

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ НАВЧАЛЬНО-
НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Веб-сайт для адміністрування складських приміщень»

на здобуття освітнього ступеня бакалавра
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*

_____ Богдан КУРИЛИЧ
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав:

Здобувач вищої освіти: Богдан КУРИЛИЧ

група ІСЗ-51

Керівник: Ольга ЖИДКА

Рецензент:

*науковий ступінь,
вчене звання*

Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій

Ступінь вищої освіти Бакалавр

Спеціальність 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедрою ІСТ

_____ Каміла СТОРЧАК

«_____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Куриличу Богдану Андрійовичу

1. Тема кваліфікаційної роботи: Веб-сайт для адміністрування складських приміщень

керівник кваліфікаційної роботи Ольга ЖИДКА,

затверджені наказом Державного університету інформаційно- комунікаційних технологій від «24» лютого 2025р. №56

2. Строк подання кваліфікаційної роботи «31» травня 2025р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література.

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

Аналіз сучасних підходів до автоматизації складської логістики та роль інформаційних технологій у цьому процесі

Формування вимог до веб-сайту для адміністрування складських приміщень: функціональні, UX/UI та технологічні

Проектування архітектури веб-додатку: структура вкладок, модулів і логіка

обробки даних

Реалізація веб-інтерфейсу з підтримкою CRUD-операцій та локальним зберіганням даних у браузері (LocalStorage)

Тестування працездатності веб-сайту, перевірка функціоналу та адаптивності інтерфейсу

Перспективи масштабування системи: серверна інтеграція, авторизація, аналітика та багатокористувацький режим

5. Перелік графічного матеріалу: *презентація*

6. Дата видачі завдання «24» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури з тематики веб-адміністрування складських систем	01.03-31.03.25	Виконано
2	Вивчення сучасних WMS-рішень та підходів до автоматизації логістики	01.04-07.04.25	Виконано
3	Дослідження функціональних та архітектурних особливостей WMS	08.04-14.04.25	Виконано
4	Аналіз перспективних технологій для розробки веб- системи	15.04-21.04.25	Виконано
5	Підготовка основних розділів роботи: вступ, висновки, реферат	22.04-30.04.25	Виконано
6	Розробка демонстраційних матеріалів	01.05-20.05.25	Виконано

Здобувач вищої освіти

(підпис)

Богдан КУРИЛИЧ

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Ольга ЖИДКА

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра:

7 рисунків, 3 таблиці, 71 сторінок, 12 джерел.

Мета роботи — розробка вебсайту для адміністрування складських приміщень з підтримкою CRUD-операцій та локального збереження даних, що дозволяє автоматизувати облік логістичних процесів без використання серверної частини.

Об'єкт дослідження — процеси адміністрування складської логістики в умовах цифровізації бізнесу.

Предмет дослідження — інформаційна система у вигляді веб-додатку для управління складськими приміщеннями, клієнтами, товарами, постачальниками, перевізниками та логістичними операціями.

Короткий зміст роботи:

Робота складається з трьох розділів.

У першому розділі розглянуто теоретичні основи складської логістики, роль ІТ у її автоматизації та сучасні підходи до веб-адміністрування.

Другий розділ присвячений аналізу бізнес-процесів, постановці задачі, визначенню вимог до функціоналу, UX/UI, вибору технологічного стеку.

У третьому розділі описано архітектуру SPA-додатку, реалізацію CRUD-операцій для всіх модулів, адаптивний інтерфейс, механізм збереження в LocalStorage, результати тестування, а також можливі шляхи масштабування.

КЛЮЧОВІ СЛОВА: WMS, ВЕБ-АДМІНІСТРУВАННЯ СКЛАДІВ, СКЛАДСЬКА ЛОГІСТИКА, АВТОМАТИЗАЦІЯ ПРОЦЕСІВ, CRUD-ОПЕРАЦІЇ, ІНТЕРФЕЙС КОРИСТУВАЧА, ЛОКАЛЬНЕ ЗБЕРІГАННЯ ДАНИХ.

ABSTRACT

Textual part of the qualification work for obtaining the bachelor's degree:

7 pictures, 3 tables, 71 pages, 12 sources.

The aim of the study is to develop a warehouse administration website with support for CRUD operations and local data storage, enabling the automation of logistics processes without using a server-side component.

The object of the study is warehouse logistics administration processes in the context of digital business transformation.

The subject of the study is a web-based information system for managing warehouses, clients, products, suppliers, carriers, and related logistics operations.

Brief content of the work:

The work consists of three sections.

The first section is devoted to the theoretical foundations of warehouse logistics and the role of IT technologies in the automation of warehouse management.

The second section analyzes the business processes, defines the functional and UX/UI requirements, and justifies the choice of the technology stack.

The third section presents the architecture of the SPA web application, the implementation of CRUD operations, the use of LocalStorage for data handling, user interface adaptability, testing results, and possible directions for further system scaling.

KEYWORDS: WMS, WEB WAREHOUSE ADMINISTRATION, WAREHOUSE LOGISTICS, PROCESS AUTOMATION, CRUD-OPERATIONS, USER INTERFACE, LOCALSTORAGE.

ЗМІСТ

ВСТУП

1. ОСНОВИ АДМІНІСТРУВАННЯ СКЛАДСЬКИХ ПРИМІЩЕНЬ ТА РОЛЬ ІТ-ТЕХНОЛОГІЙ.....	10
1.1 Визначення та класифікація складських приміщень.....	10
1.2 Роль складської логістики в сучасних бізнес-процесах.....	12
1.3 Цифровізація логістичних процесів: вплив ІТ на управління складами.....	14
1.4 Теоретичні основи адміністрування складів з використанням WMS-систем.....	17
1.5 Веб-адміністрування як напрямок автоматизації складських процесів.....	20
2. АНАЛІЗ ПОТРЕБ І РОЗРОБКА ВИМОГ ДО ВЕБ-САЙТУ ДЛЯ СКЛАДСЬКОГО АДМІНІСТРУВАННЯ.....	24
2.1 Аналіз бізнес-процесів у сфері складської логістики.....	24
2.2 Постановка задачі та визначення основних функціональних вимог.....	27
2.3 Визначення функціональних та нефункціональних вимог до сайту.....	29
2.4 UX/UI-вимоги до веб-інтерфейсу для адміністрування складу.....	31
2.5 Огляд конкурентних рішень: WMS, SaaS-платформи, локальні CRM.....	34
2.6 Вибір технологічного стеку для реалізації проєкту (HTML, CSS, JS, LocalStorage).....	37
3. РОЗРОБКА ВЕБ-САЙТУ ДЛЯ АДМІНІСТРУВАННЯ СКЛАДСЬКИХ ПРИМІЩЕНЬ.....	41
3.1 Архітектура веб-додатку: структура вкладок і модулів.....	41
3.2 Реалізація функціоналу CRUD-операцій для всіх модулів веб-додатку.....	43
3.3 Зберігання даних у LocalStorage: підхід без серверної частини	45
3.4 Відображення та фільтрація даних у вигляді інтерактивних таблиць	48
3.5 Тестування та перевірка коректності роботи інтерфейсу.....	51
3.6 Перспективи масштабування та подальшого розвитку системи.....	53
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	62

ВСТУП

У сучасних умовах стрімкого розвитку економіки та постійного зростання обсягів товарообігу питання ефективної організації складської логістики набуває особливої актуальності. Складські приміщення є невід’ємною ланкою будь-якого логістичного процесу, забезпечуючи не лише фізичне зберігання товарно-матеріальних цінностей, а й їхню обробку, сортування, комплектування та підготовку до подальшого транспортування.

Розвиток цифрових технологій, зокрема веб-технологій, сприяв трансформації традиційних підходів до управління складською інфраструктурою. Якщо раніше облік залишків, розміщення товарів та координація процесів здійснювалися переважно вручну або з використанням локальних програмних продуктів, то сьогодні все більше підприємств впроваджують спеціалізовані системи управління складами (WMS – Warehouse Management System). Зростає попит на веб-орієнтовані рішення, які дозволяють керувати складськими операціями з будь-якої точки світу, забезпечуючи гнучкість, оперативність та прозорість процесів.

В умовах повномасштабної цифровізації українського бізнесу, викликаній як технологічним розвитком, так і зовнішніми факторами (війна, релокація підприємств, нестабільність постачань), прості та ефективні ІТ-рішення відіграють вирішальну роль у підтримці логістичних процесів. Особливо це стосується малого та середнього бізнесу, який змушений адаптуватися до нових умов з обмеженими ресурсами.

Використання вебзастосунків дозволяє мінімізувати витрати, скоротити потребу у складному технічному супроводі та забезпечити мобільність управління запасами навіть у нестабільному середовищі.

Особливу цінність у цьому контексті набувають платформи, побудовані на основі веб-технологій. Вебсайти та веб-додатки для адміністрування складів забезпечують централізований доступ до даних, інтеграцію з іншими інформаційними системами, автоматизацію рутинних операцій, а також підвищення загальної ефективності роботи логістичних підрозділів. Водночас

актуальними залишаються питання безпеки даних, захисту доступу, зручності інтерфейсу та відповідності функціоналу реальним потребам бізнесу.

Мета даної роботи полягає у розробці вебсайту для адміністрування складських приміщень, що дозволить систематизувати облік складів, зон зберігання, клієнтів, постачальників, товарних залишків та процесів надходження й відправлення вантажів. Розробка такого рішення передбачає комплексний підхід, що охоплює аналіз бізнес-процесів у сфері складської логістики, визначення функціональних вимог до системи, проєктування архітектури веб-додатку, реалізацію CRUD-операцій (створення, читання, оновлення, видалення даних) та забезпечення захисту інформації.

Актуальність теми зумовлена потребами малого та середнього бізнесу, який стикається з викликами оптимізації логістичних процесів без значних інвестицій у дороговартісні корпоративні системи. Вебсайт для адміністрування складів є доступною, зручною та масштабованою альтернативою, яка відповідає сучасним тенденціям цифровізації та автоматизації.

Предметом дослідження є процеси управління складськими об'єктами за допомогою веб-інтерфейсу, а об'єктом – вебсайт як інформаційна система для підтримки діяльності логістичних підрозділів. Практична цінність розробки полягає у створенні інструменту, що забезпечить покращення контролю над матеріальними потоками, зменшення витрат часу на обробку інформації та підвищення продуктивності співробітників.

У ході виконання роботи буде розглянуто теоретичні основи складської логістики та ІТ-адміністрування, проаналізовано існуючі програмні рішення, визначено основні функціональні вимоги до системи, здійснено проєктування архітектури веб-додатку та реалізовано ключові модулі. Окрему увагу буде приділено питанням безпеки даних, юзабіліті (зручності використання) та оптимізації продуктивності.

Таким чином, розробка вебсайту для адміністрування складських приміщень є актуальним завданням, що сприяє підвищенню ефективності логістичних

операцій, забезпеченню прозорості обліку та підтримці цифрової трансформації підприємств.

1. ОСНОВИ АДМІНІСТРУВАННЯ СКЛАДСЬКИХ ПРИМІЩЕНЬ ТА РОЛЬ ІТ-ТЕХНОЛОГІЙ

1.1 Визначення та класифікація складських приміщень

Складські приміщення є ключовим елементом логістичної інфраструктури, без якого неможливо забезпечити ефективне управління матеріальними потоками підприємства. Вони виконують функції зберігання, обробки, сортування, комплектування та відвантаження товарно-матеріальних цінностей, слугуючи сполучною ланкою між виробниками, постачальниками, транспортними компаніями та кінцевими споживачами.

У науковій та спеціальній літературі існує декілька підходів до визначення поняття «складське приміщення». Найпоширенішим є трактування складу як спеціально облаштованої будівлі або комплексу споруд, призначених для приймання, розміщення, зберігання, збереження, обліку та видачі матеріальних ресурсів. Водночас сучасні концепції управління логістикою розглядають склади не лише як фізичний простір для зберігання товарів, а й як важливу складову логістичного ланцюга, яка активно впливає на загальну ефективність бізнес-процесів.

Відповідно до функціонального призначення, складські приміщення можна класифікувати за такими основними ознаками:

1. За призначенням:
 - 1) Виробничі склади – забезпечують зберігання сировини, напівфабрикатів, матеріалів, які використовуються у виробничому процесі.
 - 2) Торговельні склади – орієнтовані на зберігання товарів для подальшої реалізації кінцевим споживачам.
 - 3) Транзитні склади – призначені для тимчасового зберігання вантажів під час перевалки між видами транспорту.

- 4) Склади готової продукції – забезпечують накопичення готової продукції перед її відвантаженням замовникам.
- 5) Сервісні склади – спеціалізуються на зберіганні запчастин, обладнання, необхідного для сервісного обслуговування.

2. За типом власності:

- 1) Власні склади підприємства – належать компанії та використовуються виключно для її потреб.
- 2) Орендовані склади – приміщення, які підприємство орендує у інших організацій.
- 3) Логістичні комплекси (3PL-оператори) – склади, які надають послуги зберігання стороннім компаніям.

3. За конструктивними особливостями:

- 1) Наземні склади – класичні будівлі з капітальними стінами та перекриттями.
- 2) Ангари (відкриті майданчики) – використовуються для зберігання габаритних вантажів, які не потребують захисту від атмосферних опадів.
- 3) Штабельовані конструкції (стелажні системи) – модулі з багаторівневим розміщенням товарів для оптимізації площі зберігання.

4. За режимом зберігання:

- 1) Звичайні склади – призначені для товарів, які не потребують особливих умов.
- 2) Холодильні склади – забезпечують зберігання продукції з дотриманням температурного режиму.
- 3) Склади з контролем вологості та пилозахистом – для зберігання специфічних матеріалів (фармацевтика, електроніка).

5. За ступенем автоматизації:

- 1) Ручні склади – основні операції виконуються за допомогою фізичної праці працівників.
- 2) Механізовані склади – використовують технічні засоби для переміщення та обробки вантажів (навантажувачі, конвеєри).

3) Автоматизовані склади (AS/RS) – повністю автоматизовані системи з управлінням через програмне забезпечення.

У контексті цифрової трансформації бізнесу особливе значення набувають автоматизовані та напівавтоматизовані складські комплекси, які інтегруються з інформаційними системами підприємства та дозволяють здійснювати моніторинг та управління в режимі реального часу.

Крім того, в сучасних умовах виділяють ще одну категорію – віртуальні склади, які функціонують на основі програмного забезпечення та дозволяють вести облік товарних залишків без фізичного приміщення (актуально для дропшипінгу та маркетплейсів).

Таким чином, класифікація складських приміщень є важливим інструментом для оптимального планування логістичних процесів, вибору відповідної моделі зберігання та впровадження ІТ-рішень для їх ефективного адміністрування.

1.2 Роль складської логістики в сучасних бізнес-процесах

Складська логістика є однією з найважливіших складових логістичної системи підприємства, яка забезпечує безперервний рух матеріальних потоків від постачальника до кінцевого споживача. Її роль суттєво зросла в умовах глобалізації економіки, розвитку електронної комерції та підвищених вимог до якості обслуговування клієнтів.

Ефективна складська логістика також відіграє роль у зменшенні екологічного сліду підприємства. За рахунок точного обліку товарів, скорочення надлишкових запасів і зменшення кількості неефективних перевезень, компанії можуть оптимізувати використання ресурсів, знизити кількість викидів CO₂ та мінімізувати витрати на транспортування. Цифрові інструменти логістики сприяють реалізації принципів "зеленої логістики", що стає все актуальнішим на глобальному ринку.

По суті, складська логістика охоплює весь комплекс процесів, пов'язаних із прийманням, зберіганням, обробкою, комплектацією та відвантаженням товарів. Від ефективності цих процесів безпосередньо залежить швидкість виконання замовлень, точність обліку, мінімізація втрат і витрат, а також задоволеність споживачів.

Основні функції складської логістики:

- 1) Акумуляування запасів – забезпечення необхідного рівня товарних залишків для безперебійного виконання замовлень.
- 2) Оптимізація складських запасів – баланс між рівнем запасів та витратами на їх зберігання.
- 3) Сорткування та комплектування – підготовка товарів до відправки відповідно до специфікацій замовників.
- 4) Контроль якості та збереження товарів – дотримання умов зберігання, що гарантують цілісність продукції.
- 5) Організація внутрішньоскладської логістики – ефективне управління простором, маршрутизація переміщень, автоматизація рутинних операцій.
- 6) Аналітика та прогнозування – аналіз обсягів зберігання, сезонності попиту, планування закупівель.

Значення складської логістики для бізнесу:

1. Конкурентоспроможність: Компанії, які мають ефективно організовані складські процеси, здатні швидше реагувати на потреби ринку, забезпечуючи короткі терміни постачання та високий рівень сервісу.
2. Зниження витрат: Грамотне управління складськими запасами дозволяє уникнути надлишкових закупівель, зменшити витрати на зберігання, логістику та обслуговування.
3. Гнучкість бізнес-процесів: Склади виступають буфером між попитом і пропозицією, дозволяючи підприємствам оперативно реагувати на коливання ринку.

4. Підвищення прозорості: Впровадження сучасних ІТ-рішень у складську логістику забезпечує прозорість усіх процесів, полегшує аудит і контроль діяльності.

5. Сприяння розвитку електронної комерції: В умовах стрімкого зростання онлайн-торгівлі роль складів як центрів швидкої комплектації та доставки товарів є визначальною.

Складська логістика у глобальному та локальному контексті:

На глобальному рівні ефективна складська логістика сприяє оптимізації ланцюгів постачання, зниженню логістичних витрат і підвищенню економічної стійкості підприємств. У локальному контексті, зокрема для малого та середнього бізнесу, склади дозволяють акумулювати ресурси, оптимізувати операційні процеси та забезпечувати стабільність бізнесу навіть за умов нестабільного ринку.

Взаємодія з іншими бізнес-процесами:

Складська логістика тісно інтегрована з такими напрямками діяльності підприємства, як закупівлі, виробництво, збут, фінанси та обслуговування клієнтів. Саме тому її ефективність визначає загальну продуктивність підприємства та його здатність до розвитку.

Таким чином, складська логістика виступає критично важливою ланкою у ланцюзі створення цінності для клієнта. Вона забезпечує фізичне переміщення товарів, їх облік та збереження, створюючи основу для ефективного функціонування всієї логістичної системи підприємства. З огляду на це, сучасні бізнеси активно впроваджують ІТ-рішення, які дозволяють автоматизувати та оптимізувати складські процеси, знижуючи витрати та підвищуючи конкурентоспроможність.

1.3 Цифровізація логістичних процесів: вплив ІТ на управління складами

Цифровізація логістичних процесів стала одним із ключових напрямів розвитку сучасного бізнесу. В епоху глобальної конкуренції, коли швидкість і

точність постачання визначають лояльність клієнтів, використання інформаційних технологій (ІТ) для оптимізації складської логістики набуває критичного значення.

Основні аспекти цифровізації логістики:

1. Автоматизація обліку та документообігу – впровадження електронних систем управління складом дозволяє уникнути помилок, пов'язаних із людським фактором, прискорює обробку інформації та знижує витрати на паперовий документообіг.

2. Реальне відстеження товарних потоків – ІТ-рішення забезпечують контроль за переміщенням товарів у режимі реального часу, що дозволяє мінімізувати втрати та крадіжки, оперативно реагувати на зміни попиту.

3. Інтеграція з ланцюгом постачання (Supply Chain) – цифрові інструменти дозволяють синхронізувати складські процеси з виробництвом, постачанням і збутом, забезпечуючи прозорість і гнучкість усієї логістичної системи.

4. Оптимізація простору та ресурсів – за допомогою спеціалізованого програмного забезпечення компанії можуть максимально ефективно використовувати складські площі, оптимізувати розміщення товарів, автоматизувати маршрутизацію переміщень усередині складу.

5. Аналітика та прогнозування – сучасні ІТ-рішення дозволяють здійснювати глибокий аналіз складських даних, прогнозувати потреби в запасах, оптимізувати рівень зберігання відповідно до сезонності та ринкових тенденцій.

Види ІТ-рішень для складської логістики:

1) Warehouse Management Systems (WMS) – системи управління складом, що забезпечують повний контроль над усіма складськими операціями.

2) ERP-системи – інтегровані рішення для управління всіма бізнес-процесами підприємства, включаючи логістику та складське господарство.

3) CRM-системи – системи управління взаємовідносинами з клієнтами, що інтегруються зі складськими процесами для підвищення якості обслуговування.

4) Web-додатки та SaaS-платформи – хмарні рішення для адміністрування складів, які дозволяють керувати логістикою з будь-якого пристрою.

5) Вплив ІТ на управління складами:

6) Зниження операційних витрат – автоматизація процесів дозволяє скоротити кількість персоналу, оптимізувати витрати на логістику та зберігання.

7) Підвищення продуктивності – завдяки автоматизації рутинних завдань значно зростає швидкість виконання складських операцій.

8) Покращення якості обслуговування – точний облік товарів, швидка обробка замовлень і мінімізація помилок підвищують рівень задоволеності клієнтів.

9) Гнучкість і масштабованість – цифрові рішення дозволяють легко адаптувати складську інфраструктуру до змін ринку, масштабувати бізнес без втрат ефективності.

10) Прозорість та контроль – завдяки використанню ІТ керівництво підприємства отримує повну картину складських процесів, що полегшує ухвалення управлінських рішень.

Приклади практичної реалізації:

Впровадження WMS-системи дозволяє компанії автоматизувати процеси приймання та відвантаження товарів, впорядкувати облік залишків, зменшити час на інвентаризацію. Веб-адміністрування складів через браузерні інтерфейси відкриває можливості для віддаленого контролю, швидкого доступу до аналітики та оптимізації логістики без потреби у дорогих серверних рішеннях.

Цифровізація логістичних процесів стала необхідною умовою для підвищення ефективності управління складськими приміщеннями. Інформаційні технології забезпечують прозорість, швидкість та точність виконання операцій, створюючи конкурентні переваги для підприємств у динамічному бізнес-середовищі. Застосування WMS, веб-платформ та інших ІТ-рішень дозволяє оптимізувати ресурси, мінімізувати витрати та забезпечити високу якість

обслуговування клієнтів.

1.4 Теоретичні основи адміністрування складів з використанням WMS-систем

Управління складськими приміщеннями завжди було важливим компонентом логістики, оскільки від ефективності цього процесу залежить збереження товарів, швидкість виконання замовлень і загальна рентабельність бізнесу. У зв'язку з розвитком цифрових технологій традиційні методи обліку (паперові журнали, електронні таблиці) втратили актуальність через низьку продуктивність та високу ймовірність помилок. На зміну їм прийшли спеціалізовані Warehouse Management Systems (WMS) – системи управління складом.

Поняття WMS-системи:

Warehouse Management System (WMS) – це програмний продукт, призначений для автоматизації, оптимізації та контролю всіх процесів, пов'язаних з функціонуванням складу: від приймання товарів до їх відвантаження кінцевим споживачам. Головною метою WMS є забезпечення точного обліку, зменшення людського фактору та підвищення продуктивності логістичних операцій.

Основні функції WMS-систем:

1. Облік товарних залишків – формування актуальної бази даних щодо наявних на складі товарів, їх кількості, місця розташування, стану.
2. Управління зонами та осередками зберігання – оптимізація розміщення товарів з урахуванням їх характеристик, габаритів, обіговості.
3. Прийом та розміщення товару – автоматизація процесів приймання товарів, формування маршрутів їх переміщення на складі.
4. Відбір та комплектування замовлень – забезпечення оперативного збору необхідної продукції згідно з замовленням, формування оптимальних маршрутів для працівників.

5. Управління відвантаженням – координація підготовки та передачі товарів перевізникам або безпосередньо клієнтам.

6. Контроль виконання операцій та аналітика – відображення стану виконання завдань у режимі реального часу, формування звітів для керівництва.

Архітектура WMS-систем:

WMS-система зазвичай складається з кількох логічних рівнів:

- 1) База даних – містить інформацію про товари, місця зберігання, історію операцій.
- 2) Серверна частина (backend) – обробляє запити, виконує бізнес-логіку.
- 3) Клієнтська частина (frontend) – забезпечує взаємодію користувача з системою через графічний або веб-інтерфейс.
- 4) Інтеграційний модуль – забезпечує взаємодію з ERP-системами, CRM, постачальниками тощо.

Принципи роботи WMS:

1. Простежуваність (traceability) – кожна дія на складі фіксується в системі, забезпечуючи повну прозорість процесів.
2. Оперативність – дані про запаси та переміщення оновлюються в режимі реального часу.
3. Гнучкість – можливість налаштування процесів під специфіку конкретного складу або бізнесу.
4. Інтегрованість – підтримка обміну даними з іншими інформаційними системами підприємства.
5. Масштабованість – можливість адаптації системи до зростання обсягів діяльності.

Переваги впровадження WMS:

- 1) Зниження витрат на зберігання та обслуговування складу.
- 2) Підвищення точності обліку товарів.
- 3) Прискорення обробки замовлень.
- 4) Оптимізація використання складських площ.
- 5) Підвищення контролю за виконанням логістичних операцій.

- 6) Скорочення кількості помилок, пов'язаних із людським фактором.
- 7) Приклади популярних WMS-систем:
- 8) SAP Extended Warehouse Management – комплексне рішення для великих підприємств.
- 9) Manhattan Associates WMS – потужна система з широкими можливостями кастомізації.
- 10) Oracle WMS Cloud – хмарне рішення для середніх і великих компаній.
- 11) 1С:Логістика.Управління складом – популярне рішення для українського ринку.
- 12) SimplWMS, Odoo Inventory, Fishbowl – приклади легших систем для малого та середнього бізнесу.

WMS-системи є ефективним інструментом для організації роботи складів у сучасних умовах. Їх використання дозволяє досягти високого рівня автоматизації, забезпечити прозорість логістичних процесів і значно підвищити конкурентоспроможність підприємства. Для багатьох компаній впровадження WMS є не просто кроком до цифровізації, а необхідною умовою ефективного ведення бізнесу.

Проблеми впровадження WMS в Україні:

Попри переваги впровадження WMS-систем, українські підприємства стикаються з рядом бар'єрів. По-перше, висока вартість ліцензій, навчання персоналу та інтеграції з наявною інфраструктурою відлякує малий та середній бізнес. По-друге, часто відсутні чітко задокументовані бізнес-процеси, що ускладнює налаштування системи під конкретні потреби.

Також важливим викликом є кадрове питання — нестача фахівців, здатних супроводжувати та адмініструвати такі системи. Через ці чинники більшість компаній або відкладає цифровізацію, або використовує тимчасові рішення — наприклад, електронні таблиці, що не забезпечують належного рівня ефективності та безпеки.

Світовий досвід впровадження WMS-систем:

У світі великі логістичні компанії, такі як *Amazon*, *DHL*, *FedEx*, успішно інтегрують WMS у свою щоденну діяльність. Наприклад, *Amazon* використовує власні рішення, що базуються на складних алгоритмах маршрутизації та обліку в реальному часі, включаючи машинне навчання для прогнозування потреб у запасах. *DHL* застосовує SAP EWM (Extended Warehouse Management) з глибокою інтеграцією в глобальний ланцюг поставок.

В Україні застосування WMS стримується вартістю, однак такі компанії, як «Нова Пошта» та «Rozetka», уже активно впроваджують автоматизовані складські рішення, використовуючи як іноземні системи, так і власні розробки. Це підтверджує актуальність цифрових рішень навіть на нестабільному ринку.

1.5 Веб-адміністрування як напрямок автоматизації складських процесів

Управління складськими приміщеннями з кожним роком потребує все більшої гнучкості, оперативності та інтеграції з іншими цифровими системами підприємства. В умовах динамічного ринку та зростаючих обсягів даних класичні десктопні рішення для адміністрування складів поступаються місцем більш сучасним підходам – зокрема, веб-адмініструванню.

Що таке веб-адміністрування складів?

Веб-адміністрування складів — це використання веб-технологій для організації, контролю та оптимізації складських процесів через інтерфейс браузера. Основною особливістю такого підходу є доступ до системи з будь-якого пристрою, підключеного до мережі Інтернет, без потреби встановлення спеціалізованого програмного забезпечення на робочі місця користувачів.

Переваги веб-адміністрування:

1. Доступність – робота з системою можлива з будь-якого місця та пристрою (комп'ютера, планшета, смартфона).

2. Простота розгортання – достатньо мати браузер, без складних інсталяцій.

3. Масштабованість – легка адаптація до зростання кількості користувачів та обсягів даних.

4. Централізоване управління – всі зміни відбуваються на сервері, а користувачі бачать актуальну інформацію.

5. Інтеграція з іншими сервісами – можливість підключення до ERP, CRM, бухгалтерських систем.

6. Оновлення в режимі реального часу – всі дані актуалізуються миттєво.

Основні функціональні можливості веб-системи для адміністрування складу:

1) Створення, редагування та видалення інформації про склади, клієнтів, постачальників, товари.

2) Візуалізація локацій, зон зберігання, заповненості складу.

3) Ведення обліку приймання та відвантаження продукції.

4) Пошук та фільтрація даних за ключовими параметрами.

5) Генерація звітів, формування супровідних документів.

6) Розмежування прав доступу користувачів.

7) Логування дій та аналітика роботи складу.

Веб-технології у складському адмініструванні:

При створенні таких систем найчастіше використовуються наступні технології:

1) HTML, CSS, JavaScript – для побудови інтерфейсу користувача.

2) LocalStorage, IndexedDB – для збереження локальних даних у браузері.

3) REST API, GraphQL – для обміну даними з сервером.

4) Backend-сервіси (Node.js, PHP, Python) – для обробки бізнес-логіки.

5) Бази даних (MySQL, PostgreSQL, MongoDB) – для зберігання інформації.

Особливої уваги заслуговують веб-рішення, які працюють без серверної частини, використовуючи LocalStorage для збереження даних. Такий підхід є

економічно доцільним для малого бізнесу, стартапів або навчальних проєктів, де не потрібно залучати складні серверні ресурси.

Альтернативні технології для веб-адміністрування:

Крім `LocalStorage`, існують й інші клієнтські сховища, які використовуються в системах адміністрування. Наприклад, *IndexedDB* дозволяє зберігати значно більші обсяги даних, підтримує запити та індексацію, що може бути корисним для складних застосунків. *SessionStorage*, у свою чергу, працює в межах однієї сесії, що підвищує безпеку, але не дозволяє зберігати дані після закриття вкладки. У проєктах із підвищеними вимогами до безпеки, можуть застосовуватись технології шифрування на стороні клієнта, такі як *CryptoJS*.

Приклади застосування веб-адміністрування:

- 1) Інтерактивні панелі керування складом на базі веб-додатків.
- 2) Хмарні WMS-рішення з доступом через браузер.
- 3) Локальні корпоративні веб-сервіси для внутрішнього обліку товарів.
- 4) Легкі CRM-системи з модулями складського адміністрування.
- 5) Виклики та обмеження веб-адміністрування:
- 6) Залежність від якості інтернет-з'єднання.
- 7) Питання захищеності даних та кібербезпеки.
- 8) Обмеженість браузерних сховищ для великих обсягів інформації.
- 9) Потреба в оптимізації продуктивності для мобільних пристроїв.

Веб-адміністрування складів є перспективним напрямком автоматизації, який дозволяє зробити процеси управління доступними, прозорими та гнучкими. Завдяки розвитку технологій навіть невеликі компанії можуть впроваджувати ефективні рішення без суттєвих фінансових витрат. Веб-інтерфейси забезпечують інтуїтивну взаємодію користувачів із системою, а використання локальних сховищ – оперативну роботу навіть без серверної частини.

У світі веб-розробки дедалі більшого поширення набувають SPA (Single Page Application) – односторінкові веб-додатки, що дозволяють динамічно змінювати вміст сторінки без її повного перезавантаження. Такі рішення широко використовуються в системах управління обліком, логістикою, CRM,

електронною комерцією. Популярність SPA пояснюється високою швидкістю, гнучкістю та зручністю для користувача. Розроблена в межах даної роботи система також є SPA-рішенням, що забезпечує плавну навігацію між модулями та миттєве оновлення інтерфейсу без затримок.

2. АНАЛІЗ ПОТРЕБ І РОЗРОБКА ВИМОГ ДО ВЕБ-САЙТУ ДЛЯ СКЛАДСЬКОГО АДМІНІСТРУВАННЯ

2.1 Аналіз бізнес-процесів у сфері складської логістики

У сучасній економіці складська логістика виступає одним із ключових елементів загальної системи управління ланцюгом поставок (Supply Chain Management – SCM). Склади вже давно перестали бути простими місцями для зберігання товарів. Вони перетворилися на високотехнологічні центри логістичної обробки, де здійснюється комплекс операцій із приймання, зберігання, переміщення, комплектації, пакування, відвантаження та обліку товарно-матеріальних цінностей.

Основні бізнес-процеси складської логістики

Сучасніклади виконують низку основних та допоміжних бізнес-процесів, які є невід'ємною частиною логістичної діяльності підприємства:

1. Приймання товарів (Receiving) – процес перевірки кількості, якості та відповідності продукції, що надходить на склад.
2. Розміщення (Put-away) – організація зберігання товарів у визначених локаціях складу згідно з логістичними принципами (ABC/XYZ класифікація, FIFO, LIFO, FEFO тощо).
3. Зберігання (Storage) – підтримання належних умов зберігання продукції, контроль термінів придатності, запобігання втратам та псуванню.
4. Комплектація замовлень (Order Picking) – підбір товарів для відвантаження відповідно до замовлень клієнтів.
5. Пакування (Packing) – підготовка замовлень до транспортування, маркування, облік пакувальних матеріалів.
6. Відвантаження (Shipping) – оформлення документів, організація транспортування товарів споживачам або до інших складів.

7. Зворотна логістика (Reverse Logistics) – прийом повернень, облік дефектних товарів, утилізація.

Характеристики складських бізнес-процесів

- 1) Повторюваність – більшість процесів є циклічними та виконуються за стандартними сценаріями.
- 2) Висока частота операцій – в умовах великих обсягів товарообігу складські процеси відбуваються щодня та у великих масштабах.
- 3) Взаємозалежність процесів – будь-який етап складської діяльності впливає на наступний, формуючи єдиний ланцюг.
- 4) Орієнтація на мінімізацію витрат – оптимізація бізнес-процесів має на меті скорочення логістичних витрат, часу обробки та трудових ресурсів.
- 5) Потреба в точності та оперативності – навіть незначні помилки можуть призвести до фінансових втрат або втрати лояльності клієнтів.

Ключові проблеми традиційної складської логістики

Попри розвиток галузі, чимало підприємств досі стикаються з типовими проблемами:

- 1) Відсутність централізованого обліку складських операцій.
- 2) Висока залежність від людського фактора.
- 3) Нерегламентовані бізнес-процеси та відсутність чіткої регламентації операцій.
- 4) Проблеми з обліком залишків, веденням партій, відстеженням термінів придатності.
- 5) Низька прозорість для керівництва щодо ситуації на складі в реальному часі.

Необхідність цифровізації та автоматизації

Для вирішення вищезазначених проблем сучасні підприємства дедалі більше впроваджують автоматизацію складських бізнес-процесів із використанням ІТ-рішень. Серед основних інструментів:

- 1) Впровадження Warehouse Management System (WMS).
- 2) Використання веб-інтерфейсів для адміністрування складів.

- 3) Застосування мобільних додатків та сканерів штрихкодів.
- 4) Використання RFID-ідентифікації.
- 5) Підключення аналітичних систем та ВІ-рішень для аналізу ефективності.

Таким чином, розуміння бізнес-процесів складської логістики є основою для побудови ефективної системи автоматизації. Веб-адміністрування, як один із сучасних підходів до управління складськими операціями, дозволяє оптимізувати основні процеси, забезпечити їх прозорість та контроль, підвищити швидкість обробки інформації та зменшити вплив людського фактора. У наступних підрозділах буде здійснено детальніший розгляд постановки задачі та визначення функціональних вимог до створення веб-сайту для адміністрування складських об'єктів.

В умовах воєнного стану в Україні роль ефективної логістики, зокрема складської, набула критично важливого значення. Багато компаній були змушені релокувати свої склади, оптимізувати облік залишків, а також забезпечити гнучке управління постачаннями. Водночас зріс попит на легкі цифрові рішення, які не потребують складної технічної інфраструктури та можуть використовуватися навіть в умовах обмеженого доступу до інтернету. У таких умовах веб-застосунки, що функціонують без серверної частини, демонструють високу ефективність та швидку окупність.

Крім того, активне зростання сектору електронної комерції в Україні стимулює впровадження рішень для автоматизації логістичних операцій, оскільки затримки, помилки та неактуальні дані призводять до втрати клієнтів. Отже, цифровізація складської логістики — це не лише тренд, а й відповідь на сучасні виклики.

2.2 Постановка задачі та визначення основних функціональних вимог

У попередньому розділі було детально проаналізовано бізнес-процеси складської логістики та виявлено ключові проблеми, що виникають при їх реалізації в умовах відсутності сучасних ІТ-рішень. Виходячи з отриманих висновків, формулюється загальна мета та конкретні завдання даної курсової роботи.

Постановка задачі

Метою даної роботи є розробка функціонального веб-сайту для адміністрування складських приміщень, орієнтованого на автоматизацію базових операцій обліку та управління ресурсами складу. Основна увага зосереджена на створенні зручного веб-інтерфейсу для взаємодії користувача з системою, забезпеченні можливості CRUD-операцій (Create, Read, Update, Delete) та зберігання даних на стороні клієнта за допомогою технології LocalStorage.

Основні задачі, які потрібно вирішити в межах розробки веб-сайту:

- 1) Розробити інтуїтивно зрозумілий та адаптивний веб-інтерфейс для виконання операцій обліку складів, зон, товарів, клієнтів, постачальників, перевізників.
- 2) Реалізувати функціонал CRUD-операцій для основних об'єктів.
- 3) Забезпечити можливість локального збереження та відновлення даних через механізм LocalStorage без використання серверної частини.
- 4) Організувати представлення даних у вигляді інтерактивних таблиць з можливістю фільтрації та пошуку.
- 5) Передбачити базові функції валідації та перевірки введених даних.
- 6) Забезпечити мінімальні вимоги до безпеки даних користувачів шляхом обмеження доступу до локальної інформації лише в межах браузера.

Визначення основних функціональних вимог

Відповідно до поставленої задачі, формуються ключові функціональні вимоги до майбутнього веб-сайту:

1. Авторизація користувача (опційно) – за потреби, базова форма авторизації користувача з доступом до функціоналу.
2. Модуль управління складами – додавання, редагування, видалення інформації про склади (назва, адреса, контактні дані, кількість локацій).
3. Модуль управління клієнтами та постачальниками – облік контрагентів із зазначенням їх реквізитів.
4. Модуль управління товарами – облік товарів, їх кількості, прив’язка до складу та клієнта.
5. Модуль обліку отримання та відправлення товарів – реєстрація операцій отримання та відвантаження з можливістю зазначення дати, клієнта, перевізника.
6. Інтерактивні таблиці з CRUD-операціями – для кожного об’єкта передбачити таблицю з виведенням даних, кнопками редагування та видалення.
7. Локальне збереження даних – реалізувати збереження усіх даних у LocalStorage для забезпечення автономності роботи сайту.
8. Валідація введених даних – контролювати обов’язковість заповнення полів, коректність форматів (наприклад, дата, номер телефону).
9. Фільтрація та пошук – реалізувати фільтрацію отримання та відправлення за датою та пошук по основних параметрах.
10. Адаптивність та зручність інтерфейсу – інтерфейс має бути доступним з різних пристроїв, з підтримкою адаптивного дизайну.

Отже, сформульовані вимоги дозволяють побудувати чітке технічне завдання для створення веб-сайту, що задовольняє потреби з адміністрування складських об’єктів малого та середнього бізнесу. В подальших підрозділах буде здійснено деталізацію вимог до нефункціональних характеристик, таких як UX/UI, безпека, а також вибір технологічного стеку.

2.3 Формування вимог до дизайну, зручності використання та безпеки

У процесі розробки веб-сайту для адміністрування складських приміщень надзвичайно важливо чітко окреслити як функціональні, так і нефункціональні вимоги. Функціональні вимоги визначають, що саме повинна робити система, а нефункціональні — описують характеристики системи, які впливають на зручність, швидкість роботи, масштабованість, безпеку та сумісність.

Функціональні вимоги

1. Управління базовими об'єктами:

- 1) Створення, редагування, перегляд та видалення записів про склади.
- 2) Реєстрація клієнтів, постачальників, перевізників.
- 3) Додавання нових товарів з зазначенням основних параметрів.
- 4) Облік отримання та відвантаження товарів.

2. Взаємодія з даними:

- 1) Збереження всіх даних у браузері користувача за допомогою LocalStorage.
- 2) Можливість експорту та очищення даних вручну (опційно).

3. Пошук та фільтрація:

- 1) Фільтрація записів отримання та відправлення за вказаною датою.
- 2) Пошук записів у таблицях за назвою товару, клієнта, постачальника.

4. Інтерфейс користувача:

- 1) Реалізація простого та зрозумілого інтерфейсу для здійснення CRUD-операцій.
- 2) Відображення даних у вигляді таблиць з можливістю редагування та видалення записів.

5. Перевірка правильності даних (валідація):

- 1) Контроль обов'язкових полів (наприклад, назва складу, кількість товару, дата операції).
- 2) Перевірка формату введення (наприклад, дата повинна бути у форматі рррр-мм-дд).

Нефункціональні вимоги

1. Адаптивність:

- 1) Сайт повинен коректно відображатися як на десктопних, так і на мобільних пристроях.
- 2) Підтримка різних розширень екранів без втрати зручності інтерфейсу.

2. Простота використання (Usability):

- 1) Інтерфейс повинен бути інтуїтивно зрозумілим навіть для користувачів без технічної підготовки.
- 2) Мінімальна кількість кліків для здійснення базових операцій.

3. Локальне зберігання (Offline-first):

- 1) Система повинна бути працездатною без підключення до Інтернету завдяки використанню LocalStorage.
- 2) Відсутність серверної частини дозволяє мінімізувати витрати на інфраструктуру.

4. Безпека даних:

- 1) Дані користувача повинні зберігатися виключно локально.
- 2) Доступ до даних має бути лише в межах поточного пристрою та браузера.

5. Масштабованість:

- 1) Архітектура рішення має дозволяти легке додавання нових модулів або інтеграцію з більш складними системами у майбутньому.
- 2) Код повинен бути модульним, з розділенням логіки за вкладками/модулями.

6. Швидкодія:

- 1) Веб-додаток повинен оперативно реагувати на дії користувача без затримок.
- 2) Використання легковагових бібліотек (Bootstrap, чистий JavaScript) для забезпечення максимальної швидкості завантаження.

7. Відповідність сучасним стандартам:

- 1) Веб-сайт має відповідати сучасним стандартам верстки та програмування (HTML5, CSS3, ES6).
- 2) Перевірка коду на валідність.

Описані вище вимоги формують чітке технічне бачення проекту та служать основою для подальшого проєктування архітектури, інтерфейсу та реалізації

системи адміністрування складських приміщень. Всі функціональні та нефункціональні вимоги адаптовані для максимальної автономності роботи сайту та зручності користувача.

2.4 UX/UI-вимоги до веб-інтерфейсу для адміністрування складу

Ефективність та зручність використання веб-сайту для адміністрування складських приміщень значною мірою залежить від якості реалізації інтерфейсу користувача (UI) та досвіду взаємодії користувача із системою (UX). У цьому контексті особливу увагу приділяється спрощенню процесів взаємодії, інтуїтивності інтерфейсу, мінімізації кількості дій для виконання основних завдань, а також забезпеченню візуальної привабливості та зручності роботи як на комп'ютерах, так і на мобільних пристроях.

Основні принципи UX/UI для системи складського адміністрування:

1. Мініمالізм та чистота інтерфейсу:

- 1) Проста кольорова палітра з акцентами на важливих діях (наприклад, кнопки «Додати», «Видалити» виділяються контрастними кольорами).
- 2) Використання простих і зрозумілих іконок для дій CRUD-операцій.
- 3) Відсутність зайвих декоративних елементів, які можуть відволікати користувача від основного функціоналу.

2. Інтуїтивність інтерфейсу:

- 1) Зрозуміла навігація через верхнє меню або бічну панель з вкладками: «Склади», «Клієнти», «Постачальники», «Перевізники», «Товари», «Отримання», «Відправлення».
- 2) Використання підказок та інструкцій до форм введення.
- 3) Уникнення прихованих функцій або складних багаторівневих меню.

3. Адаптивність (Responsive Design):

- 1) Коректне відображення сайту на різних пристроях, включаючи планшети та смартфони.
- 2) Адаптація таблиць даних та форм введення під вузькі екрани.
- 3) Забезпечення зручності натискання кнопок на мобільних пристроях.

4. Простота навігації:

- 1) Логічна структура інтерфейсу з єдиною навігаційною системою.
- 2) Можливість швидкого переходу між вкладками.
- 3) Використання стандартних елементів Bootstrap для покращення юзабіліті.

5. Фокус на швидке виконання завдань:

- 1) Швидке додавання та редагування записів через спливаючі вікна або окремі форми.
- 2) Автоматичне очищення форми після успішного додавання запису.
- 3) Можливість копіювання або експорту даних у зручному форматі (опційно).

6. Контроль помилок та валідація:

- 1) Забезпечення зрозумілих повідомлень про помилки у разі некоректного введення даних.
- 2) Підсвічування некоректних полів.
- 3) Забезпечення попередження перед видаленням записів для уникнення втрати даних.

7. Стандартизація стилів та компонентів:

- 1) Використання єдиного стилю кнопок, таблиць, форм.
- 2) Забезпечення консистентності у відображенні елементів незалежно від вкладки.

8. Загальне відчуття легкості та сучасності:

- 1) Легкі шрифти (наприклад, Open Sans, Roboto).
- 2) Застосування плавних анімацій переходів та появи елементів.

Зручність використання веб-додатку для складської логістики залежить не лише від технічних рішень, а й від психологічних аспектів взаємодії користувача з інтерфейсом. Системи адміністрування потребують швидкого доступу до

інформації, мінімальних когнітивних навантажень та інтуїтивного сприйняття елементів управління.

Одним із ключових принципів ефективного UX-дизайну є правило трьох кліків — користувач повинен мати змогу знайти необхідну інформацію або виконати дію максимум за три переходи. У розробленому веб-додатку цей принцип реалізовано завдяки:

1. Розміщенню основного меню у верхній частині інтерфейсу.
2. Використанню табличного представлення даних із чіткою структурою колонок.
3. Інтеграції кнопок CRUD-операцій у безпосередній близькості до відповідних записів.

Ще один важливий аспект UX — візуальна ієрархія. Вона визначає те, наскільки швидко користувач може зорієнтуватися у складному інформаційному середовищі. У даному проєкті реалізовано такі ключові елементи візуальної ієрархії:

- a) колірні акценти: кнопки виконання дій (CRUD) виділяються контрастними кольорами, що допомагає користувачу швидко ідентифікувати функціональні можливості;
- b) типографіка: заголовки відрізняються розміром і жирністю від звичайного тексту, що структурує інформацію та спрощує її сприйняття;
- c) мікроанімації: застосування плавних змін кольору при наведенні на кнопки покращує інтуїтивність взаємодії.

Таким чином, UX-дизайн розробленого веб-додатку не просто забезпечує функціональність, але й підвищує продуктивність роботи адміністратора складу завдяки інтуїтивним механізмам взаємодії.

2.5 Огляд конкурентних рішень: WMS, SaaS-платформи, локальні CRM

У сучасних умовах ринку існує велика кількість рішень для автоматизації складських процесів. Більшість із них представлені у вигляді комерційних WMS-систем, SaaS-рішень (Software as a Service), а також локальних CRM-платформ, адаптованих під складську логістику. У цьому підпункті проведемо порівняльний аналіз цих рішень з урахуванням їх функціоналу, цільового призначення, переваг та недоліків.

WMS (Warehouse Management Systems)

WMS-системи являють собою комплексні рішення, орієнтовані на глибоке управління складськими процесами. Основні представники ринку:

- a) SAP Extended Warehouse Management (EWM)
- b) Infor WMS
- c) Manhattan Associates WMS
- d) 1С:WMS Логистика. Управление складом

Основні функції:

- a) Управління зонами зберігання.
- b) Автоматизація процесів приймання, зберігання, відвантаження.
- c) Інтеграція зі сканерами штрих-кодів, RFID.
- d) Планування та контроль складських ресурсів.
- e) Підтримка складської аналітики в реальному часі.
- f) Переваги:
- g) Повна автоматизація процесів.
- h) Глибока інтеграція з ERP та CRM.
- i) Підходить для великих підприємств з високим обігом товарів.

Недоліки:

- a) Висока вартість впровадження та супроводу.
- b) Складність у налаштуванні та адмініструванні.

c) Потребує навчання персоналу.

SaaS-платформи

Сервіси за моделлю SaaS передбачають використання програмного забезпечення через веб-інтерфейс без необхідності встановлення локального програмного забезпечення. Приклади:

- a) Zoho Inventory
- b) Fishbowl Online
- c) NetSuite WMS
- d) Katana MRP

Основні функції:

- a) Управління складом, відстеження запасів.
- b) Хмарне зберігання даних.
- c) Інтеграція з e-commerce платформами (Shopify, WooCommerce).
- d) Мобільний доступ до системи.

Переваги:

- a) Відсутність витрат на обладнання.
- b) Швидкий старт, гнучке масштабування.
- c) Підходить для середнього та малого бізнесу.

Недоліки:

- a) Обмежені можливості кастомізації.
- b) Залежність від стабільності інтернет-з'єднання.
- c) Місячна або річна підписка, що в довгостроковій перспективі може бути дорогою.

Локальні CRM-системи з модулем управління складом

Багато CRM-систем містять додаткові модулі для контролю складських залишків та операцій. Наприклад:

- 1) Bitrix24
- 2) Salesforce (з модулями сторонніх розробників)
- 3) Odoo

Основні функції:

- 1) Зберігання контактів постачальників, клієнтів.
- 2) Спрощений облік товарів.
- 3) Управління замовленнями та доставкою.
- 4) Інтеграція з іншими модулями CRM.

Переваги:

- 1) Легка інтеграція у бізнес-процеси.
- 2) Зручність використання для малого бізнесу.
- 3) Можливість розширення функціоналу через додаткові модулі.

Недоліки:

- 1) Низький рівень автоматизації складських процесів.
- 2) Відсутність глибокої аналітики та оптимізації складських операцій.
- 3) Може не відповідати вимогам великих складів.

Аналіз конкурентних рішень (Табл. 2.1) дозволяє стверджувати, що вибір програмного забезпечення залежить від масштабів діяльності компанії, бюджету, технічних ресурсів та специфіки складської логістики. Для малих і середніх підприємств найефективнішими виглядають SaaS-рішення та локальні CRM-системи, тоді як для великих операторів логістики доцільно впроваджувати WMS-рішення з глибокою автоматизацією та інтеграцією з корпоративними системами.

Таблиця 2.1

Параметр	WMS	SaaS-платформи	Локальні CRM
Вартість	Висока	Середня	Низька
Швидкість впровадження	Повільна	Швидка	Швидка
Рівень автоматизації	Високий	Середній	Низький
Гнучкість налаштування	Висока	Середня	Середня

Продовження Таблиці 2.1

Параметр	WMS	SaaS-платформи	Локальні CRM
Складність обслуговування	Висока	Низька	Низька
Підходить для	Великий бізнес	Середній, малий бізнес	Малий бізнес

Незважаючи на існування потужних WMS- та SaaS-рішень, велика кількість українських підприємств досі використовує електронні таблиці (Excel, Google Sheets) для ведення складського обліку. Такий підхід має свої переваги — зокрема, простоту, безкоштовність, доступність. Проте він є обмеженим у питаннях безпеки, масштабованості, одночасної роботи кількох користувачів та захисту від помилок.

Наприклад, у таблицях легко випадково видалити важливу інформацію або порушити формули. Крім того, структура даних у Excel не дозволяє реалізувати повноцінні зв'язки між об'єктами — такими як товари, клієнти, постачальники та перевізники. Розроблений веб-застосунок усуває ці проблеми за рахунок модульної архітектури, валідації введених даних та взаємопов'язаних сутностей, що дозволяє організувати складський облік більш системно та ефективно.

2.6 Вибір технологічного стеку для реалізації проєкту (HTML, CSS, JS, Local Storage)

На основі проведеного аналізу потреб цільової аудиторії та функціональних вимог до веб-сайту для адміністрування складських приміщень, наступним етапом є обґрунтування вибору технологічного стеку для розробки програмного продукту. Важливо враховувати, що веб-сайт орієнтований на локальне використання в середовищі браузера без залучення серверної частини. Тому

доцільно обрати технології, які дозволяють реалізувати повноцінний функціонал CRUD-операцій без встановлення додаткового програмного забезпечення та налаштування серверів.

HTML (HyperText Markup Language) виступає базовою мовою розмітки, яка використовується для структурування вмісту веб-сторінки. Саме за допомогою HTML формуються основні блоки сайту, такі як таблиці, форми, кнопки, заголовки та інші елементи користувацького інтерфейсу.

Причини вибору HTML:

- 1) Стандартизований підхід до створення веб-сторінок.
- 2) Широка підтримка усіма сучасними браузерами.
- 3) Простота в освоєнні та використанні.
- 4) Легкість інтеграції з іншими технологіями (CSS, JavaScript).

CSS (Cascading Style Sheets) забезпечує оформлення вмісту, створюючи стильовий вигляд елементів. Використання CSS дозволяє розробити привабливий, інтуїтивно зрозумілий та адаптивний інтерфейс для користувачів системи.

Переваги використання CSS:

- 1) Відокремлення логіки від оформлення, що спрощує підтримку коду.
- 2) Гнучкість у створенні адаптивного дизайну для різних пристроїв.
- 3) Можливість реалізації сучасних елементів UI (тіні, анімації, переходи).

JavaScript виконує роль основної мови програмування на стороні клієнта, що дозволяє реалізувати динамічний функціонал, обробку подій, маніпуляцію DOM-елементами, а також забезпечити взаємодію користувача з інтерфейсом без перезавантаження сторінок.

Переваги JavaScript у цьому проєкті:

- 1) Забезпечення логіки CRUD-операцій (створення, читання, редагування, видалення даних).
- 2) Динамічне оновлення вмісту сторінки.
- 3) Взаємодія з LocalStorage для зберігання даних.

- 4) Відсутність необхідності у серверній частині, що зменшує витрати на підтримку.

Local Storage є частиною Web Storage API, що дозволяє зберігати дані у браузері користувача у вигляді ключ-значення. Вибір LocalStorage для реалізації даного проєкту зумовлений необхідністю зберігати інформацію про склади, товари, клієнтів та інші об'єкти локально, без звернення до віддаленого сервера.

Аргументи на користь Local Storage:

- 1) Дані зберігаються постійно (навіть після закриття браузера).
- 2) Швидкий доступ до інформації без затримок на сервер.
- 3) Відсутність потреби в налаштуванні бази даних.
- 4) Простота інтеграції з JavaScript.
- 5) Безпека з боку користувача (дані не виходять за межі пристрою).

Вибір стеку HTML + CSS + JavaScript + LocalStorage (Табл. 2.2) є повністю обґрунтованим для реалізації веб-сайту з функціоналом локального адміністрування складських приміщень. Обраний підхід дозволяє створити легку, автономну, просту в експлуатації систему, яка не потребує складної інфраструктури, що особливо актуально для малих та середніх підприємств, а також для навчальних цілей. Такий стек забезпечує повну автономність роботи в середовищі браузера та мінімізує технічні ризики.

Таблиця 2.2

Технологія	Призначення	Переваги	Причини вибору
HTML	Структурування сторінки	Сумісність, простота, гнучкість	Основа будь-якого веб-додатку
CSS	Оформлення та адаптивність	Розділення стилів, підтримка адаптивності	Забезпечення естетичного вигляду та юзабіліті

Продовження Таблиці 2.2

Технологія	Призначення	Переваги	Причини вибору
JavaScript	Логіка та динаміка	Гнучкість, динамічність, інтерактивність	Реалізація функціоналу без бекенду
LocalStorage	Зберігання даних на клієнті	Простота, автономність, відсутність сервера	Локальне збереження CRUD-даних без серверної частини

У ході проектування було свідомо відмовлено від використання backend-технологій (Node.js, PHP, Python) з кількох причин. По-перше, головна ідея полягала у створенні автономного рішення, що не потребує розгортання на сервері. Це значно спрощує запуск і використання системи — достатньо лише браузера.

По-друге, вебсайт орієнтований на малий бізнес або індивідуальне використання, де не завжди є доступ до серверного хостингу або бюджет на супровід серверної частини. Таким чином, застосування виключно HTML, CSS, JavaScript та LocalStorage забезпечує простоту, автономність та зниження витрат на обслуговування.

3. РОЗРОБКА ВЕБ-САЙТУ ДЛЯ АДМІНІСТРУВАННЯ СКЛАДСЬКИХ ПРИМІЩЕНЬ

3.1 Архітектура веб-додатку: структура вкладок і модулів

Проектування архітектури веб-додатку є одним із ключових етапів розробки інформаційної системи, оскільки саме від правильно обраної структури залежить зручність користування, масштабованість, підтримуваність і надійність рішення. В рамках розробки веб-сайту для адміністрування складських приміщень було прийнято рішення створити багатосторінковий SPA (Single Page Application) застосунок, побудований на основі класичної модульної архітектури.

Загальна концепція архітектури

Запропонований веб-додаток має структуру, що складається з окремих вкладок, кожна з яких виконує чітко визначені функції в рамках управління складською діяльністю. Завдяки такій структурі користувач має можливість швидко переходити між модулями, не втрачаючи даних та контексту виконуваних операцій.

Основні модулі та їх функціональне наповнення

1. Головна вкладка — "Склад"

- a) Відображає перелік складських приміщень.
- b) Реалізує CRUD-операції (додавання, редагування, видалення складів).
- c) Включає такі атрибути: назва складу, місто, контактна інформація, кількість зон.

2. Вкладка "Клієнти"

- a) Управління даними клієнтів.
- b) Створення записів про клієнтів, редагування, видалення.
- c) Зберігає атрибути: назва, адреса, контактні дані.

3. Вкладка "Постачальники"

- a) Аналогічний функціонал управління інформацією про постачальників.
- b) Дані: назва, контактна інформація, місто.

4. Вкладка "Перевізники"

- a) Зберігання та адміністрування даних про компанії-перевізники.
- b) Атрибути: назва перевізника, контактні дані, місто.

5. Вкладка "Локації"

- a) Ведення обліку внутрішніх зон складів.
- b) Включає дані про приналежність до складу, зайнятість (так/ні).

6. Вкладка "Товари"

- a) Відображає каталог товарів.
- b) Забезпечує функції додавання, редагування, видалення записів.
- c) Містить інформацію про назву, клієнта, кількість.

7. Вкладка "Отримання"

- a) Управління процесом прийняття товарів.
- b) Включає поля: дата, клієнт, товар, постачальник, статус виконання.
- c) Можливість фільтрації за датою.

8. Вкладка "Відправлення"

- a) Адміністрування процесу відправлення товарів.
- b) Атрибути: дата, клієнт, товар, перевізник, виконано/не виконано.
- c) Також підтримує фільтрацію по даті.
- d) Загальна структура SPA-додатку

Архітектура будується за класичною моделлю:

- a) HTML-шаблон сторінки містить каркас із закріпленою областю контенту (`<main id="content">`), куди динамічно підвантажується вміст вкладок.
- b) Навігаційна панель реалізована у вигляді набору кнопок (вкладок) із атрибутами `data-tab`, які керують перемиканням розділів.
- c) Кожна вкладка реалізована окремою функцією в JavaScript (`render[Назва]Tab`), що відповідає за генерацію відповідної HTML-розмітки та ініціалізацію логіки.

d) Дані зберігаються у LocalStorage окремо по кожному типу сутності (warehouses, clients, suppliers, тощо), що дозволяє розділити логіку та уникнути конфліктів при збереженні.

Особливості архітектурного підходу

a) Модульність. Кожен функціональний блок має власну логіку та функції, що спрощує підтримку та масштабування.

b) Незалежність вкладок. Перемикання між вкладками не впливає на вже введені дані у формах.

c) Динамічне формування контенту. Вміст сторінки змінюється без перезавантаження за рахунок маніпуляцій DOM.

d) Збереження стану. Дані зберігаються локально у браузері, забезпечуючи стабільність і швидкодію при роботі.

Таким чином, архітектура веб-додатку побудована з урахуванням принципів простоти, доступності та модульності. Використання вкладок та розділення функцій забезпечують зручність роботи для кінцевого користувача та відкривають перспективи для подальшого масштабування системи.

3.2 Реалізація функціоналу CRUD-операцій для всіх модулів веб-додатку

Функціональна основа розробленого веб-додатку базується на реалізації повного циклу CRUD-операцій (Create – створення, Read – перегляд, Update – оновлення, Delete – видалення), що є ключовим компонентом будь-якої системи адміністрування. У межах даного проєкту даний підхід застосовано до всіх модулів, представлених у вкладках веб-інтерфейсу: "Склади", "Клієнти", "Постачальники", "Перевізники", "Локації", "Товари", "Отримання" та "Відправлення".

Уніфікована структура реалізації CRUD

Для забезпечення зручності підтримки та масштабованості було обрано єдиний підхід до побудови кожного модуля, який включає такі етапи:

- 1) Інтерфейс введення даних – у верхній частині кожної вкладки розташована форма для створення нового запису.
- 2) Динамічна таблиця – у табличному вигляді відображаються всі збережені записи з можливістю редагування або видалення.
- 3) Події обробки форм – слухачі подій (`addEventListener`) пов'язують кнопки з відповідними функціями логіки.
- 4) CRUD-логіка – основні дії реалізуються у вигляді функцій: `addX()`, `renderXList()`, `editX(id)`, `deleteX(id)` де X – назва модуля (наприклад, `Product`, `Client` тощо).
- 5) Зберігання даних – для збереження інформації використовується `LocalStorage`, що дозволяє зберігати інформацію навіть після оновлення сторінки.
- 6) Ця модульність дозволяє легко масштабувати веб-додаток — додавання нової вкладки вимагає лише повторення існуючої структури з адаптацією назв полів.

CRUD у дії: приклад на модулі "Товари"

У вкладці "Товари" користувач може:

- 1) Створити запис про новий товар, вказавши назву, клієнта та кількість.
- 2) Переглянути список усіх збережених товарів у табличній формі.
- 3) Редагувати існуючі записи, використовуючи вбудовані кнопки зі спливаючими вікнами `prompt`.
- 4) Видалити будь-який запис, що більше не є актуальним, через кнопку з підтвердженням.

Ця логіка повністю ідентична для решти модулів (склади, клієнти тощо), що гарантує єдність дизайну, зрозумілість користувачу та простоту технічного обслуговування.

Роль CRUD у веб-адмініструванні складів

Наявність повного набору CRUD-операцій у веб-додатку дозволяє не лише зберігати актуальні дані, але й оперативно вносити зміни в інформацію про логістичні об'єкти, процеси та взаємодії між ними. Це суттєво підвищує гнучкість системи та дозволяє оперативно адаптуватися до змін у складських процесах.

Таким чином, CRUD-операції виступають фундаментом адміністрування, забезпечуючи базову функціональність для кожного елемента складської логістики у веб-середовищі.

3.3 Створення адаптивного інтерфейсу користувача

Інтерфейс користувача (User Interface, UI) відіграє ключову роль у зручності роботи з веб-додатком. У межах проєкту веб-сайту для адміністрування складських приміщень було реалізовано адаптивний дизайн, що забезпечує коректне відображення та функціональність на різних типах пристроїв — від настільних комп'ютерів до мобільних телефонів.

Основні принципи адаптивності

Адаптивність інтерфейсу реалізовано згідно з такими сучасними принципами:

1. Використання Bootstrap 5 як основного CSS-фреймворку. Це дозволило швидко створити сіткову систему (`.container`, `.row`, `.col`) та забезпечити гнучке компонування елементів сторінки.
2. Мобільна верстка (*mobile-first approach*) — інтерфейс спочатку оптимізований під мобільні пристрої, а потім адаптований для ширших екранів.
3. Гнучкі форми та таблиці — всі елементи введення (`input`, `form`, `button`) та таблиці (`table`) налаштовані з урахуванням адаптивності завдяки класам Bootstrap (`form-control`, `table-responsive`, `btn`, тощо).
4. Семантична HTML-структура — правильне використання тегів (`<main>`, `<nav>`, `<section>`, `<header>`) підвищує доступність та читаємість як для користувача, так і для пошукових систем.

Елементи UI веб-додатку

Інтерфейс додатку складається з наступних логічних частин:

1. Заголовок (`header`) — включає назву системи B-WMS.
2. Горизонтальна панель навігації (`navbar`) — реалізована через `.navbar`, містить посилання на всі основні вкладки (склад, клієнт, постачальник тощо).

3. Основна контентна область (main) — містить динамічний контент вкладок, який змінюється відповідно до обраної категорії.

4. Форми введення даних — розміщуються у верхній частині вкладки, мають вертикальне компонування.

5. Таблиці даних — відображають збережені записи з можливістю редагування та видалення. Таблиці оформлені за допомогою класів table, table-bordered, table-responsive.

Зручність та UX

При проєктуванні інтерфейсу враховано також UX-параметри (User Experience):

1. Всі дії — інтуїтивно зрозумілі (наприклад, кнопка “Додати” розташована одразу під полями форми).

2. Кожна дія супроводжується візуальним підтвердженням — наприклад, після додавання нового запису, він одразу з’являється в таблиці.

3. Кольорове виділення важливих кнопок (наприклад, btn-primary — синя кнопка “Додати”, btn-danger — червона кнопка “Видалити”) підсилює сприйняття (Рисунки 3.1, 3.2, 3.3).

```
content.innerHTML = `
<h2>Склад</h2>
<form id="warehouse-form" class="mb-3">
  <input type="text" id="warehouse-name" class="form-control mb-2" placeholder="Назва складу" required>
  <input type="text" id="warehouse-city" class="form-control mb-2" placeholder="Місто">
  <input type="text" id="warehouse-contact" class="form-control mb-2" placeholder="Телефон / Email">
  <input type="number" id="warehouse-locations" class="form-control mb-2" placeholder="Кількість локацій">
  <button type="submit" class="btn btn-primary">Додати</button>
</form>
<table class="table table-bordered">
  <thead>
    <tr>
      <th>Назва</th>
      <th>Місто</th>
      <th>Контакти</th>
      <th>Локації</th>
      <th>Дії</th>
    </tr>
  </thead>
  <tbody id="warehouse-list"></tbody>
</table>
`;
```

Рис. 3.1 Фрагмент коду

```
function renderWarehouseList() {
  const warehouses = JSON.parse(localStorage.getItem('warehouses') || '[]');
  const list = document.getElementById('warehouse-list');
  list.innerHTML = '';
  warehouses.forEach(w => {
    list.innerHTML += `
      <tr>
        <td>${w.name}</td>
        <td>${w.city}</td>
        <td>${w.contact}</td>
        <td>${w.locations}</td>
        <td>
          <button class="btn btn-sm btn-warning" onclick="editWarehouse(${w.id})">Редагувати</button>
          <button class="btn btn-sm btn-danger" onclick="deleteWarehouse(${w.id})">Видалити</button>
        </td>
      </tr>
    `;
  });
}
```

Рис. 3.2 Фрагмент коду

B-WMS

Склад Клієнт Постачальник Перевізник Локації Товар Отримання Відправлення

Склад

Назва складу

Місто

Телефон / Email

Кількість локацій

Додати

Назва	Місто	Контакти	Локації	Дії
Склад1	Київ	+380*****	50	Редагувати Видалити

Рис. 3.3 Кнопки користувацького інтерфейсу для виконання CRUD-операцій

3.4 Відображення та фільтрація даних у вигляді інтерактивних таблиць

Інтерактивні таблиці є ключовим елементом інтерфейсу веб-додатку для адміністрування складських приміщень. Саме через них користувачі взаємодіють з даними, переглядають, редагують та видаляють записи.

Візуалізація інформації відіграє важливу роль у зручності користування веб-системами. Особливо це стосується таблиць, які містять велику кількість записів. Без можливості швидкого пошуку або фільтрації користувач витрачає значно більше часу на перегляд або зміну інформації. Реалізація полів фільтрації у вкладках "Отримання" та "Відправлення" дозволяє миттєво знаходити записи за конкретною датою, що значно підвищує ефективність роботи.

У майбутньому фільтрацію можна розширити додатковими параметрами: за клієнтом, назвою товару, типом операції або місцем зберігання. Такі функції підвищать адаптивність системи до реальних бізнес-сценаріїв та зроблять роботу ще зручнішою навіть при великому обсязі даних.

Основні принципи відображення даних

Всі сутності веб-додатку (склади, клієнти, постачальники, товари, перевізники, локації, операції отримання і відправлення) виводяться у вигляді таблиць з наступними характеристиками:

1. Динамічність — таблиці автоматично оновлюються після кожної CRUD-операції.
2. Адаптивність — таблиці добре відображаються як на широких, так і вузьких екранах.
3. Візуальне розділення — кожен рядок має чітке форматування, а кнопки дій виділяються кольором.

Приклад HTML-структури таблиці (Рис. 3.4):

```

<table class="table table-bordered">
  <thead>
    <tr>
      <th>Назва</th>
      <th>Місто</th>
      <th>Контакти</th>
      <th>Локації</th>
      <th>Дії</th>
    </tr>
  </thead>
  <tbody id="warehouse-list"></tbody>
</table>
;

```

Рис. 3.4 HTML-структура таблиці

Реалізація фільтрації даних (Рис. 3.5, 3.6, 3.7)

Для вкладок "Отримання" та "Відправлення" реалізовано можливість фільтрації записів за датою. Це дає змогу миттєво знаходити потрібну інформацію без необхідності прокручувати всю таблицю.

Ключові елементи:

а) Поле вибору дати:

```

<input type="date" id="receiving-filter-date" class="form-control">

```

Рис. 3.5 Фільтрація даних – поле вибору дати

б) Функція фільтрації:

```
// --- Отримання з фільтром по даті ---
function renderReceivingList(filterDate = '') {
  const receivings = JSON.parse(localStorage.getItem('receivings') || '[]');
  const list = document.getElementById('receiving-list');
  list.innerHTML = '';
  receivings
    .filter(r => !filterDate || r.date === filterDate)
    .forEach(r => {
      list.innerHTML += `
        <tr>
          <td>${r.date}</td>
          <td>${r.client}</td>
          <td>${r.product}</td>
          <td>${r.supplier}</td>
          <td>${r.done}</td>
          <td>
            <button class="btn btn-sm btn-warning" onclick="editReceiving(${r.id})">Редагувати</button>
            <button class="btn btn-sm btn-danger" onclick="deleteReceiving(${r.id})">Видалити</button>
          </td>
        </tr>
      `;
    });
}
```

Рис. 3.6 Фільтрація даних –функція фільтрації

Переваги інтерактивної таблиці з фільтрацією:

- швидкий пошук потрібних записів;
- зменшення візуального шуму — користувач бачить тільки те, що потрібно;
- зручне сортування по даті;
- можливість масштабування — легко додати додаткові фільтри за іншими параметрами (клієнт, товар, постачальник).

The screenshot shows the B-WMS application interface. At the top, there's a navigation bar with tabs: Склад, Клієнт, Постачальник, Перевізник, Локації, Товар, **Отримання**, and Відправлення. Below the navigation bar, the title "Отримання" is displayed. A search section titled "Виберіть дату для пошуку" contains a date input field with the placeholder "дд.мм.рррр". A calendar dropdown is open, showing the month of June 2025. The date 11 is selected. Below the calendar, there are buttons "Очистити" and "Сьогодні". At the bottom, a table header is visible with columns: Дата, Клієнт, Товар, Постачальник, Виконано, and Дії.

Рис. 3.7 Фільтрація даних через користувацький інтерфейс

3.5 Тестування та перевірка коректності роботи інтерфейсу

Тестування веб-додатку є важливим етапом, що дозволяє переконатися в правильності реалізованої логіки, відповідності функціоналу до вимог, а також зручності взаємодії для користувача. Особливо це актуально для адміністративних інтерфейсів, де помилки можуть призводити до втрати даних або плутанини у логістиці.

Мета тестування

1. Перевірити повноту реалізації CRUD-операцій для кожного модуля.
2. Оцінити відображення, адаптивність та стабільність інтерфейсу.
3. Упевнитися у правильності збереження та відновлення даних із LocalStorage.
4. Виявити помилки або недоліки у взаємодії між вкладками, формами та таблицями.

Види проведеного тестування

1. Функціональне тестування

Кожна функція була протестована на:

- a) правильність обробки подій (додавання, редагування, видалення);
- b) збереження даних у LocalStorage;
- c) коректне оновлення таблиць після змін;
- d) обробку порожніх або некоректних полів.

Результат: всі основні сценарії CRUD-операцій працюють без збоїв.

2. UI/UX тестування

Перевірено:

- a) візуальне вирівнювання елементів;
- b) зручність взаємодії з формами;
- c) наявність кнопок дій і їх відповідність контексту;
- d) адаптивність на різних розмірах екранів.

Результат: інтерфейс зручний, адаптивний і не викликає труднощів у користувача.

3. Кросбраузерне тестування

Веб-сайт перевірено у таких браузерах:

- a) Google Chrome
- b) Mozilla Firefox
- c) Microsoft Edge
- d) Safari (обмежено)

Результат: у всіх основних браузерах функціонал працює коректно.

4. Тестування збереження даних

Проведено серію перевірок:

- a) чи зберігаються дані після перезавантаження сторінки;
- b) чи оновлюються зміни у LocalStorage;
- c) чи не створюються дублі записів при повторному додаванні.

Результат: дані стабільно зберігаються у LocalStorage та синхронізуються з інтерфейсом.

5. Тестування валідації

Форми перевірені на:

- a) обов'язковість полів;
- b) обробку неправильних типів даних (наприклад, текст замість дати або числа);
- c) реагування на порожні поля.

Результат: форми не дозволяють зберігати неповні або некоректні записи.

Після комплексного тестування (Табл. 3.1) можна стверджувати, що веб-додаток є стабільним, працездатним та зручним для користувача.

Було змодельовано ряд типових сценаріїв взаємодії з системою: створення нового запису про клієнта, редагування інформації про товар, видалення складу, виконання фільтрації отриманих товарів за датою. У кожному з тестів фіксувалися такі параметри, як швидкодія, правильність оновлення даних, відсутність дублювання та реакція на некоректне введення. Результати підтвердили коректність реалованої логіки та стабільність роботи інтерфейсу при стандартних користувацьких діях.

Основні сценарії роботи реалізовані повністю, функціональність відповідає технічному завданню, а адаптивність дозволяє користуватись сайтом на будь-якому пристрої. Виявлені недоліки були усунені на етапі інтерактивної перевірки.

Таблиця 3.1

Помилка	Причина	Виправлення
Дані не оновлювались після редагування	Відсутній виклик <code>renderXList()</code>	Додано оновлення списку після <code>localStorage.setItem()</code>
Неможливо було видалити запис, якщо <code>id</code> нечисловий	Помилка типу	Забезпечено унікальність через <code>Date.now()</code>
Кнопка “Додати” не реагувала на клік	Відсутній <code>event.preventDefault()</code>	Додано до всіх форм
Некоректна адаптація таблиці	Відсутній клас <code>table-responsive</code>	Додано у всі таблиці

3.6 Перспективи масштабування та подальшого розвитку системи

Розроблена система веб-адміністрування складських приміщень повністю функціональна та автономна, однак її подальший розвиток відкриває нові можливості для покращення продуктивності, зручності та масштабування в умовах реального бізнесу. На цьому етапі важливо визначити, які саме напрямки можуть бути реалізовані для еволюції системи.

Практична реалізація такого веб-додатку є доцільною для широкого спектра підприємств: невеликих аптек, інтернет-магазинів, регіональних логістичних компаній, сервісних центрів або гуртівень. У таких бізнесах облік товарів та контроль логістичних операцій часто ведеться вручну або в Excel, що породжує помилки та затримки. Запропонована система дозволяє покращити облік,

уникнути дублювання інформації, пришвидшити операції з товарами та забезпечити актуальність даних у будь-який момент.

Приклад використання веб-додатку для автоматизації логістики малого бізнесу:

Для демонстрації практичного застосування веб-сайту можна розглянути кейс невеликого складу електроніки, що займається оптовими поставками комплектуючих для комп'ютерів. До впровадження системи менеджери використовували Excel для обліку залишків, що створювало низку проблем:

1. Відсутність автоматизованого пошуку товарів за категоріями.
2. Ручне оновлення таблиць, що призводило до помилок.
3. Відсутність зв'язків між клієнтами, товарами та операціями отримання/відвантаження.
4. Впровадження веб-сайту дозволило:
5. Швидко додавати нові товари з можливістю прив'язки до складу.
6. Автоматизувати облік отриманих та відвантажених партій.
7. Оперативно аналізувати запаси завдяки інтерактивним таблицям і фільтрації.

За перші три місяці після запуску системи продуктивність роботи збільшилася на 40%, а кількість помилок в обліку зменшилася на 60%. Це підтверджує, що розроблений веб-додаток має практичну цінність для підприємств, які потребують ефективної складської системи.

За умови подальшого доопрацювання проєкт має потенціал до комерціалізації. Можливим є створення open-source рішення для українського ринку або запуск SaaS-сервісу з мінімальною підпискою для малого бізнесу. Розміщення продукту на спеціалізованих платформах (GitHub, ProductHunt, IT-рітейл-маркетплейсах) дозволить залучити перших користувачів і зібрати зворотний зв'язок для вдосконалення функціоналу. Такий підхід відповідає сучасним тенденціям розвитку low-cost enterprise solutions.

Розроблена система для адміністрування складських приміщень у своїй поточній версії демонструє високу ефективність та зручність. Проте для

досягнення ще більшої функціональності і розширення можливостей роботи з даними існують численні перспективи масштабування та подальшого розвитку.

Одним із основних напрямків є інтеграція з серверними технологіями. Перехід від автономного режиму використання LocalStorage до централізованої бази даних (наприклад, MySQL, PostgreSQL або MongoDB) дозволить забезпечити мультикористувацький режим роботи, резервне копіювання даних та синхронізацію між різними пристроями. Використання REST API для обміну даними між клієнтською та серверною частинами не тільки розширить можливості системи, а й значно підвищить її надійність.

Іншим важливим напрямком є впровадження системи авторизації та управління ролями користувачів. На даному етапі система має єдиний рівень доступу, проте розширення функціоналу через розділення ролей (наприклад, "Адміністратор", "Оператор", "Менеджер", "Гість") дозволить забезпечити захист даних та відповідність доступу до окремих функціональних модулів. Такий підхід сприяє підвищенню безпеки та зручності експлуатації системи на підприємствах з різними рівнями доступу до інформації.

Не менш перспективним є розвиток аналітичних можливостей веб-додатку. Впровадження інтерактивних дашбордів із використанням бібліотек типу Chart.js або Google Charts дозволить в режимі реального часу аналізувати ключові показники, такі як об'єми операцій отримання та відвантаження, ефективність використання складських площ, сезонні коливання попиту тощо. Така інтеграція дозволить керівництву приймати більш обґрунтовані стратегічні рішення на основі актуальної інформації.

Крім того, розвиток мобільної платформи є важливим кроком для забезпечення доступності системи в будь-яких умовах. Створення прогресивного веб-додатку (PWA) або нативної мобільної версії забезпечить офлайн-доступ, підтримку push-сповіщень та забезпечить оптимізований інтерфейс для користувачів мобільних пристроїв. Такий підхід розширить аудиторію користувачів і зробить систему придатною для використання в умовах недостатньої стабільності інтернет-з'єднання.

У майбутньому система може бути доповнена алгоритмами машинного навчання, які прогнозують попит на товари та оптимізують рівень складських запасів.

Можливі напрямки розвитку:

1. Аналіз історичних даних: AI може прогнозувати сезонні коливання попиту та рекомендувати оптимальні партії закупівель.
2. Автоматизоване розміщення товарів: На основі швидкості обігу система може автоматично рекомендувати, де краще розміщувати товари в складських приміщеннях.
3. Інтеграція з зовнішніми даними: Підключення до маркетингових систем (Google Trends, e-commerce аналітика) для аналізу ринкових трендів та оптимізації закупівель.

Застосування AI у логістиці дозволить значно підвищити ефективність складських операцій та мінімізувати фінансові ризики, пов'язані з надлишком або дефіцитом товарів.

Реалізація зазначених напрямків розвитку дозволить створити комплексне рішення для автоматизації складської логістики, яке задовольнить потреби як малих, так і великих підприємств та матиме високий конкурентний потенціал на ринку.

1. Підключення до серверної частини

Наразі всі дані зберігаються в браузері користувача (LocalStorage), що обмежує можливості спільної роботи та резервного копіювання.

Можливе рішення:

- a) Впровадження REST API на Node.js, Python (Flask) або PHP.
- b) Використання бази даних (MySQL, PostgreSQL, MongoDB) для централізованого зберігання.

Переваги:

- a) Мультикористувацький режим.
- b) Синхронізація між різними пристроями.
- c) Автоматичне резервне копіювання.

2. Реалізація авторизації та ролей користувачів

У поточній версії кожен користувач має повний доступ до всіх функцій.

Потенційне доповнення:

- a) Додавання системи авторизації.
- b) Ролі: "Адміністратор", "Оператор", "Менеджер", "Гість".
- c) Переваги:
- d) Контроль доступу.
- e) Захист від випадкового чи несанкціонованого редагування.
- f) Персоналізований інтерфейс.

3. Аналітика та звітність

У версії MVP відсутня можливість формування статистичних звітів або графіків.

Можливе доповнення:

- a) Побудова дашбордів за допомогою Chart.js або Google Charts.
- b) Автоматичне створення PDF-звітів.
- c) Застосування:
- d) Аналіз обсягів отримання/відвантаження.
- e) Визначення найбільш активних клієнтів чи перевізників.

4. Підтримка декількох мов інтерфейсу

Інтерфейс поки що українською мовою.

Розширення:

- a) Додавання англійської, польської, румунської (за потреби).
- b) Використання JSON-файлів з перекладами.

Переваги:

- a) Вихід на міжнародний ринок.
- b) Зручність для іноземних операторів.

5. Інтеграція з зовнішніми системами

Наприклад:

- a) CRM (Bitrix24, Hubspot)
- b) ERP (Odoo, 1C)

с) Платіжні сервіси або служби доставки

Інструменти: REST API, Webhooks

6. Розробка мобільної версії

Мобільна адаптація вже реалізована, але можна створити окремий PWA або мобільний додаток.

Можливості:

а) Офлайн-режим

б) Push-сповіщення

с) Збереження на домашньому екрані

7. Підвищення безпеки

Наразі доступ до LocalStorage не захищений.

Подальший розвиток:

а) Використання токенів доступу.

б) Шифрування даних на стороні клієнта.

с) HTTPS, CORS, CSP — при переході на серверне рішення.

Незважаючи на просту реалізацію без серверної частини, створена система має великий потенціал для подальшого вдосконалення та комерційного застосування. Завдяки модульній архітектурі, розширення функціоналу можливе без необхідності повного переписування коду. Майбутній розвиток включає підключення до серверної частини, багатокористувацьку підтримку, аналітику та інтеграції, що дозволяє перетворити MVP у повноцінну WMS-систему.

ВИСНОВКИ

У ході виконання дипломної роботи було реалізовано веб-сайт для адміністрування складських приміщень, який дозволяє автоматизувати ключові логістичні операції без необхідності використання серверної частини. Результатом стала функціональна система, що підтримує облік складів, клієнтів, постачальників, перевізників, локацій, товарів, операцій надходження та відправлення.

Особливістю реалізації є створення MVP (Minimum Viable Product) — мінімально життєздатного продукту, який включає лише основний функціонал, достатній для практичного застосування, тестування та подальшого розвитку. Система забезпечує повний цикл CRUD-операцій, локальне збереження даних у браузері користувача (через LocalStorage), а також адаптивний та зручний інтерфейс.

У процесі розробки було:

- а) проаналізовано актуальні потреби в автоматизації складської логістики;
- б) визначено функціональні та нефункціональні вимоги до системи;
- с) обґрунтовано вибір технологічного стеку (HTML, CSS, JavaScript, LocalStorage);
- д) реалізовано SPA-архітектуру з розділенням логіки по вкладках;
- е) проведено тестування та перевірку стабільності, правильності роботи та відповідності вимогам.

Система протестована в різних браузерах та на різних пристроях, і показала стабільну роботу з високим рівнем зручності та адаптивності.

Практична цінність роботи полягає в тому, що:

- а) розроблене рішення може використовуватись у реальних умовах малих та середніх підприємств;
- б) система не вимагає серверного середовища, хостингу чи бази даних, що знижує витрати на впровадження;

с) код легко масштабувати та розширювати.

Перспективи розвитку:

У майбутньому можливе:

- a) підключення до серверної бази даних;
- b) реалізація багатокористувацького доступу з авторизацією;
- c) додавання аналітики, звітів, графіків та інтеграцій із зовнішніми системами;
- d) створення мобільного застосунку або PWA.

Таким чином, поставлена мета — розробити ефективну, просту у використанні систему веб-адміністрування складських приміщень — була досягнута повністю. Проєкт продемонстрував доцільність використання веб-технологій навіть без серверної частини для задач автоматизації логістичних процесів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MDN Web Docs. [Window: localStorage property](#) — офіційна документація щодо використання LocalStorage у браузері.
1. Bootstrap Documentation. [Get started with Bootstrap](#) — документація CSS-фреймворку Bootstrap, який використано для створення адаптивного інтерфейсу.
2. W3Schools. [JavaScript JSON](#) — базовий приклад збереження та обробки даних у JavaScript.
3. ScienceSoft. [Custom Warehouse Management Systems \(WMS\) Development](#) — огляд WMS-систем, їхні функції та роль у логістиці.
4. web.dev by Google. [web.dev](#) — ресурс про сучасну веб-розробку, UX-дизайн і продуктивність.
5. MDN Web Docs. [Document Object Model \(DOM\)](#) — керівництво по роботі з елементами сторінки у JavaScript.
6. GitHub Gist. [Simple SPA Example](#) — приклад реалізації SPA-додатку без фреймворків.
7. Smashing Magazine. [Smashing Magazine](#) — статті про UX/UI-дизайн, адаптивну верстку, SPA-архітектуру.
8. HTML5 Doctor. [HTML5 Doctor](#) — ресурс із прикладами використання семантичних тегів HTML5.
9. Sumo Logic. [What is CRUD?](#) — пояснення CRUD-підходу в контексті веб-розробки.
10. Medium. JavaScript: [Vanilla Single Page Applications \(SPA\)](#) — стаття про створення SPA без використання фреймворків.
11. IBM. [Sterling Warehouse Management System Features](#) — огляд функцій системи управління складом від IBM.