

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
АВТОМАТИЗОВАНИХ СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА**

на тему: «Оптимізація логістики за допомогою інформаційної  
технології на основі IoT»

на здобуття освітнього ступеня магістра  
зі спеціальності 126 Інформаційні системи та технології  
(код, найменування спеціальності)  
освітньо-професійної програми Інформаційні системи та технології  
(назва)

*Кваліфікаційна робота містить результати власних досліджень. Використання  
ідей, результатів і текстів інших авторів мають посилання  
на відповідне джерело*

\_\_\_\_\_ Дмитро МІРОНОВ  
(підпис) *Ім'я, ПРИЗВИЩЕ здобувача*

Виконав:  
здобувач вищої освіти  
група ІСДМ-61

Дмитро МІРОНОВ

Керівник:  
*науковий ступінь,  
вчене звання*

Аліна ТУШИЧ  
д.ф., доцент

Рецензент:  
*науковий ступінь,  
вчене звання*

\_\_\_\_\_  
Ім'я, ПРИЗВИЩЕ

**Київ 2023**

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**  
**Навчально-науковий інститут інформаційних технологій**

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти Магістр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

**ЗАТВЕРДЖУЮ**

Завідувач кафедру ІІЗАС

\_\_\_\_\_ Каміла СТОРЧАК

«\_\_\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

\_\_\_\_\_ Міронову Дмитру Сергійовичу

*(прізвище, ім'я, по батькові здобувача)*

1. Тема кваліфікаційної роботи: Оптимізація логістики за допомогою інформаційної технології на основі IoT

керівник кваліфікаційної роботи Аліна ТУШИЧ д.ф., доцент,

*(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)*

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145

2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, протоколи передачі даних, вимоги до інформаційних технологій.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Інформаційна система та технології. Види інформаційної системи. Логістична інформаційна система;

4.2 Функціональна й організаційна структура логістичної інформаційної системи;

- 4.3 Переваги використання веб-платформ у сфері логістики;  
 4.4 Шляхи оптимізації обробки даних за допомогою ІТ-технологій;  
 4.5 Вимоги до інформаційних систем в логістиці та принципи їх побудови;  
 4.6 ІоТ та його види. Застосування ІоТ в логістиці;  
 4.7 Види протоколів передачі даних. Інтеграція систем з ІоТ;  
 4.8 Розробка та оптимізація інформаційної системи на основі ІоТ;  
 4.9 Опис веб-платформи та її алгоритмів.
5. Перелік графічного матеріалу: *презентація*
- 5.1 Діаграм – 2;  
 5.2 Рисунків – 4б;  
 5.3 Таблиць – 4.
6. Дата видачі завдання «19» жовтня 2023 р.

### **КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	28.10-01.11.23	
2	Аналіз інформаційних систем	05.11-15.11.23	
3	Аналіз системи логістики	15.11-20.11.23	
4	Аналіз протоколів передачі даних	20.11-27.11.23	
5	Розробка та налаштування WebScket	27.11-30.11.23	
6	Розробка та програмування ІоТ частини	30.11-02.12.23	
8	Розробка системи на основі ІоТ	02.12-20.12.23	
9	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
10	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

\_\_\_\_\_

(підпис)

Дмитро МІРОНОВ

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

\_\_\_\_\_

(підпис)

Аліна ТУШИЧ

(Ім'я, ПРІЗВИЩЕ)





## РЕФЕРАТ

Магістерська атестаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел і має 95 сторінок основного тексту, 46 рисунків, 4 таблиць. Загальний обсяг роботи 87 сторінки.

**Об'єктом дослідження** є процес взаємодії логістики з цифровою індустрією та взаємодія системи з IoT технологіями.

**Предмет дослідження** є інформаційна система, яка оптимізована за допомогою IoT.

**Мета роботи** дослідження та розробка інформаційної системи логістики на основі IoT.

**Методи дослідження** оптимізація логістики за допомогою удосконаленої інформаційної технології на основі IoT.

**Актуальність роботи** – керувати процесом руху та зберігання товарів підприємства, своєчасно доставляти необхідну кількість товару від місця походження до місця споживання з найменшою вартістю та найкращим сервісом, автоматизувати процес прийняття товарів, створення логістичного процесу. Оптимізація всіх процесів за допомогою IoT.

**Рекомендаціями використання результатів роботи** є використання програми як основа для подальшого удосконалення системи та впровадження більше приладів IoT.

**Сфера застосування:** запропонована система може бути використана, впровадженна на підприємствах різного рівня чи спецефікації яка використовує логістику.

## ABSTRACT

The master's thesis consists of an introduction, four chapters, general conclusions, a list of used sources and has 96 pages of the main text, 46 figures, 4 tables. The total volume of work is 87 pages.

**Object of the research** is the process of interaction of logistics with the digital industry and the interaction of the system with IoT technologies.

**Subject of research** is an information system that is optimized with the help of IoT.

**Purpose of research and development** of an IoT-based logistics information system.

**Methods of research** optimization of logistics using advanced information technology based on IoT.

**The relevance of the work** is to manage the process of movement and storage of the company's goods, timely delivery of the necessary amount of goods from the place of origin to the place of consumption with the lowest cost and the best service, to automate the process of accepting goods, creating a logistics process. Optimizing all processes with IoT.

**Recommendations for using the results of the work** are to use the program as a basis for further improvement of the system and the introduction of more IoT devices.

**Scope:** the proposed system can be used, implemented in enterprises of different levels or specifications that use logistics.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ</b> .....	9
<b>ВСТУП</b> .....	10
<b>1 ЗНАЧЕННЯ ТА ОРГАНІЗАЦІЯ ЛОГІСТИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ</b> .....	12
1.1 Інформаційна система. Види інформаційної системи. Логістична інформаційна система.....	12
1.2 Функціональна й організаційна структура логістичної інформаційної системи.....	20
1.3 Класифікація інформаційних потоків у логістиці .....	25
1.4 Переваги використання веб-платформ у сфері логістики .....	30
<b>2 ОСОБЛИВОСТІ ТА ТРУДНОЩІ ЗАСТОСУВАННЯ СУЧАСНИХ ІТ-ТЕХНОЛОГІЙ У ЛОГІСТИЦІ</b> .....	32
2.1 Шляхи оптимізації обробки даних за допомогою ІТ-технологій .....	32
2.2 Труднощі пов'язані із сучасними технологіями управління логістикою та можливі шляхи подолання цих проблем .....	39
<b>3 ШЛЯХИ ОПТИМІЗАЦІЇ ЛОГІСТИКИ НА ОСНОВІ ІОТ</b> .....	44
3.1 Що таке ІоТ та його види.....	44
3.2 Застосування ІоТ в логістиці.....	46
3.3 Інтеграція ІоТ для покращення інформаційної технології.....	48
<b>4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЛОГІСТИКИ НА ОСНОВІ ІОТ</b> .....	56
4.1 Вимоги до інформаційних систем в логістиці та принципи їх побудови.....	56
4.2 Створення та налагодження веб-платформи.....	60
4.3 Опис веб-платформи та її алгоритмів.....	75
<b>ВИСНОВКИ</b> .....	93
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	95
<b>ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ</b> .....	97



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

ЛІС	–	логістична інформаційна система.
Http	–	<u>протокол</u> передачі даних.
DOM	–	англ. Document Object Model – об'єктна модель документа.
XML	–	<u>англ.</u> Extensible Markup Language, скорочено ХМ – розширювана мова розмітки.
API	–	<u>англ.</u> Application Programming Interface, API – прикладний програмний інтерфейс.
AJAX	–	англ. Asynchronous JavaScript and XML – асинхронний JavaScript и XML – підхід до побудови користувацьких інтерфейсів веб-застосунків.
URL	–	<u>англ.</u> Uniform Resource Locator – єдиний вказівник на ресурс, URL) – стандартизована адреса певного ресурсу.
UML	–	<u>англ.</u> Unified Modeling Language – уніфікована мова моделювання.
HTML	–	англ. HyperText Markup Language – мова розмітки гіпертексту) – це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет.
JavaScript (JS)	–	динамічна, <u>об'єктно-орієнтована прототипна мова програмування.</u>
NodeJs	–	платформа з відкритим кодом для виконання високопродуктивних мережових застосунків, написаних мовою JavaScript
ІоТ	–	Інтернет речей (англ. Internet of Things, IoT) концепція мережі, що складається із взаємозв'язаних фізичних пристроїв

## ВСТУП

Актуальність теми. В сучасному світі важко представити товар без доставки. Сама доставка включає в себе певний перелік дій, однією з яких є логістика. Як запам'ятати велику кількість вантажоперевезень? Чи як знайти найоптимальніший маршрут? Інформаційна система на основі IoT має всі можливості для реалізації на вигляд легкої задачі.

В процесі діяльності підприємства відбувається постійний обмін інформацією, яка необхідна для ефективного управління. Недостатнє забезпечення управлінського комплексу підприємства інформацією може призвести до порушення взаємодії між різними гілками управління.

Успішна діяльність будь-якого підприємства неможлива без ефективної взаємодії між його підрозділами. Зокрема, важливим є взаємодія між відділом збуту, маркетинговим відділом, планово-економічним, фінансовим відділами, бухгалтерією та іншими підрозділами. Приблизно 20% усіх логістичних витрат витрачаються на логістичні операції, в той час як складання вибіркової інформації забирає від 55% до 65% вартості складських операцій. Це показує, що застосування інформаційних систем може значно знизити витрати, поліпшуючи процеси навчання нових чи тимчасових працівників, сприяючи кращому плануванню розміщення складських вантажів та покращуючи процес набору персоналу.

Також моніторинг і контроль проходження інформації в процесі діяльності – це система, яка постійно відстежує, як готується, обробляється та супроводжується інформація про стан діяльності всіх структурних підрозділів підприємства.

Одним із способів підвищення ефективності логістики є використання технологій Інтернету речей (IoT). IoT дозволяє об'єднувати фізичні об'єкти з цифровим світом, надаючи доступ до інформації про їхній стан та поведінку.

Об'єкт дослідження є процес взаємодії логістики з цифровою індустрією та взаємодія системи з IoT технологіями.

Предметом дослідження є інформаційна системи яка оптимізована за допомогою IoT. Метою магістерської роботи є розробка практичних підходів щодо формування ефективної логістичної інформаційної системи, розробка комунікації між системою та IoT пристроями, покращення існуючих методів для забезпечення роботи логістики.

Завдання дослідження є проаналізувати процеси цифровізації в роботі логістичних підприємств. Виокремити окремі логістичні операції, які потребують упровадження інноваційних інформаційних систем.

# 1 ЗНАЧЕННЯ ТА ОРГАНІЗАЦІЯ ЛОГІСТИЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

## 1.1 Інформаційна система. Види інформаційної системи. Логістична інформаційна система

Наразі, інформаційні системи мають різноманітні варіанти реалізації. Раніше вони зазвичай представляли собою картотеки, але з величезним обсягом сучасних даних, ручна обробка стала неможливою. Це призвело до розвитку та впровадження автоматизованих інформаційних систем (АІС). Залежно від способу представлення та структурування інформації, ці системи можуть бути фактографічними, документальними або геоінформаційними. [7]

Фактографічні АІС зберігають дані у формі багатьох типів елементів, що відображають різні факти або події. Кожен елемент або їх поєднання містить окрему інформацію, незалежну від інших даних. Структура кожного типу об'єкта має обмежений набір деталей, що описують основні аспекти цієї інформації в конкретній області. Наприклад, фактографічна АІС містить дані про окремих осіб, кожна з яких представлена записом із визначеними даними, такими як ім'я, прізвище, рік народження, місце роботи, освіта і т.д. Отримання сховища інформації в такій системі часто вимагає процесу формування вхідних даних із документальних джерел, а структурування полягає в виділенні різних типів об'єктів і вказівці реквізитів для їх заповнення.

У документній АІС окремий інформаційний елемент — це документ, який не поділений на менші елементи, а інформація зазвичай є неструктурованою або лише частково структурованою під час введення (вхідний документ). Під час введення документа ви можете встановити кілька формальних позицій, наприклад дату створення, виконавця та тему. Деякі типи АІС документів забезпечують встановлення логічних зв'язків між реалізованими документами. Супідрядність за

смісловим змістом, перехресне посилання на конкретні документи тощо. Визначення та встановлення таких зв'язків є складним аналітичним завданням, яке включає багато критеріїв та аспектів і не може бути повністю формалізовано.

В АІС географічної інформації дані організовані як окремі інформаційні об'єкти (з певним набором деталей), пов'язані із загальною електронною топографічною базою. Геоінформаційні системи використовуються для підтримки інформації у суміжних галузях, структурах інформаційних об'єктів і процесів, які мають геопросторову складову, наприклад транспортні маршрути, управління містом тощо.

Розробка та проектування інформаційної системи починається зі створення концептуальної моделі її використання. Такі концептуальні моделі передусім визначають обсяг конкретних завдань і функцій, які система забезпечує для створення та функціонування інформаційної системи та збору, збору та публікації інформації.

Тому існує ще один критерій класифікації можливостей АІС і вирішуваних завдань, мабуть, найважливішими з яких є наступні:

- довідкові;
- пошукові;
- розрахункові;
- технологічні.

Довідкові функції — найпоширеніший тип функцій в інформаційних системах, які дозволяють абонентам системи отримувати довідкові дані для певних класів об'єктів (людей, організацій, телефонів, адрес тощо), а також набір фіксованої або довільно визначеної інформації. Типами інформаційних систем, що виконують суто довідкові функції, є різні електронні записи, картотеки, «електронні блокноти» програмного або апаратного забезпечення та їх більш розвинені аналоги у вигляді так званих персональних інформаційних систем.

Логістична інформаційна система (ЛІС) - це певним чином організована сукупність взаємопов'язаного комп'ютерного обладнання, різноманітних інструкцій і необхідних засобів програмування, що дозволяє вирішувати

конкретні функціональні завдання управління матеріальними потоками. [10]

Будь-яка логістична система являє собою складну багатокomпонентну структуру, що складається з взаємопов'язаних елементів. Ці елементи виконують певні функції, необхідні для ефективного функціонування логістичної системи. Одним із таких факторів є автоматизоване робоче місце адміністративного персоналу (АРПУ). АРПУ — це ІТ-додаток, який дозволяє керівному персоналу отримувати доступ до інформації та інструментів, необхідних для управління логістичними процесами. Іншим елементом логістичної системи є інформаційний підрозділ системи організаційного управління (ІМСУ). ІМСУ є підрозділом, відповідальним за розробку, розгортання та підтримку інформаційних систем, що використовуються в логістичних системах. Нарешті, логістична система може складатися з групи управлінського персоналу, об'єднаного спільними інформаційними функціями. Ці працівники можуть відповідати за такі завдання, як управління запасами, планування транспортування та складування.

Інформаційна система повинна складатися з взаємозалежних елементів у впорядкованому порядку та мати повний набір характеристик. Поділ інформаційної системи на її складові елементи можна здійснити різними способами. Більшість інформаційних систем поділяють на дві підсистеми: функціональну та підтримку.

У загальному вигляді логістична інформаційна система – це автоматизована система управління логістичними процесами. Отже, математичне забезпечення в логістичній інформаційній системі являє собою комплексну програму з набору засобів програмування, які забезпечують рішення для управління логістикою, загального збору даних та роботи обладнання.

Логістичні інформаційні системи управління матеріальними потоками можуть бути створені як на мікрорівні, так і на макрорівні.

Логістичні інформаційні системи мають різні функції та підсистеми, які дозволяють їм виконувати різні завдання. Функціональні підсистеми мають різний склад завдань. Підсистема забезпечення може відрізнитися за всіма своїми елементами (тобто технічним, інформаційним, математичним забезпеченням).

На вищому рівні управління створюються планові інформаційні системи, які використовуються для прийняття довгострокових стратегічних рішень.

Завдання, які необхідно вирішити:

- Організація та оптимізація етапів логістичного ланцюга;
- Керування змінними даними;
- Планування виробництва;
- Цілісне управління запасами;
- Контроль над резервами та інші аспекти управління.

Інформаційні системи планування розробляються з урахуванням загальновизнаних стандартів, що робить їх більш універсальними та гнучкими. Це дозволяє налаштувати стандартне програмне забезпечення відповідно до потреб вашої організації з меншими витратами часу та зусиль.

Інформаційна система обладнання призначена для управління логістичними процесами на рівні складу та магазину. Вони дозволяють вирішувати такі завдання, як:

- Детальне управління запасами, включаючи відстеження рівня запасів, переміщення запасів між складами та місцями зберігання, а також очищення запасів;
- Керувати транспортуванням на складі або всередині заводу, включаючи планування маршрутів, управління вантажами та відстеження стану товарів;
- Підбір товарів за замовленнями, в тому числі формування комплектів товарів під конкретне замовлення, а також відвантаження товарів клієнтам.

В інформаційних системах пристроїв адаптованість до стандартних програмних пакетів низька. Це пояснюється кількома причинами, а саме:

- виробничий процес підприємства зумовлений історією і навряд чи зазнає серйозних змін з точки зору стандартизації;
- структури даних, які обробляються різними користувачами, сильно відрізняються.

Інформаційні системи управління будуються на рівні управління або рівні

оперативного управління. Швидкість обробки інформації в логістичних інформаційних системах обмежується швидкістю надходження інформації до комп'ютера. Це так званий режим роботи в реальному часі, який дозволяє отримувати необхідну інформацію про рух вантажів в даний момент, дозволяючи своєчасно управляти і впливати на керовані об'єкти. Ці системи можуть вирішувати різноманітні завдання, такі як контроль матеріальних потоків, технічне обслуговування виробництва та управління операційною поведінкою.

Загалом для управління інформаційною системою управління використовується лише одне програмне забезпечення.

Відповідно до логістичної концепції інформаційні системи різних груп об'єднуються в одну інформаційну систему.

Існує вертикальна інтеграція і горизонтальна інтеграція.

Вертикальна інтеграція — це зв'язок систем планування, доставки та виконання через вертикальні потоки інформації.

Горизонтальна інтеграція означає з'єднання різних наборів завдань у системах прийняття та виконання рішень через горизонтальні інформаційні потоки.

У загальному вигляді переваги інтегрованих інформаційних систем можна виразити так:

- підвищення швидкості обміну інформацією;
- зменшилась кількість облікових помилок;
- менш ефективно завантаження паперу;
- об'єднати раніше різні блоки інформації.

У сучасному світі дуже популярні такі інформаційно-логістичні системи:

1. Urreg. Створений для задоволення надзвичайних вимог організацій, Urreg пропонує низку функцій, призначених для згладження планування маршрутів, розкладу та оптимізація. З Urreg легко додавати різні зупинки. Незалежно від того, чи є у вас скромна група доставки чи розширений перелік пунктів призначення, Urreg спрощує процес. Розставляйте пріоритети для важливих поставок і легко призначайте точні часові вікна, щоб гарантувати



своєчасне прибуття, покращуючи логістичні операції. Дана програма зображена на рисунку 1.1

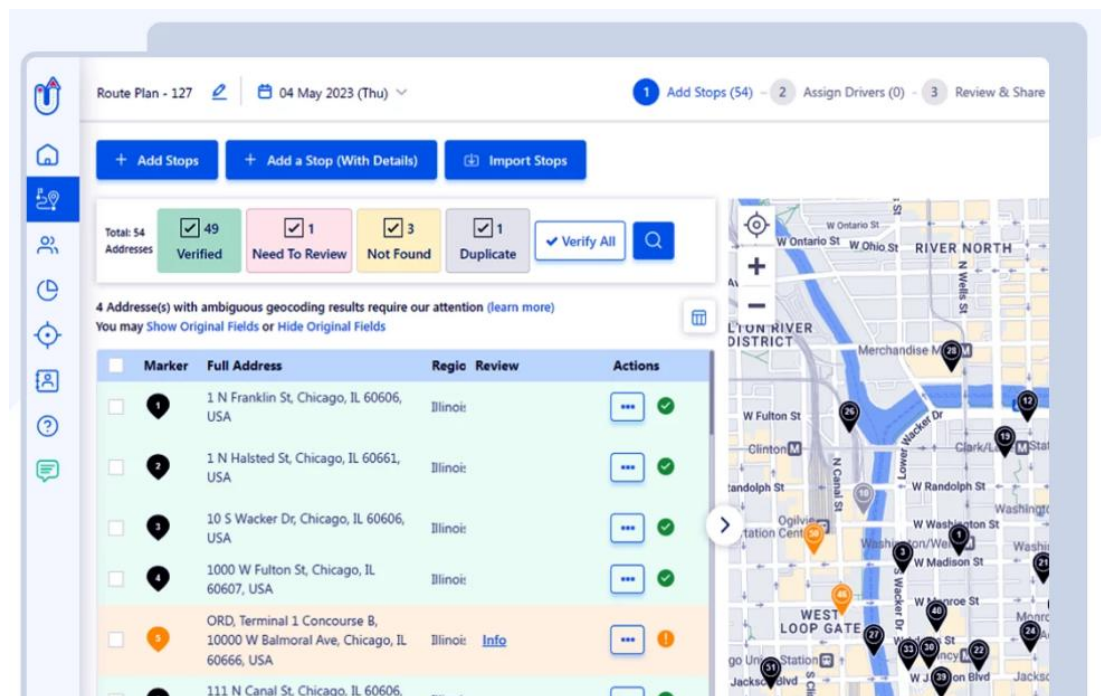


Рисунок 1.1 Upper

2. Brightpearl – це програмне забезпечення для управління бізнесом, спеціально створене для роздрібних та оптових компаній. Воно інтегрує в себе різноманітні функції, включаючи управління складом, замовленнями, фінансами, CRM та інші ключові аспекти бізнесу. Ця платформа дозволяє автоматизувати багато процесів у бізнесі, збільшуючи ефективність та знижуючи ризики помилок. Brightpearl дозволяє підприємствам краще відстежувати запаси, обробляти замовлення, керувати відносинами з клієнтами та виконувати фінансовий облік. Дана програма зображена на рисунку 1.2

The screenshot displays the Brightpearl Sales interface. At the top, there is a navigation bar with menu items: Marketing, Channels, Customers, Helpdesk, Sales, Purchases, Accounting, Products, Departments, Reports. The user is identified as Bob Thompson. On the right, there are links for App Store, Settings, and Help, along with a search bar and a 'Customers' dropdown. Below the navigation bar, the 'Sales' section is active, showing a list of orders. The interface includes buttons for 'Update status', 'Send email', 'Send SMS', 'Print', 'Delete', 'Invoice', 'Allocate', 'Fulfill', and 'More actions'. A table of orders is shown with columns for Order ID, Reference, Status, Channel, and Delivery ZIP Code. The status column contains various icons and text labels such as 'New eCommerce order', 'Back order', 'Ready to Ship', and 'Invoiced'. The table lists 10 orders with their respective IDs and references.

Order ID	Reference	Status	Channel	Delivery ZIP Code
5822	200001440676602	New eCommerce order	ebay	20505
5821	200001440676368	Back order	ebay	90210
5820	2000014404578912	Back order	amazon	26784
5819	2000014406764165	New eCommerce order	Shopify	22313
5818	2000014312240446	Ready to Ship	Big Commerce	20505
5817	2000014012089746	Invoiced	ebay	90210
5816	2000014414578319	Invoiced	amazon	26784
5815	2000014321516894	Invoiced	Magento	22313

Рисунок 1.2 Brightpearl

3. Forecast NOW! - це програмне рішення, яке допомагає прогнозувати попит на товари, керувати запасами продукції та оцінювати їхню ефективність. Ця програма складається з математичної моделі та інтерфейсу, який можна налаштувати під потреби користувача. Вона призначена для торгових компаній з широким асортиментом товарів у різних сферах, таких як продукти, побутова техніка, будівельні матеріали, медичне обладнання та інше. Forecast NOW! допомагає розраховувати оптимальний обсяг замовлення для кожного товару, враховуючи його характер попиту, сезонність, акції, аналогічні товари та обмеження в поставках.

4. Descartes – надає численні програмні рішення для роздрібної торгівлі, транспорту та логістики, виробництва, розповсюдження та обслуговування на місцях продажу. Якщо ви працюєте в роздрібній торгівлі, ви можете використовувати їх програмне забезпечення для керування складом, розповсюдженням у магазині та доставкою додому. Так само, якщо ви займаєтесь транспортним бізнесом, ви можете використовувати Descartes для свого обліку. Програмне забезпечення допоможе надати вам інтегровані інструменти фінансового управління та звітності, що стосуються багатьох підрозділів. Дана програма зображена на рисунку 1.3

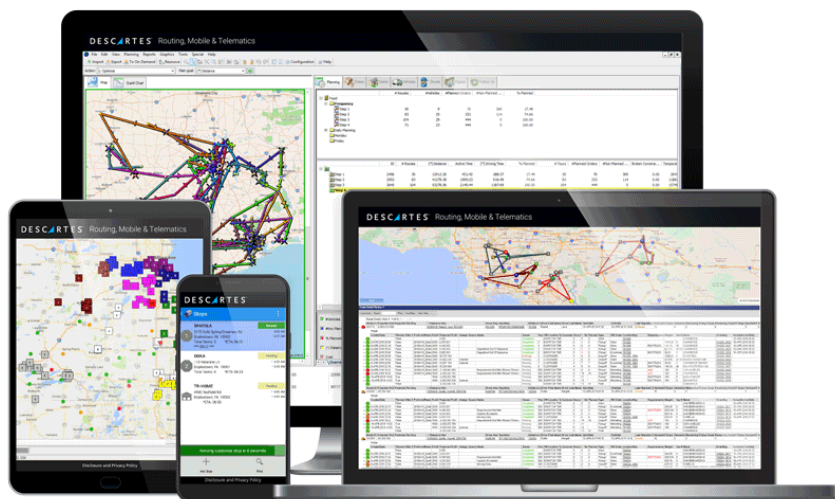


Рисунок 1.3 Махортра

Використання ІТ в управлінні ліками може зменшити транспортні витрати, забезпечити з'єднання різних частин і контролювати всі операції всередині та за межами фармацевтичної компанії.

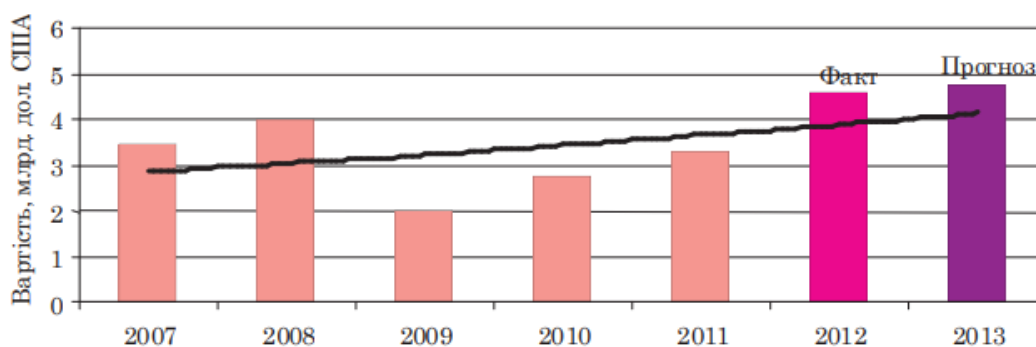


Рисунок 1.4 Динаміка впровадження ІТ у використанні вітчизняними підприємствами та її прогнозування

## 1.2 Функціональна й організаційна структура логістичної інформаційної системи

Функціональна структура логістичної інформаційної системи має форму піраміди. Він базується на операційній системі між кожною ланкою логістичної системи та визначає відносини між функціональними підрозділами компанії (з точки зору реалізації логістичних функцій), логістичними посередниками та споживачами компанії. На аналітичному рівні регіональні адміністративні або логістичні менеджери компанії використовують інформацію переважно для тактичних маркетингових цілей і прогнозування фінансових і операційних результатів. На найвищому стратегічному рівні логістика визначає стратегію управління, яка тісно пов'язана зі стратегічним плануванням і місією компанії.

Системні характеристики функціональної структури логістичних інформаційних систем пов'язані з найбільшою успішністю окремих стратегічних партнерів, тактичними цілями підприємств, конкурентними перевагами.

Організаційна структура логістичної інформаційної системи (ЛІС) складається з чотирьох взаємопов'язаних підсистем, які забезпечують інформаційно-комп'ютерну підтримку всіх функцій логістичного менеджменту та зв'язку з мікро- та макрологістичним середовищем.

Ці підсистеми:

- Управління замовленнями відповідає за обробку замовлень від клієнтів, відстеження їх виконання та управління запасами;
- Дослідження та комунікації збирає та обробляє інформацію про ринок, конкурентів та зовнішні умови;
- Підтримка логістичних рішень надає інструменти для аналізу даних та моделювання;
- Вихідна таблиця та формування звітів забезпечує доступ користувачів до інформації, необхідної для аналізу та планування логістичної діяльності.

В організаційній структурі логістичної інформаційної системи підсистему

управління програмою замовлення називають однією з основних підсистем, яка має безпосередній контакт зі споживачами в процесі обробки та виконання замовлення. Ключовим тут є поняття «електронний обмін даними» та використання заснованих на ньому стандартів.

Дослідницько-комунікаційна підсистема відображає вплив зовнішнього та внутрішнього середовища компанії на процес управління логістикою та здійснює зв'язок між усіма аспектами логістичної системи та функціями управління:

- підготовка або придбання технічних рішень і процедур для використання посередників;
- взаємодія управління логістикою з іншими корпоративними функціями;
- стратегічне налаштування логістичних систем і організації персоналу;
- акцент на продуктивності та якості обслуговування в індустрії логістики;
- адаптація до фізичної форми логістичних ланцюгів, каналів і мереж та умов виконання функцій управління;
- інтеграція логістичного планування та корпоративного планування;
- інтеграція інформаційних технологій.

Уже згадані підсистеми відображають зміни та вимоги зовнішнього та внутрішнього середовища підприємства. Ця підсистема дозволяє менеджерам з логістики сканувати мікро- та макросередовище компанії чотирма способами:

- непрямий розгляд, коли немає конкретної мети, загальний аналіз отриманої інформації;
- безпосередній розгляд, коли така інформація про зовнішнє та внутрішнє середовище компанії активно аналізується з наперед визначеною метою;
- неформальні дослідження обмежених і неструктурованих даних;
- формальні дослідження в обробці та аналізі отриманої інформації з використанням заздалегідь розроблених планів, процедур і методів.

Щоб оцінка впливу зовнішнього та внутрішнього середовища підприємства на поведінку логістичної системи була максимально ефективною, менеджери

логістики повинні використовувати ключові інформаційні джерела підсистем у процесі моніторингу. Тут слід розглянути два аспекти. По-перше, інформація використовується для аналізу ефективності логістичних процесів. Наприклад, аналіз бухгалтерської інформації або даних про ціни на аналогічний продукт конкурентів може дати повну картину щодо ефективності управління. Також можна використовувати дані про розмір відвантажень, які здійснює відділ доставки компанії, щоб отримати цінну інформацію для оцінки та аналізу ефективності управління логістикою та виробництвом. По-друге, логістичні партнери компанії також можуть використовувати інформацію з підсистем логістичної інформаційної системи для того, щоб краще координувати свої дії та уникати дублювання зусиль.

Третім компонентом є підсистема підтримки логістичних рішень (ПСПЛ) — це ключовий компонент логістичної інформаційної системи (ЛІС). Вона забезпечує менеджерам логістики доступ до інформації та аналітичних моделей, необхідних для прийняття обґрунтованих рішень у сфері логістики. ПСПЛ включає в себе чотири типи баз даних:

- Основні документи містять зовнішню та внутрішню інформацію, необхідну для прийняття логістичних рішень. Ця інформація може включати дані про попит на товари, витрати на логістику, а також інформацію про конкурентів і зовнішнє середовище;
- Після прийняття рішення ключові фактори визначають основні дії, цілі та обмеження. Ці фактори можуть включати такі параметри, як час доставки, рівень запасів та рівень обслуговування клієнтів.
- Політики або параметри охоплюють основні процедури логістики в ключових сферах операцій. Ці процедури можуть включати такі процеси, як управління запасами, транспортування та складування.
- Файли рішень зберігають інформацію про попередні (періодичні) рішення різних логістичних функцій. Ця інформація може бути використана для аналізу тенденцій і розробки нових рішень.

Ця підсистема логістичної системи використовує економіко-математичні моделі та методи для прийняття рішень. Ці моделі та методи можна розділити на

три категорії: оптимізаційні, евристичні та імітаційні. Оптимізаційні моделі використовують операційні обчислювальні методи для пошуку найкращого рішення. Евристичні моделі використовують спрощені методи для пошуку прийняттого рішення. Імітаційні моделі використовують комп'ютерне моделювання для дослідження поведінки системи. Логістична система виконує широкий спектр функцій, включаючи:

- Організація виробництва, транспортування та обробки товарів з метою оптимального пересування матеріалів та інформації від постачальників до кінцевих споживачів;
- Розміщення предметів у виробництві, розподілі та складуванні для оптимізації використання ресурсів і мінімізації витрат;
- Побудову оптимальних логістичних ланцюгів, каналів, мереж для забезпечення безперервного і ефективного руху матеріалів і інформації;
- Побудову оптимальної організаційної структури логістичної системи для забезпечення ефективної взаємодії між різними підрозділами компанії;
- Вибір оптимального маршруту для доставки вантажів з мінімальними витратами часу і коштів;
- Визначення оптимальної тривалості складових логістичних циклів для забезпечення високого рівня обслуговування клієнтів;
- Оптимізацію процедур збору, агротехніки та виконання замовлень для підвищення ефективності логістичних процесів;
- Оптимізацію параметрів системи управління запасами для забезпечення достатнього рівня запасів, але без надмірних витрат;
- Вибір найкращих перевізників, експедиторів, постачальників тощо для забезпечення ефективного і надійного логістичного обслуговування.

У підсистемах широко використовуються інтерактивні (діалогічні) програми, засновані на інформації для прийняття логістичних рішень.

Четвертим компонентом в організаційній структурі логістичної інформаційної системи є підсистема формування вихідних таблиць і звітів.

Інформаційні системи матеріально-технічного забезпечення вищевказаних

функцій повинні бути належним чином організовані. Особливістю цієї системи є те, що вона може впливати на всі функціональні підсистеми логістичної організації в процесі експлуатації. На цій основі можливі три організаційні форми: централізована, децентралізована і спеціалізована.

Централізована організація інформаційного забезпечення передбачає, що всі функції та процеси, пов'язані з інформаційними системами та технологіями, зосереджені в одному підрозділі, який підпорядковується безпосередньо вищому керівництву компанії. Перевагою такого підходу є можливість ефективного впровадження нових інформаційних систем і технологій. Це пов'язано з тим, що централізоване управління дозволяє забезпечити єдину політику в галузі інформаційного забезпечення, а також координувати діяльність різних підрозділів компанії в цій сфері. Недоліком централізованої організації інформаційного забезпечення є висока вартість утримання менеджерів. Це пов'язано з тим, що в централізованому підрозділі зазвичай працює велика кількість кваліфікованих фахівців, які отримують високу заробітну плату.

Завдяки децентралізації організації підсистеми інформаційного забезпечення фахівці різних функціональних підрозділів беруть на себе функцію управління інформаційними потоками в межах своїх спеціалізацій. Перевага цього методу організації полягає в тому, що він забезпечує передові знання сфери компетенції адміністратора інформаційних систем, але недолік полягає в тому, що він перекриває подібні завдання та функції уряду в різних частинах організації.

Через спеціалізований підхід організація не має відділу інформаційних систем (технологій). Коли мова йде про розробку та впровадження нових інформаційних систем, робота передається спеціалізованим компаніям (аутсорсинг).

Це типова невелика організація, яка може не мати штатних ІТ-експертів і консультантів. Перевага цього методу організації систем інформаційного забезпечення полягає в тому, що він є дуже науковим і методологічно складним, але його недоліком є те, що важко врахувати конкретні властивості об'єкта.

Спосіб вибору організацією тієї чи іншої системи інформаційного



забезпечення залежить від багатьох факторів, в першу чергу від розміру організації, поточних бізнес-процесів і доступності оплати. Слід зазначити, що зараз системи інформаційного забезпечення досягли такого рівня спеціалізації, який потребує особливої уваги до їх організації – це добре розуміють сучасні менеджери. Тому будь-яка невелика організація має інформаційні служби. Інформаційна система, необхідна для повноцінного виконання логістичних функцій, повинна відповідати наступним вимогам:

- інформаційний потік має бути сумісним з точки зору інформації;
- внутрішні зв'язки та взаємозалежності інформаційного потоку мають бути причинно-наслідковими;
- ієрархічна приналежність інформаційного потоку повинна бути чіткою.

Формування логістичної інформаційної системи здійснюється за принципом ієрархії, кількість рівнів логістичної інформаційної системи починається з найнижчого рівня. Цей принцип прийнятий для забезпечення можливості побудови інформаційних систем на більш високому рівні і, за необхідності, включення їх як підсистем у системи та мережі на загальному рівні на високому рівні, ніж.

### **1.3 Класифікація інформаційних потоків у логістиці**

Для створення інформаційно-логістичної системи необхідно уявити, як вона буде працювати, з точки зору логістики. Для цього потрібно зрозуміти, як працює система управління виробництвом, і як вона взаємодіє з іншими підрозділами підприємства.

У структурі підприємства розрізняють об'єкти, що перебувають у стані спокою (буфери), та технічні частини, що перебувають у русі. Це охоплює всі етапи життєвого циклу. Потім визначається, які конкретні дані потрібно зібрати, обробити та передати для досягнення оптимального управління логістикою.

Дані збираються по кожному об'єкту, а в буферних і технічних зонах створюються пункти зведення та роздачі інформації. Запис має містити наступні дев'ять інформаційних елементів для кожного елемента роботи: Тип пропозиції. Кількість (обсяг); Джерело відправлення. його місцезнаходження; час прибуття до місця проживання; Час вибуття з місця проживання. Транспортування; Доставка; Бронювання. Отримані дані розподіляються між двома комп'ютерними системами. Перша система контролює матеріальний потік і управління матеріалами. Інша система безпосередньо відповідає за нагляд за виробництвом і управлінням запасами матеріалів. Його кількість залежить від вимог виробничого процесу.

Логістична інформаційна система (ЛІС) обробляє інформацію за допомогою прикладних програм або вручну. Дані ЛІС можуть використовуватися іншими підрозділами підприємства.

Зміст інформації визначається потребами підрозділів підприємства та вироблюваних управлінських рішень. Для інформації встановлюються такі вимоги:

- стислість;
- своєчасність;
- точність;
- вірогідність;
- правильний добір початкових відомостей.

Отже, ІСЛУ відповідає за обробку та передачу інформації, необхідної для логістичного управління підприємством. [9]

Інформація є ключем до прийняття ефективних рішень. Щоб забезпечити повний потік інформації, необхідно дотримуватися двох принципів: Дані слід збирати якомога ближче до виробничої та маркетингової зони, де походить подія. Дані мають бути представлені у форматі, придатному для перетворення та порівняння.

ІСЛУ повинні забезпечувати горизонтальну (по функціональних підсистемах) і вертикальну (по ієрархіях) інтеграцію логістичної системи. Головною проблемою в управлінні логістикою є організація процесів, які забезпечують своєчасну та достатню інформацію про ринкові умови.

Потік інформації має бути сформований таким чином, щоб відповісти на запитання: для чого потрібна ця інформація? На яку внутрішню інформацію ви можете покластися та наскільки вона повна та надійна? Які зовнішні інформаційні факти насправді доступні, яка вторинна інформація може бути надійно використана і як? Які технології, люди та ресурси можна використовувати для створення та споживання інформаційних каналів? Які вимоги до оперативності збору інформації та її достовірності?

Інформаційні потоки в інформаційній системі логістики мають:

- співпрацювати між собою;
- бути взаємопов'язаними, включаючи причинно-наслідкові зв'язки;
- мати визначену ієрархію та організовані взаємозв'язки;
- в цілому мати здатність до інтеграції.

Отже, інформаційний потік – це рух даних, призначений для виконання певної функції. Він може бути як формальним, так і неформальним, і може відбуватися як всередині організації, так і між організаціями. [7]

Інформація – двигун логістичної системи, завжди відкритий і здатний адаптуватися до нових умов. У зв'язку з цим одним з основних понять в логістиці є поняття інформаційного потоку. Загалом, інформаційний потік - це рух у середовищі даних, представлених у структурованій формі. Для логістики інформаційний потік - це сукупність інформації, що циркулює між логістичною системою та зовнішнім середовищем логістичної системи, необхідної для управління та контролю логістичної діяльності.

У логістиці існує кілька видів інформаційних потоків:

1. Залежно від типу системи, до якої підключений потік:
  - а) горизонтальні, що входять до основного рівня логістичної системи;
  - б) вертикальні - від верхнього до нижнього рівня в цій системі.
2. За місцем проходження:
  - а) зовнішні, що обмінюються між логістичною системою та зовнішнім середовищем;

- b) внутрішні, що циркулюють всередині системи або окремих її частин.

3. За напрямком системи:

- a) вхідні;
- b) вихідні.

4. За типом носіїв інформації:

- a) паперові;
- b) магнітні;
- c) електронні;
- d) інші.

5. За призначенням:

- a) директивні (управляючі);
- b) нормативно-довідкові;
- c) облік і аналіз;
- d) допоміжні.

Інформаційний потік можна виміряти за кількістю інформації, що передається за одиницю часу. Цей потік може бути паперовим або електронним, і його можна виміряти в одиницях паперових документів, рядках документів або бітах.

Параметри, які характеризують інформаційний потік:

- місце походження;
- напрямок руху;
- регулярність;
- форма існування;
- швидкість передачі та отримання;
- інтенсивність потоку тощо.

Можна управляти потоком інформації шляхом:

- змінюючи напрямок потоку;
- регулювання швидкості передачі, відповідно до швидкості прийому;
- обмеження обсягу потоку до максимальної пропускної здатності

окремих точок або частин маршруту передачі.

Інформація допомагає координувати, планувати та контролювати повсякденну діяльність. Зазвичай зміст інформаційного потоку відображає зміст документа. Недостовірна та неточна інформація або відсутність інформації може призвести до проблем і помилок на всіх рівнях логістичної системи. Наприклад, затримка в отриманні або обробці інформації про товари, доставлені комерційному підприємству, може порушити нормальну роботу приймального відділу, спричинити простої транспортного засобу та, зрештою, уповільнити потік продуктів у логістичний ланцюг.

Таблиця 1.1 Класифікація інформаційних потоків у логістиці

№ з/п	Класифікаційна ознака	Різновид інформаційного потоку	Характеристика інформаційного потоку
1	Залежно від виду систем, зв'язаних потоком	горизонтальні	належать до одного рівня ієрархії логістичної системи
		вертикальні	від верхнього рівня логістичного менеджменту до нижчого
2	За часом виникнення інформації	регулярні	відповідні до регламентованого в часі передавання даних
		періодичні	із жорстким обмеженням на час передачі
		оперативні	забезпечують зв'язок абонентів в інтерактивному та діалоговому режимах «On line» і «off line»

## 1.4 Переваги використання веб-платформ у сфері логістики

Веб-платформи — це комплекти інструментів та технологій, які дозволяють розробляти та запускати веб-додатки. Ці інструменти та технології розроблені та підтримуються відкритими спільнотами, такими як Консорціум World Wide Web (W3C). Веб-додаток — це програма, яка працює в веб-браузері. Браузер є клієнтською частиною додатка, а веб-сервер — серверною частиною. Логіка програми централізована на сервері, а функція браузера полягає в основному в тому, щоб відображати інформацію, завантажену з сервера, і відправляти дані користувача назад. Однією з переваг веб-додатків є те, що вони не прив'язані до конкретної операційної системи користувача. Це означає, що веб-додаток можна використовувати на будь-якому пристрої з веб-браузером, незалежно від того, яку операційну систему він використовує. Через цю універсальність і відносну простоту розробки веб-додатків вони стали широко популярними в кінці 1990-х і на початку 2000-х років. [16]

Однією зі сфер розвитку Всесвітньої павутини є управління даними та вмістом. Ця сфера включає в себе керування великими даними, керування даними хмарних обчислень, керування мультимедійними даними та інші сфери, такі як веб-майнінг, кластеризація веб-даних, класифікація та аналіз, моделювання візуалізації веб-вмісту, семантична мережа тощо. Завдання ефективного управління веб-контентом стало важливим для багатьох галузей, які виростили з веб-інфраструктури, включаючи електронну комерцію, освітні та дистанційні веб-ресурси, а також управління великими інформаційними порталами, розробку та підтримку корпоративних інформаційних порталів, електронну мережу. ресурсів, управління, особист. підтримка сайтів і блогів тощо.

До популярних програмних рішень для управління контентом відносяться такі системи CMS: Drupal, Joomla, Wordpress, Plone та ін. Moodle і aTutor зазвичай використовуються для дистанційного навчання, а MediaWiki є інструментом для створення освітніх вікі-проектів. Перевагами цих систем є відкритий код, велика

кількість програм і безкоштовні ліцензії, що зумовило їх популярність. Запатентовані системи: Magento, Microsoft SharePoint, Adobe Business Catalyst (інтегрована система SaaS); Дошка (система дистанційного навчання).

Незважаючи на те, що існує багато програмних систем, що використовуються для управління контентом, залишається актуальним завдання дослідження, розробки та вдосконалення загального управління інформаційними об'єктами веб-ресурсів, яке має бути простим, оптимізувати створення нових ресурсів різного призначення та забезпечить ефективні механізми для управління їх змістом. обслуговування та налаштування. Розглянуто основні принципи створення сучасних веб-орієнтованих інформаційних систем.

Web-додатки в порівнянні з локальними програмами мають наступні переваги:

- легкий доступ - будь-яка особа з комп'ютером та доступом до Інтернету може користуватися web-додатком;
- простота встановлення - після розробки web-додатків їх не треба встановлювати на комп'ютерах користувачів. Достатньо надати URL-адресу. Зміни в додатку відразу впроваджуються для усіх користувачів;
- надійність мережі та web-технологій - високий рівень розвитку та надійності мережевих з'єднань та web-технологій сприяє стабільності роботи web-додатків web-додатки легше обслуговувати, оскільки вони мають спільну кодову базу на кількох мобільних платформах;
- легше обслуговування - Web-додатки зручніші в обслуговуванні завдяки спільній кодовій базі для кількох мобільних платформ;
- розробники web-додатків не зобов'язані подавати свої додатки на розгляд до магазину додатків;
- web-додатки не підлягають схваленню магазином додатків, тому їх можна публікувати без жодних обмежень. Розробники мають повну свободу вибору часу та форми випуску своїх додатків.

Користувачі можуть отримувати доступ до найновішої версії веб-додатку безпосередньо, без необхідності відвідувати магазин додатків.

## 2 ОСОБЛИВОСТІ ТА ТРУДНОЩІ ЗАСТОСУВАННЯ СУЧАСНИХ ІТ-ТЕХНОЛОГІЙ У ЛОГІСТИЦІ

### 2.1 Шляхи оптимізації обробки даних за допомогою ІТ-технологій

Технічний процес обробки інформації - це сукупність взаємопов'язаних технічних операцій, що виконуються над інформацією в певному порядку. Потіки процесу в процесі можна комбінувати різними способами. При цьому визначається тип технологічного потоку (оперативний, обов'язковий, змішаний). [30]

Основні типи інформації, які є найбільш важливими для обчислювальної техніки з точки зору представлення, кодування та методів зберігання:

- графіка або зображення - перший вид, спосіб збереження інформації про навколишній світ у вигляді петрогліфів, потім у вигляді зображень, фотографій, схем, на папері, полотні, мармурі тощо. досягати. Матеріал представляє реальне зображення;

- звуковий - навколишній світ наповнений звуками, винахід звукозаписуючої апаратури в 1877 р. вирішив проблему їх зберігання та відтворення, а його різновид включає музичну інформацію - адже цей вид винайшов спосіб кодувати її спеціальними символами та інформацією;

- текст - це спосіб кодування людської мови за допомогою спеціальних символів - алфавітів, різні народи мають різні мови і використовують різні набори літер для позначення мови; Цей спосіб набув особливого значення після винайдення паперу та друкарства;

- числове значення - кількісне вимірювання об'єктів та їх властивостей у світі; З розвитком торгівлі, економіки та іноземної валюти він набуває особливого значення; Подібно до відображення текстової інформації, методи кодування за допомогою спеціальних символів - за допомогою чисел, системи кодування можуть бути різними;



- відеоінформація – спосіб збереження "живих" картин навколишнього світу, що з'явився з винаходом кіно.

Існують також ще не винайдені способи кодування та зберігання інформації: тактильна – через органи чуття, сенсорна – через нюх і смак.

Інформаційні процеси, такі як збір, обробка та передача інформації, завжди були необхідними для розвитку науки, техніки та суспільства. Навіть у міру розвитку людської цивілізації ці процеси залишаються незмінними за своєю суттю, але способи їх виконання постійно автоматизуються. [30]

Збір інформації — це процес отримання знань про щось. Це може бути зроблено різними способами, наприклад, шляхом читання, спостереження, опитування або експериментування. [9]

Інформацію можна зібрати самостійно або за допомогою технічних пристроїв. Наприклад, користувач може отримати інформацію про роботу суден або літаків, ознайомившись з розкладами безпосередньо від іншої особи або через документацію, встановлену цією особою, або використовуючи технічні засоби (супровід авто, телефон тощо). Завдання збору інформації може вирішуватися незалежно від інших завдань, у тому числі й завдання обміну (передачі) інформацією.

Обробка інформації – це систематизоване перетворення інформації відповідно до певних правил, щоб отримати бажаний результат. [30]

Після вирішення завдання обробки інформації результати повинні бути представлені кінцевому користувачеві в бажаному форматі. Цей процес реалізується в результаті завдання Опублікувати інформацію. Розповсюдження інформації зазвичай здійснюється через зовнішнє комп'ютерне обладнання, таке як текст, таблиці та графіка.

У процесі обробки інформації вирішується кілька інформаційних завдань, які можуть бути задані в умовній формі. Тобто мені потрібно отримати конкретний результат. Набір вихідних даних. Процес перетворення входу на вихід є процесом обробки. Об'єкти переробки називають експортерами переробки.

Для успішної обробки інформації фахівець (фізична чи юридична особа)

повинен розуміти алгоритм обробки, тобто послідовність операцій, які необхідно виконати для досягнення бажаного результату.

Існує два способи обробки інформації. Перший спосіб – це отримання нової інформації або знань. Наприклад, це може бути розв'язання математичних задач, аналіз ситуації або виконання інших завдань, які вимагають нових знань або розуміння. Другий тип обробки інформації – це зміна форми інформації, але не її змісту. Наприклад, це може бути переклад тексту з однієї мови на іншу, перекодування інформації з одного формату в інший або зміна способу подання інформації.

Для зберігання, передачі та обробки інформації її необхідно перетворити в символну форму. Цей процес називається кодуванням. Кодування може здійснюватися за допомогою технічних засобів, таких як телеграф, радіо, комп'ютер. Інший вид обробки інформації - структурування даних. Цей процес полягає в тому, щоб надати даним певний порядок, класифікацію або каталогізацію. Структурування даних полегшує їх зберігання, пошук та обробку.

Пошук інформації здійснюється за допомогою алгоритму пошуку. Алгоритм пошуку - це послідовність дій, які виконуються для пошуку інформації. Алгоритм пошуку залежить від способу організації інформації в сховищі. Алгоритм пошуку наведено на рисунку 2.1. [8]



Рисунок 2.1 Схеми обробки інформації

Важливо відзначити, що в центрі уваги проектування процесу – реалізація. Модель, реалізована цією технологією, залежить від обсягу завдань та їх часових

характеристик: частоти та терміновості, вимог до швидкості обробки інформації та можливостей апаратної моделі (особливо комп'ютерної).

Бувають: пакетний режим, істинно пропорційний режим, метод розподілу часу, система модерації, голосування, діалог, дистанційна обробка, інтерактивний, програмний, мультипрограмний (багатопроесорний).

Для користувачів фінансово-кредитних систем найбільш актуальними є такі режими: пакетний, сеансовий та миттєвий. втрачений час.

Пакетний режим. Якщо використовується цей режим, користувач не зможе підключитися безпосередньо до комп'ютера. Час, необхідний для збору, запису, введення та обробки інформації, не є постійним. Спочатку користувач збирає інформацію та групує її за типом завдання або будь-яким іншим типом. Код (Це, як правило, невиконавче завдання, де наслідки рішення є довгостроковими). Після отримання повідомлення воно вводиться та обробляється, що призводить до затримок обробки. Цей режим часто використовується для централізованої обробки інформації.

Діалоговий режим (запиту), який дозволяє користувачам безпосередньо взаємодіяти з комп'ютерними системами під час роботи. Програми обробки даних постійно зберігаються в пам'яті комп'ютера, поки комп'ютер доступний у певний час або протягом певного періоду часу, поки комп'ютер доступний користувачеві. Взаємодія користувача з комп'ютерною системою у формі розмови може бути багатоплановою і визначається багатьма факторами: мовою спілкування, активною або пасивною роллю користувача, який є ініціатором розмови: користувач або комп'ютер. ; Іноді відповідь, структура розмови тощо. Якщо користувач ініціює розмову, він повинен розуміти використання програмного забезпечення, формат даних тощо. Якщо ініціатором є комп'ютер, то пристрій сам підкаже, що робити на кожному кроці, а також параметри. Такий спосіб роботи називається «вибір меню». Він підтримує операції користувача та його команди. При цьому користувачі недостатньо підготовлені.

Діалоговий режим вимагає від користувача наявності певного рівня технічного оснащення, тобто терміналу або персонального комп'ютера з каналом

зв'язку, підключеного до центральної комп'ютерної системи. Цей режим використовується для доступу до інформації, комп'ютерних ресурсів або програм. Можливість роботи в голосовому режимі може бути обмежена часом початку та закінчення.

Розмовний режим вимагає від користувача наявності певного рівня технічного обладнання, тобто терміналу або персонального комп'ютера з каналом зв'язку, підключеного до центральної комп'ютерної системи. Цей режим використовується для доступу до інформації, комп'ютерних ресурсів або програм. Можливість роботи в голосовому режимі може бути обмежена часом початку та закінчення. Реальний час означає здатність комп'ютерної системи взаємодіяти з керованими процесами або контролювати швидкість цих процесів. Час відгуку комп'ютера має відповідати швидкості контрольованого процесу або вимогам користувача з мінімальною затримкою. Ця модель зазвичай використовується для децентралізованих і розподілених обчислень. Приклад: комп'ютер встановлений на столі оператора, і вся операційна інформація вводиться в комп'ютер через комп'ютер.

Крім перерахованих, існує кілька способів обробки даних: централізований, децентралізований, децентралізований та інтегрований.

Централізація гарантує існування ОЦ. Таким чином, користувач передає вихідну інформацію на комп'ютер і отримує результати обробки з дійсними файлами. Такий підхід характеризується складністю і труднощами встановлення швидких і безперебійних з'єднань, великим навантаженням інформації ОС (через її розмір), термінами виконання та організацією безпеки системи для запобігання несанкціонованому доступу авторизованих.

Децентралізована обробка. Такий підхід пов'язаний з появою персонального комп'ютера, який дозволив автоматизувати конкретні робочі місця. В даний час децентралізовані обчислювальні технології в основному поділяються на три категорії.

Перший тип системи базується на автономних персональних комп'ютерах, які не пов'язані локальною мережею. Дані зберігаються в окремих файлах і на

окремих дисках. Для отримання метрик інформація переноситься з одного комп'ютера на інший. Мінуси: відсутність координації завдань, нездатність обробляти великий обсяг інформації, низький захист від несанкціонованого доступу.

Другий тип системи базується на персональних комп'ютерах, які об'єднані в локальну мережу. Це дозволяє зберігати всі дані в єдиному файлі. Однак цей файл може містити лише обмежену кількість інформації, оскільки він повинен бути доступним для всіх комп'ютерів у мережі.

Третій тип системи базується на персональних комп'ютерах, які об'єднані в локальну мережу. Ця мережа включає спеціальні сервери, які відповідають за зберігання і обробку даних.

Розподілений підхід до обробки даних заснований на розподілі функцій обробки між різними комп'ютерами, підключеними до мережі. Цей метод можна реалізувати двома способами: перший полягає в установці комп'ютерів на кожному вузлі мережі (або на кожному рівні системи), а дані обробляються одним або кількома комп'ютерами, залежно від фактичної продуктивності мережі. . . Комп'ютерні навички. система . та їхні поточні потреби. Іншим рішенням є об'єднання великої кількості різних процесорів в єдину систему. Цей шлях використовується для систем обробки банківської та фінансової інформації (установ, офісів тощо), які потребують комп'ютерної мережі. Переваги децентралізованого підходу: Можливість керувати будь-якою кількістю наборів даних; Висока надійність, оскільки при виході з ладу певного технічного засобу його можна негайно замінити іншою технологією; Зменшити час і вартість передачі даних; Підвищення гнучкості системи; Оптимізовані процеси розробки програмного забезпечення тощо. Децентралізований підхід передбачає набір спеціалізованих процесорів, тобто кожен комп'ютер призначений для вирішення конкретного завдання або рівня.

Наступний метод обробки даних - інтегрований. Це передбачає створення інформаційної моделі для керованих об'єктів, тобто створення розподіленої бази даних. Цей спосіб пропонує користувачам максимальну зручність. З одного боку,

сховища дозволяють колективне використання та централізоване управління. З іншого боку, великий обсяг інформації та різноманітність завдань вимагають децентралізації бази даних. Інтеграція методів обробки інформації може підвищити якість, надійність і швидкість обробки, оскільки обробка здійснюється на окремих рядках інформації після їх введення з комп'ютера. Особливістю цього підходу є поділ технічних і часових процедур збору, підготовки та введення даних.

Сучасні системи обробки інформації використовують цифрові технології, які дозволяють виключити паперовий документообіг. Вони обмінюються даними по мережах між робочими станціями, що дозволяє співробітникам об'єднуватися для вирішення будь-яких завдань. Для цього використовуються такі технології, як робочі групи, телеконференції, електронна пошта та електронні дошки оголошень. Для таких систем, що охоплюють роботу підприємства в цілому, застосовується термін «корпоративні системи управління бізнес-процесами». Вони використовують технологію «клієнт-сервер», що дозволяє підключати віддалених користувачів через глобальну мережу Інтернет. Такі системи можуть об'єднувати понад 40 тисяч користувачів, які розташовані в різних країнах і на континентах. Наприклад, компанія McDonalds використовує корпоративну систему управління бізнес-процесами для об'єднання своїх підрозділів по всьому світу, включаючи Україну.

## **2.2 Труднощі пов'язані із сучасними технологіями управління логістикою та можливі шляхи подолання цих проблем**

Необхідно більше уваги приділяти вивченню специфічних характеристик наркотиків, а не тільки пов'язаних з послугами транспортування чи переробки, оскільки наркотики включають і інші чинники: постачання, виробництво, реалізацію та зберігання. Має бути створена модель управління, яка враховує всі ці аспекти та дозволяє подолати перешкоди, з якими стикаються компанії та їхні ліки

при застосуванні ІТ. Але у зв'язку зі швидким розвитком ІТ-індустрії та її значенням у бізнес-операціях потреба у подоланні проблем застосування та розгортання постійно зростає, і необхідні подальші дослідження цієї галузі.

Однак у процесі впровадження ІТ у логістичний сектор все ще існують проблеми. Перешкоди для впровадження ІТ у фармацевтичному секторі бізнесу включають:

- більше ускладнює вивчення швидко мінливих інформаційних потоків у фармацевтиці, включаючи зміни власності, корпоративну диверсифікацію та складні ринкові відносини;
- розробляти інформаційні системи та програмне забезпечення для автоматизації комплексного управління бізнесом;
- необхідність удосконалення систем мобільного зв'язку;
- Доцільність постійного використання Інтернет-технологій в організаціях та забезпечення контролю мобільності в ЛС.

Основним напрямком розвитку комп'ютеризації логістики є інтеграція інформаційного потоку та комунікаційного забезпечення транспортування вантажів. Ці галузі пов'язані з процесом економічної інтеграції розвинених країн і представляють нову науково-практичну галузь - телекомунікації. З такими проблемами в Україні стикається розвиток цієї галузі, орієнтованої на активне використання комп'ютерних систем та інформаційних мереж:

- організація якості, обслуговування та ремонту обладнання;
- інтеграція інформаційних процесів;
- навчання персоналу;
- технологічні зміни в процесах, пов'язаних з обробкою та використанням наземної інформації;
- ціни на персональні комп'ютери та периферійні пристрої, засоби зв'язку;
- програмне забезпечення.

Головним фактором управління є швидкість обробки даних, що надходять безперервно, і прийняття необхідних рішень. Інформаційний потік істотно впливає



на ефективність управління логістичною системою.

Оскільки ЛС є сукупністю функціонально обмежених підсистем, питання ІТ-управління ЛС необхідно розглядати окремо для кожної підсистеми. Цілі впровадження ІТ наведені в таблиці 2.1. Сучасні інформаційні системи підвищують продуктивність і розширюють спектр виконуваних ними функцій. Інформаційні системи, які відіграють важливу роль у системах прийняття управлінських рішень у сфері логістики, мають переваги та недоліки. Однак загальна особливість полягає в тому, що ІТ активно використовуються між усіма ланками ЛС компанії, щоб інформаційні системи могли швидко реагувати на вимоги ринку і водночас створювати синергетичні зв'язки.

Таблиця 2.1 Мета впровадження ІТ кожної з ланок ЛС

Постачання	Забезпечення відсутності накопичення матеріалів і налагодження зв'язків з постачальниками та клієнтами
Виробництво	Забезпечення інформаційного зв'язку між усіма етапами логістичної системи та зниження простоїв у виробництві
Транспортування	Співпраця всіх ланок ланцюга поставок, зменшення транспортних витрат і скорочення часу на транспортування
Збут	Зменшення запасів готової продукції на складі та налагодження зв'язків з постачальниками
Складування	Ефективне керування запасами, забезпечення узгодженості між постачальниками та споживачами, оптимізація процесів зберігання готової продукції

Система підтримки прийняття рішень (СППР) є автоматизованою системою, що використовує дані та моделі для виявлення проблем та ухвалення рішень. Традиційна СППР складається з двох основних компонентів: бази даних і системи управління цією базою даних, яка допомагає приймати рішення та вибирати оптимальні варіанти, зображено на рисунку 2.2.

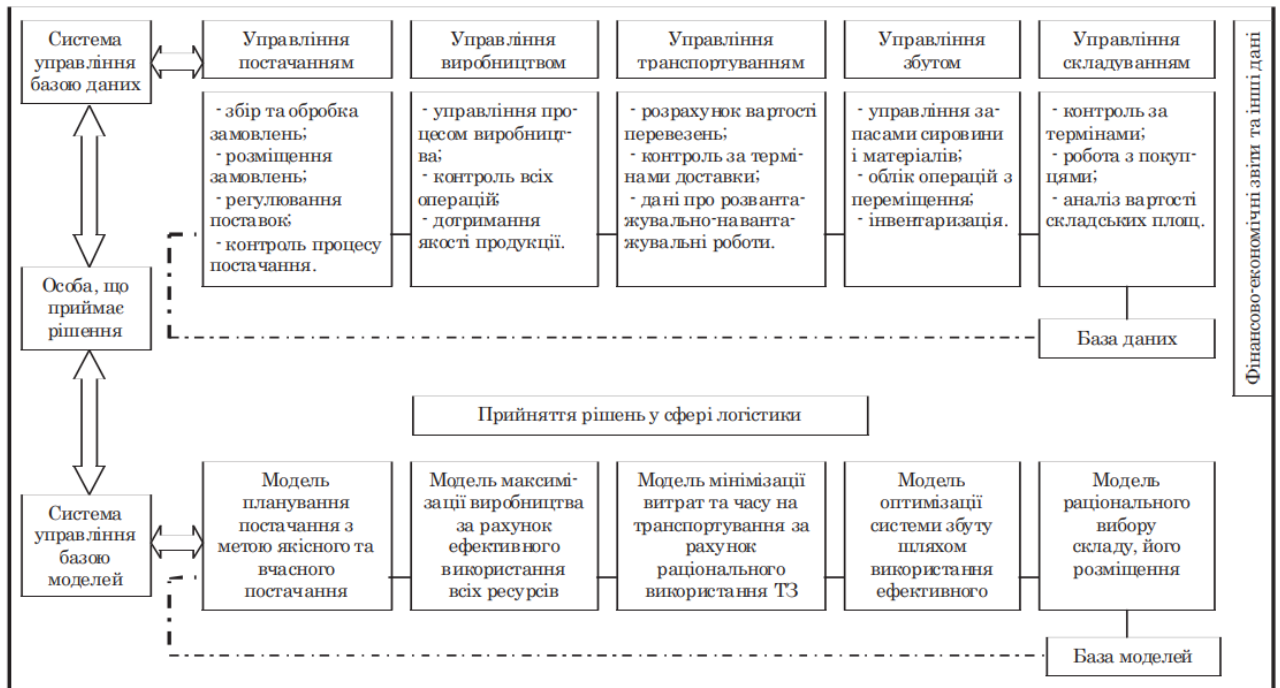
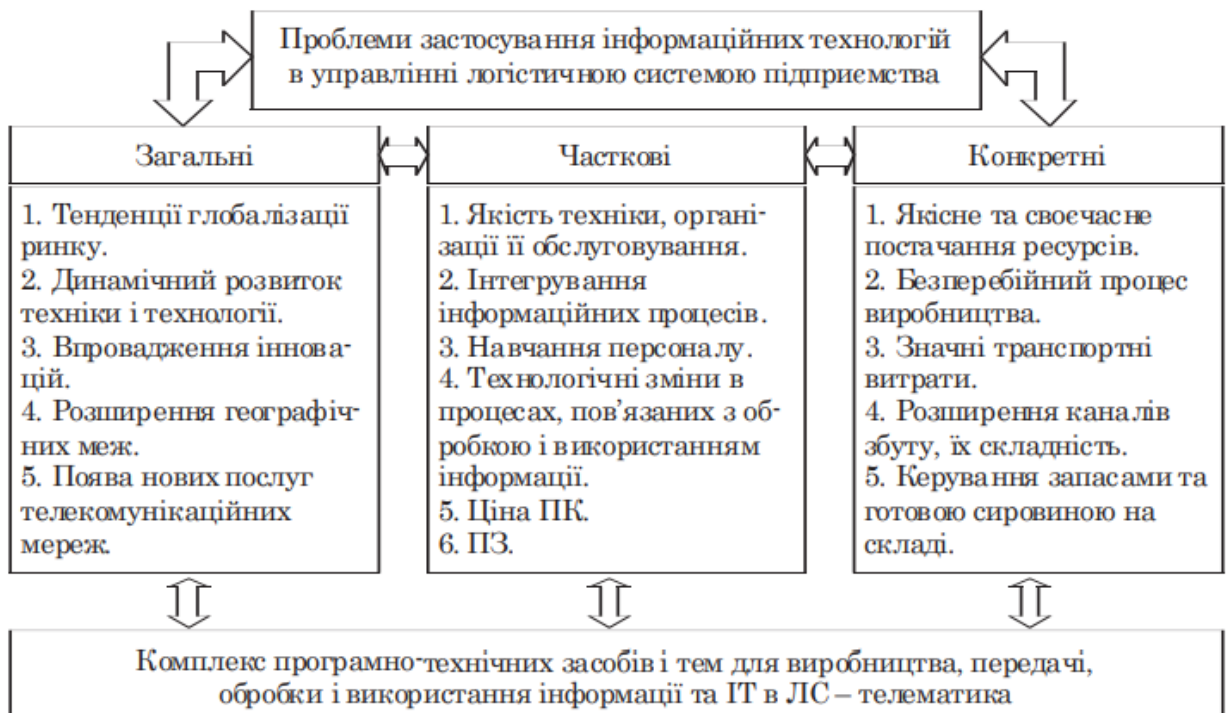


Рисунок 2.2 Структура управління логістичною системою підприємства

Після аналізу перспектив, напрямів розвитку, перешкод і можливостей застосування ІТ в логістиці, можна виділити загальні, часткові та конкретні проблеми застосування ІТ в управлінні логістичними системами підприємств, проблеми застосування ІТ зображено на рисунку 2.3.



### Рисунок 2.3 Проблеми застосування ІТ в логістичних системах

Загострення економічної кризи, злиття логістичних компаній та зміна керівництва можуть призвести до зміни масштабів проблем використання ІТ в управлінні логістичними системами підприємств. Щоб вирішити ці проблеми доцільно розробити такі шляхи їх подолання (від загальних до конкретних):

- розвиток українського інформаційного сектору;
- формування інтелектуальних ресурсів;
- стимули для активного фінансування ІТ-індустрії;
- тісно співпрацювати з ІТ-фахівцями;
- збільшити кількість управлінських рішень, які можна автоматизувати;
- навчання бізнес-персоналу для підвищення його здібностей та підвищення їх довіри до сучасних ІТ;
- забезпечення роботи фармацевтичних підрозділів в єдиному інформаційному середовищі;
- можливість поетапного впровадження ІТ-оцінок та контролю, економія ресурсів;
- розробляти програмні продукти, пов'язані з фармацевтикою компанії.

Майбутні дослідження у цій області мають на меті створення всеосяжної ІТ-моделі, яка значно підвищить ефективність управління логістичними системами підприємства.

## 3 ШЛЯХИ ОПТИМІЗАЦІЇ ЛОГІСТИКИ НА ОСНОВІ ІОТ

### 3.1 Що таке ІоТ та його види

Інтернет речей (ІоТ) — це нова парадигма, яка змінила традиційний спосіб життя на високотехнологічний стиль життя. Розумне місто, розумні будинки, контроль забруднення, енергозбереження, розумний транспорт, розумні індустрії – це такі трансформації завдяки ІоТ. Було проведено багато важливих досліджень і розслідувань, щоб покращити технологію через ІоТ. Однак існує ще багато викликів і проблем, які необхідно вирішити, щоб повністю розкрити потенціал ІоТ. Ці виклики та проблеми необхідно розглядати з різних аспектів ІоТ, таких як додатки, виклики, передові технології, соціальні та екологічні наслідки тощо.

Оскільки ІоТ експоненціально зростає з року в рік, компанії використовують величезні бізнес-цінності, які він може запропонувати. Ось деякі з найважливіших переваг ІоТ:

- Для створення нових бізнес-моделей і джерел доходу;
- Покращити бізнес-рішення за допомогою статистичних даних із даних ІоТ ;
- Підвищення продуктивності та ефективності господарської діяльності;
- Щоб покращити клієнтський досвід.

ІоТ реалізується за допомогою апаратних пристроїв – це такі як датчики, гаджети, прилади та інші машини, які збирають дані та обмінюються ними через Інтернет. Вони запрограмовані для певних програм і можуть бути вбудовані в інші пристрої ІоТ. Наприклад, пристрій ІоТ у вашому автомобілі може визначити затори попереду та автоматично надіслати повідомлення людині, з якою ви збираєтеся зустрітися, про вашу неминучу затримку.

Апаратні пристрої мають різні форми та різне призначення але всі вони мають схожість у тому, як вони працюють. По-перше, пристрої ІоТ — це фізичні

об'єкти, які сприймають події, що відбуваються у фізичному світі. Вони містять вбудований центральний процесор, мережевий адаптер і мікропрограму та зазвичай підключаються до сервера протоколу динамічної конфігурації хоста. Для роботи в мережі також потрібна IP-адреса.

Більшість пристроїв IoT налаштовуються та керуються через програмне забезпечення. Наприклад, програма на вашому смартфоні для керування освітленням у вашому домі. Деякі пристрої також мають вбудовані веб-сервери, що усуває потребу у зовнішніх програмах. Наприклад, світло включається відразу, коли ви входите в кімнату.

Основні приклади реалізації пристроїв IoT можуть бути:

- Охорона будинку є ключовою рушійною силою розумних і безпечних будинків є IoT. Різноманітні датчики, освітлення, сигналізація та камери (усіма з яких можна керувати зі смартфона) підключені через IoT для забезпечення безпеки 24x7;

- Трекери активності це розумні домашні камери безпеки які забезпечують сповіщення та спокій. Також це сенсорні пристрої, які можуть контролювати та передавати ключові показники здоров'я в режимі реального часу. Можна відстежувати та контролювати свій артеріальний тиск, апетит, фізичні рухи та рівень кисню;

- Промислова безпека може бути системи виявлення, датчики та камери з підтримкою IoT їх можна розмістити в зонах обмеженого доступу для виявлення порушників. Вони також можуть визначити підвищення тиску та невеликі витoki небезпечних хімічних речовин і усунути їх до того, як вони стануть серйозними проблемами;

- Окуляри доповненої реальності окуляри з підтримкою комп'ютера, які можна носити, допомагають отримувати додаткову інформацію, таку як 3D-анімація та відео, для сцен реального світу користувача. Інформація представлена в лінзах окулярів і може допомогти користувачам отримати доступ до Інтернет-додатків

- Виявлення руху можуть виявляти вібрацію в будівлях, мостах, дамбах

та інших великих спорудах. Ці пристрої можуть ідентифікувати аномалії та порушення в структурах, які можуть призвести до катастрофічних збоїв. Їх також можна використовувати в районах, чутливих до повеней, зсувів.

### 3.2 Застосування IoT в логістиці

Компанії використовують IoT для покращення логістичних процесів — як на складах, так і за їх межами. Сьогодні галузь логістики стикається з багатьма проблемами, які перешкоджають ефективності та збільшують витрати. Відсутність видимості в режимі реального часу рівня запасів, місць відправлення та використання активів є основною проблемною точкою. Ручні паперові процеси ускладнюють відстеження активів і своєчасне отримання даних про відправлення. Збої в ланцюжках поставок легко порушують терміни доставки. По суті, недостатня видимість і неефективні процеси відстеження створюють вузькі місця, які знижують продуктивність логістики.

Саме тут на допомогу приходить IoT. Рішення Інтернету речей (IoT) можуть забезпечити вкрай необхідну видимість у всьому ланцюжку поставок. Датчики на палетах, вантажівках і контейнерах підключають активи до Інтернету та передають дані в реальному часі про місцезнаходження, стан тощо. GPS забезпечує точне відстеження відправлень. Мітки RFID автоматизують управління запасами. Усі ці дані збираються в хмарі та доступні через зручні інформаційні панелі. Озброєні багатими знаннями, менеджери з логістики можуть виявити та усунути неефективність.

Крім того, IoT оптимізує управління територією складу, оптимізує складські операції та забезпечує обслуговування обладнання залежно від стану. Профілактичне обслуговування скорочує час простою. Датчики також контролюють температуру вантажу, вологість тощо, щоб запобігти псуванню. IoT трансформує логістичні операції, покращуючи використання активів,

автоматизуючи процеси, зменшуючи затримки та скорочення, а також дозволяючи приймати рішення на основі даних.

IoT змінює правила гри для модернізації логістичних операцій, подолання існуючої неефективності та забезпечення управління ланцюгом поставок нового покоління. Логістична галузь потребує IoT, щоб задовольнити вимоги клієнтів щодо видимості та оперативності в умовах динамічного ринку.

Ключові цілі використання IoT в логістиці:

- Покращена безпека та виявлення крадіжок. Наприклад підключені програми підвищують контроль над тим, хто входить на склад у будь-який момент часу, допомагають відстежувати всі товари та сповіщають бізнес-менеджера, якщо щось зникне;
- Вища безпека працівників. Вплив IoT на логістику включає захист працівників шляхом виявлення проблем з обладнанням задовго до того, як вони стають помітними;
- Наскрізне відстеження продукту. Підвищення прозорості процесу доставки є основною метою, яку бізнес-менеджери хочуть досягти шляхом інтеграції рішень IoT для логістики;
- Надання бізнес-менеджерам розширеної аналітики. Завдяки широкому спектру програм і перевагам візуалізації великих даних та інструментам аналітики Інтернет речей надає менеджерам бізнесу великі зображення способу обробки всіх операцій;
- Поліпшення доставки. Інтернет речей має багато способів покращити керування доставкою. RFID-мітки та підключені GPS-датчики допомагають керівникам компаній відстежувати доставку аж до її кінцевої стадії.

Розглянемо основні випадки використання IoT в логістиці:

- Системи відстеження запасів. Завдяки підключеним датчикам власники компаній зможуть забезпечити безпечне зберігання продуктів, заощадити час, маючи можливість миттєво знайти потрібний предмет, і мінімізувати людські помилки. Підключена система реалізований на Amazon є доказом того, як технологія може покращити відстеження активів і управління складом;

- Системи прогнозої аналітики. Використовуються для прокладки найкоротших маршрутів доставки, виявлення ранніх ознак несправності обладнання та нагадування персоналу, коли настав час замінити компоненти обладнання;

- Інструменти управління розташуванням. Інструменти управління місцем розташування допомагають логістичним компаніям: підвищення точності та автоматизація відстеження активів, усунення втрат часу, підвищення ефективності керування транспортними засобами завдяки відстеженню автопарку та розумним рішенням для паркування.

### **3.3 Інтеграція IoT для покращення інформаційної технології**

В ролі IoT пристроїв для інтеграції з датчиками можуть бути такі варіанти: Arduino, Raspberry Pi, ESP32 і STM32. Є інші дошки, такі як Teensy, Particle, AdaFruit Feather, BeagleBone та багато інших.

Ці комплекти рідко використовуються для виробництва через їх високу вартість і великий розмір, але вони є фантастичними варіантами для створення раннього функціонального прототипу.

Arduino на сьогоднішній день найпопулярнішою серією плат розробки є Arduino, і доступно багато різних плат Arduino. Зараз Arduino пропонує кілька плат, які набагато краще підходять для багатьох проектів. Це включає серію MKR і серію Nano 33. Arduino Nano 33 IoT поєднує в собі форм-фактор Arduino Nano з простою точкою входу до основних додатків IoT і пікомереж. Плата Nano 33 зображена на рисунку 3.1.



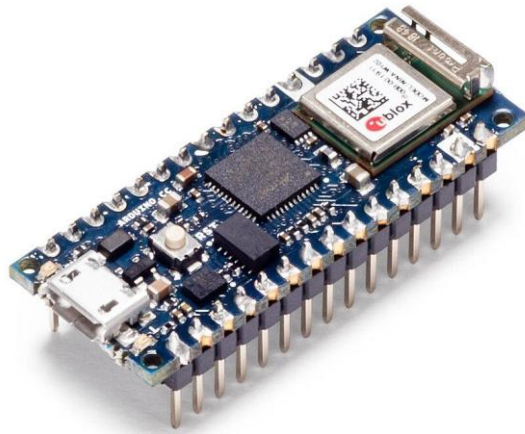


Рисунок 3.1 Плата Arduino Nano 33

Raspberry Pi довгий час означав майже одне – це була високопродуктивна мікропроцесорна плата, яка могла запускати повноцінну операційну систему. Можна підключити монітор і клавіатуру та використовувати їх як дуже маленький комп'ютер. Це чудово для створення проектів для розваги, але не дуже, якщо ви хочете розширити масштаб виробництва. Це здебільшого тому, що Raspberry Pi є надмірним для більшості продуктів. Плата Raspberry Pi зображена на рисунку 3.2.

Якщо дійсно не потрібен високий рівень обчислювальної потужності, використання Raspberry Pi надто ускладнить проект, зробить його дорожчим для масштабування, різко завищить вартість виробництва та розрядить батареї набагато швидше. Для більшості проектів мікроконтролер є набагато кращим рішенням. Якщо дійсно потрібна така продуктивність обробки, тоді можна розглянути обчислювальні модулі Raspberry Pi, які є меншими версіями їхніх саморобних плат, які фактично розроблені для використання в комерційних продуктах. Одна з проблем із використанням Raspberry Pi у проекті, навіть якщо вам потрібна така велика обчислювальна потужність, полягає в тому, що їх може бути складно знайти надійно в наявності, особливо з будь-яким обсягом.

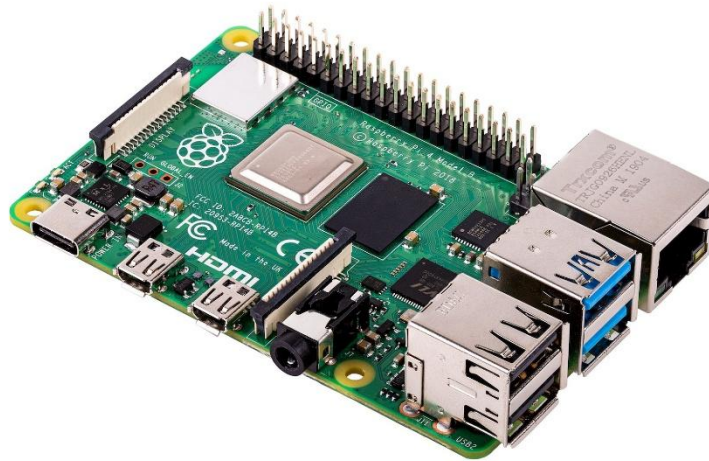


Рисунок 3.2 Плата Raspberry Pi

Незважаючи на те, що Raspberry Pico W і Arduino Nano 33 IoT пропонують можливості WiFi, вони обидва роблять це шляхом підключення модуля WiFi на додаток до поточного чіпа мікроконтролера. Набагато кращим рішенням є Espressif ESP32, який є надшвидким мікроконтролером, який вбудовує як Wi-Fi, так і Bluetooth на одному вбудованому чіпі. Це набагато краще рішення для більшості проектів, оскільки наявність усього в одному чіпі зменшує складність, вартість і розмір. Однією з причин він має надзвичайно простий шлях міграції від раннього прототипування до масового виробництва. Плата ESP32 зображена на рисунку 3.3.

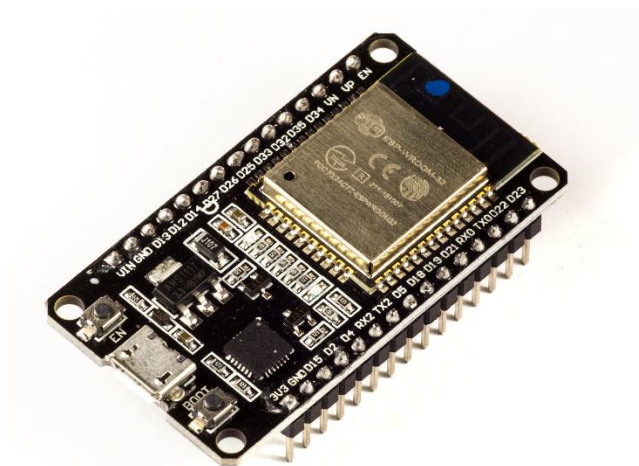


Рисунок 3.3 Плата ESP32

Якщо продукту не потрібен Wi-Fi або Bluetooth, і йому не потрібна обчислювальна потужність повноцінної мікропроцесорної плати, то в більшості випадків обирають серію STM32.

Одна з речей з головних переваг STM32, полягає в тому, що доступні сотні різних версій із майже кожним набором функцій і рівнями продуктивності мікроконтролера. STM32, ймовірно, найпоширеніший мікроконтролер, який використовується в комерційних продуктах, яким не обов'язково потрібен WiFi або Bluetooth. Хоча багато конструкцій STM32 включають бездротове підключення через окремий модуль. Плата STM32 зображена на рисунку 3.4.

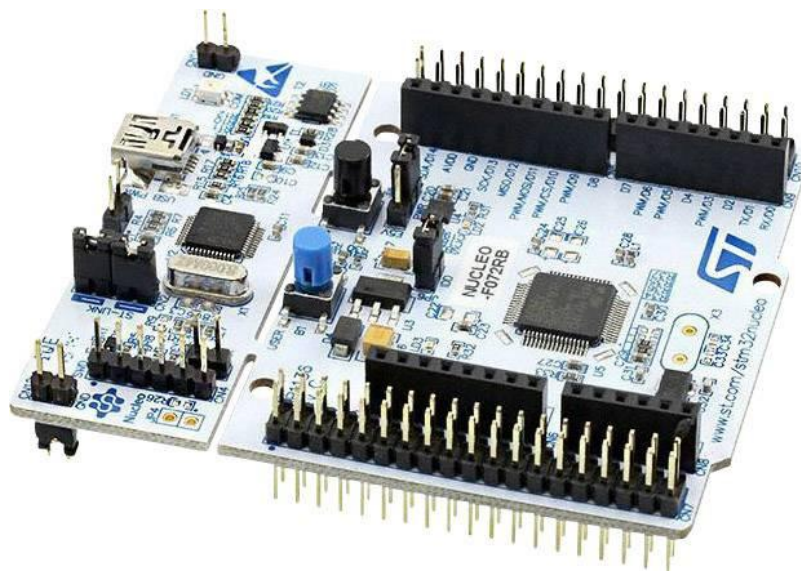


Рисунок 3.4 Плата STM32

Після того як обрали плату яка має доступ до WiFi потрібно обрати протокол обміну даними. Основними протоколами передачі даних є: WebSocket, gRPC, HTTP/HTTPS, MQTT.

WebSocket - протокол на основі TCP, який використовується для зв'язку між браузером користувача та сервером. Цей протокол дозволяє програмістам передавати дані між браузером і сервером двостороннім способом без повторних запитів. WebSocket підтримується в усіх сучасних браузерах і є однією з найпростіших комунікаційних технологій у сфері розробки веб-додатків. Використовуючи WebSocket, можна обмінюватися даними між браузером і

сервером для стабільної передачі даних і інформації. Іншими словами, протокол WebSocket дозволяє програмістам постійно передавати дані між браузером і сервером без необхідності повторного надсилання запитів. Використовуючи цю веб-функцію, ви можете отримати більше переваг від зв'язку між програмами. Цей протокол використовується для багатьох веб-додатків, таких як програми чату, онлайн-ігри, регулярні оновлення тощо. Прикладом використання є жива чат-система, яка використовує WebSocket для негайної передачі повідомлень між користувачами. Принцип роботи протокола WebSocket [15] зображено на рисунку 3.5.

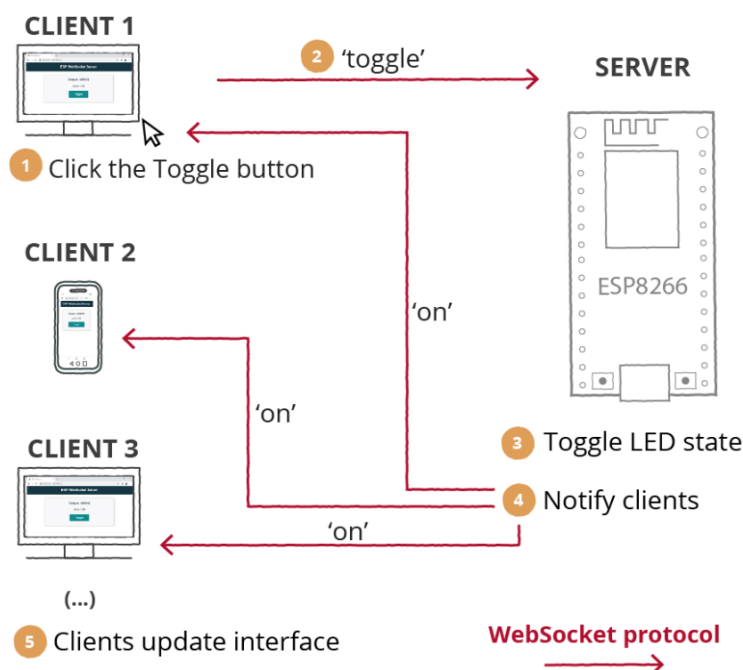


Рисунок 3.5 Схема роботи протокола WebSocket

gRPC - платформа з відкритим кодом, розроблена на основі Remote Procedure Call (RPC) і може бути реалізована в усіх середовищах розробки. Ця технологія забезпечує можливість чіткого, зручного та двонаправленого зв'язку та координації між клієнтом і сервером, а також спрощує побудову підключених систем. Ця технологія розроблена на основі HTTP/2. Однією з цікавих особливостей цієї технології є зв'язок між службами та всіма центрами обробки даних з можливістю відстеження, балансування навантаження перевірки рівня доступу та працездатності згаданої служби. Прикладом використання є

мікросервісна архітектура, де gRPC використовується для забезпечення ефективного міжсервісного зв'язку [14]. Принцип роботи протокола gRPC зображено на рисунку 3.6.

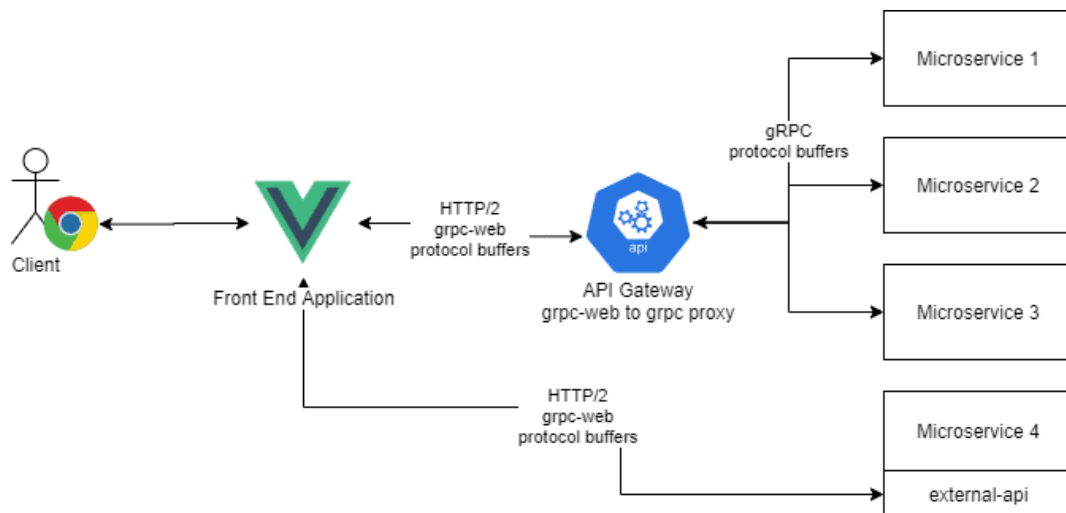


Рисунок 3.6 Схема роботи протокола gRPC

MQTT - це простий протокол обміну повідомленнями, розроблений для обмежених пристроїв, мереж із високою затримкою та пристроїв із низькою пропускною здатністю. Таким чином, це відповідне рішення для обміну даними для зв'язку між машинами (M2M) і відіграє важливу роль в Інтернеті речей (IoT). Комунікація MQTT працює як система публікації та підписки. Пристрої транслюють повідомлення на певну тему. Усі пристрої, які підписалися на цю тему, отримують це повідомлення. Одним із основних застосувань є надсилання повідомлень для керування виходами, читання та публікація даних із датчиків, підключених до Інтернету. Прикладом використання є інтернет речей (IoT), де MQTT використовується для передачі даних між пристроями IoT та серверами. Наприклад, системи контролю за розумними будинками, де пристрої використовують MQTT для взаємодії з центральним сервером [14]. Принцип роботи протокола MQTT зображено на рисунку 3.7.

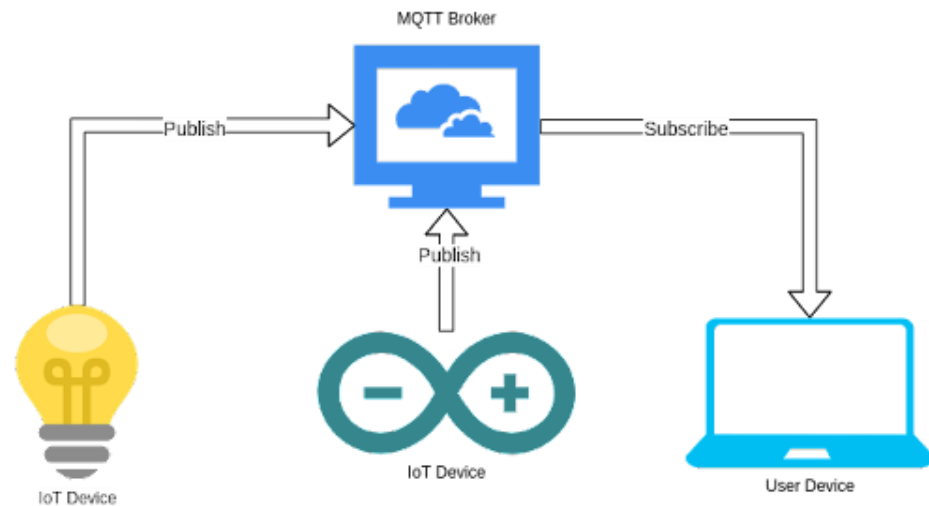


Рисунок 3.7 Схема роботи протокола MQTT

HTTP (Hypertext Transfer Protocol) - це протокол передачі гіпертексту, який використовується для обміну даними між веб-серверами та веб-клієнтами, такими як веб-браузери. HTTP є основою комунікації у Всесвітній павутині, де клієнт (наприклад, веб-браузер) відправляє запити на сервери, а сервери відповідають на ці запити, надаючи різноманітний веб-контент. Протокол HTTP використовується для передачі різних типів веб-ресурсів, таких як HTML-сторінки, зображення, аудіо та відеофайли, стилі CSS, скрипти JavaScript та інші дані, що відображаються на веб-сторінках. HTTP визначає, як клієнт і сервер обмінюються повідомленнями. Зазвичай, клієнт (наприклад, браузер) надсилає запит серверу за певним ресурсом за допомогою URL (Uniform Resource Locator), а сервер надсилає відповідь з необхідними даними. HTTP базується на моделі клієнт-сервер, де клієнт відправляє запит, а сервер надає відповідь, що дозволяє відображати веб-сторінки та інші ресурси. Основна відмінність між HTTP та HTTPS полягає в застосуванні шифрування. Коли ви використовуєте HTTPS, дані шифруються перед відправленням через мережу, що робить їх важкими для зловмисників перехопити та прочитати. Це забезпечує захист особистої інформації, такої як паролі, особисті дані та фінансова інформація, під час передачі через Інтернет. Шифрування в HTTPS забезпечується за допомогою SSL (Secure Socket Layer) або його наступника, TLS (Transport Layer Security). Ці протоколи дозволяють клієнтові та



серверу встановлювати захищене з'єднання, що забезпечує конфіденційність та цілісність переданих даних. У результаті, використання HTTPS допомагає забезпечити безпеку й приватність користувачів в Інтернеті [11]. Принцип роботи протокола HTTP зображено на рисунку 3.8.

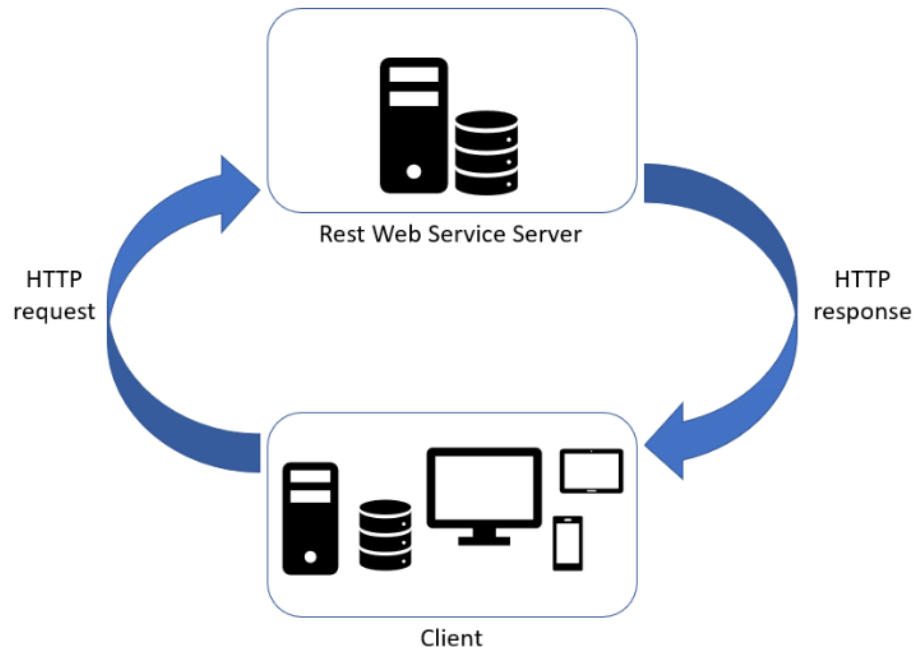


Рисунок 3.8 Схema роботи протокола HTTP

Для зручності написання кода на Arduino для підключення до обраного протокола використовують різні бібліотеки, а саме для:

- WebSocket - ArduinoWebsockets, Arduino-WebSocket, SocketIO;
- gRPC – simpleRPC;
- MQTT - PubSubClient, Arduino MQTT Library, Adafruit MQTT Library;
- HTTPS - ArduinoHttpClient, ArduinoBearSSL, WiFiClientSecure, TinyGSM.

## 4 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЛОГІСТИКИ НА ОСНОВІ ІОТ

### 4.1 Вимоги до інформаційних систем в логістиці та принципи їх побудови

Системний підхід розглядає систему як ціле, яке складається з взаємозалежних частин. Для розуміння системи необхідно вивчити її зв'язок із зовнішнім середовищем, а також її структуру. Розробка логістичної інформаційної системи (ЛІС) також повинна відповідати принципам системного підходу. Перед розробкою ЛІС необхідно вивчити її зв'язок із зовнішнім середовищем, а також її структуру. Зовнішнє середовище ЛІС включає в себе постачальників, клієнтів, конкурентів, а також законодавче та нормативне середовище. Структура ЛІС включає в себе її компоненти, такі як бази даних, програмне забезпечення та обладнання. Вивчення зв'язку ЛІС із зовнішнім середовищем дозволяє розробникам зрозуміти потреби та вимоги користувачів ЛІС. Це допомагає розробникам створити систему, яка буде відповідати потребам користувачів і буде ефективно працювати в реальному світі. Вивчення структури ЛІС дозволяє розробникам зрозуміти, як різні компоненти системи взаємодіють один з одним. Це допомагає розробникам створити систему, яка буде ефективно обробляти інформацію і забезпечувати взаємодію користувачів із системою. Відповідність принципам системного підходу є важливою для розробки ефективної ЛІС.

З погляду системного підходу до логістичних процесів можна виділити три рівні.

Перший рівень — це робоче місце, де здійснюються логістичні операції, тобто переміщення, розвантаження, пакування тощо. Одиниця товару, частина або будь-який інший елемент матеріального потоку.

Другий рівень — це місця розташування, магазини та склади, де



транспортуються або використовуються товари.

Третій рівень — це загальна система транспортування та переміщення, яка включає серію подій, які вважаються моментом доставки постачальником сировини. Ланцюжок закінчується, коли готовий продукт потрапляє до кінцевого споживача.

Робочі місця, де переміщуються одиниці товару, частини матеріального потоку або будь-які інші предмети; зони та склади, де здійснюється транспортування вантажів; система руху та система мобільності в цілому.

Інформаційна система планується зв'язати систему логістики із загальним потоком матеріалів. При цьому здійснюється наскрізне планування за ланцюгом «збут-виробництво-постачання», за допомогою якого може бути створена ефективна система організації виробництва і запущена необхідна система попиту на логістичний ринок. запит. Тому система планування як би "прив'язує" логістичну систему до зовнішнього середовища, тобто до всієї логістичної діяльності. Система операційної системи деталізує планування та забезпечує його виконання на різних конкретних виробничих ділянках, складах і робочих місцях.

Відповідно до концепції логістики інформаційні системи, що відносяться до різних груп, об'єднуються в єдину інформаційну систему. Існує вертикальна і горизонтальна інтеграція.

Вертикальна інтеграція - це процес об'єднання різних систем і процесів в одній організації, щоб забезпечити плавний потік інформації та даних від планування до виконання. Принципова схема вертикального інформаційного потоку, що з'єднує системи планування, утилізації та виконання, наведена в таблиці 3.1.

Горизонтальна інтеграція - це процес об'єднання різних систем і процесів в одній організації, щоб забезпечити плавний потік інформації та даних між ними.

Переваги інтегрованої інформаційної системи включають наступне:

- підвищення швидкості обміну інформацією;
- зменшення кількості помилок у обліку;
- зменшення використання непродуктивної паперової документації;

- об'єднання окремих блоків інформації для більш ефективного використання.

Таблиця 3.1 Принципова схема інформаційного потоку в мікрологістичній системі

Вид звітності	Вид інформаційної системи	Рівень керівництва	Розв'язувані завдання
Річний звіт	Планові	Вище керівництво	Вироблення стратегії і тактики. Доведення цілей
Щотижневий, місячний, квартальний звіт	Диспозитивні	Середній менеджмент	Визначення способу дій доведення правил, інструкції. Завдань
Щоденний звіт	Виконавчі	Безпосередні виконавці	Виконання інструкцій Обробка і групування первинної інформації.

Для того, щоб комп'ютерна логістична інформаційна система (ЛІС) була ефективною, необхідно дотримуватися певних принципів її розробки.

1. Принципи використання апаратно-програмних модулів. Апаратний модуль – це єдиний функціональний вузол електронного пристрою, виготовлений як окремий виріб. Програмний модуль можна розглядати як уніфіковану незалежну частину програмного забезпечення, яка виконує певну функцію в рамках загального програмного забезпечення. Дотримання інструкцій з використання апаратно-програмних модулів дозволить:

- забезпечення сумісності комп'ютерів і програмного забезпечення на всіх рівнях управління;
- підвищити ефективність логістичних інформаційних систем;
- скоротити витрати;
- прискорити їх будівництво.

2. Принцип можливості створення прогресивних систем. Як і інші автоматичні системи управління, комп'ютеризовані логістичні інформаційні

системи також постійно розвиваються. Це означає, що при їх проектуванні необхідно забезпечити можливість безперервного збільшення кількості об'єктів автоматизації, можливість розширення поєднання функцій, які виконує інформаційна система, і кількості вирішуваних завдань. Слід пам'ятати, що етап визначення створення системи, тобто вибір пріоритетів, має великий вплив на подальший розвиток і ефективність логістичної інформаційної системи.

3. Чітке правило встановлення стиків. «На перетині потік документів та інформації перетинає межі повноважень і відповідальності різних бізнес-одиниць або межі незалежних організацій. Забезпечення безперешкодного проїзду через транспортні розв'язки є одним із важливих завдань логістичної галузі».

4. Принцип гнучкості системи враховує специфічні вимоги конкретного застосування.

5. Принципи прийняття системи для діалогу «людина-машина» між користувачами.

Існує різноманітні функції, які має виконувати логістична інформаційна система:

- функція обслуговування, спрямована на задоволення потреб користувачів інформацією;
- функція планування та управління, що спрямована на розробку планів та ефективне управління ресурсами;
- координаційні функції та інші, що сприяють координації різних елементів логістичної системи.

Функції логістичної системи інформаційної системи розглянуті на табл. 3.2.

Таблиця 3.2 - Функції логістичної інформаційної системи

Функції планування	Управління запасами: <ul style="list-style-type: none"> <li>- на кшталт продукції споживача</li> <li>- по розміщенню</li> <li>- прогнозування попиту</li> <li>- стратегічне планування</li> </ul>
Функції координування	Складання графіків виробництва; <u>п</u> ланування матеріальних споживачів; п <u>л</u> анування продажів (маркетинг)
Функція обслуговування (у тому числі інформаційного споживача)	Статус замовлення споживача Наявність запасів: <ul style="list-style-type: none"> <li>- на кшталт продукції</li> <li>- по розміщенню</li> <li>- статус входять навантажень</li> </ul>
Функція управління	Рівні обслуговування; функціонування торговця; функціонування транспортного агентства; функціонування системи

## 4.2 Створення та налагодження веб-платформи

Для розробки великих продуктів необхідно ретельно підібрати технології та інструменти, які будуть відповідати потребам проекту.

У роботі використані технології:

- JavaScript;
- ReactJs;
- NodeJs;
- HTML;
- CSS;
- Ajax;
- C++.

Почнемо з першої технології JavaScript та що це таке. JavaScript — це мова програмування, яку розробники використовують для створення інтерактивних веб-сторінок [3]. Від оновлення каналів соціальних мереж до відображення анімацій та інтерактивних карт, функції JavaScript можуть покращити взаємодію з веб-сайтом. Як мова сценаріїв на стороні клієнта, це одна з основних технологій Всесвітньої павутини. Наприклад, під час перегляду веб-сторінок щоразу, коли ви бачите карусель зображень, спадне меню, що відкривається, або динамічну зміну кольорів елементів на веб-сторінці, ви бачите ефекти JavaScript.

Для чого використовується JavaScript?

Історично веб-сторінки були статичними, подібними до сторінок у книзі. Статична сторінка здебільшого відображала інформацію у фіксованому макеті й не робила того, що ми зараз очікуємо від сучасного веб-сайту. JavaScript виник як технологія на стороні браузера, щоб зробити веб-додатки більш динамічними. Використовуючи JavaScript, браузери могли реагувати на взаємодію користувача та змінювати макет вмісту на веб-сторінці.

Як працює JavaScript?

Усі мови програмування працюють шляхом перекладу англійського синтаксису в машинний код, який потім виконує операційна система. JavaScript загалом класифікується як мова сценаріїв або інтерпретована мова. Код JavaScript інтерпретується, тобто безпосередньо перекладається механізмом JavaScript у базовий код машинної мови. З іншими мовами програмування компілятор компілює весь код у машинний код на окремому етапі. Таким чином, усі мови програмування є мовами програмування, але не всі мови програмування є мовами сценаріїв.

Двигун JavaScript

Механізм JavaScript — це комп'ютерна програма, яка виконує код JavaScript. Перші двигуни JavaScript були простими інтерпретаторами, але всі сучасні двигуни використовують оперативну компіляцію або компіляцію під час виконання для підвищення продуктивності.

JavaScript на стороні клієнта

Клієнтський JavaScript стосується того, як JavaScript працює у вашому браузері. У цьому випадку механізм JavaScript знаходиться всередині коду браузера. Усі основні веб-браузери мають власні вбудовані механізми JavaScript.

Розробники веб-додатків пишуть код JavaScript із різними функціями, пов'язаними з різними подіями, наприклад клацанням миші або наведенням миші. Ці функції вносять зміни до HTML і CSS.

Ось огляд того, як працює JavaScript на стороні клієнта:

- Браузер завантажує веб-сторінку, коли ви її відвідуєте;
- Під час завантаження браузер перетворює сторінку та всі її елементи, такі як кнопки, написи та спадні поля, у структуру даних, яка називається об'єктною моделлю документа (DOM);
- Механізм JavaScript браузера перетворює код JavaScript на байт-код. Цей код є посередником між синтаксисом JavaScript і машиною;
- Різні події, такі як клацання мишею на кнопці, запускають виконання відповідного блоку коду JavaScript. Потім механізм інтерпретує байт-код і вносить зміни в DOM;
- Браузер відобразить новий DOM.

Javascript DOM (Document Object Model) — це інтерфейс програмування, який перетворює веб-сторінку в об'єктну модель, зрозумілу браузеру. Коли браузер завантажує веб-сторінку, він аналізує вміст сторінки та створює цей вміст у пам'яті у формі дерева DOM. Ця структура дерева містить усі елементи сторінки (наприклад, заголовки, абзаци, зображення, посилання), і ці елементи пов'язані один з одним, структура DOM зображена на рисунку 4.1.

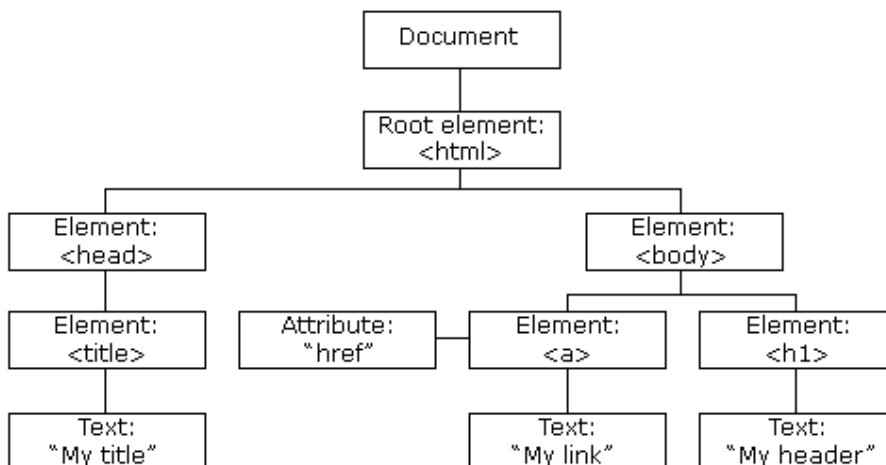


Рисунок 4.1 структура DOM

### Які переваги JavaScript?

- Легко вивчати та використовувати синтаксис JavaScript був створений на основі мови програмування Java, його легко вивчати та кодувати. Розробники використовують JavaScript майже на кожному веб-сайті та в мобільній програмі для створення сценаріїв на стороні клієнта. За останнє десятиліття Node.js також набув значної популярності для серверного кодування. У Node.js закодовано багато основних поточкових і відеоплатформ;

- Отримайте незалежність від платформи на відміну від інших мов програмування, ви можете вставити JavaScript на будь-яку веб-сторінку та використовувати його з багатьма іншими фреймворками та мовами веб-розробки. Написавши його, ви можете запускати код JavaScript на будь-якій машині. Таким чином, JavaScript робить платформу розробки додатків незалежною;

- Зменшити навантаження на сервер завдяки цьому ви можете використовувати JavaScript, щоб зменшити навантаження на сервер і перевантаження мережі, оскільки він може виконувати логічні операції та виконувати велику роботу сервера над самим клієнтом. Для прикладу розглянемо процес заповнення реєстраційної форми. JavaScript швидко перевіряє, чи ввели ви 10-значний номер у полі мобільного телефону. Якщо ці запити надсилатимуться на сервер, ваша сторінка перезавантажуватиметься для кожної помилки, що зробить процес реєстрації дуже повільним і виснажливим;

- Покращте інтерфейс користувача створює елегантні веб-сайти, на яких зручно знаходити та обробляти складну інформацію. Розробники застосовують JavaScript, щоб розширити функціональність і читабельність, а також зробити взаємодію з користувачем сайту ефективнішою;

- Підтримка паралельності - це може виконувати кілька різних наборів інструкцій паралельно. На серверній частині Node.js може обробляти та обробляти високомасштабовані відповіді сервера, не споживаючи тієї ж кількості пропускну здатності.

Що таке React? Це популярний фреймворк JavaScript, розроблений Facebook. Він надає інструменти для створення інтерфейсів користувача з використанням компонентної архітектури. Однією з ключових ідей React є декларативний опис інтерфейсу користувача за допомогою компонентів, що робить код більш читабельним і легким у супроводі. У цій статті ми надамо огляд фреймворку React.

React був вперше представлений у 2013 році. Він був створений, щоб вирішити проблему ефективного оновлення інтерфейсу користувача в режимі реального часу без необхідності повного перезавантаження сторінки. Спочатку розроблений для веб-додатків, React швидко набув поширення також у мобільній розробці завдяки використанню React Native., синтаксис бібліотеки зображена на рисунку 4.2 [24]

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Привіт, {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Петро" />,
  document.getElementById('hello-example'),
);
```

Рисунок 4.2 синтаксис бібліотеки React



React – це декларативна бібліотека JavaScript, яка використовується для створення інтерактивних інтерфейсів. Декларативно означає, що ви просто описуєте, як виглядає ваш інтерфейс, а React сам оновлює його, коли ваші дані змінюються. Це робить код більш передбачуваним і набагато легше налагоджувати.

Переваги React:

- Віртуальний DOM є внутрішнім представленням реального DOM. Це дозволяє фреймворку React оптимізувати процес оновлення інтерфейсу, застосовуючи зміни лише до тих частин, які фактично змінилися, тим самим покращуючи продуктивність. Замість того, щоб безпосередньо оновлювати реальну DOM під час змін, React створює внутрішнє представлення віртуальної DOM. Коли відбуваються зміни, React порівнює зміни у віртуальній DOM із поточним станом і застосовує лише необхідні зміни до реальної DOM. Це зменшує кількість маніпуляцій з DOM, що призводить до покращення продуктивності;

- Компонентний підхід є однією з ключових переваг React, що сприяє багаторазовому використанню коду та створенню модульних програм. Розбиття інтерфейсу користувача на невеликі компоненти відіграє важливу роль у структуруванні та покращенні читабельності коду. Такий підхід спрощує процес обслуговування коду та додавання нових функцій, оскільки кожен компонент має чітко визначене завдання;

- Підтримка Facebook яка розроблено та підтримується командою інженерів у Facebook, забезпечуючи переваги стабільності та надійності. Команда активно інвестує в розвиток React, забезпечуючи постійне вдосконалення фреймворку та виправлення помилок.

Що таке JSX в React ?

Розширення синтаксису JSX (JavaScript XML) React дозволяє програмістам створювати код JavaScript, схожий на HTML. Пропонуючи більш декларативні & інтуїтивно зрозумілий підхід, це спрощує процес створення інтерфейсу користувача. Розробники можуть визначати повторно використовувані компоненти React за допомогою JSX, поєднуючи JavaScript і синтаксис, як-от

HTML. Перш ніж код JSX буде представлено в DOM, він перетворюється на стандартний JavaScript за допомогою таких програм, як Babel. Це дає можливість створювати динамічний матеріал і ефективно його оновлювати.

Наступною технологією є Node JS — це середовище JavaScript з відкритим кодом для розробки на стороні сервера. Середовище було вперше запущено в 2009 році Райаном Далем і з тих пір стало основним продуктом у світі розробки. Node JS є надзвичайно швидким за своєю природою, оскільки він використовує двигун V8 JavaScript і використовує ядро браузера Google Chrome поза ним.

Node.js був розроблений у 2009 році шанованим програмістом Раяном Далем. Це середовище виконання JavaScript з відкритим вихідним кодом, створене на основі libuv і механізму Google V8 JavaScript.

Можливо, єдина найкорисніша річ, яку приносить Node.js, — це його унікальний набір API. Веб-браузери також мають різноманітні DOM/Web API, які допомагають вам взаємодіяти з інтерфейсом користувача, за винятком того, що вони не є революційними для можливостей реального часу. Синтаксис NodeJs зображено на рисунку 4.3.

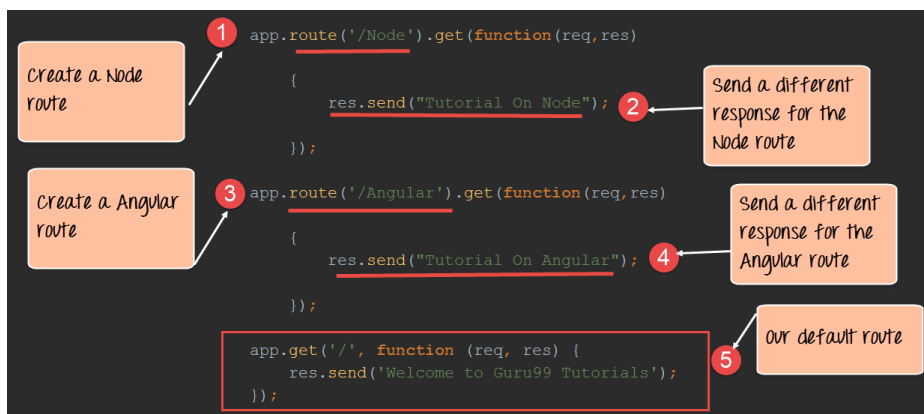


Рисунок 4.3 синтаксис мови NodeJs

Переваги NodeJs:

- Крос-функціональність дозволяє покращити продукт та продуктивність команди, адже JavaScript є найпопулярнішою мовою розробки інтерфейсу, майже всі розробники знайомі з синтаксисом. Подібним чином Node.js працює на тій же мові, що робить перехід на нього на сервері безпроблемним;

- Повний стек JavaScript дає можливість можна найняти JavaScript-програмістів, які можуть писати як бекенд, так і інтерфейс веб-додатків і зробити розгортання програми Node простішим та ефективнішим;

- Масштабоване середовище це однопотоковий цикл подій Node.js. Це робить систему швидкою, а також легкою, дозволяючи їй обробляти кілька одночасних з'єднань. Масштабованість сприяє зростанню програм разом із бізнесом. Оскільки комп'ютерні процеси продовжують ускладнюватися, додатки, розроблені за допомогою Node.js, можна масштабувати горизонтально та вертикально, а кластерні модулі можуть допомогти розробникам розподілити навантаження на декілька комп'ютерів;

- Одночасна обробка запитів дає можливість забезпечувати неблокуючу парадигму введення-виведення, яка зменшує час відповіді між клієнтом і веб-сервером, що робить його оптимальним для веб-додатків у реальному часі, які стосуються інтенсивної обробки на стороні клієнта;

- Програми реального часу може швидко синхронізувати дані сервера та клієнта, а також обробляти численні запити клієнтів. Він також дозволяє ділитися та повторно використовувати пакети коду бібліотеки, що робить його найкращим вибором для RTA, таких як програми для чату чи обміну миттєвими повідомленнями, а також програми, де користувачі можуть спостерігати за змінами редагування в режимі реального часу.

Наступною технологією є HTML розшифровується як Hypertext Markup Language, і це широко поширена мова програмування, яка використовується для розробки веб-сторінок. HTML - це мова розмітки, яка використовується для створення веб-сторінок. Він визначає, як виглядає веб-сторінка та як відобразити вміст за допомогою елементів. Він формує або визначає структуру нашої веб-сторінки, таким чином, він формує або визначає структуру нашої веб-сторінки. Синтаксис мови гіпертекстової розмітки HTML зображено на рисунку 4.4.

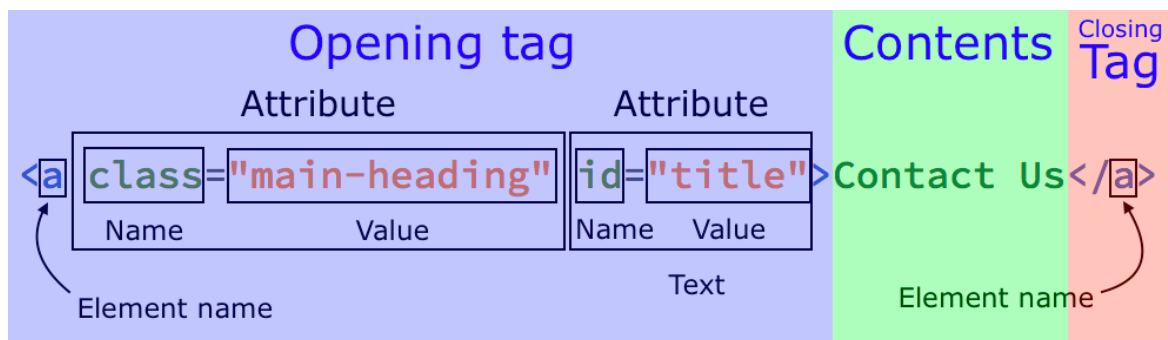


Рисунок 4.4 синтаксис мови гіпертекстової розмітки HTML

Також було використано Cascading Stylesheets (CSS, каскадні таблиці стилів). Коротше кажучи, CSS — це мова дизайну, яка робить веб-сайт привабливішим, ніж просто звичайні чи ненадихаючі фрагменти тексту. Тоді як HTML значною мірою визначає текстовий вміст, CSS визначає візуальну структуру, макет і естетику. HTML — це мова розмітки, а CSS — мова таблиць стилів. [3]

CSS, подібно до HTML, не належить до традиційних мов програмування. Вона функціонує як мова таблиць стилів, що дозволяє визначати вигляд обраних елементів у HTML документах. Проще кажучи, CSS є технологією, яка описує зовнішній вигляд веб-сторінок, що написані на HTML. Основне призначення CSS - розділити код сторінки від її візуального оформлення. Синтаксис мови CSS зображено на рисунку 4.5.

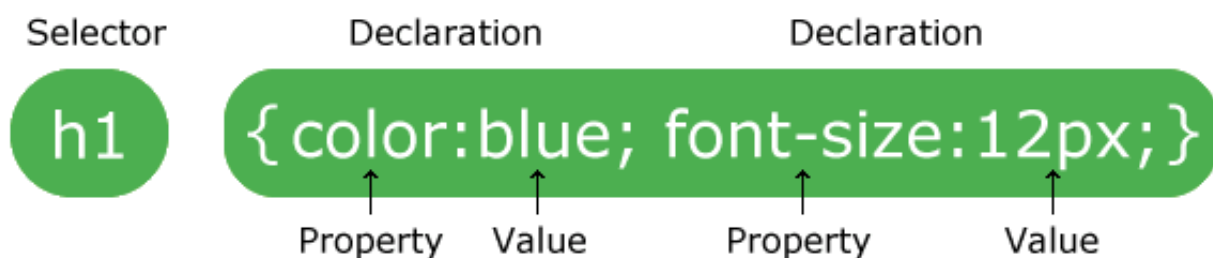


Рисунок 4.5 синтаксис CSS

Наступна технологія це Ajax - Термін AJAX розшифровується як асинхронний JavaScript і XML. Розуміння AJAX є важливим для веб-розробників, оскільки він зазвичай використовується у веб-додатках.

AJAX походить від функції JavaScript, щоб забезпечити більшу інтерактивність. JavaScript створює, додає та керує динамічною структурою, відстежуючи, який вміст потребує оновлення в реальному часі, коли відвідувач отримує доступ до веб-сайту. Тим часом Extensible Markup Language (XML) транспортує дані між веб-сервером і браузером. Окрім XML, для обміну даними часто використовуються інші формати, наприклад звичайний текст і JSON.

JavaScript і Extensible Markup Language працюють разом в AJAX, щоб забезпечити асинхронне оновлення вмісту веб-сторінки. Іншими словами, за допомогою AJAX вміст сайту можна оновлювати без перезавантаження всієї сторінки.

AJAX містить такі технології:

- XHTML і CSS – для представлення інформації;
- Об'єктна модель документа (DOM) – для динамічного відображення даних і їх взаємодії;
- XML, HTML та XSLT – для обміну даними та обробки. Однак багато розробників замінили XML на JSON, оскільки він походить від JavaScript;
- Об'єкт XMLHttpRequest – дозволяє асинхронний зв'язок із веб-сервером;
- JavaScript – мова програмування, яка поєднує всі ці веб-технології.

Загальні принципи AJAX прості. Порівняти звичайну модель веб програм та з веб програми з AJAX на рисунку 4.6.

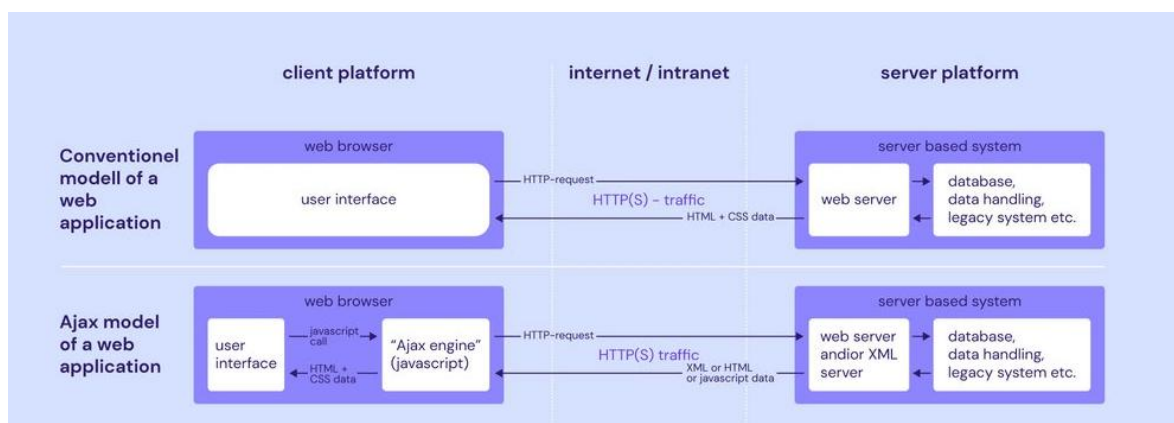


Рисунок 4.6 схема роботи AJAX та порівня роботи без AJAX

Прикладом AJAX є функція Google Autocomplete. Ця функція пропонує пропозиції щодо ключових слів, допомагаючи користувачам завершити свій пошуковий запит під час введення в рядок пошуку. Зміни відбуваються в реальному часі, але веб-сторінка залишається незмінною. AJAX дозволяє рівню обміну даними та презентації працювати одночасно, не заважаючи функціям один одного. Це не було на початку 90-х років, коли ця нова технологія ще не була представлена. Вам довелося перезавантажити Google, щоб нова рекомендація з'явилася на вашому екрані. Також прикладом використання може бути: системи голосування та рейтингу, чати, соціальні медіа.

#### Переваги AJAX:

- Дозволяє браузеру оновлювати фрагменти вмісту без перезавантаження чи відкриття нової сторінки;
- Зменшує трафік сервера, дозволяючи веб-додаткам відтворюватися без даних;
- Використовує меншу пропускну здатність, оскільки отримує невеликі фрагменти вмісту;
- Дозволяє браузерам здійснювати асинхронні виклики до веб-сервера без перезавантаження всієї сторінки;
- Об'єкт XMLHttpRequest в AJAX встановлює незалежне з'єднання між сервером веб-сайту та машиною на стороні клієнта;
- Створює чуйну взаємодію. Рухи миші, які не є клацаннями користувача, також можуть викликати події.

#### Мінуси AJAX:

- Якщо ваш браузер не підтримує або не вмикає JavaScript, він не завантажуватиме веб-сторінки за допомогою AJAX належним чином;
- Будь-хто може переглянути вихідний код веб-додатку, розробленого за допомогою AJAX, що робить його менш безпечним;

- Коли користувач натискає кнопку «Назад» у своєму браузері, він може не повернутися до попереднього стану сторінки. Послідовні запити AJAX не зберігаються в історії браузера.

Наступна технологія це API - слугує програмним посередником, що дозволяє двом програмам спілкуватися одна з одною. Однак вони також можуть зробити набагато більше [12].

Інтерфейси програмного інтерфейсу (API) дозволяють двом програмним програмам спілкуватися за допомогою попередньо встановлених правил і протоколів. API слугує програмним інтерфейсом для двох різних програм, яким необхідно спілкуватися. API може передавати дані між двома програмами, отримувати доступ до функцій чи служб інших програм або створювати програму.

Для створення API потрібен набір інструкцій на основі програмування, які дозволяють програмним програмам спілкуватися. Клієнт API ініціює запит API, який надходить на сервер. API отримує запитані дані із зовнішнього сервера або програми, які повертаються клієнту. Окрім отримання даних, API також можуть запускати функції, передавати інформацію назад на сервер для керування та зберігання або повертати інформацію в режимі реального часу, таку як ціни чи наявність. Для успішного виконання API необхідні наступні компоненти:

- Клієнт API це може бути користувачем API, який ініціює запит API, або запит автоматично активується зовнішньою подією чи сповіщенням від служби чи програми;
- Ключ API це унікальний пароль, що містить літери та цифри, який надає доступ до API;
- API запити це повідомлення, надіслане до програми, яке запитує сервер про інформацію або послугу;
- Кінцева точка це спеціальний уніфікований покажчик ресурсу (URL), який вказує на розташування ресурсу на сервері. Кінцева точка API дозволяє різним системам і програмам спілкуватися, надсилаючи та отримуючи інформацію з інструкціями;

- Спосіб запиту це конкретні операції, які клієнт хоче виконати над ресурсом URL. REST API використовує метод HTTP, який може виконувати такі дії:

- GET – отримує дані з сервера;
- POST - додає нові дані;
- PUT - замінює весь ресурс новою інформацією;
- PATCH - використовується для часткового оновлення наявного URL-ресурсу додатковою інформацією;
- DELETE - використовується для видалення даних із бази даних;

- Параметри – це змінні, що передаються кінцевій точці API, щоб надати чіткі інструкції для обробки сервером API. Параметри можна включити як частину запиту API в рядок запиту URL-адреси або в поле тіла запиту;

- Заголовки запиту надають важливу інформацію для обробки запиту сервером, а інформація заголовка міститься в тілі повідомлення;

- Сервер API — це програмне забезпечення, яке знаходиться безпосередньо на сервері. Сервер API знаходиться між клієнтом і джерелом даних, тому веб-API знаходяться між додатком користувача та веб-сервером. Схема роботи арі зображена на рисунку 4.7.

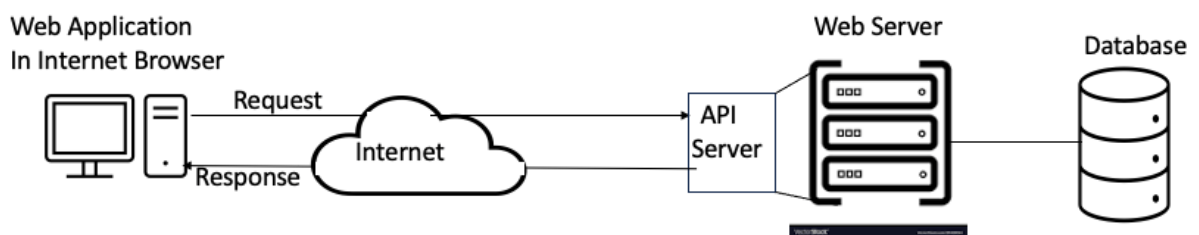


Рисунок 4.7 Схема роботи арі

Мова програмування C++ яка використовується для Arduino з метою оптимізації — це потужна та універсальна мова програмування, яку можна використовувати для різноманітних програм, таких як розробка операційних систем, відеоігор та наукового моделювання. Це розширення мови програмування



C із додатковими функціями, такими як об'єктно-орієнтоване програмування та шаблони. [25]

Синтаксис C++ подібний до синтаксису C, з операторами та виразами, написаними в певному порядку для досягнення потрібної функції. Програми C++ зазвичай починаються з включення бібліотек, які надають попередньо визначені функції, які можна використовувати в програмі.

Основна функція - це місце, де починається виконання програми, і вона укладена у фігурні дужки. Змінні можуть бути оголошені та ініціалізовані за допомогою різних типів даних, таких як цілі числа, числа з плаваючою точкою та символи. Оператори керування, такі як if-else, switch, і цикли, такі як for, while і do-while, використовуються для керування потоком програми та виконання повторюваних завдань.

Об'єктно-орієнтоване програмування в синтаксисі C++ передбачає створення класів, які інкапсулюють дані та функції, а також об'єктів, які є екземплярами цих класів. Спадкування, поліморфізм і інкапсуляція є одними з ключових функцій об'єктно-орієнтованого програмування, які підтримуються в C++. Шаблони — це ще одна потужна функція синтаксису C++, яка дозволяє загальне програмування.

Для емуляції датчиків будемо використовувати сервіс wokwi. Будемо використовувати такі пристрої та датчики:

- ESP32 (мікроконтролер)
- GPS
- OLED display ssd1306 (пристрій виведення)
- Матрична клавіатура 4x4 (пристрій введення)
- Датчик вагів hx711

При розробці IoT частини було використані такі бібліотеки:

Основні бібліотеки:

- ArduinoWebsockets.h – взаємодія з протоколом WebSocket
- ArduinoJson.h – формування даних у тип json
- WiFi.h – підключення плати до Wifi

Бібліотеки для датчиків:

- HX711.h – зчитування значення з вагів
- NMEA.h – бібліотека та комбінований стандарт для роботи з GPS
- Keypad.h – для роботи з мембранною клавіатурою
- Adafruit\_SSD1306.h, Wire.h, Adafruit\_SSD1306.h – для взаємодії з OLED екраном

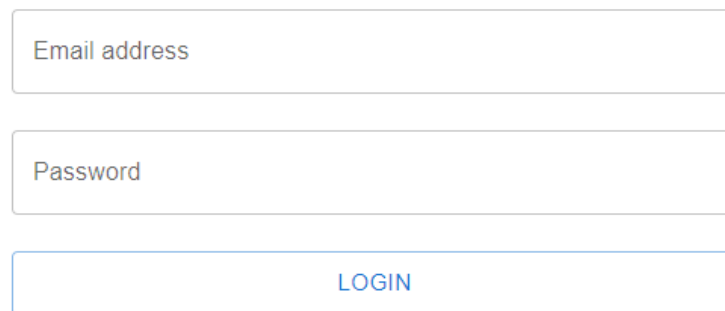
Для взаємодії системи та IoT використовуємо протокол передачі даних WebSocket. WebSocket реалізований на мові NodeJs за допомогою бібліотеки express. WebSocket використовуємо тільки як можливість передачі даними між системами, основна логіка опрацювань відбувається на стороні SPA застосунку.

### 4.3 Опис веб-платформи та її алгоритмів

Після запуску програми користувачеві необхідно авторизуватися, щоб отримати доступ до її функцій, авторизація зображено на рисунку 4.8

## Sign in to Logistick

Enter your details below.



The image shows a login form for the Logistick system. It consists of three main components: a title 'Sign in to Logistick', a subtitle 'Enter your details below.', and three input fields. The first field is for 'Email address', the second for 'Password', and the third is a 'LOGIN' button. The fields are arranged vertically and are separated by small gaps. The 'LOGIN' button is a wide, thin rectangle with a blue border and the text 'LOGIN' in blue capital letters.

Рисунок 4.8 форма авторизації в систему

Система забезпечує зворотний зв'язок користувачеві, якщо він вводить неправильні облікові дані, зображено на рисунку 4.9

# Sign in to Logistick

Enter your details below.

Email address

dora

Incorect login

Password

.....

Incorect password

LOGIN

Рисунок 4.9 повідомлення про неправильний пароль чи логін

Після успішного входу в систему користувач побачить головну сторінку, яка містить кількісну інформацію в системі, яка зображена на рисунку 4.10:

- кількість всього водії;
- кількість всього транспорту;
- кількість транспортні перевезення;
- інформаційний текст.

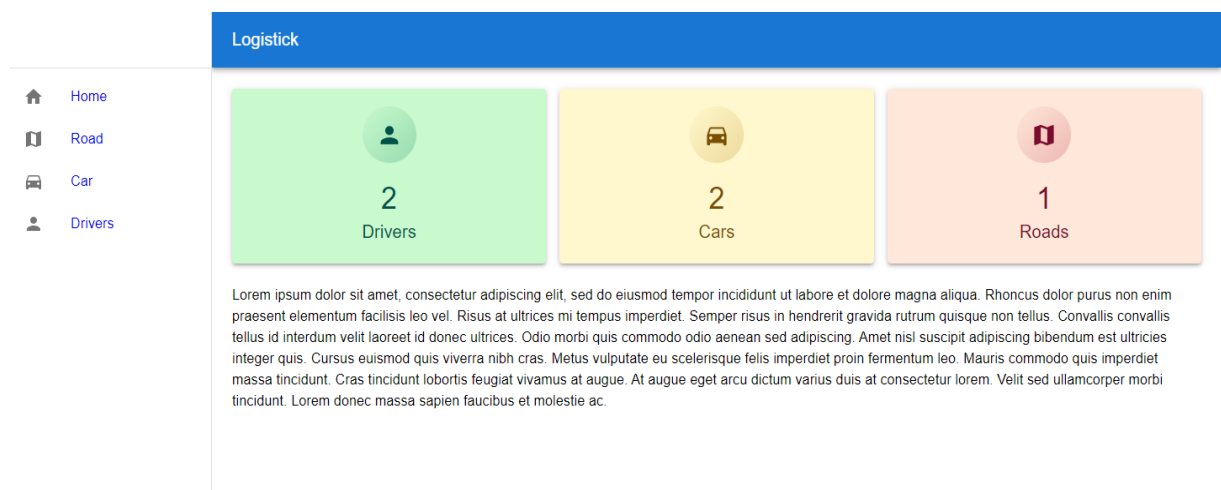


Рисунок 4.10 головна сторінка системи

В меню система знаходиться 4 посилання на сторінки, на яких є певний функціонал, а саме:

- головна сторінка;
- транспортні перевезення;
- машини;
- водії;
- склад.

Першим етапом для організації перевезення є внесення в систему інформації про водія та автомобіль. При переході на посилання “Drivers” спостерігаємо список з усіма водіями, пошук водіїв та кнопка для створення нової карточки водія, зображено на рисунку 4.11.

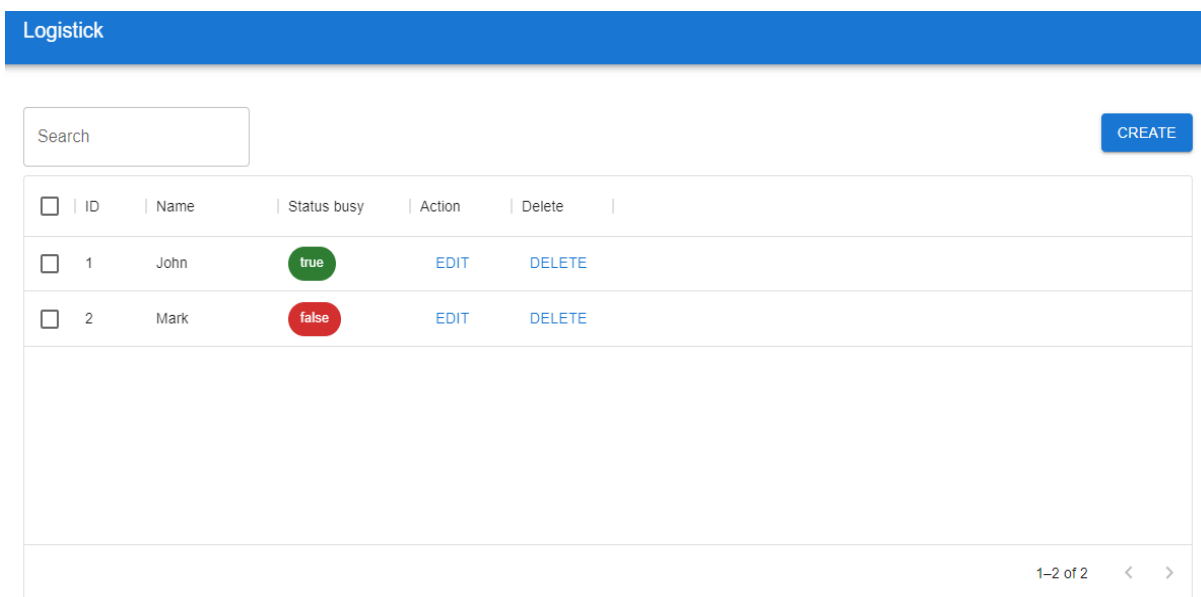
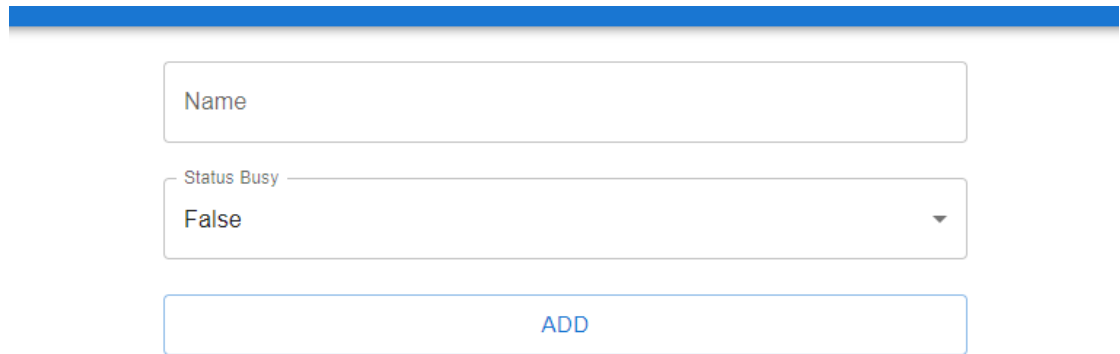


Рисунок 4.11 сторінка список водіїв

Для створення карточки водія потрібно натиснути на кнопку “Create”, після натискання система відобразить форму для створення, зображено на рисунку 4.12.

Для заповнення форми необхідно вказати наступну інформацію:

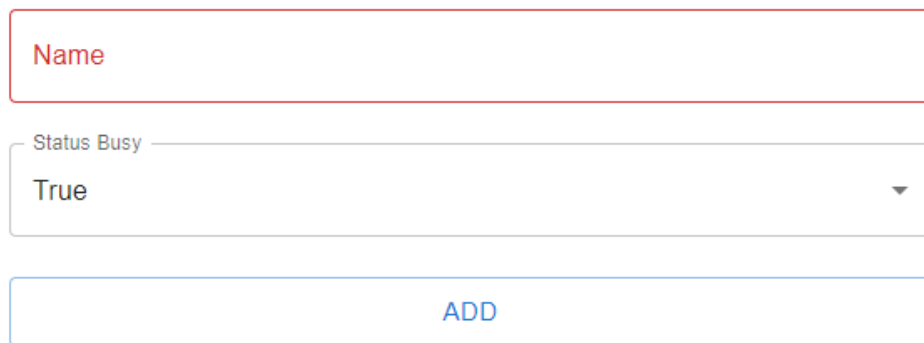
- ім'я;
- статус зайнятості.



The form consists of three main components: a text input field labeled 'Name', a dropdown menu labeled 'Status Busy' with 'False' selected, and a blue 'ADD' button.

Рисунок 4.12 форма створення водія

Якщо поля в формі будуть порожніми, система виділить поля червоним кольором, зображено на рисунку 4.13



The form is identical to Figure 4.12, but the 'Name' input field is highlighted with a red border, indicating a validation error. The 'Status Busy' dropdown is now set to 'True'.

Рисунок 4.13 виділення порожнього поля при створенні

Після успішної валідації форми запис буде створений та доданий до файла на сервері, зображено на рисунку 4.14

<input type="checkbox"/>	ID	Name	Status busy	Action	Delete
<input type="checkbox"/>	1	John	true	EDIT	DELETE
<input type="checkbox"/>	2	Mark	false	EDIT	DELETE
<input type="checkbox"/>	3	Kevin	true	EDIT	DELETE

Рисунок 3.14 створений запис

Даний запис можна відредагувати чи видалити. При редагуванні буде відкрита форма для створення запису, зображено на рисунку 4.13, але вже з заповненими даними того запису, що обрали для редагування, зображено на рисунку 4.16

Name	Kevin
Status Busy	True

Рисунок 4.16 редагування запису

На сторінці “Car” є список всіх машин. Функціонал та дизайн такий як на сторінці “Drivers”, зображено на рисунку 4.11, але з іншими полями.

Для створення потрібно заповнити такі поля, зображено на рисунку 4.17:

- назву машини;
- номер машини;
- вантажопідйомність;
- витрата палива;

- статус зайнятості.

---

The image shows a web form for creating a vehicle record. It consists of five input fields stacked vertically, each with a label and a value. The labels are: Name, Number car, Load capacity, Fuel, and Status Busy. The values are: MAN, A234H443, 1000, 500, and False. Below the fields is a blue button labeled ADD.

Name	MAN
Number car	A234H443
Load capacity	1000
Fuel	500
Status Busy	False

ADD

Рисунок 4.17 форма створення

Також у формі при створенні машини присутня валідація. Приклад валідації зображено на рисунку 4.13

Після створення записів про машину та водіїв користувач, може здійснити запис про вантажоперевезення. На сторінці "Road" розміщена таблиця, що містить інформацію про перевезення, зображено на рисунку. 4.18



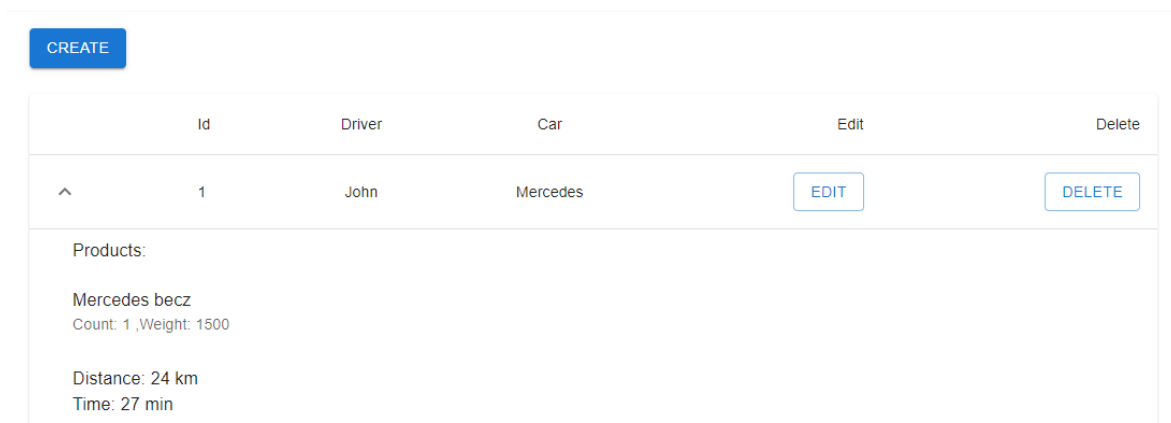


Рисунок 4.18 сторінка списку вантажоперевезень

Для створення перевезення потрібно натиснути на кнопку “Create” на сторінці “Road”, після чого користувачу буде виведений перший етап створення перевезення, зображено на рисунку. 4.19

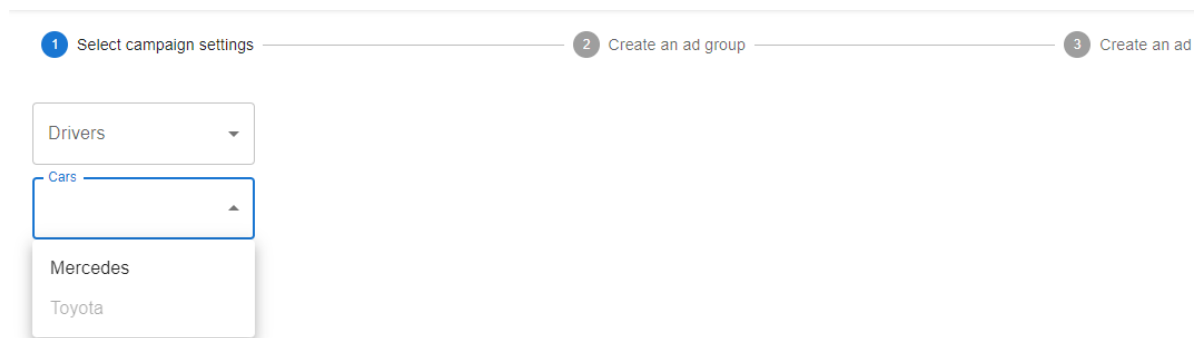


Рисунок 4.19 перший етап створення перевезення

Якщо машина чи водій будуть зі статусом “зайнятий”, то в списку він буде не доступний. Якщо поля будуть порожніми, користувачу система повідомить про це, приклад валідації поля зображено на рисунку 4.13. Після вибору параметрів на першому етапі, користувач переходить до другого кроку, де йому потрібно вказати список продуктів, які будуть транспортовані. Для додання товару до списку потрібно заповнити 3 поля, зображено на рисунку 4.20:

- назву;
- кількість;
- вага.

Name  Count  Weight | kg

Id	Name	Count	Weight	Delete
1	Box	2	10	<a href="#">DELETE</a>

1-1 of 1 < >

Рисунок 4.20 другий етап створення перевезення

Для оптимізації системи будемо використовувати IoT пристрої. Отже для даного етапа зі сторони IoT можна оптимізувати вимір ваги та автоматичне підставлення значення у відповідний елемент введення інформації. На емуляторі використовуємо прилад HX711 який вимірює вагу. Після чого мікроконтроллер ESP32 відправляє дані до системи. Оптимізація вирахування вагів за допомогою IoT зображено на рисунку 4.21 та рисунку 4.22.

Select campaign settings     
  Create an ad group     
  Create an ad

Name:      
 Count:      
 Weight | kg:      

Id	Name	Count	Weight	Delete
No rows				

Sources    **Network**    Performance    Memory    Application    Lighthouse    AdBlock

Chrome log     Disable cache    No throttling     Invert     Hide data URLs     Hide extension URLs   
                                            

Blocked requests     3rd-party requests     Group by frame     Screenshots

200 ms    300 ms    400 ms    500 ms    600 ms    700 ms    800 ms

X Headers    **Messages**    Initiator    Timing

All   

Data	Length	Time
["type":"weight","value":619,"arduinoId":1]	43	15:35:54.506
["type":"weight","value":380,"arduinoId":1]	43	15:35:54.613
["type":"weight","value":190,"arduinoId":1]	43	15:35:54.818
["type":"weight","value":142,"arduinoId":1]	43	15:35:57.461
["type":"weight","value":47,"arduinoId":1]	42	15:35:57.763
["type":"weight","value":0,"arduinoId":1]	41	15:35:57.837
["type":"weight","value":47,"arduinoId":1]	42	15:35:58.800
["type":"weight","value":95,"arduinoId":1]	42	15:35:59.106

Рисунок 4.21 системає отримує дані від IoT та підставляє у відповідне поле

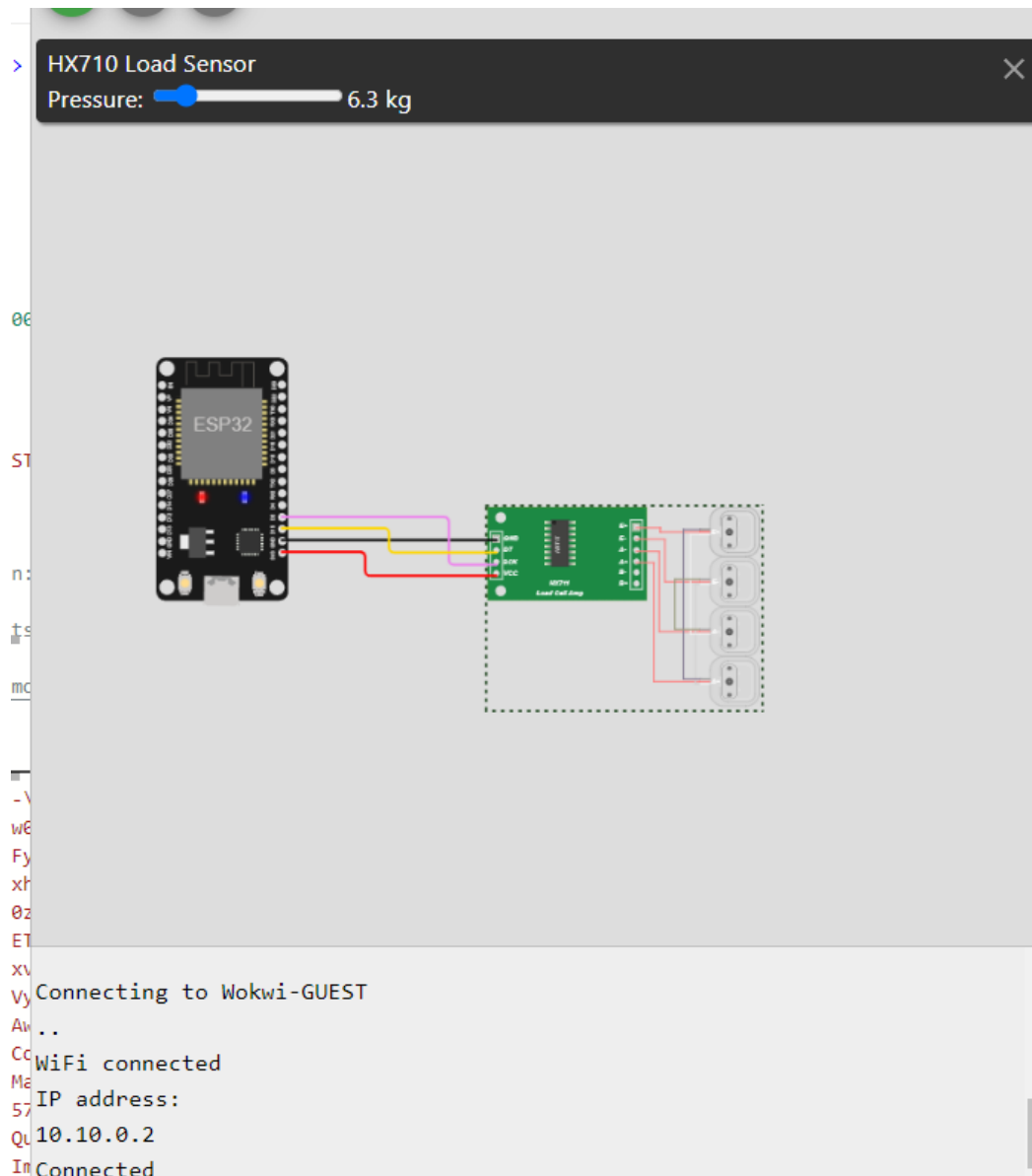


Рисунок 4.22 датчки HX711 вираховує дані, та ESP32 відправляє дані до системи

Якщо вага продуктів перевищить максимальну вантажопідйомність, система надішле повідомлення користувачу, зображено на рисунку 4.23.

Name  Count  Weight | kg

Id	Name	Count	Weight	Delete
1	Box	2	10	<a href="#">DELETE</a>
2	Table	1	1000	<a href="#">DELETE</a>
1-2 of 2 < >				

Products weight 1020 is more than load capacity car 500

Рисунок 4.23 сповіщення про зайву вагу товару

Якщо є надлишок ваги товарів, користувачу потрібно видалити товари, натиснувши на кнопку “Delete” на той товар, який завеликий чи змінити машину, яка буде відповідати іншими характеристиками. Після успішного завершення другого етапу валідації форми, користувач переходить на останній етап процесу. На 3-му етапі користувач обирає з якої точки вирушить машина до кінцевої точки, зображено на рисунку 4.24.

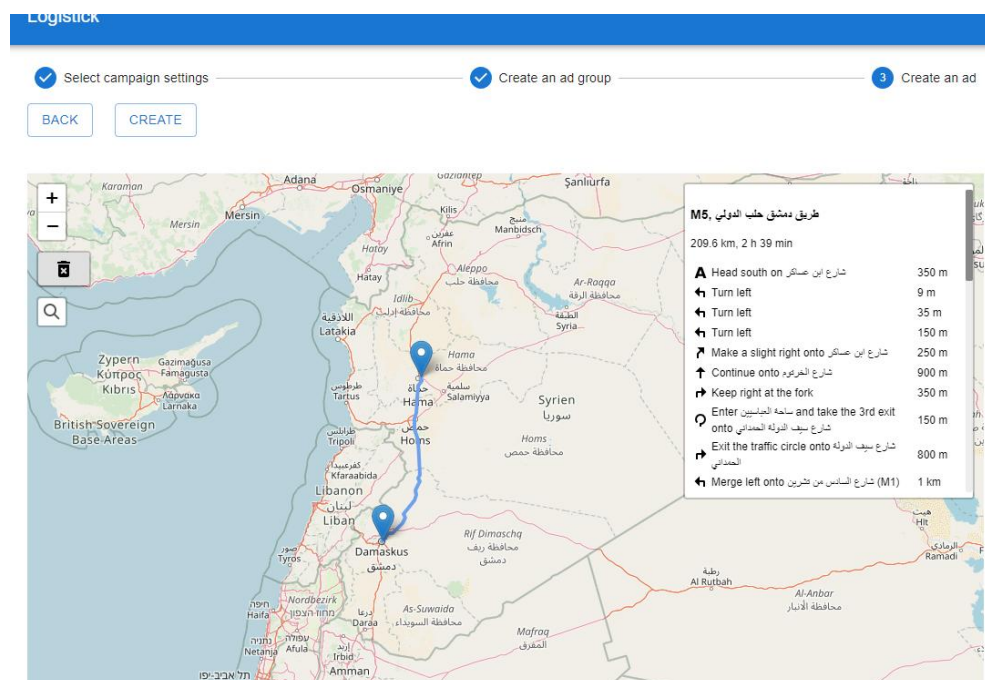


Рисунок 4.24 вибір маршруту

Після заповнень всіх даних запис може бути створений та буде відображений в загальному списку вантажоперевезень, зображено на рисунку 3.18.

Ще одним датчиком для оптимізації системи став GPS трекер, який дозволяє відстежувати місцеперебування вантажівки. Датчик передає дані координат для системи після чого система відображає перебування транспорту на карті. Отримання даних GPS та зміна місцезнаходження вантажівки відображено на рисинку 4.25 та 4.26.



Рисунок 4.25 відображення вантажівка за даними від GPS

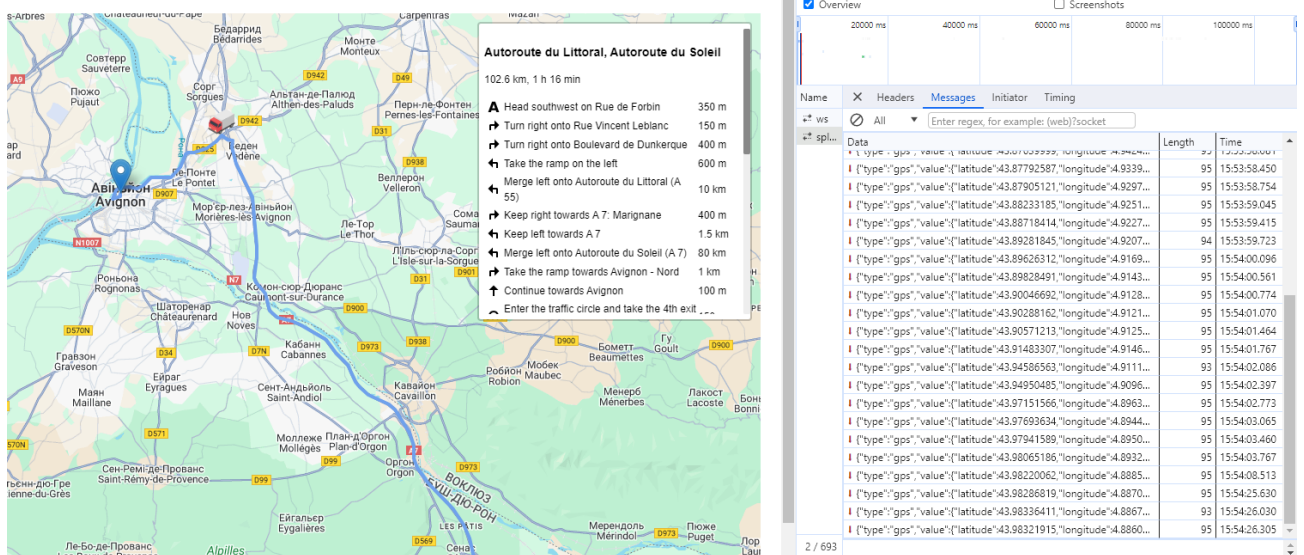


Рисунок 4.26 відображення вантажівки в іншому місці від попереднього результату яке зображено на рисунку 4.25 після отримання додаткових даних від GPS

В системі також є функціонал для відстежування місця зберігання товарів на складі. Сторінка відображення товарів на складі зображена на рисунку 4.27.



Рисунок 4.27 сторінка відображення товарів на складі

Для оптимізації отримання товарів на склад за допомогою IoT було оптимізовану систему використовуючи пристрій клавіатури та олед дисплею. Користувач буде вводити ідентифікатор товару через клавіатури після чого натискаючи на підтвердження введеного коду на клавіатурі. Зображено на рисунку 4.28. До системи буде відправлено ідентифікатор товару. В системі буде визначено місце для товару супроводжуючи повідомленням в системі про

визначення місце для товару. В системі буде підсвічуватися місце для товару після чого місце стане червоним. Після чого користувач буде відразу знати куди потрібно відвезти товар. Зоображено на рисунку 4.29 та 4.30.

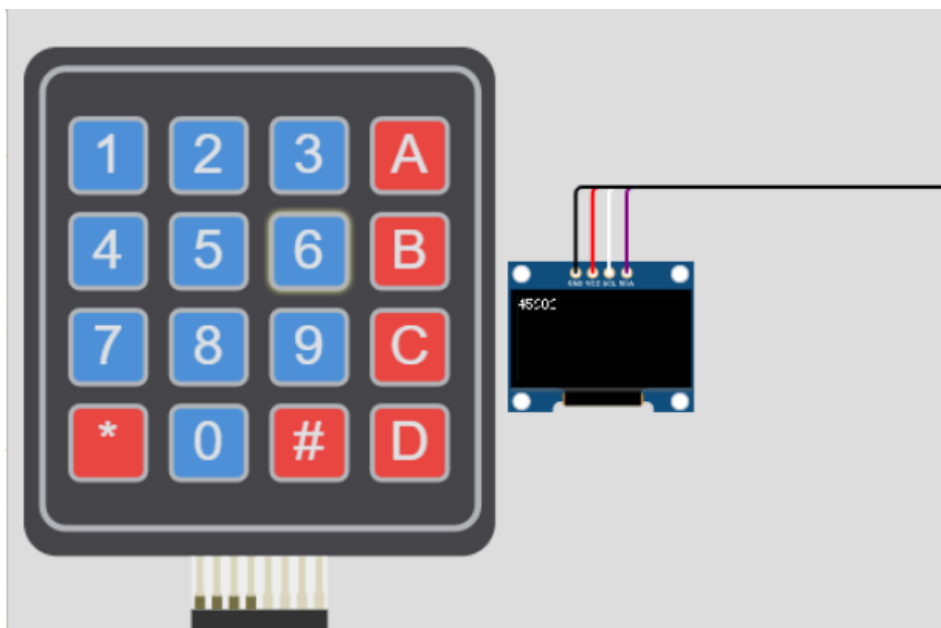


Рисунок 4.28 за допомогою клавіатури вводимо ідентифікатор товару



Рисунок 4.29 від датчика отримали повідомлення про визначення місця для нового товару



Number place: 1 Free	Number place: 2 Free	Number place: 3 Free	Number place: 4 Product ID: 64189765	Number place: 5 Product ID: 236
Number place: 6 Free	Number place: 7 Free	Number place: 8 Free	Number place: 9 Free	Number place: 10 Product ID: 41825528
Number place: 11 Free	Number place: 12 Product ID: 9845	Number place: 13 Product ID: 45902	Number place: 14 Free	Number place: 15 Free

Рисунок 4.30 система відзначила місце товару червоним кольором

Розглянемо докладніше основні функції програми, які представлені у вигляді діаграма прецедентів на рисунку 4.31.

На діаграмі продемонстровано можливості користувача.

Можливості адміністратора :

- Авторизація
- Перегляд загальної інформації системи
- Модифікація запису водія
  - 1) Редагування
  - 2) Видалення
  - 3) Створення
- Модифікація запису машини
  - 1) Редагування
  - 2) Видалення
  - 3) Створення
- Модифікація запису перевезення
  - 1) Редагування
  - 2) Видалення
  - 3) Створення
- Перегляд наявності місць на складі

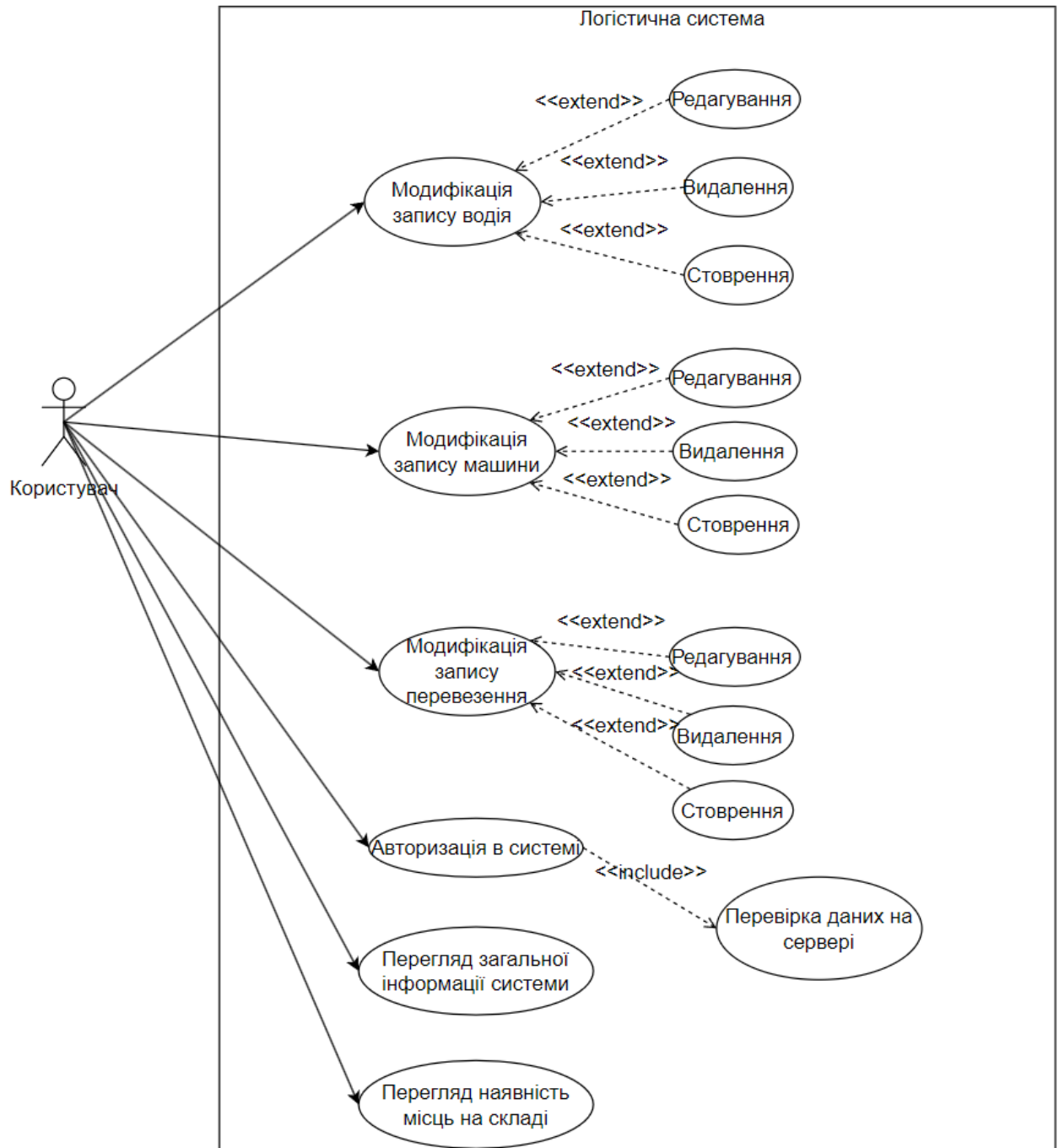


Рисунок 4.31 UML діаграма прецедентів

Опис діаграми послідовності функціоналу робота зі створенням запису про перевезення показує послідовність дій оптимізації логістики за допомогою інформаційної технології та IoT, зображено на рисунку 4.32.

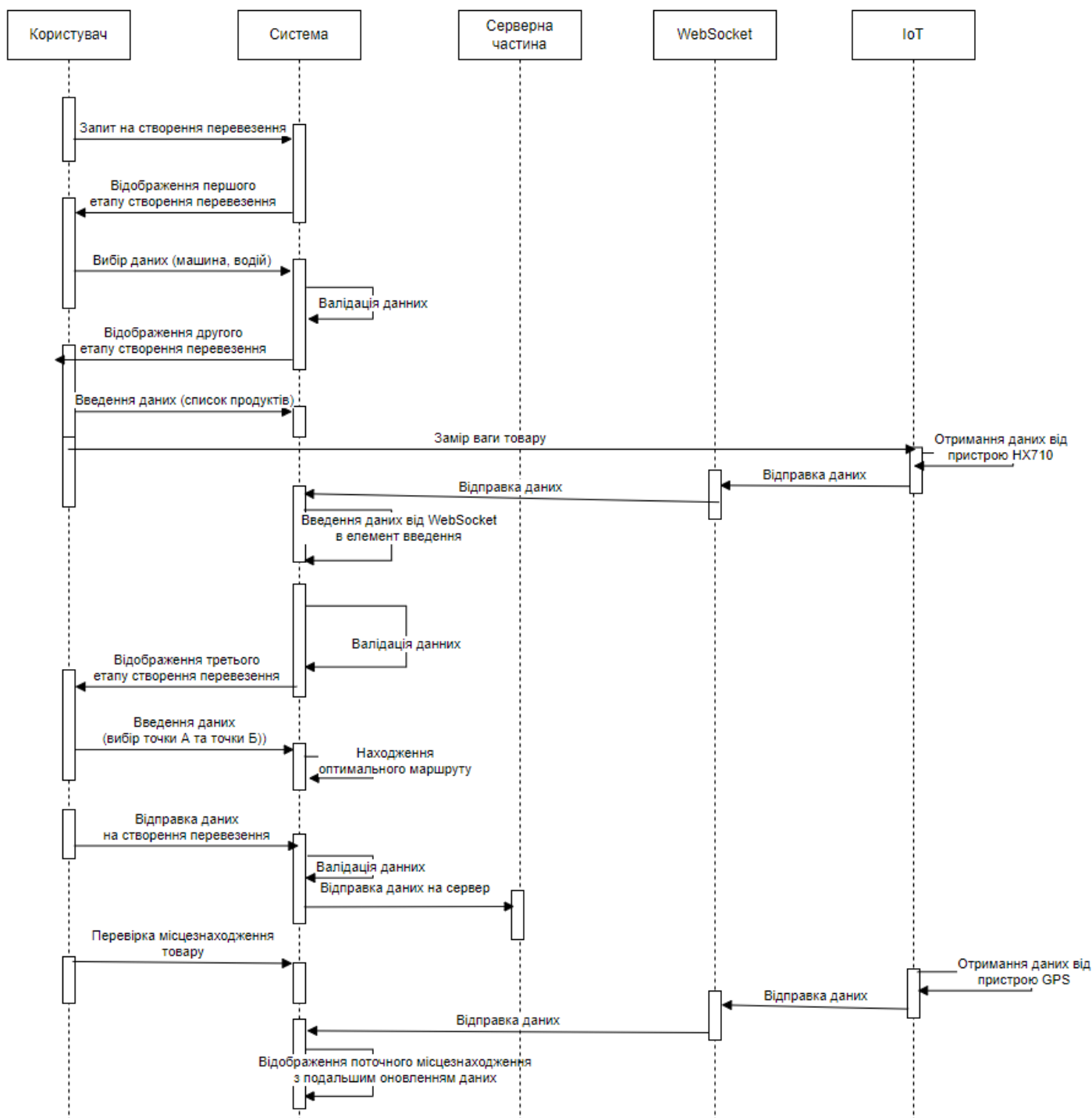


Рисунок 4.32 UML діаграма послідовності

Послідовність дій:

- Запит на створення перевезення;
- Відображення першого етапу створення перевезення;
- Вибір даних (машина, водій);

- Валідація даних;
- Відображення другого етапу створення перевезення;
- Введення даних (список продуктів);
- Валідація даних;
- Замір ваги товару;
- Отримання даних від пристрою HX710;
- Відправка даних;
- Введення даних від WebSocket в елемент введення;
- Валідація даних;
- Відображення третього етапу створення перевезення;
- Введення даних (вибір точки А та точки Б));
- Находження оптимального маршруту;
- Відправка даних на створення перевезення;
- Валідація даних;
- Відправка даних на сервер та збереження;
- Перевірка місцезнаходження товару;
- Отримання даних від пристрою GPS;
- Відправка даних;
- Відображення поточного місцезнаходження з подальшим оновленням даних.

## ВИСНОВКИ

В магістерській роботі розглянуто оптимізацію логістики за допомогою інформаційної технології на основі IoT. Досліджено основні потреби логістики задля створення системи на основі IoT.

У першому розділі статті розглянуто існуючі інформаційні системи логістики, виявлено їхні недоліки та переваги. За основу розробки нової системи взято основні функції, які необхідні для ефективного управління логістичними процесами. Для системи обрана веб-платформа, оскільки вона забезпечує доступність з будь-яких пристроїв і платформ. Це дозволяє швидко та зручно виконувати поставлені завдання на підприємстві. Розглянуто класифікацію інформаційних систем логістики та їхніх потоків. Класифікація допомагає визначити основну мету системи та її функціонал.

Інформаційний потік був проаналізований з метою визначення того, як інформація передається в системі користувачам для управління та контролю логістичними операціями.

У другому розділі розглядаються характеристики та проблеми, пов'язані з використанням сучасних IT-технологій у логістиці. Від його обробки залежить напрямок оптимізації тексту. Тому було визначено типи обробки інформації та обрано перший тип, який передбачає отримання нової інформації, що зробить систему більш автоматизованою. Також досліджуються прогалини в системі інформаційної логістики, щоб уникнути цих прогалин у процесі формування системи.

У третьому розділі розглянуто шляхи оптимізації логістики на основі IoT. Було дослідження типи мікропроцесорів IoT та їх переваги. Було досліджено шляхи та сфери застосування IoT технологій в логістиці. Також дослідження варіанти інтеграції IoT пристроїв, а саме протоколи передачі даних.

У четвертому розділі сформульовано основні мови програмування JavaScript, ReactJs, NodeJs, C++ та ряд допоміжних бібліотек з роботою IoT

пристроїв. Крім цього, використано різноманітні технології AJAX, API, WebSocket для створення системи. Також наведено діаграми та блок-схеми, на основі яких відображається функціонал системи та оптимізації роботи системи за допомогою IoT. Описано та показано основну послідовність подій у системі.

За результатами проведеної роботи створено та побудовано модель кінцевого продукту, Інформаційну технологію на основі IoT. Дана система дозволяє покращати ефективність управління логістичною системою на новий рівень. Покращення якості інформаційних систем сприяє ефективному вирішенню проблем, пов'язаних з перевезенням.

Пройдено тестування всіх модулів і виправлено помилки. Цей проект можна покращити шляхом додавання більшої кількості IoT пристроїв та нових секцій у системі.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Оксана Коляда, Валерій Докуніхін. Логістичні системи доставки специфічних вантажів. – С.531. - 978-966-388-623-7.
- 2 Cody Lindley. DOM Enlightenment. Exploring JavaScript and the Modern DOM, 2013. – С.180. – 978-1449342845.
- 3 MDN Web Docs [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/>.
- 4 Marco Russo, Alberto Ferrari. Tabular Modeling in Microsoft SQL Server Analysis Services (Developer Reference) 2nd Edition, 2017. С.512. - 978-1509302772.
- 5 Mike Amundsen, Mehdi Medjaoui, Erik Wilde, Ronnie Mitra. Continuous API Management: Making the Right Decisions in an Evolving Landscape 1st Edition, 2018. С.290. – 978-1492043553.
- 6 Elliotte Rusty Harold. JavaMail API Sending and Receiving Email with Java, С.98.
- 7 Тетяна Борщ, Ігор Пістунов, Інформаційні системи в фінансово-кредитних установах, 2013. – С.234. - 978-617-673-139-9.
- 8 Ольга Перевозчикова, Інформаційні системи і структури даних, 2007. – С.288. - 978-966-518-230-7.
- 9 Олена Струтинська, Інформаційні системи та мережеві технології. - Університет "Україна" 2008. – С.211. - 978-966-388-215-4.
- 10 Валентина Іванова, Управління інформаційними потоками туристичних підприємств. - Університетська книга 2017. – С.200. - 978-966-680-802-1
- 11 Роберт Сесіл Мартін. Clean Architecture. – Фабула, 2019. – 368 с. - 978-617-09-5286-8.
- 12 Копі Дж. Болл. Hacking Apis. Breaking Web Application Programming Interfaces - No Starch Press, 2022. – 368 с. – 9781718502444.
- 13 Andrew Lombardi. WebSocket Lightweight Client-Server Communications - O'Reilly Media, 2015. – 144 с. – 978-1-4493-6927-9.
- 14 Leonard Richardson. RESTful Web APIs: Services for a Changing World 1st Edition - O'Reilly, 2013. – 406 с. – 978-1449358068.
- 15 Introduction | Socket.IO [Електронний ресурс] – Режим доступу до ресурсу: <https://socket.io/docs/v4/>.
- 16 Джеймс Е. IoT System Design: Project Based Approach / Еліс Джеймс., 2022. – 279 с. – 97830308586501.
- 17 Arduino Docs | Arduino Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.arduino.cc/>.
- 18 LEARN ESP32 WITH ARDUINO - PRACTICAL PROJECT: Esp32 JTAG Debug, Programming Without Coding, IoT Project, GDB, MQTT and more, 2023. – 239 с. – B0BTKXY398.

- 19 Developing IoT Projects with ESP32: Automate your home or business with inexpensive Wi-Fi devices. – 474 с. – 978-1838641160.
- 20 Робсон Е. Head First. Програмування на JavaScript / Е. Робсон, Е. Фрімен., 2022. – 672 с. – 978-617-522-047-4.
- 21 Introduction to Node.js | Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/en/>.
- 22 Springer S. Node.js: The Comprehensive Guide to Server-Side JavaScript Programming (Rheinwerk Computing) / Sebastian Springer., 2022. – 834 с. – (978-1493222926).
- 23 Express - Node.js web application framework [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/>.
- 24 React – JavaScript-бібліотека для створення користувацьких інтерфейсів [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.legacy.reactjs.org/>.
- 25 Stroustrup B. Tour of C++, A (C++ In-Depth Series) / Bjarne Stroustrup., 2022. – 320 с. – 978-0136816485.
- 26 W. Kernighan B. C Programming Language / Brian W. Kernighan., 1988. – 272 с. – 978-0131103627.
- 27 Map creation and interactions | React Leaflet [Електронний ресурс] – Режим доступу до ресурсу: <https://react-leaflet.js.org/docs/>.
- 28 Hanes D. IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things / David Hanes., 2017. – 576 с. – 978-1587144561.
- 29 Логістика, 2019. – 392 с. – 978-617-67-386-7.
- 30 Сазонець І. Міжнародний бізнес і логістика. Понятійно-термінологічний словник / Ігор Сазонець., 2021. – 288 с. – 9786110124119.
- 31 Rudd J. A Practical Guide to Logistics: An Introduction to Transport, Warehousing, Trade and Distribution / Jerry Rudd., 2019. – 384 с. – 978-0749486310.



## ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



Державний університет інформаційно-  
комунікаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

“Оптимізація логістики за допомогою інформаційної технології на основі IoT”

на здобуття освітнього ступеня магістра  
зі спеціальності 126 Інформаційні системи та технології  
освітньо-професійної програми Інформаційні системи та технології

Виконав: здобувач вищої освіти гр. ІСДМ-61  
Міронов Дмитро  
Керівник: д.ф., доцент кафедри ІПЗАС  
Тушич Аліна Миколаївна

2

## Мета, Об'єкт, Предмет, Актуальність

Мета дипломного проекту оптимізація логістики за допомогою удосконаленої інформаційної технології на основі IoT.

Об'єктом дослідження є процес взаємодії логістики з цифровою індустрією та взаємодія системи з IoT технологіями.

Предмет дослідження є інформаційна система яка оптимізована за допомогою IoT.

Актуальність роботи є керувати процесом руху та зберігання товарів підприємства, своєчасно доставляти необхідну кількість товару від місця походження до місця споживання з найменшою вартістю та найкращим сервісом, автоматизувати процес прийняття товарів, створення логістичного процесу. Оптимізація всіх процесів за допомогою IoT.

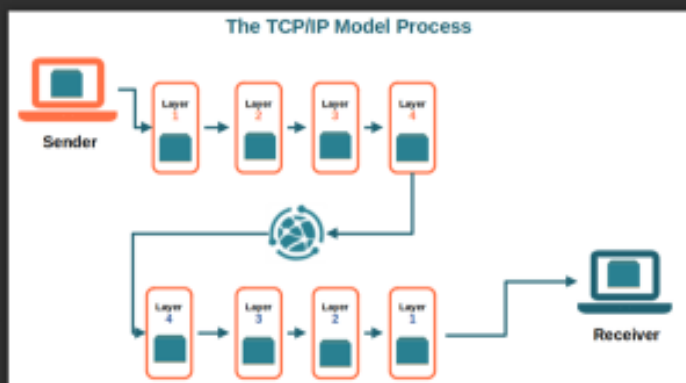


## 3

## Вибір способу комунікації між системою та IoT

Інтернет-протоколи для передачі даних між пристроями – це стандартні правила, які визначають, як пристрої можуть спілкуватися один з одним. Вони забезпечують узгодженість та сумісність між пристроями, що дозволяє їм обмінюватися даними незалежно від їх виробника чи моделі.

- SOAP
- REST
- GraphQL
- gRPC
- WebSocket
- WebHook
- MQTT



## 4

## Вибір мов програмування для системи та IoT

JavaScript (Js) – мова програмування, використовується для взаємодії з користувачем відповіді на дії користувача, та отримання інформації з серверу.

React — це декларативна, ефективна і гнучка JavaScript-бібліотека, призначена для створення інтерфейсів користувача. Вона дозволяє компонувати складні інтерфейси з невеликих окремих частин коду — “компонентів”

Node.js - це серверна платформа, обгорнута мовою JavaScript для створення масштабованих програм, керованих подіями.

C++ — це об'єктно-орієнтована мова програмування, яка надає чітку структуру програмам і дозволяє повторно використовувати код, знижуючи витрати на розробку.



5

## Можливості та оптимізація системи

- Створення, редагування, видалення записів про транспорт
- Створення, редагування, видалення записів про водіїв
- Створення, редагування, видалення записів транспортних перевезень
- Передача даних від GPS та відображення в реальному часі місцезнаходження вантажу
- Передача даних від вагів в форму заповнення товарів
- Автоматичний вибір місця на складі для товарів при отриманні який сканується через qr код або введенням за допомогою клавіатури

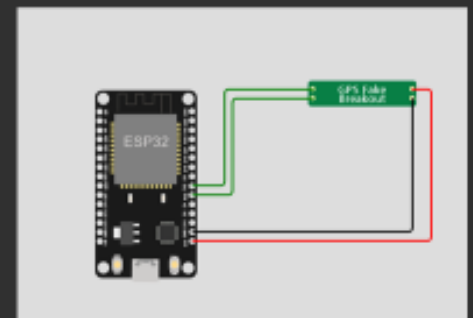
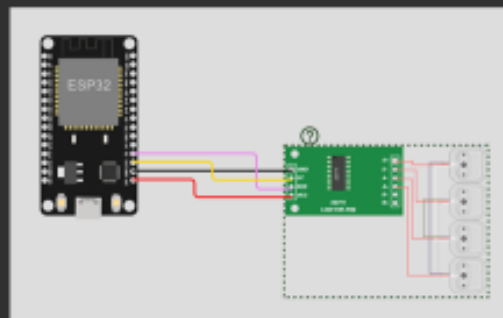
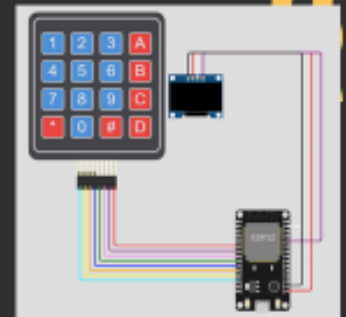
6

## Оптимізація на основі IoT

Оптимізація логістики з боку IoT це – використання датчиків які взаємодіють з платою яка запрограмована на певні дії.

Для оптимізації системи використовується такі датчики:

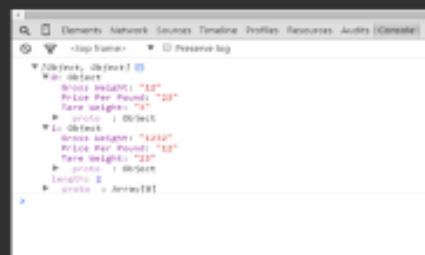
- GPS
- Ваги hx711
- Мембранна клавіатура або сканер qr
- Oled LCD SSD1305



7

## Логіка програми

Основним принципом системи є SPA ( Single Page Application ) дуже поширений спосіб програмування веб-сайтів в наші дні: ідея в тому, що сайт завантажує весь потрібний для досвіду користувача HTML/JS відразу ж при першому відвідуванні головної сторінки



При наступних переходах по сторінках браузер лише переглядає вміст заново, не оновлюючи сайт. Оптимізація системи основана за допомогою IoT.

8

## Логіка програми IoT

Логіка IoT полягає у зборі даних від датчиків та відправка даних. IoT комунікують за протоколом WebSocket. Передача даних відбувається у форматі JSON.

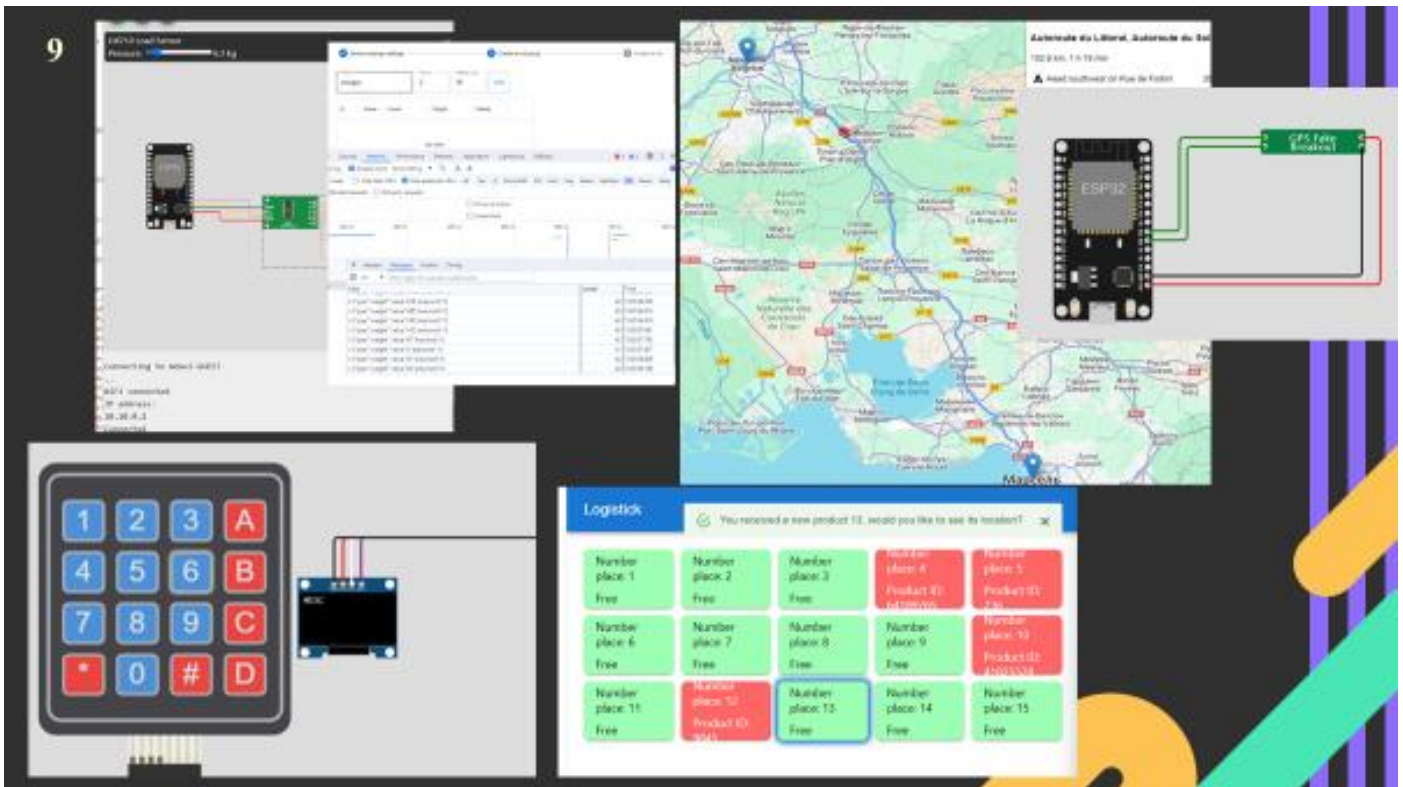
Для програмування IoT пристроїв використовуються такі бібліотеки:

Основні бібліотека:

- ArduinoWebsockets.h – взаємодія з протоколом WebSocket
- ArduinoJson.h – формування даних у тип json
- WiFi.h – підключення плати до Wifi

Бібліотеки для датчиків:

- HX711.h – зчитування значення з вагів
- NMEA.h – бібліотека та комбінований стандарт для роботи з GPS
- Keypad.h – для роботи з мембранною клавіатурою
- Adafruit\_SSD1306.h, Wire.h, Adafruit\_SSD1306.h – для взаємодії з OLED екраном



10

## Висновки

- проаналізовано основні визначення і поняття інформаційної логістичної системи;
- проаналізовано основні методи оптимізації за допомогою IoT;
- проаналізовано інтернет протоколи для передачі даних між пристроями;
- описано традиційні інформаційно-логістичні системи та їхні модифікації;
- описано технології розроблення Web системи;
- створено та побудовано модель кінцевого продукту, інформаційної логістичної системи для Web системи.
- програмування IoT пристроїв

## 11

**Апробація результатів дослідження:**

1. Міронов Д.С. «Оптимізація логістики за допомогою інформаційної технології на основі IoT». Тези доповіді на IV Науково-технічній конференції «Сучасний стан та перспектива розвитку IoT» - Київ, 7 квітня 2023 р.
2. Міронов Д.С. «Дослідження архітектури для зв'язку між клієнтом та сервером». Стаття у загальногалузевому науково-виробничому журналі «Зв'язок», м.Київ.

**Дякую за увагу!**