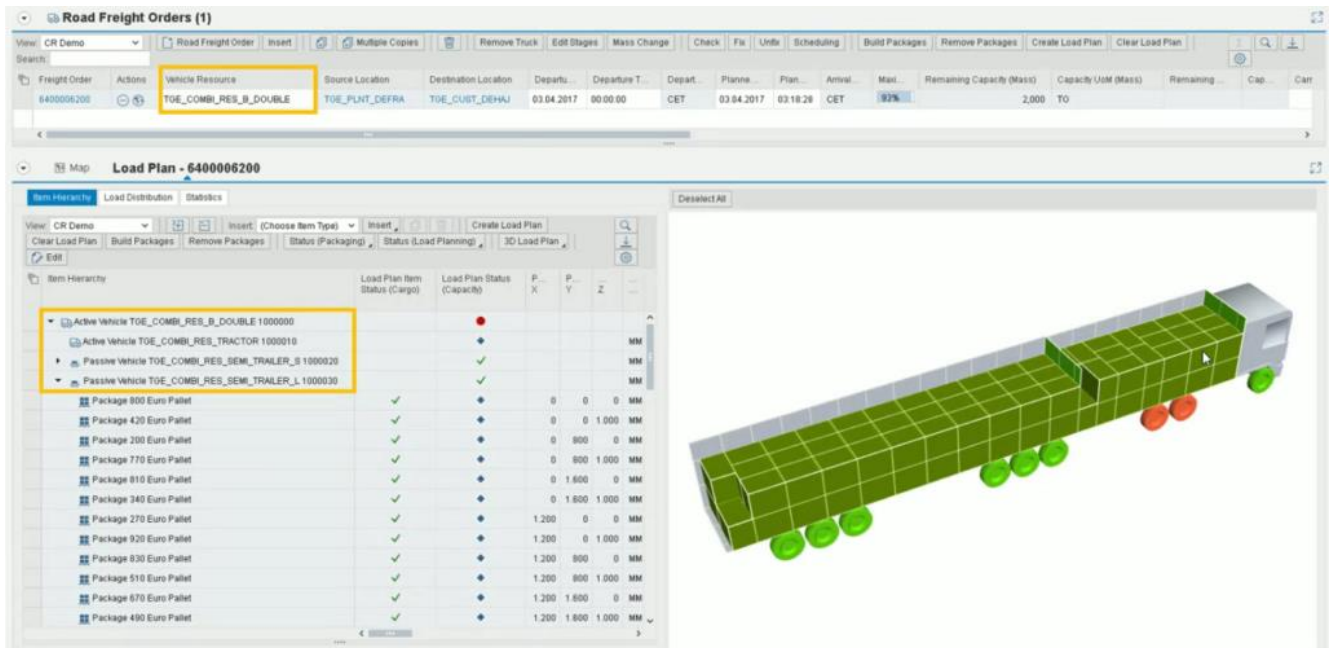


Рисунок 1.2 – Інтерфейс застосунку

Таблиця 1.4 – Переваги системи та їх роз'яснення

Інтеграція з ERP	Тісна інтеграція з SAP ERP дозволяє створювати єдину інформаційну систему.
Автоматизація	Зменшує ручну працю завдяки автоматичному плануванню маршрутів, тарифів, документів.
Оптимізація витрат	Дає змогу зменшити витрати на логістику за рахунок оптимізації маршрутів та перевізників.
Масштабованість	Підходить як для середніх, так і для великих підприємств з міжнародними перевезеннями.
Аналітика	Вбудовані інструменти ВІ для аналізу ефективності транспорту.

Таблиця 1.5 – Недоліки системи та їх роз'яснення

Висока вартість	Ліцензії SAP і впровадження можуть бути дуже дорогими.
Складність у впровадженні	Необхідна участь сертифікованих SAP-консультантів, часто тривале налаштування.
Не надто гнучкий інтерфейс	Менш інтуїтивно зрозумілий порівняно з сучасними хмарними рішеннями.
Потребує навчання	Користувачам потрібно пройти навчання через складну термінологію та логіку модулів.



Рисунок 1.3 – Логотип компанії

Oracle Transportation Management (OTM) - це потужна система для управління логістичними перевезеннями, розроблена компанією Oracle. Вона входить до складу Oracle Supply Chain Management (SCM) і орієнтована на автоматизацію, оптимізацію та контроль усіх аспектів транспортної логістики як для вантажовідправників, так і для логістичних операторів.

Основні можливості OTM:

- планування перевезень: оптимізація маршрутів, складання графіків, вибір перевізників;
- виконання перевезень: управління транспортними замовленнями, документообігом, інвойсами;

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

- фінансовий контроль: розрахунок вартості перевезень, аудит рахунків, автоматизовані платежі;
- аналітика і звітність: вбудовані ВІ-інструменти та дашборди для оцінки ефективності логістики;
- підтримка міжнародних перевезень: врахування митних, податкових та регуляторних вимог;
- хмарна архітектура: підтримує як локальне, так і хмарне розгортання.

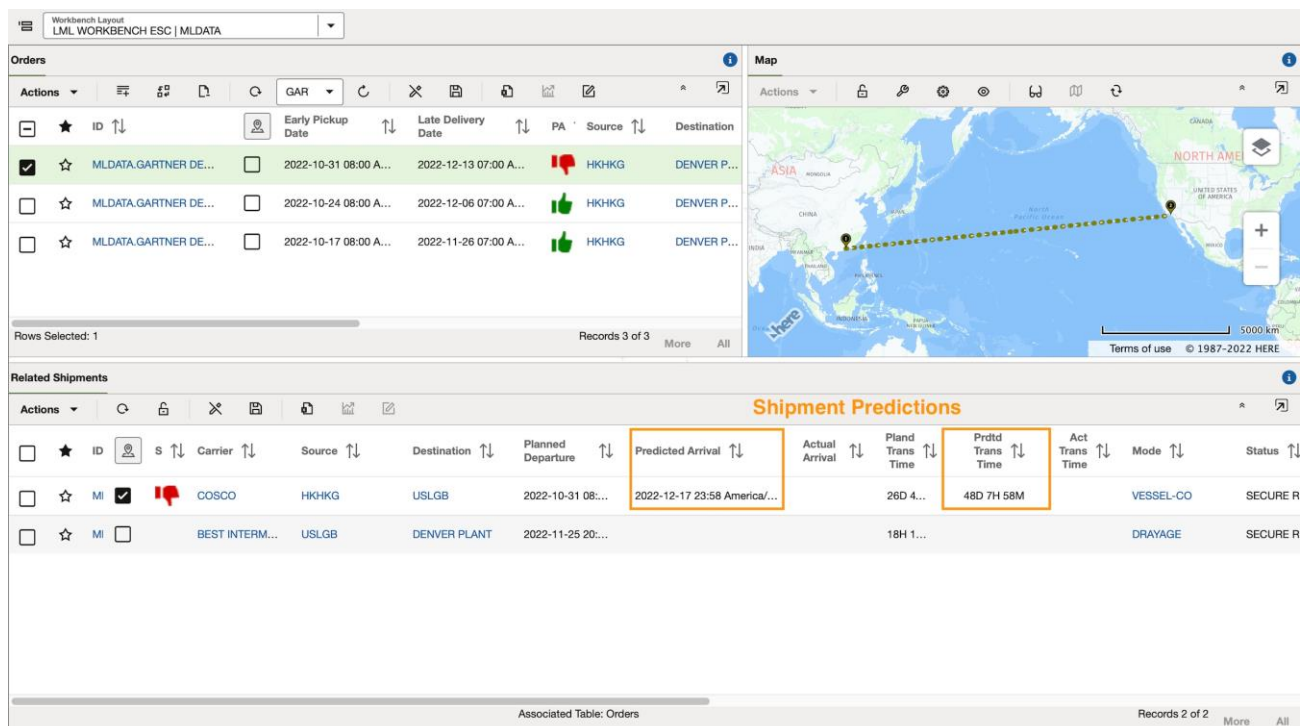


Рисунок 1.4 – Інтерфейс застосунку

Таблиця 1.6 – Переваги системи та їх роз'яснення

Гнучкість та масштабованість	Підходить як для середнього, так і для великого бізнесу, легко масштабувати при зростанні обсягів.
Хмарні технології	Забезпечує доступ з будь-якої точки світу, швидке оновлення функцій, зменшення витрат на ІТ-інфраструктуру.

Кінець таблиці 1.6

Глибока аналітика	Потужні інструменти візуалізації, прогнозування, виявлення проблем у ланцюгу поставок.
Інтеграція	Добре інтегрується з ERP-системами (зокрема Oracle E-Business Suite, Oracle Fusion Cloud ERP), WMS, CRM.
Автоматизація	Можливість створювати складні транспортні сценарії, що знижують вплив людського фактора.

Таблиця 1.7 – Недоліки системи та їх роз'яснення

Висока вартість ліцензування	Як і інші продукти Oracle, система дорога у придбанні та обслуговуванні.
Складність налаштування	Вимагає кваліфікованих спеціалістів для впровадження та підтримки.
Інтерфейс	Деякі користувачі відзначають, що інтерфейс потребує покращення з точки зору зручності.
Велика кількість параметрів	Через велику гнучкість система може бути перевантаженою для простих завдань.
Залежність від хмари	Для хмарної версії потрібен стабільний інтернет, що може бути проблемою у деяких регіонах.

Microsoft Dynamics 365 Supply Chain Management - це потужна ERP-система, яка дозволяє компаніям ефективно управляти ланцюгами постачання, включаючи логістику, виробництво, складування, поповнення запасів і планування. Це рішення є частиною платформи Microsoft Dynamics 365 і тісно інтегрується з іншими продуктами Microsoft, такими як Power BI, Microsoft Teams та Azure IoT.



Рисунок 1.5 – Логотип компанії

Основні можливості Dynamics 365 Supply Chain Management:

- управління запасами: контроль залишків у реальному часі, поповнення на основі попиту;
- транспортна логістика: управління вантажами, транспортними замовленнями, маршрутами та перевізниками;
- складська логістика: розширені функції WMS (Warehouse Management System) ;
- виробництво: підтримка дискретного, безперервного, проектного та змішаного виробництва;
- прогнозування попиту: використання AI для прогнозування замовлень та оптимізації закупівель;
- аналітика: вбудовані звіти та візуалізації на основі Power BI;
- інтеграція з IoT: моніторинг стану обладнання та автоматизація технічного обслуговування.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

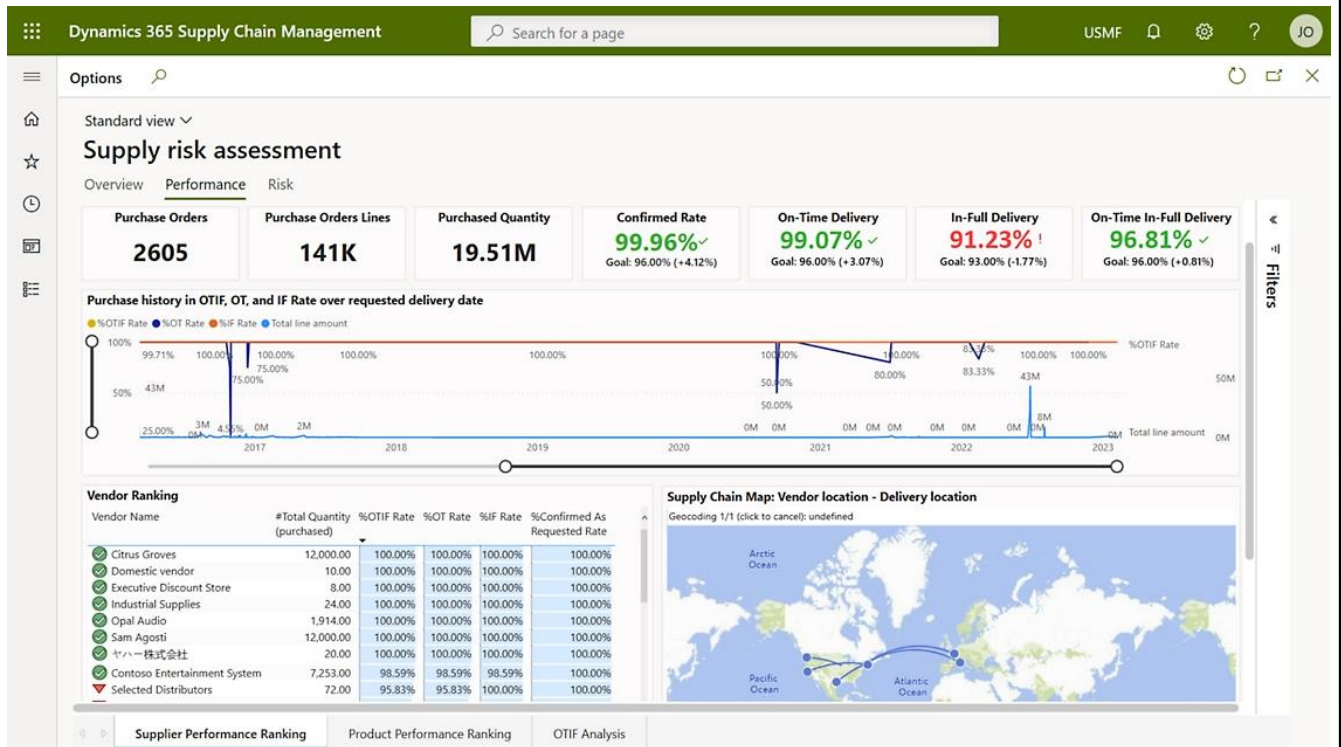


Рисунок 1.6 – Інтерфейс застосунку

Таблиця 1.8 – Переваги системи та їх роз'яснення

Хмарна та гібридна інфраструктура	Можливість працювати у хмарі, на локальному сервері або у змішаній конфігурації.
Гнучке налаштування	Можливість адаптації під специфіку бізнесу без значних витрат.
AI та машинне навчання	Використання вбудованих інтелектуальних сервісів для прогнозування попиту та оптимізації ланцюга постачань.
Зрозумілий інтерфейс	Інтерфейс, знайомий користувачам Microsoft Office, зменшує час на навчання.
Регулярні оновлення	Microsoft активно розвиває Dynamics 365, додаючи нові функції та вдосконалення.

Таблиця 1.9 – Недоліки системи та їх роз'яснення

Висока загальна вартість володіння	Абонентська плата, розширення та інтеграції можуть бути дорогими для малого бізнесу.
Початкові витрати на впровадження	Налаштування і адаптація потребують часу та фахівців.
Не підходить для дуже складної логістики	У порівнянні з SAP TM або Oracle OTM, менше «глибини» в логістичних модулях.
Залежність від Azure	Вся хмарна інфраструктура прив'язана до Microsoft Azure, що обмежує гнучкість у виборі хостингу.
Може бути надмірним для малого бізнесу	Повноцінна система має багато функцій, не потрібних для простих операцій.

### 1.3 Підходи до вирішення задачі за темою дослідження

У вирішенні задачі розробки та впровадження інформаційної системи управління логістичними перевезеннями (ІСУЛП) важливо враховувати як загальні принципи інформаційного моделювання, так і специфіку логістичних процесів. Оскільки логістика охоплює динамічні та взаємопов'язані компоненти транспортування, складування, облік ресурсів, маршрутизацію, підхід до вирішення має бути комплексним, гнучким та технологічно адаптивним. Одним із базових методологічних підходів є системний підхід, який передбачає цілісне бачення логістичного процесу як сукупності взаємодіючих підсистем.

Інформаційна система управління логістичними перевезеннями (ІСУЛП) повинна забезпечувати гармонійну та злагоджену взаємодію модулів транспортного планування, моніторингу доставки, управління складськими запасами, аналітики, звітності, а також координації взаємодії з клієнтами і

постачальниками. Це дозволяє забезпечити повну прозорість всіх етапів логістичного ланцюга, суттєво зменшити витрати часу і ресурсів та оперативно адаптуватися до змін зовнішнього середовища.

В умовах сучасності ефективно управління перевезеннями значною мірою залежить від активного застосування цифрових технологій. Серед таких технологій варто виокремити використання GPS-трекерів для постійного моніторингу транспорту, застосування передових алгоритмів оптимізації для створення маршрутів з урахуванням завантаженості доріг і часу доставки, електронний документообіг, який значно скорочує витрати часу на обробку заявок і звітності, а також впровадження мобільних додатків, що використовуються водіями та диспетчерами. Ці інноваційні рішення підвищують оперативність, знижують ризик людської помилки та створюють цілісний цифровий простір для ефективної взаємодії.

Для максимальної результативності ІСУЛП має бути тісно інтегрована з іншими інформаційними системами підприємства, такими як ERP (системи управління ресурсами підприємства), CRM (системи управління відносинами з клієнтами) і WMS (системи управління складами). Завдяки такій інтеграції формується єдина інформаційна екосистема, в якій усі компоненти взаємодіють у режимі реального часу, а дані централізовано обробляються. Це дозволяє уникнути дублювання інформації, сприяє швидкому прийняттю рішень на основі комплексного аналізу даних та поліпшує взаємодію між логістичним, закупівельним, торговельним та клієнтським відділами підприємства.

Крім того, ефективна ІСУЛП має характеризуватися модульною архітектурою, що забезпечує поетапне впровадження компонентів, таких як управління доставкою чи складські операції, дозволяє легко адаптуватися до розвитку компанії або змін у логістичних схемах, а також інтегрувати новітні технології, такі як штучний інтелект для прогнозування можливих затримок чи блокчейн для підвищення безпеки транзакцій. Модульність також спрощує процеси технічного обслуговування та оновлення системи, суттєво зменшуючи

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

ризика зупинки всієї логістичної діяльності через можливий збій одного з елементів.

В останній час особливого значення набуває зручність інтерфейсу користувача. Це передбачає розробку інтуїтивно зрозумілих і простих інструментів для диспетчерів, водіїв і менеджерів, наявність доступних мобільних додатків, візуально зрозумілих дашбордів, автоматизацію рутинних завдань (наприклад, автоматичне заповнення звітів чи маршрутних листів) і підтримку багатомовності для забезпечення комфортної роботи користувачів у міжнародному середовищі.

Орієнтація на зручність користування підвищує швидкість навчання персоналу, ефективність використання системи та рівень задоволення співробітників.

#### 1.4 Висновки до першого розділу

У ході дослідження було розглянуто ключові аспекти побудови сучасної логістичної ІС, проаналізовано предметну область, виявлено актуальні проблеми галузі, порівняно існуючі рішення та окреслено методологічні підходи до вирішення задачі.

На етапі аналізу предметної області встановлено, що сучасна логістика є надзвичайно складною системою, яка вимагає високого рівня координації між усіма ланками транспортного процесу - від планування маршрутів і управління автопарком до контролю складських залишків. Основні проблеми, з якими стикаються компанії, включають низьку прозорість процесів, слабку інтеграцію інформаційних потоків, втрати часу при ручній обробці даних та недостатню гнучкість у реагуванні на змінні умови.

У порівняльному аналізі існуючих рішень було розглянуто як готові програмні продукти (наприклад, Oracle Transportation Management, Microsoft Dynamics 365 Supply Chain Management), так і можливості власних розробок. Встановлено, що хоча промислові рішення забезпечують широкий функціонал і масштабованість, вони мають високу вартість і вимагають адаптації під специфіку

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

бізнесу. У той час як власні розробки дозволяють повністю врахувати потреби компанії, але потребують значних ресурсів для розробки та супроводу.

У розділі підходів до вирішення задачі було обґрунтовано необхідність використання системного та інтеграційного підходів, цифровізації логістичних процесів, модульної архітектури та орієнтації на зручність користувача. Впровадження таких підходів дозволяє створити ефективну ІС, яка забезпечує цілісність і оперативність управління перевезеннями, скорочує витрати та підвищує якість обслуговування клієнтів.

На основі постановки задачі було сформульовано ключові цілі, яких має досягати інформаційна система: автоматизація процесів планування та виконання перевезень, підвищення точності обліку, забезпечення прозорості логістичного ланцюга, інтеграція з іншими обліковими та управлінськими системами.

Отже, впровадження інформаційної системи управління логістичними перевезеннями є стратегічно важливим кроком для підприємств, які прагнуть підвищити свою конкурентоспроможність, забезпечити надійність логістичних операцій та адаптуватися до викликів цифрової економіки. Рациональний вибір платформи, грамотна постановка завдань та впровадження сучасних технологій є запорукою успішної цифрової трансформації логістики.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

# ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПЕРЕВЕЗЕННЯМИ

## 2.1 Визначення вимог і технічних характеристик

Проєктування інформаційної системи управління логістичними перевезеннями (ІСУЛП) є критично важливим етапом у створенні ефективного цифрового рішення, що забезпечує управління транспортними і складськими процесами сучасного підприємства. Даний процес передбачає комплексний підхід до розробки технологічного інструменту, здатного істотно оптимізувати логістичні операції, підвищити прозорість кожного етапу виконання завдань, а також суттєво покращити ефективність використання наявних ресурсів.

На початковому етапі проєктування необхідно детально визначити стратегічні цілі інформаційної системи, що повинні корелювати зі загальною стратегією розвитку компанії. Головною метою впровадження ІСУЛП є автоматизація рутинних та ресурсомістких операцій, таких як формування оптимальних маршрутів доставки, управління графіками та маршрутами перевезень, контроль стану та місцезнаходження транспортних засобів, а також ефективне управління складськими запасами. Це сприятиме мінімізації людських помилок, забезпеченню високої точності і швидкості прийняття управлінських рішень, що, у свою чергу, підвищить загальну ефективність логістичного управління.

Ключовим аспектом проєктування ІСУЛП є формування загальної архітектури системи. Пропонована архітектура має бути багаторівневою, з чітко визначеними компонентами та зонами відповідальності. Клієнтська частина повинна забезпечувати легкий і зрозумілий веб-інтерфейс, а також мобільні застосунки, адаптовані для зручного використання різними категоріями співробітників — диспетчерами, логістами, водіями та керівниками. Серверна компонента має бути потужною і надійною, відповідати за виконання основної бізнес-логіки, обробку великих обсягів даних, інтеграцію з зовнішніми сервісами

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

та іншими корпоративними інформаційними системами. Така організація сприяє стабільності роботи, дозволяє швидко і надійно обробляти дані у реальному часі та значно скорочує ризики збоїв та простоїв системи.

Особливу увагу слід приділити проектуванню бази даних, яка є центральним компонентом інформаційної системи. База даних, що базується на реляційній моделі, забезпечує ефективне зберігання великих обсягів інформації, що стосується транспортних засобів, водіїв, маршрутів, замовлень та клієнтів. Всі ці об'єкти взаємопов'язані та дозволяють швидко та ефективно виконувати аналітичні запити, генерувати звітність і здійснювати оперативний контроль за станом перевезень. Використання реляційних баз даних гарантує підтримку цілісності та достовірності даних протягом усього життєвого циклу системи.

Важливою частиною проектування є забезпечення зручності використання інформаційної системи для всіх категорій кінцевих користувачів. Інтерфейс системи повинен бути простим і зрозумілим, з чіткою логікою побудови, забезпечувати швидке введення, пошук та перегляд інформації. Інтерфейс має бути інтуїтивно зрозумілим, враховувати специфіку роботи різних користувачів та бути доступним на різних пристроях — від персональних комп'ютерів до мобільних телефонів та планшетів.

Окрім цього, особливу роль у проектуванні відіграють сценарії використання системи (Use Cases). Докладне опрацювання сценаріїв дозволяє систематизувати та врахувати всі можливі варіанти використання системи: створення замовлень, їх підтвердження та призначення, контроль за процесом перевезення, завершення замовлення та формування регулярних звітів. Таке детальне описання сценаріїв значно полегшує подальше тестування системи, забезпечує її відповідність визначеним вимогам, а також сприяє швидкій адаптації та ефективній роботі персоналу з новою інформаційною системою.

Отже, якісне проектування інформаційної системи управління логістичними перевезеннями забезпечує створення потужного, зручного та надійного

інструменту, здатного оптимізувати логістичні процеси підприємства, покращити управлінські рішення та підвищити загальну ефективність роботи компанії.

## 2.2 Опис середовища розробки Visual Studio Code

Visual Studio Code (VS Code) - це потужне кросплатформне середовище розробки, створене компанією Microsoft. Воно поєднує простоту текстового редактора з широкими можливостями інтегрованої IDE. Це середовище відзначається високою продуктивністю, гнучкістю та активною спільнотою розробників.

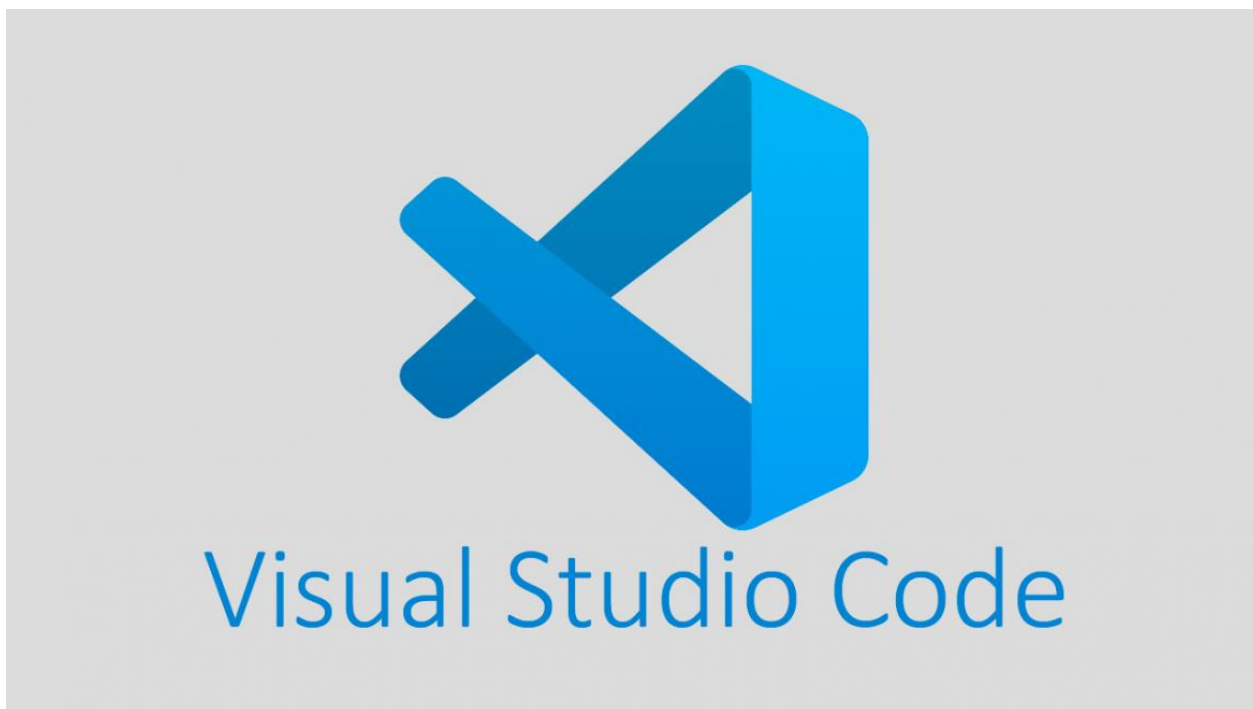


Рисунок 2.1 - Логотип програмного середовища

На відміну від традиційних середовищ розробки, VS Code орієнтований на модульність. Користувачі самостійно формують набір необхідних інструментів шляхом встановлення розширень. Система дозволяє розробляти програмне забезпечення практично будь-якої складності - від веб-додатків до складних корпоративних рішень.

Таблиця 2.1 – основні функціональні можливості VS Code

Категорія	Опис
Підтримка мов	Python, JavaScript, C/C++, Java, PHP, HTML, CSS та інші
Інтелектуальне доповнення	IntelliSense: підказки, автозаповнення, визначення типів, імпортів
Вбудований термінал	Можливість запуску команд без виходу із середовища
Налагодження коду	Точки зупинки, перегляд змінних, керування виконанням
Інтеграція з Git	Коміти, гілки, злиття, перегляд історії змін
Плагіни та розширення	Підтримка тисячі плагінів: від лінтерів до інструментів моделювання

Інтерфейс Visual Studio Code простий та інтуїтивний. Він оптимізований для максимально ефективної роботи з кодом, залишаючи достатньо простору для навігації та роботи з проєктами. Умовно інтерфейс поділяється на шість основних областей:

Таблиця 2.2 - Структура інтерфейсу користувача

Компонент інтерфейсу	Опис
Редактор (Editor)	Основна зона для редагування файлів. Підтримує вертикальне й горизонтальне розділення вкладок.
Основна бічна панель	Включає переглядач файлів (Explorer), що допомагає навігувати по структурі проєкту.
Додаткова бічна панель	Знаходиться навпроти основної. За замовчуванням містить вікно Chat.

## Кінець таблиці 2.2

Рядок стану (Status Bar)	Виводить інформацію про відкритий проєкт, мову, Git-гілку, повідомлення про помилки.
Панель активності	Ліворуч. Дозволяє перемикатися між режимами перегляду, Git, розширеннями тощо.
Нижня панель (Panel)	Знаходиться під редактором. Містить термінал, журнал виводу, дебаг-повідомлення.

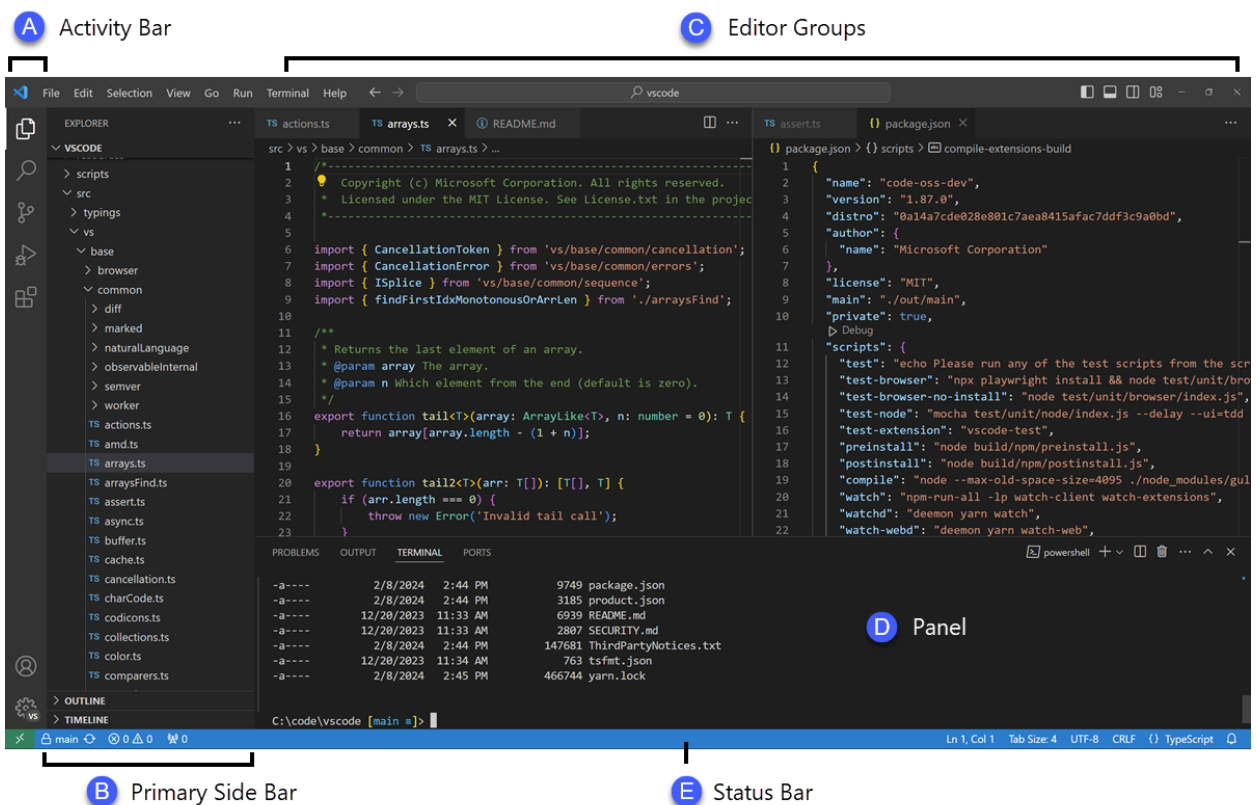


Рисунок 2.2 - Інтерфейс програмного середовища

У процесі розробки інформаційної системи управління логістичними перевезеннями (ІСУЛП) надзвичайно важливо вибрати ефективне та зручне

середовище розробки, яке дозволить комплексно охопити всі етапи - від проектування бази даних до реалізації функціонального інтерфейсу користувача. Visual Studio Code (VS Code) є саме таким інструментом. Завдяки своїй гнучкості, відкритості та великій кількості розширень, це середовище оптимально підходить для реалізації проєктів подібного масштабу.

Під час реалізації ІСУЛП, Visual Studio Code дозволяє ефективно поєднувати роботу з серверною частиною (backend), клієнтським інтерфейсом (frontend), базами даних, REST API та системами контролю версій. Наприклад, серверна логіка системи, яка обробляє маршрутизацію, авторизацію користувачів та взаємодію з базою даних, може бути реалізована за допомогою фреймворків Django або Flask на мові Python. Для цього у VS Code використовуються розширення, що забезпечують автодоповнення синтаксису, відладку коду, підсвітку помилок та запуск серверу безпосередньо з редактора.

Фронтенд-частина системи - інтерфейс користувача, з яким взаємодіють диспетчери, водії та менеджери - може бути реалізована на базі JavaScript-фреймворків, зокрема React або Vue.js. VS Code забезпечує зручність роботи з HTML, CSS і JavaScript/TypeScript, дозволяючи швидко тестувати компоненти інтерфейсу та інтегрувати їх із серверною частиною.

Також важливу роль відіграє робота з базою даних. Для ІСУЛП це зазвичай реляційна база типу PostgreSQL або MySQL, яка містить інформацію про маршрути, замовлення, водіїв та транспортні засоби. За допомогою розширення SQLTools розробник має змогу створювати структуру таблиць, писати SQL-запити, здійснювати тестування та перевірку зв'язків між сутностями бази даних.

Важливою особливістю є тестування REST API, що відповідає за обмін даними між компонентами системи. Завдяки плагіну REST Client у VS Code можна тестувати HTTP-запити прямо в редакторі, не використовуючи зовнішні інструменти, такі як Postman.

Крім того, для документування архітектури системи, створення UML-діаграм та ER-моделей у Visual Studio Code використовуються розширення типу PlantUML

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

або Draw.io Integration. Це дозволяє тримати як код, так і документацію в межах єдиного середовища, що особливо зручно для командної розробки.

Окремо варто згадати інтеграцію з системою контролю версій Git. Вона дозволяє зручно відстежувати зміни в коді, створювати гілки, виконувати злиття та працювати з репозиторіями, наприклад GitHub чи GitLab. Це особливо корисно при роботі над проектом кількома учасниками.

Таким чином, Visual Studio Code виступає не просто текстовим редактором, а повноцінним середовищем для комплексної розробки ІСУЛП. Його можливості значно спрощують усі етапи створення інформаційної системи - від моделювання до тестування та підтримки.

### 2.3 Використання Tailwind CSS у проекті ІСУЛП

Tailwind CSS – це популярний, сучасний фреймворк для створення стилів веб-додатків і сайтів, що використовує утилітарний підхід. Він надає розробникам велику кількість попередньо визначених класів CSS, які можна безпосередньо застосовувати в HTML, не створюючи додаткових CSS-файлів.

Tailwind CSS відрізняється від традиційних фреймворків, таких як Bootstrap або Foundation, своєю концепцією, яка дозволяє будувати кастомні, гнучкі та продуктивні дизайни без зайвих стилів.



Рисунок 2.3 – Лого

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 32
Зм.	Арк.	№ докум.	Підпис	Дата		

## Основні принципи Tailwind CSS.

### Утилітарні класи

У Tailwind CSS майже немає попередньо створених компонентів (як кнопки чи меню), натомість є набір базових утилітарних класів, які керують такими властивостями, як:

- розміри (ширина, висота, padding, margin);
- типографія (шрифт, розмір шрифту, жирність) ;
- кольори (тексту, фону, рамок) ;
- гнучкість (flexbox, grid) ;
- адаптивність (responsive design).

### Адаптивність (Responsive Design)

Tailwind CSS має вбудовану підтримку адаптивних класів. Використовуючи префікси, ви можете налаштувати елементи для різних розмірів екранів:

sm:, md:, lg:, xl:, 2xl: – відповідають різним розмірам екранів від мобільних пристроїв до великих моніторів.

### Тематизація та конфігурація

Tailwind CSS дуже гнучкий і легко конфігурується. Основні параметри дизайну можна налаштовувати через спеціальний файл конфігурації (tailwind.config.js), де визначаються кольори, шрифти, розміри тощо.

Завдяки утилітарним класам значно скорочується час на написання стилів. Розробники менше часу витрачають на перемикання між HTML та CSS-файлами, а більше – на власне реалізацію функціоналу.

Утилітарні класи створюють єдиний стильовий простір, в якому легко підтримувати консистентність інтерфейсу, особливо при роботі в команді.

Tailwind CSS використовує систему "tree-shaking" (очищення від зайвого коду), що дозволяє створювати максимально оптимізовані CSS-файли. Це сприяє швидшому завантаженню сторінок і кращій продуктивності веб-додатків.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

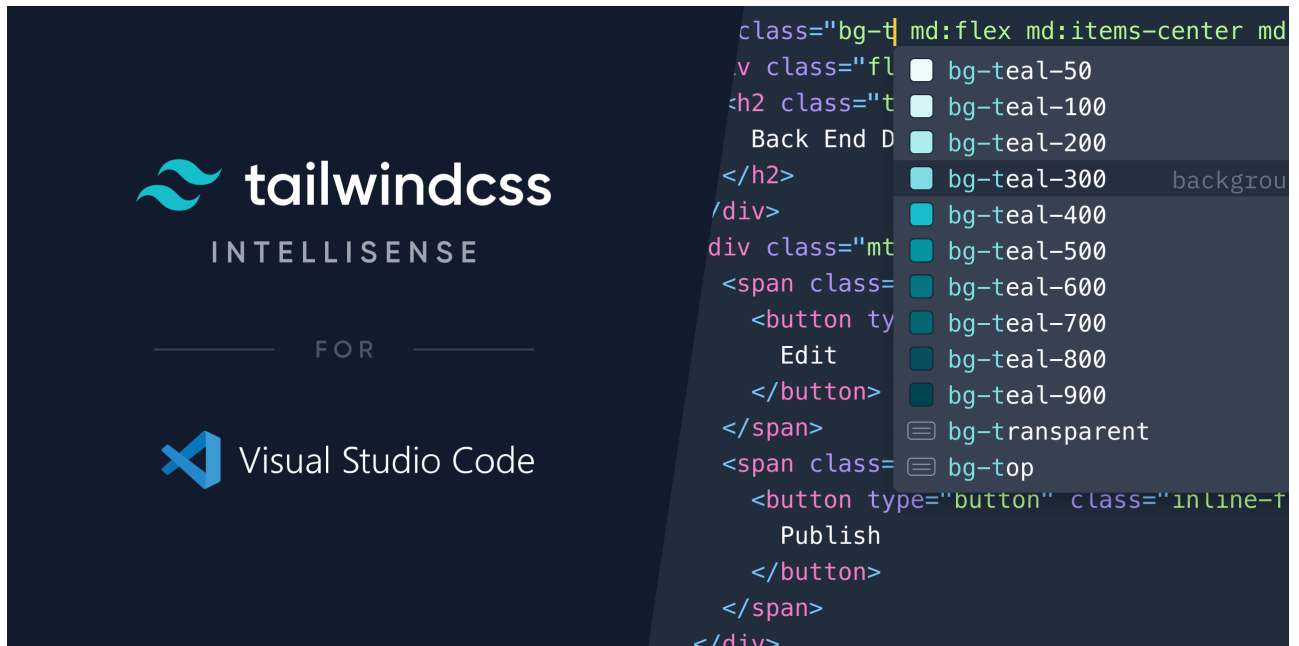


Рисунок 2.4 – Інтеграція Tailwind CSS в VS Code

Конфігурація Tailwind CSS дуже гнучка. Ви легко можете додавати власні кольори, шрифти та інші властивості, що дозволяє пристосовувати стилі до будь-якого проєкту.

Tailwind CSS можна легко інтегрувати у ваші проєкти через популярні інструменти збірки (Webpack, Vite), або навіть CDN для швидкого тестування.

Tailwind має розвинуту екосистему, що включає додаткові інструменти:

- Tailwind UI – преміум-компоненти, готові для використання у ваших проєктах.
- Headless UI – набір доступних та стилізованих компонентів (dropdown, modal, tab).
- Tailwind Play – інтерактивний онлайн-редактор, що дозволяє швидко тестувати стилі без налаштування локального середовища.

Tailwind CSS є ефективним рішенням для сучасної веб-розробки, яке дозволяє значно прискорити процес створення стильових рішень, забезпечити консистентність та продуктивність вашого застосунку. Його гнучкість та адаптивність роблять цей фреймворк універсальним вибором як для невеликих проєктів, так і для складних веб-застосунків.

## 2.4 JavaScript та React: основа сучасної веб-розробки

У сучасному світі веб-розробки JavaScript посідає центральне місце серед мов програмування. Його значення виходить далеко за межі простої реалізації інтерактивності на сайтах. Вперше представлений у 1995 році, JavaScript розпочав своє існування як інструмент для покращення динаміки веб-сторінок. Він швидко набув популярності завдяки своїй універсальності та простоті інтеграції у браузерне середовище. З розвитком Інтернету JavaScript еволюціонував у повноцінну багатопарадигмнену мову програмування, яка сьогодні використовується як на клієнтській, так і на серверній стороні.

Мова JavaScript має низку важливих особливостей. Вона є інтерпретованою, тобто код виконується безпосередньо у браузері без попередньої компіляції. Крім того, вона підтримує асинхронність виконання, що дозволяє створювати швидкі та ефективні додатки, які не блокують інтерфейс під час обробки великих обсягів даних чи очікування відповіді від сервера. Ще однією ключовою характеристикою є динамічна типізація – розробник може оголосити змінну без чітко визначеного типу, що спрощує синтаксис та прискорює процес розробки.

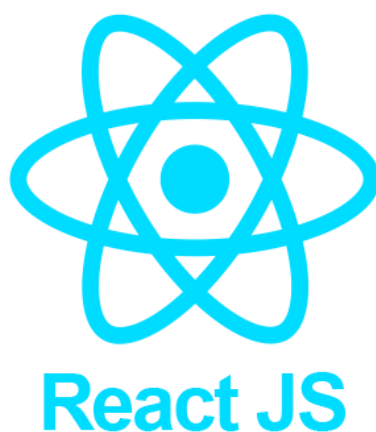


Рисунок 2.5 – Лого

Особливе значення має тісна інтеграція JavaScript із DOM (Document Object Model), яка дозволяє в реальному часі змінювати структуру та вигляд веб-сторінки.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 35
Зм.	Арк.	№ докум.	Підпис	Дата		

Саме завдяки цьому з'явилася можливість створення так званих SPA (Single Page Applications) - односторінкових застосунків, які змінюють вміст без повного перезавантаження сторінки.

На основі JavaScript було створено чимало бібліотек і фреймворків, однак однією з найвідоміших і найпотужніших є React. Це відкрита JavaScript-бібліотека, розроблена у стінах компанії Facebook у 2013 році. React від самого початку був спрямований на спрощення побудови складних інтерфейсів за рахунок використання компонентного підходу. Суть цього підходу полягає в тому, що вся веб-сторінка або веб-додаток розбивається на окремі незалежні компоненти, кожен з яких відповідає за власну частину інтерфейсу. Наприклад, кнопка, форма введення чи навігаційне меню - усе це окремі компоненти, які можна створити, протестувати й повторно використовувати.

Важливою особливістю React є те, що він використовує декларативний стиль програмування. Замість того, щоб поетапно описувати всі дії, необхідні для зміни інтерфейсу (імперативний стиль), розробник просто вказує, який вигляд повинен мати компонент у конкретному стані. React самостійно обчислює, які зміни потрібно внести до реального DOM, використовуючи проміжний рівень - так званий Virtual DOM. Це дозволяє мінімізувати кількість операцій із DOM, які є відносно повільними, і значно покращити продуктивність веб-додатків.

Однією з ключових концепцій React є використання JSX - спеціального синтаксису, що поєднує HTML-подібний код із JavaScript. JSX дозволяє описувати структуру інтерфейсу у звичній для верстальника формі, але при цьому дає змогу використовувати логіку мови програмування. Наприклад, можна створити динамічний список або змінювати вигляд компонента залежно від переданих параметрів.

```

15
16  const LOCALE = globalThis.navigator.language
17
18  const div = document.body.appendChild(document.createElement('div'))
19  const list = div.appendChild(document.createElement('ol'))
20
21  const dayNames = new Map()
22
23  for (let i = 0; i < 7; ++i) {
24    const d = Temporal.PlainDate.from({
25      year: Temporal.Now.plainDateISO().year,
26      month: 1,
27      day: i + 1,
28    })
29
30    dayNames.set(d.dayOfWeek, d.toLocaleString(LOCALE, { weekday: 'long' }))
31  }
32
33  for (const num of [...dayNames.keys()].sort((a, b) => a - b)) {
34    list.appendChild(Object.assign(
35      document.createElement('li'),
36      { textContent: dayNames.get(num) },
37    ))
38  }
39

```

Рисунок 2.6 – Синтаксис JavaScript

React-компоненти бувають функціональними або класовими. Хоча історично використовувались обидва типи, з появою хуків (useState, useEffect та інших) функціональні компоненти стали основним способом створення інтерфейсів у сучасних React-додатках. Вони є більш лаконічними, легкими для тестування та підтримки. Стан компонентів, або state, визначає їхню поведінку та вигляд. Наприклад, лічильник, що збільшується при натисканні кнопки, змінює свій стан, а отже й оновлює інтерфейс.

Крім локального стану, React дозволяє працювати з глобальним станом, що особливо актуально у великих застосунках. Для цього існують вбудовані механізми, як-от Context API, а також сторонні бібліотеки, наприклад, Redux, MobX або Recoil. Ці інструменти дозволяють організувати складні зв'язки між компонентами, уникати зайвих пропсів і забезпечують централізоване управління даними.

Ще однією важливою складовою React-додатків є роутінг - навігація між сторінками. У класичних веб-додатках кожна сторінка відповідає окремому HTML-

файлу. У React використовується підхід до SPA, коли навігація реалізується без перезавантаження. Це можливо завдяки бібліотеці React Router, яка дозволяє створювати маршрути, обробляти переходи й передавати параметри між сторінками.

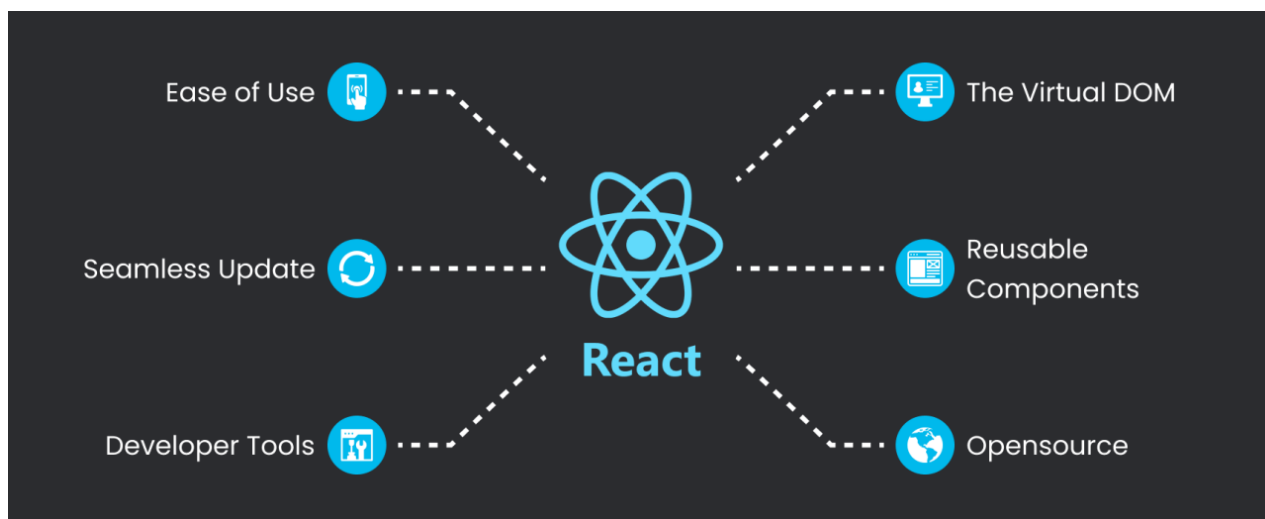


Рисунок 2.7 – Особливості React

React також прекрасно інтегрується із запитам до зовнішніх API. Зазвичай для цього використовують вбудовану функцію `fetch` або сторонню бібліотеку `axios`. Запити до серверів здійснюються всередині хука `useEffect`, що дозволяє виконувати асинхронні дії під час завантаження компонента або зміни певних залежностей.

У процесі розробки React-проектів активно використовуються різноманітні інструменти для створення, тестування та розгортання. Наприклад, `Create React App` - це стандартне середовище, що забезпечує базову конфігурацію для нових застосунків. Більш сучасним і продуктивним рішенням є `Vite` - збірник, що забезпечує блискавичний старт і миттєве оновлення під час розробки. У випадках, коли потрібна серверна генерація сторінок або повноцінна SEO-оптимізація, розробники звертаються до фреймворку `Next.js`, який ґрунтується на React, але додає можливості `SSR` (Server Side Rendering) і `SSG` (Static Site Generation).

Завершуючи, слід зазначити, що тандем JavaScript і React є надзвичайно потужним інструментом для створення сучасних, адаптивних та швидкодіючих

веб-додатків. Завдяки великій спільноті, постійному оновленню та багатій екосистемі бібліотек ці технології залишаються на передовій веб-інженерії, і їхнє вивчення є надзвичайно корисним для кожного майбутнього розробника.

## 2.5 Висновки до другого розділу

У процесі розробки інформаційної системи управління логістичними перевезеннями (ІСУЛП) особливу увагу було приділено вибору ефективних інструментів програмування та технологій, що забезпечують реалізацію функціональної, гнучкої та масштабованої інформаційної системи. На основі проведеного аналізу можна зробити низку важливих висновків щодо доцільності використання Visual Studio Code та Tailwind CSS у межах даного проєкту.

Visual Studio Code зарекомендував себе як універсальне середовище розробки, здатне забезпечити повний цикл програмної реалізації - від написання коду до тестування і документування. Його модульна архітектура, підтримка різних мов програмування, інтеграція з системами контролю версій, а також велика кількість розширень дозволили ефективно реалізувати як серверну, так і клієнтську частини системи. Завдяки плагінам типу SQLTools, REST Client, PlantUML та іншим, розробка була значно спрощена і прискорена.

Обидва інструменти - Visual Studio Code та Tailwind CSS - ефективно взаємодіють у межах загальної архітектури системи, забезпечуючи узгоджену роботу фронтенду, бекенду та бази даних. Їхнє використання дозволило не лише реалізувати повноцінний прототип ІСУЛП, а й створити умови для подальшого розширення системи відповідно до зростаючих потреб підприємства.

Отже, обрані програмні засоби повністю задовольняють потреби розробки інформаційної системи, що робить їх оптимальними для використання у проєкті.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 39
Зм.	Арк.	№ докум.	Підпис	Дата		

# ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ЛОГІСТИЧНИМИ ПЕРЕВЕЗЕННЯМИ

## 3.1 Опис розробки інформаційної системи управління логістичними перевезеннями

Розробка інформаційної системи управління логістичними перевезеннями є важливим завданням для сучасних підприємств, що займаються перевезеннями товарів і вантажів. Основними цілями такої системи є оптимізація процесів управління замовленнями, контроль за транспортними засобами та водіями, а також забезпечення високої ефективності та зручності використання.

Першим етапом розробки стало визначення вимог та аналіз предметної області. В процесі аналізу були виділені ключові функції, необхідні для ефективного управління. Основними серед них є реєстрація та управління інформацією про водіїв, облік та управління замовленнями, автоматичне збереження інформації та її швидке відновлення, простий і зрозумілий користувацький інтерфейс, а також можливість масштабування та інтеграції з іншими сервісами і системами.

Архітектура системи базується на принципі односторінкового додатку (SPA), що дозволяє перемикатися між розділами без перезавантаження сторінки, забезпечуючи високу швидкість роботи і комфорт користування.

Інтерфейс системи складається з кількох ключових елементів, кожен з яких відіграє окрему роль у процесі логістичного управління.

Після запуску вебінтерфейсу інформаційної системи управління логістичними перевезеннями, відбувається ініціалізація ключових структур та вкладок інтерфейсу. Це забезпечує миттєву готовність користувача до взаємодії із системою. Компонент App() автоматично створює три функціональні розділи: панель керування, замовлення та водії. Ці вкладки реалізовані як частини єдиного динамічного інтерфейсу, який не потребує перезавантаження сторінки під час перемикання.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

При запуску додаток завантажує попередньо збережені дані з локального сховища браузера — localStorage.

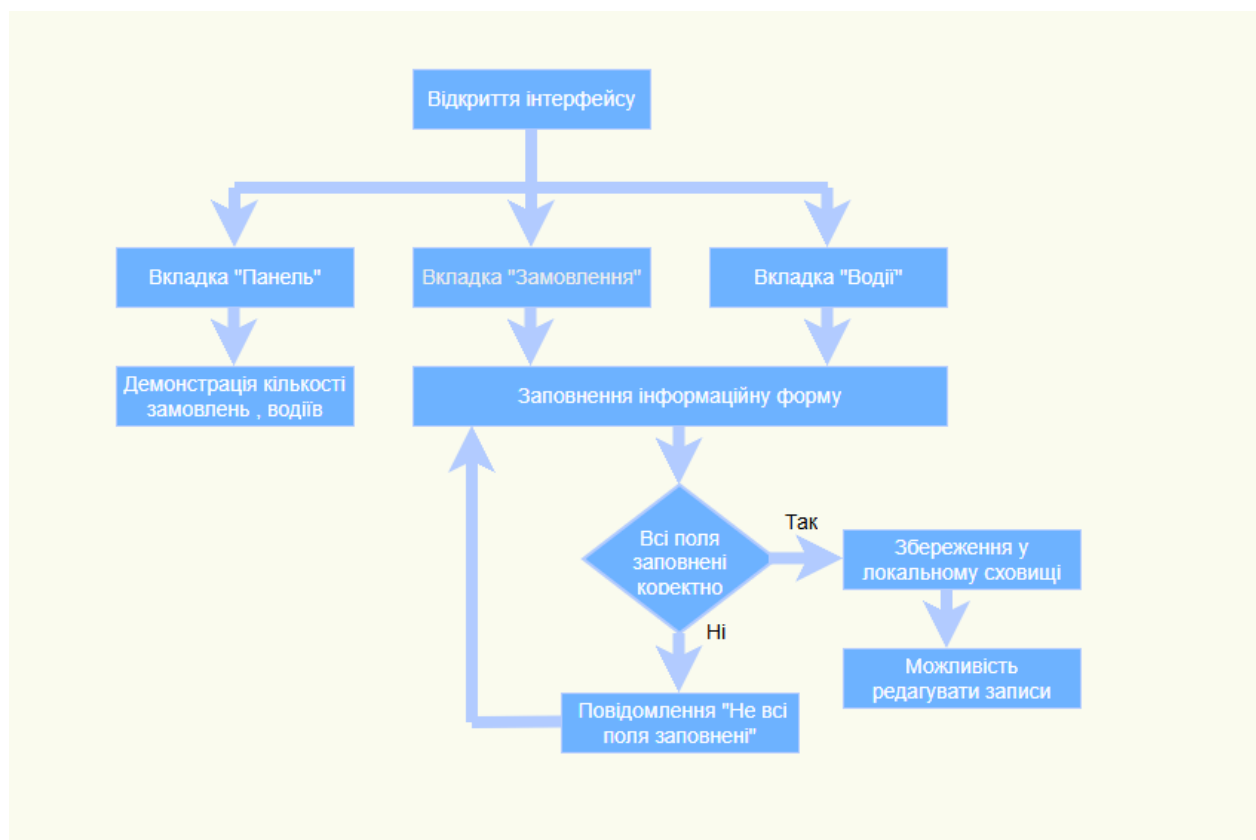


Рисунок 3.1 – Алгоритм роботи інформаційної системи

Користувач може обирати між вкладками за допомогою інтуїтивного меню навігації. Вибір вкладки «Панель» дозволяє переглянути актуальну статистику: скільки наразі зареєстровано водіїв і скільки створено замовлень. Якщо обрати розділ «Замовлення», відкривається форма, яка дозволяє додавати нові логістичні заявки, а також переглядати та редагувати наявні. У вкладці «Водії» користувач має змогу керувати даними про водіїв, зокрема вносити інформацію про ім'я, транспортний засіб і номер телефону.

Процес додавання нового запису відбувається через заповнення спеціальної форми, де передбачено всі необхідні поля. Система вимагає обов'язкового заповнення всіх полів: у разі, якщо хоча б одне з них залишиться порожнім, з'являється сповіщення про помилку. Це гарантує правильність структури даних і

виключає появу неповних записів. Якщо форма заповнена коректно, система створює новий об'єкт з унікальним ідентифікатором, додає його до відповідного масиву (orders або drivers) і оновлює локальне сховище. Усі дані зберігаються безпосередньо у браузері, що дозволяє продовжити роботу з ними навіть після закриття сторінки або перезапуску системи.

Редагування запису також реалізовано зручно і безпечно. При натисканні на кнопку «Редагувати» у таблиці, вибрані дані завантажуються у форму. Користувач може змінити будь-які поля та зберегти зміни. У цьому випадку система оновлює масив даних у пам'яті і відповідно синхронізує зміни з localStorage.

Видалення записів реалізовано через інтерфейс таблиці. Для кожного рядка передбачено кнопку «Видалити». Після її натискання запис одразу виключається з масиву, а зміни синхронізуються з локальним сховищем. Цей процес не потребує перезавантаження сторінки і виконується миттєво.

Окрему увагу варто приділити механізму автоматичного збереження. Усі дії, пов'язані з редагуванням, додаванням або видаленням записів, супроводжуються негайною фіксацією змін у localStorage завдяки використанню React-хука useEffect. Це рішення гарантує збереження інформації незалежно від дій користувача чи стану сеансу браузера.

Візуальна складова системи реалізована за допомогою бібліотеки Tailwind CSS. Це дозволяє створити адаптивний, сучасний інтерфейс, у якому всі таблиці замовлень і водіїв миттєво оновлюються після кожної дії користувача. Панель керування, що відображає кількість записів, завжди демонструє актуальні дані. Інтерфейс оптимізовано для взаємодії як з десктопних, так і з мобільних пристроїв.

Загалом, інформаційна система логістичних перевезень забезпечує надійну обробку даних у реальному часі, інтуїтивну взаємодію з користувачем, просту структуру компонентів та збереження інформації без необхідності серверної обробки. Такий підхід ідеально підходить для використання в невеликих логістичних компаніях або як основа для більш масштабного проекту з базою даних, аналітикою та інтеграцією з зовнішніми API.

### 3.2 Модулі та їх функціональність у системі

Головний модуль - App()

Центральним компонентом є App, який забезпечує координацію роботи всієї системи. Він виконує функцію диспетчера, тобто керує станом програми, зберігає усі ключові масиви даних - замовлення (orders) та водії (drivers), а також визначає, який розділ інтерфейсу наразі активний. Завдяки використанню useState, App відслідковує зміни у цих даних, а за допомогою useEffect - синхронізує їх із локальним сховищем браузера (localStorage).

## Логістика

Панель

Замовлення

Водії

Рисунок 3.2 – Вкладки інтерфейсу

Основні особливості App():

- реалізовано локальне збереження даних, щоб вони не зникали після перезавантаження;
- інтерфейс підтримує три вкладки - «Панель», «Замовлення», «Водії»;
- функції add, edit, delete передаються до дочірніх компонентів як колбеки;

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

- початкові дані завантажуються з localStorage.

Крім того, App відповідає за відображення вкладок і за передачу функцій та масивів даних у відповідні дочірні компоненти (DashboardContent, OrderForm, DriverForm).

#### Інформаційна панель - DashboardContent

Цей модуль - простий, але інформативний. Його основне завдання - виводити коротку статистику, тобто кількість зареєстрованих замовлень та водіїв. Він не містить логіки керування даними, а лише приймає orders та drivers як властивості (props) від батьківського компонента App.

Основні елементи компонента:

- заголовок "Панель керування" - візуально виділяє розділ;
- два інформаційні блоки;
- використовуються стилі Tailwind CSS для оформлення (shadow, rounded, bg-white, hover:bg-gray-50 тощо);
- компонент оновлюється автоматично, коли змінюються orders або drivers, завдяки реактивності React.

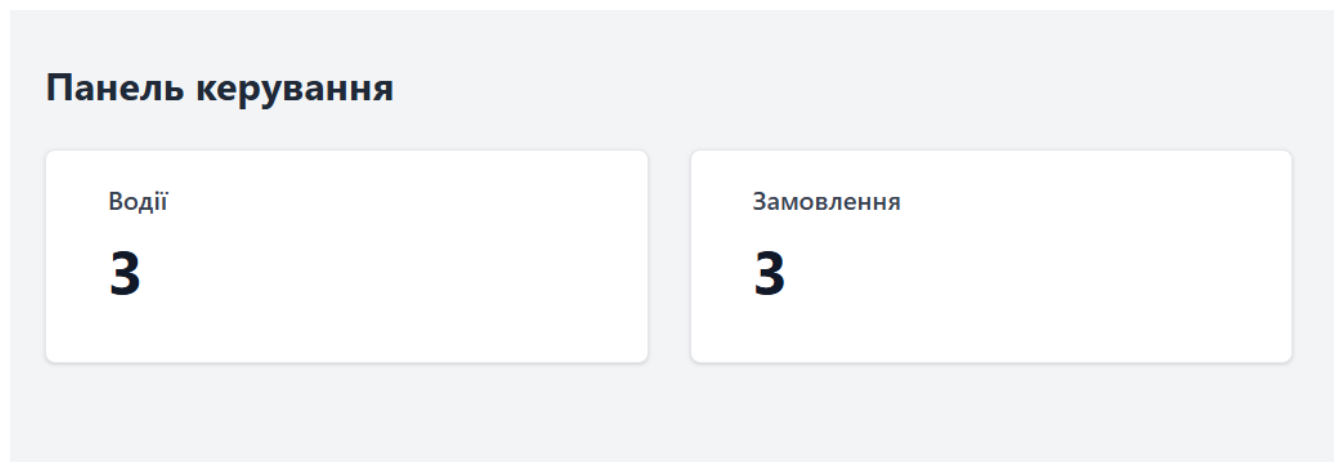


Рисунок 3.3 – Коротка статистика активних замовлень та водіїв

Інформація подається у вигляді графічних карток, що створює зручне візуальне уявлення про стан логістичного процесу. Це особливо важливо для швидкої оцінки роботи системи.

#### Форма керування замовленнями - OrderForm

Цей компонент дозволяє користувачу створювати нові заявки на перевезення вантажів, редагувати існуючі або видаляти непотрібні. Він підтримує локальний стан форми - тобто запам'ятовує введені значення до моменту збереження. У структурі форми передбачено чотири поля: товар, вага, пункт відправлення та пункт призначення.

Нове замовлення

Подушки 5 кг

Львів Пункт отримання

Не всі поля заповнені

Додати

Рисунок 3.4 – Демонстрація не коректного заповнення

Коли користувач натискає кнопку «Додати» або «Зберегти», дані перевіряються на повноту. Якщо всі поля заповнені - запис або додається до масиву замовлень, або оновлюється у разі редагування. Компонент тісно взаємодіє з App - саме через нього передаються колбек-функції для зміни глобального стану.

У нижній частині компонента розташована таблиця, яка відображає всі внесені замовлення. Це дозволяє одразу бачити зміни та взаємодіяти з кожним записом окремо.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

ID	Товар	Вага	Пункт відпр.	Пункт отрим.	Дії
#1	Пральна машина	7 кг	Одеса	Київ	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
#2	Шорти	1 кг	Самчики	Біла церква	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
#3	Автомобільні деталі	4.6 кг	Хмельницький	Львів	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рисунок 3.5 - Таблиця з повною інформацією про замовленнями

### Форма керування водіями - DriverForm

Аналогічно до попереднього модуля, DriverForm відповідає за обробку інформації про водіїв. Його поля включають ім'я, транспортний засіб та номер телефону. Він також має власний локальний стан, у якому зберігається інформація до моменту підтвердження змін.

Цей компонент також надає користувачу можливість швидкого редагування або видалення даних водія, а всі зміни моментально передаються в App через передані функції. Після натискання кнопки «Додати» дані оновлюються у глобальному стані та зберігаються у localStorage.

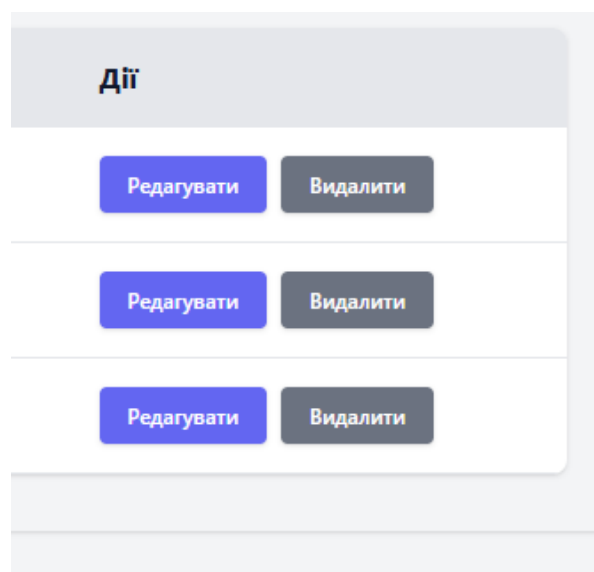


Рисунок 3.6 – Кнопки редагування та видалення замавлень, активних водіїв

Таблиця, розташована під формою, дозволяє переглядати актуальний список водіїв. Користувач може легко вносити зміни або видаляти записи, не переходячи до іншої вкладки чи перезавантажуючи сторінку.

Повторно використовувані UI-компоненти

У проєкті активно застосовуються багаторазові візуальні компоненти, які створюють узгоджений інтерфейс користувача. Найголовніші з них:

Card та CardContent: використовуються для відображення інформаційних блоків на панелі. Завдяки стилізації в Tailwind CSS, вони мають привабливий вигляд і чудово адаптуються до змін вмісту.

Button: цей компонент є базовим для виконання дій - таких як додавання, редагування або скасування. Завдяки уніфікованому дизайну він однаково виглядає у всіх частинах інтерфейсу, що підвищує зручність для користувача.

Характеристика архітектури

Загальна структура системи базується на принципах компонентного підходу React. Усі модулі чітко поділені за функціональністю, що дозволяє легко масштабувати систему, додавати нові функції або змінювати існуючі без порушення загальної логіки.

Кожен модуль виконує власну, чітко визначену роль, що зменшує кількість залежностей між компонентами. При цьому головний компонент (App) залишається єдиним координатором стану, завдяки чому досягається централізоване управління даними. Реактивність забезпечує миттєве оновлення інтерфейсу при зміні стану.

### 3.3 Взаємодії компонентів у системі

В основі роботи інформаційної системи лежить модульна архітектура, де кожен компонент має чітко визначену функціональність та відповідальність. Взаємодія між окремими частинами побудована таким чином, щоб забезпечити збереження даних, миттєве їх відображення в інтерфейсі та максимально зручну роботу з боку користувача. Центральним вузлом усієї системи є компонент App,

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

який виступає як диспетчер взаємодій, управляє активною вкладкою, а також слугує точкою зберігання масивів замовлень і водіїв. Саме цей компонент забезпечує синхронізацію між локальним сховищем браузера (localStorage) та станом додатку.

Під час завантаження сторінки компонент App ініціалізує початковий стан програми. Він відповідає за визначення активної вкладки — чи то панель керування, чи розділ із замовленнями, чи форма керування водіями. Усі зміни, які відбуваються всередині дочірніх компонентів, такі як створення нового запису, редагування чи видалення, спрямовуються до App за допомогою спеціальних колбеків. App обробляє ці зміни, оновлює відповідні масиви та негайно записує оновлену інформацію у локальне сховище. Такий підхід дозволяє уникнути втрати даних після оновлення сторінки або закриття браузера.

Для представлення зведеної інформації про кількість наявних замовлень і зареєстрованих водіїв використовується компонент DashboardContent. Він не має власної логіки збереження або обробки, проте приймає дані у вигляді властивостей від головного компонента. Його основною функцією є візуалізація кількісної інформації у вигляді інформативних карток. Завдяки цьому користувач одразу може отримати загальне уявлення про стан логістичних операцій.

Форми для керування замовленнями та водіями реалізовані у вигляді окремих компонентів OrderForm та DriverForm. Кожен із них виконує схожі завдання, але працює зі своїм типом даних. Обидва компоненти мають власний локальний стан форми, що дозволяє користувачеві вводити або змінювати дані перед їхнім збереженням. У випадку із замовленням користувач заповнює інформацію про вантаж, його вагу та пункти відправлення й отримання. У розділі водіїв, відповідно, вказуються ім'я, тип транспорту та контактний номер. Якщо користувач натискає кнопку додавання, перевіряється повнота заповнення полів. У разі успішної валідації новий запис надсилається до компонента App, який додає його до загального списку, присвоює унікальний ідентифікатор і оновлює сховище.

Редагування записів відбувається за схожим сценарієм. Коли користувач натискає на кнопку «Редагувати», дані з таблиці завантажуються назад у форму, де їх можна змінити. Після збереження оновлені значення потрапляють до App і замінюють старі. Видалення реалізовано також через взаємодію з головним компонентом: при натисканні кнопки «Видалити» відповідний запис вилучається зі списку і відразу зникає з інтерфейсу.

Щоб забезпечити зручність і однорідність дизайну, вся система побудована на базі багаторазово використовуваних візуальних компонентів, таких як Card, CardContent і Button. Компонент Card застосовується для створення інформаційних блоків у панелі керування, Button — для реалізації дій користувача (додавання, редагування, збереження, скасування тощо). Це дозволяє підтримувати єдиний стиль в усьому інтерфейсі та спрощує масштабування проєкту.

Дані оновлюються миттєво завдяки реактивній природі фреймворку React. Усі зміни, здійснені користувачем, одразу візуалізуються в таблицях, що містяться нижче форми, та в загальній статистиці на панелі керування. Завдяки механізму автоматичного запису у localStorage, система зберігає поточний стан без потреби у додаткових базах даних чи серверних обробниках.

Весь цей процес — від взаємодії користувача з інтерфейсом до збереження даних — створює цілісну модель, де кожен компонент несе конкретну відповідальність. Такий розподіл забезпечує зручну структуру коду, легке тестування, розширення функціоналу й адаптацію системи до нових вимог. Це дозволяє розглядати запропоновану інформаційну систему не лише як навчальний приклад, а як цілком робочий інструмент для малого чи середнього логістичного підприємства.

### 3.4 Результати впровадження системи

У процесі вибору інструментів для проєктування інформаційної системи управління логістичними перевезеннями було враховано низку сучасних вимог, які висуваються до веборієнтованих додатків. Особливу увагу приділено таким

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 49
Зм.	Арк.	№ докум.	Підпис	Дата		

параметрам, як адаптивність інтерфейсу, продуктивність при роботі з великими обсягами даних, легкість обслуговування системи, а також можливість її швидкої модифікації та розширення. На тлі аналізу доступних технологій було обрано JavaScript-бібліотеку React, яка зарекомендувала себе як ефективний інструмент для створення динамічних та швидких користувацьких інтерфейсів. Її перевага полягає в можливості інтелектуального оновлення лише тих частин інтерфейсу, які змінюються, без потреби повного перезавантаження сторінки, що забезпечує високу швидкодію та позитивний користувацький досвід.

## Логістика

Панель

Замовлення

Водії

### Нове замовлення

[Додати](#)

ID	Товар	Вага	Пункт відпр.	Пункт отрим.	Дії
#1	Пральна машина	7 кг	Одеса	Київ	<a href="#">Редагувати</a> <a href="#">Видалити</a>
#2	Шорти	1 кг	Самчики	Біла церква	<a href="#">Редагувати</a> <a href="#">Видалити</a>
#3	Автомобільні деталі	4.6 кг	Хмельницький	Львів	<a href="#">Редагувати</a> <a href="#">Видалити</a>

Рисунок 3.8 – Форма для керування даними про замовлення клієнтів

Процес розробки системи умовно можна поділити на кілька взаємопов'язаних етапів, кожен з яких відіграв ключову роль у формуванні повноцінного цифрового рішення. Все починалося з поглибленого аналізу предметної галузі логістичних перевезень, що дозволило краще зрозуміти специфіку логістичних процесів, особливості документообігу, потреби менеджерів і водіїв, а також знайти вузькі місця в уже існуючих аналогах. На основі отриманих висновків були сформульовані вимоги до майбутньої системи, зокрема щодо функціоналу, ергономіки інтерфейсу та основних користувацьких сценаріїв.

## Логістика

Панель

Замовлення

Водії

### Панель керування

Водії

**3**

Замовлення

**3**

Рисунок 3.9 – Коротка інформація про кількість активних замовлень та водіїв

Архітектурне проектування велося з урахуванням принципів модульності та масштабованості, що дозволяє легко адаптувати систему до нових умов експлуатації та підключення до сторонніх сервісів або баз даних. Застосування концепції односторінкового додатку (SPA) дозволило суттєво покращити зручність навігації: перемикання між розділами, такими як «Панель», «Замовлення» чи «Водії», здійснюється миттєво без оновлення сторінки. При побудові внутрішньої логіки застосовувалися сучасні можливості React, зокрема використання хуків `useState` та `useEffect`, що забезпечило ефективне керування станом додатку та синхронізацію з локальним сховищем браузера.

Система демонструє високу адаптивність, тому може однаково зручно використовуватись як на десктопних, так і на мобільних пристроях. Завдяки логічно продуманій структурі інтерфейсу користувачі швидко освоюють систему без необхідності у складному навчанні. Візуальна простота, підкріплена функціональністю, дозволяє персоналу оперативно знаходити необхідні дані, додавати нові записи чи редагувати існуючі, що значно скорочує час виконання рутинних завдань.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 51
Зм.	Арк.	№ докум.	Підпис	Дата		

## Логістика

Панель

Замовлення

Водії

### Новий водій

Додати

ID	Ім'я	Транспортний засіб	Телефон	Дії
#1	Олексій	skoda	+380780567894	<a href="#">Редагувати</a> <a href="#">Видалити</a>
#2	Олександр	mercedes	+380500789558	<a href="#">Редагувати</a> <a href="#">Видалити</a>
#3	Валерій	skoda	+380680983456	<a href="#">Редагувати</a> <a href="#">Видалити</a>

Рисунок 3.10 – Форма для керування даними про водіїв

## Логістика

Панель

Замовлення

Водії

### Редагувати замовлення

Не всі поля заповнені

Зберегти

Скасувати

ID	Товар	Вага	Пункт відпр.	Пункт отрим.	Дії
#1	Пральна машина	7 кг	Одеса	Київ	<a href="#">Редагувати</a> <a href="#">Видалити</a>
#2	Шорти	1 кг	Самчики	Біла церква	<a href="#">Редагувати</a> <a href="#">Видалити</a>
#3	Автомобільні деталі	4.6 кг	Хмельницький	Львів	<a href="#">Редагувати</a> <a href="#">Видалити</a>

Рисунок 3.11 – Демонстрація некоректного вводу даних

## Новий водій

Рисунок 3.12 – Демонстрація форми для заповнення нового водія

## Редагувати замовлення

ID	Товар	Вага	Пункт відпр.	Пункт отрим.	Дії
#1	Пральна машина	7 кг	Одеса	Київ	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
#2	Шорти	1 кг	Самчики	Біла церква	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>
#3	Автомобільні деталі	4.6 кг	Хмельницький	Львів	<input type="button" value="Редагувати"/> <input type="button" value="Видалити"/>

Рисунок 3.13 – Редагування створеного замовлення

Надійність системи підвищено за рахунок механізму збереження усіх змін у локальному сховищі браузера. Це рішення дозволяє уникати втрати важливої інформації навіть при неочікуваному завершенні роботи чи відключенні мережі. Окрім цього, такий підхід забезпечує стійкість системи до потенційних збоїв з

Зм.	Арк.	№ докум.	Підпис	Дата

КВРІСТ 220168.22.01.05 ПЗ

Арк.

53

боку сервера або інтернет-з'єднання, що є критично важливим у сфері логістики, де точність і оперативність мають першорядне значення.

У перспективі система має значні можливості для вдосконалення та розширення. Завдяки модульній архітектурі можливе підключення до баз даних, реалізація аналітичних панелей для візуалізації логістичних процесів, генерація звітів у реальному часі, розширення функціоналу з урахуванням специфіки конкретних підприємств. Крім того, можна реалізувати інтеграцію з картографічними сервісами для відстеження маршрутів транспорту та автоматизовану обробку замовлень із зовнішніх джерел.

Отже, розроблена інформаційна система управління логістичними перевезеннями є не просто реалізацією поточних потреб користувачів, а платформою, яка поєднує сучасні вебтехнології, зручність використання та гнучкість у подальшому розвитку. Такий підхід дозволяє зробити вагомий внесок у цифрову трансформацію логістичних процесів, забезпечуючи підприємствам конкурентні переваги у галузі.

### 3.5. Висновки до третього розділу

Процес вибору інструментів для створення інформаційної системи управління логістичними перевезеннями є важливим етапом, який суттєво впливає на майбутню якість та ефективність розробленого рішення. У рамках цього процесу було детально враховано низку сучасних вимог, що пред'являються до веборієнтованих додатків. Значну увагу приділено адаптивності інтерфейсу, яка передбачає коректну та зручну роботу користувачів із системою незалежно від типу пристрою, який вони використовують. Іншим важливим критерієм стала продуктивність, яка особливо важлива при обробці великих обсягів інформації, характерних для логістичних задач. Крім цього, значна увага приділялася простоті підтримки системи, легкості оновлення та можливості подальшого розширення функціоналу.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

Після аналізу сучасних технологій і рішень, наявних на ринку веброзробки, вибір було зроблено на користь JavaScript-бібліотеки React. Дана бібліотека вже давно зарекомендувала себе як надзвичайно ефективний інструмент для створення високошвидкісних, інтерактивних та користувацьких інтерфейсів. Головна перевага React полягає в можливості динамічно оновлювати лише необхідні частини інтерфейсу без перезавантаження сторінки цілком. Завдяки цьому забезпечується максимальна швидкодія, що в умовах інтенсивного використання логістичних систем суттєво покращує користувацький досвід, а також зменшує навантаження на серверну частину.

Розробка системи складалася з кількох важливих та взаємопов'язаних етапів, кожен із яких зробив свій внесок у формування остаточного продукту. Початковий етап передбачав глибокий аналіз предметної галузі, зокрема вивчення специфіки роботи логістичних процесів, документообігу та взаємодії між менеджерами, операторами та водіями. Це дозволило чітко окреслити потреби кінцевих користувачів та виявити потенційні слабкі місця вже наявних аналогічних рішень. На основі цього аналізу були визначені основні вимоги до майбутньої інформаційної системи, що стосуються функціональності, ергономічності інтерфейсу та основних сценаріїв взаємодії користувачів з додатком.

Наступним етапом було архітектурне проектування системи, при якому важливу роль відігравали принципи модульності та масштабованості. Саме модульна архітектура дозволяє ефективно інтегрувати нові функції, оперативно здійснювати адаптацію системи до змінних умов роботи або нових бізнес-вимог. Використання концепції односторінкового додатку (SPA) забезпечило додаткові переваги, зокрема швидкість і плавність навігації між різними розділами системи, такими як «Панель управління», «Замовлення», «Водії» та інші. Завдяки цьому користувачі можуть миттєво отримувати доступ до необхідної інформації без зайвих затримок.

Технологічно система спирається на сучасні можливості React, зокрема на використання хуків useState та useEffect. Це дозволило значно покращити управління станом додатку, забезпечити синхронізацію даних з локальним сховищем браузера, а також зробити інтерфейс більш динамічним та інтуїтивно зрозумілим. Крім цього, завдяки адаптивній верстці інтерфейсу, система однаково зручна для використання як на настільних комп'ютерах, так і на мобільних пристроях, що суттєво розширює можливості її використання у різних робочих умовах.

Зручність та ергономіка інтерфейсу дозволяють користувачам швидко освоювати роботу з системою без додаткового навчання. Завдяки простій та логічній структурі навігації, користувачі можуть оперативно знаходити потрібні дані, додавати нові записи, а також швидко вносити зміни у вже існуючі. Це суттєво прискорює виконання щоденних завдань та мінімізує кількість помилок, пов'язаних з ручним введенням даних.

Важливим аспектом надійності системи є використання локального сховища браузера для зберігання проміжних змін. Такий підхід дозволяє уникнути втрати важливих даних у випадках несподіваного відключення системи, перебоїв із мережею або інших технічних несправностей. Це є особливо критичним для логістичних систем, де втрата інформації може призвести до значних збитків.

Таким чином, інформаційна система управління логістичними перевезеннями, створена із застосуванням сучасних вебтехнологій на основі React, забезпечує не тільки задоволення поточних потреб користувачів, але й пропонує платформу, здатну ефективно розширюватися та модифікуватися в майбутньому. Це робить її важливим елементом цифрової трансформації логістичних процесів, надаючи підприємствам значні конкурентні переваги.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень було розроблено, детально обґрунтовано і змодельовано інформаційну систему управління логістичними перевезеннями. За результатами моделювання було підтверджено правильність прийнятих рішень щодо вибору апаратних компонентів і архітектури системи.

У першому розділі проведено аналіз предметної області, здійснено вивчення наявних рішень та існуючих проблем, визначено актуальність та перспективи розвитку інформаційної системи. Також було визначено основні функціональні й нефункціональні вимоги до системи, що стали основою для подальшого проектування.

У другому розділі у процесі розробки інформаційної системи управління логістичними перевезеннями (ІСУЛП) особливу увагу було приділено вибору ефективних інструментів програмування та технологій, що забезпечують реалізацію функціональної, гнучкої та масштабованої інформаційної системи. На основі проведеного аналізу можна зробити низку важливих висновків щодо доцільності використання Visual Studio Code та Tailwind CSS у межах даного проекту.

У третьому розділі розробка інформаційної системи управління логістичними перевезеннями складалася з кількох ключових етапів: визначення вимог, проектування архітектури, реалізації основних функцій та тестування. На початковому етапі був проведений детальний аналіз предметної області, на основі якого були визначені ключові вимоги до функціоналу та інтерфейсу системи.

Отже, результати виконаної роботи підтверджують успішність та доцільність реалізації проекту інформаційної системи, а також створюють міцну базу для майбутньої фізичної реалізації й впровадження розробленого рішення на практиці.

					КвРІСТ 220168.22.01.05 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Крикавський Є. В. Логістика: навч. посіб. Львів: ЛНУ імені Івана Франка, 2021. 512 с.
2. Бублик М. І. Інформаційні системи в логістиці: навч. посіб. Київ: КНЕУ, 2021. 228 с.
3. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 12.06.2025).
4. React – Official Documentation. URL: <https://reactjs.org/docs> (дата звернення: 12.06.2025).
5. Vite – Frontend Tooling. URL: <https://vitejs.dev/> (дата звернення: 12.06.2025).
6. MySQL Documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 12.06.2025).
7. Firebase Documentation. URL: <https://firebase.google.com/docs> (дата звернення: 12.06.2025).
8. Microsoft Dynamics 365 SCM. URL: <https://dynamics.microsoft.com> (дата звернення: 12.06.2025).
9. Oracle Transportation Management. URL: <https://www.oracle.com/scm/transportation-management/> (дата звернення: 12.06.2025).
10. Software Advice – TMS. URL: <https://www.softwareadvice.com/tms/> (дата звернення: 12.06.2025).
11. Supply Chain Digital. URL: <https://www.supplychaindigital.com/> (дата звернення: 12.06.2025).
12. Logistics Management Magazine. URL: <https://www.logisticsmgmt.com/> (дата звернення: 12.06.2025).
13. Docker Docs. URL: <https://docs.docker.com/> (дата звернення: 12.06.2025).
14. JavaScript.info. URL: <https://javascript.info/> (дата звернення: 12.06.2025).

					КВРІСТ 220168.22.01.05 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

15. Mozilla Developer Network (MDN). URL: <https://developer.mozilla.org/> (дата звернення: 12.06.2025).
16. SelectHub: Logistics Software. URL: <https://www.selecthub.com/logistics/> (дата звернення: 12.06.2025).
17. Investopedia – Supply Chains. URL: <https://www.investopedia.com> (дата звернення: 12.06.2025).
18. Code Visual Studio Docs. URL: <https://code.visualstudio.com/docs> (дата звернення: 12.06.2025).
19. Open Source Logistics Software – SourceForge. URL: <https://sourceforge.net/> (дата звернення: 12.06.2025).
20. ERP vs TMS Systems – Comparison. URL: <https://www.erpsoftwareblog.com/> (дата звернення: 12.06.2025).
21. Logistics Tech Outlook. URL: <https://www.logisticstechoutlook.com/> (дата звернення: 12.06.2025).
22. TMS Software Guide. URL: <https://www.capterra.com/transportation-management-software/> (дата звернення: 12.06.2025).
23. FreightWaves. URL: <https://www.freightwaves.com/> (дата звернення: 12.06.2025).
24. TMS vs WMS: Key Differences. URL: <https://www.gep.com/> (дата звернення: 12.06.2025).
25. McKinsey & Company. Digitizing Logistics. URL: <https://www.mckinsey.com/> (дата звернення: 12.06.2025).
26. Accenture. Digital Logistics Platforms. URL: <https://www.accenture.com/> (дата звернення: 12.06.2025).
27. Harvard Business Review. Logistics Tech. URL: <https://hbr.org/> (дата звернення: 12.06.2025).
28. MIT Center for Transportation & Logistics. URL: <https://ctl.mit.edu/> (дата звернення: 12.06.2025).

					КВРІСТ 220168.22.01.05 ПЗ	Арк. 59
Зм.	Арк.	№ докум.	Підпис	Дата		

29. Logistics Trends 2024 – DHL. URL: <https://www.dhl.com/global-en/home/insights.html> (дата звернення: 12.06.2025).

30. IBM Supply Chain Solutions. URL: <https://www.ibm.com/supply-chain> (дата звернення: 12.06.2025).

31. Open Logistics Foundation. URL: <https://openlogisticsfoundation.org/> (дата звернення: 12.06.2025).

32. Власюк О. С. Інформаційні технології в управлінні підприємствами. Київ: НАУ, 2022. 200 с.

33. Стеценко С. О. Інформаційні системи в економіці. Київ: Центр учбової літератури, 2021. 304 с.

34. Українська академія друкарства. Лекції з логістики. Львів: УАД, 2022. 160 с.

35. Гончар О.О. Електронна логістика Дніпро: ДНУ, 2021. 244 с.

36. Сергієнко О. В. Програмування на JavaScript Київ: Ліра-К, 2023. 280 с.

37. Іваненко П. М. Підручник з веброзробки: HTML, CSS, JS Харків: Ранок, 2022 192 с.

38. Дорошенко А. Розробка інтерфейсів у Tailwind CSS Львів: ЛНУ, 2023. 148 с.

39. Cloud TMS Solutions 2023 Report. URL: <https://www.g2.com/categories/transportation-management> (дата звернення: 12.06.2025).

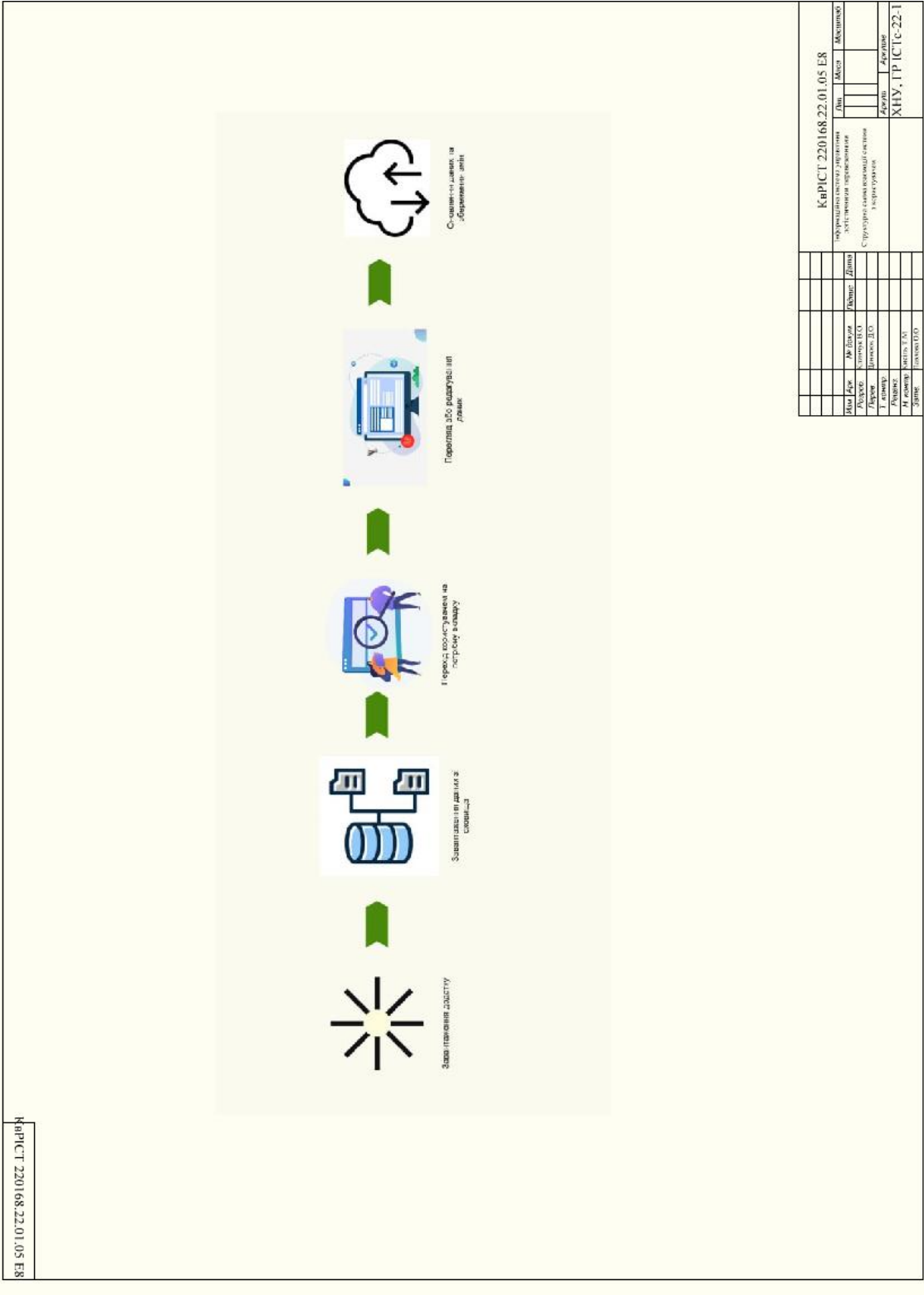
40. Oracle Transportation Management. URL: <https://www.oracle.com/scm/transportation-management/> (дата звернення: 12.06.2025).

					КВРІСТ 220168.22.01.05 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		



**Додаток Б**  
(обов'язковий)

**КОПІЯ КРЕСЛЕННЯ «СТРУКТУРНА СХЕМА ВЗАЄМОДІЇ СИСТЕМИ З КОРИСТУВАЧЕМ»**



№РПСТ 220168.22.01.05.Е8

№РПСТ 220168.22.01.05.Е8		Дат.	Місц.	Масштаб
позначення документа		дет. техніч. розробки		
Нам. Рпр.	Дир. Бюро	Рисув.	Діагн.	
Листов.	Листов.	Виснос. ДГО		
Т. архив.				
Листов.				
Діагн.				
Діагн.				
ХНУ, ГР ІС 22-1				



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система управління логістичними перевезеннями

Автор: Клинчук Валентин

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Денисюк Дмитро Олександрович, старший викладач

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:




- 1) Запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі збіги представлені загальноживаними фразами, наприклад: «на рисунку зображено», «загальна структура системи», «висновки до розділу» тощо.
- 4) Якість запозичень відповідає технічним особливостям дослідження: виявлено збіги в кодах, формулах і термінах, які є вихідними даними до великої кількості задач і не можуть вважатися авторськими порушеннями.
- 5) Система зафіксувала технічні модифікації тексту, зокрема: заміну окремих символів, скорочення індексів у формулах, зміну розміщення символів. Це є наслідком форматування або експорту документа, а не цілеспрямованого уникнення перевірки.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 3.07% і адресується до 25 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КПС

Дмитро ДЕНИСЮК

Єлизавета ГНАТЧУК

Ольга ПАВЛОВА

Завідувачу кафедри КПС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ  
Валентин КЛИНЧУК  
ПІБ здобувача вищої освіти

---

ФІТ, 3 курсу, групи ІСТс-22-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

15.06 2025 року



## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Валентин КЛИНЧУК

**Співавтор:**

**Назва:** Клинчука\_Інформаційна система управління логістичними перевезеннями

**Експерт:**

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 3.1%

**Коефіцієнт подібності 2:** 2.1%

**Мікропробіли:** 4

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2025-06-16 15:24:37.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вини освіти, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-16

Дата



Доцент Андрій Нічепорук

експерт

## Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 1.0%**

**Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 9%**

ID: 246109 Title: БКР Інформаційна система управління логістичними перевезеннями Added in a DB: 2025-06-16 Authors: Валентин КЛИНЧУК Heads: Дмитро ДЕНИСЮК Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	72124	576	1873 (3%)	25 (4%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Клинчук Валентин Олександрович

Тема: Інформаційна система управління логістичними перевезеннями

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   56  

1. Короткий зміст роботи та прийнятих рішень:

Метою кваліфікаційної роботи є проектування та реалізація інформаційної системи для ефективного управління логістичними перевезеннями, що охоплює автоматизацію планування маршрутів, облік ресурсів, моніторинг доставки та аналітику логістичних процесів. У процесі роботи було прийнято рішення щодо побудови адаптивної модульної архітектури, яка забезпечує гнучке масштабування та інтеграцію з існуючими ІТ-системами підприємства.

2. Висновок про відповідність роботи дипломному завданню:

виконана робота повністю відповідає завданню, сформульованому в дипломному проєкті. Усі поставлені цілі досягнуто, що підтверджується результатами реалізації та тестування системи.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи:

У першому розділі здійснено комплексний аналіз предметної області управління логістикою. Проведено огляд існуючих програмних рішень у сфері TMS (Transportation Management Systems), порівняно їх функціональні можливості, а також вивчено сучасні технології для впровадження таких систем, зокрема хмарні платформи, API-інтеграції та цифрові інструменти аналітики. Це дозволило обґрунтувати вибір технологічної бази та програмного середовища для реалізації ІСУЛП. У другому розділі розроблено архітектуру системи, яка охоплює користувацький інтерфейс, модуль обробки замовлень, транспортне планування, моніторинг та аналітичну частину. Обрано засоби реалізації веб-інтерфейсу (React),

серверної логіки (Node.js). Розглянуто методи інтеграції з GPS-трекерами для відстеження транспорту й алгоритми оптимізації логістичних маршрутів. Також акцентовано увагу на мікросервісному підході, що дозволяє підвищити масштабованість системи. У третьому розділі представлено практичну реалізацію системи, проведено тестування функціональних модулів у змодельованому логістичному середовищі. Наведено приклади роботи інтерфейсу, алгоритмів розрахунку маршрутів і моніторингу транспорту. Результати тестування підтверджують ефективність запропонованого рішення: зменшено час обробки замовлень і покращено точність доставки. Отримані результати задокументовано та підкріплено відповідними аналітичними оцінками.

4. Позитивні сторони роботи: робота демонструє високий рівень практичної реалізації та глибоке опрацювання предметної області. Використано сучасні технології розробки, зокрема мікросервіси, хмарну інфраструктуру та інструменти аналізу даних. Реалізована система має перспективи до масштабування й впровадження в реальних умовах логістичного підприємства.

5. Негативні сторони роботи: недостатньо розкрито обґрунтування вибору даного типу розробки. Аспекти кібербезпеки системи залишилися поза увагою.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: робота виконана на належному науково-технічному рівні.

8. Інші зауваження: немає

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Триворська Наталія Іванівна, доцент кафедри МЗ, к. ед. наук

"17" червня 2025 р.

 (підпис)