

**ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ**

Кафедра комп'ютерних наук та інформаційних технологій

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

**«Обґрунтування та розробка засобу автоматизації
складського обліку підприємства»**

Студента 4 курсу, 10 групи,
спеціальності
122 «Комп'ютерні науки»

підпис студента

Шкара Дмитро
Олексійович

Науковий керівник

підпис керівника

Парацак Олексій
Миколай

Гарант освітньої програми
кандидат технічних наук, професор

підпис керівника

Демідов Павло
Георгійович

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____ **Затверджую**
Пурський О.І.
«12» грудня 2022р.

Завдання
на випускну кваліфікаційну роботу (проект) студента

Шкари Дмитра Олексійовича

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)

«Обґрунтування та розробка засобу автоматизації складського обліку підприємства»

Затверджена наказом ректора від «09» грудня 2022 р. № 3332

2. Строк здачі студентом закінченої роботи 26 травня 2023 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: Обґрунтування та розробка функціональної моделі засобу автоматизації складського обліку підприємства

Об'єкт дослідження: є інформаційні системи в області складського обліку та їх вплив на ефективність управління складськими процесами.

Предмет дослідження: технології автоматизації складського обліку на підприємстві

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Паращак О.М.	15.12.2022 р.	15.12.2022 р.
2	Паращак О.М.	15.12.2022 р.	15.12.2022 р.
3	Паращак О.М.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ПРОБЛЕМИ І ТЕНДЕНЦІЇ РОЗВИТКУ ІНФОРМАЦІЙНИХ СИСТЕМ В ОБЛАСТІ СКЛАДСЬКОГО ОБЛІКУ

1.1. Проблеми впровадження систем управління складськими процесами

1.2. Актуальність автоматизації

1.3 Тенденції розвитку інформаційних систем в області складського обліку

РОЗДІЛ 2. ПРЕДМЕТНА ОБЛАСТЬ ТА МОДЕЛЮВАННЯ ПРОЦЕСУ АВТОМАТИЗАЦІЇ

2.1 Опис предметної області

2.2 Огляд існуючих програмних систем управління складом в Україні

2.3 Моделювання процесу складського обліку

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗАЦІЇ

3.1 Функціональна модель складського обліку

3.2 Архітектура Web-додатку

3.3 Програмна реалізація

3.4 Функціональні можливості Web-додатку

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК

7. Календарний план виконання роботи

№ Пор	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	Вибір теми випускної кваліфікаційної роботи	01.10.2022	04.10.2022
2	Розробка та затвердження завдання на	15.12.2022	15.12.2022

	<i>випускню кваліфікаційну роботу</i>		
3	<i>Вступ</i>	03.02.2023	03.02.2023
4	<i>РОЗДІЛ 1. Проблеми і тенденції розвитку інформаційних систем в області складського обліку</i>	28.02.2023	28.02.2023
5	<i>РОЗДІЛ 2. Предметна область та моделювання процесу автоматизації</i>	06.04.2023	06.04.2023
6	<i>РОЗДІЛ 3. Програмна реалізація Web-додатку складу магазину вікон</i>	12.05.2023	12.05.2023
7	<i>Висновки</i>	15.05.2023	15.05.2023
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	15.05.2023	15.05.2023
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	26.05.2023	31.05.2023 -01.06.2023
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	27.05.2023	02.06.2023
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	30.05.2023	05.06.2023
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «15» грудня 2022 р.

Керівник випускної кваліфікаційної роботи (проекту)

Парацак О. М.

(прізвище, ініціали, підпис)

Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

Завдання прийняв студент-дипломник

Шкара Д. О.

(прізвище, ініціали, підпис)

9. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)

30.05.2023 р.

(підпис, дата)

10. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента

Шкара Д.О.

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми

Демідов П.Г.

(підпис, прізвище, ініціали)

Завідувач кафедри

Пурський О.І.

(підпис, прізвище, ініціали)

« » 2023 р.

Анотація

У випускній кваліфікаційній роботі досліджуються проблеми і тенденції розвитку інформаційних систем в області складського обліку. Теоретично обґрунтовано проблеми, пов'язані з впровадженням систем управління складськими процесами, а також актуальність автоматизації цього процесу. Розглядається опис предметної області, існуючі програмні системи управління складом в Україні та математичне моделювання процесу складського обліку. У третьому розділі розробляється система автоматизації складського обліку. Проводиться детальний аналіз вимог і функціональності системи, розробляються необхідні алгоритми та програмні модулі. Також проводиться тестування розробленої системи для перевірки її працездатності.

Ключові слова: складський облік, інформаційні системи, автоматизація, математичне моделювання.

Anotation

The graduation qualification work explores the problems and trends in the development of information systems in the field of warehouse accounting. Theoretical foundations are provided for the problems related to the implementation of warehouse management systems, as well as the relevance of automating this process. The subject area is described, existing software systems for warehouse management in Ukraine are reviewed, and mathematical modeling of the warehouse accounting process is considered. In the third chapter, an automated warehouse accounting system is developed. A detailed analysis of system requirements and functionality is conducted, necessary algorithms and software modules are designed. Testing of the developed system is also carried out to verify its functionality.

Keywords: warehouse accounting, information systems, automation, mathematical modeling.



ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ПРОБЛЕМИ І ТЕНДЕНЦІЇ РОЗВИТКУ ІНФОРМАЦІЙНИХ СИСТЕМ В ОБЛАСТІ СКЛАДСЬКОГО ОБЛІКУ	12
1.1 Проблеми впровадження систем управління складськими процесами.....	12
1.2. Актуальність автоматизації.....	16
1.3 Тенденції розвитку інформаційних систем в області складського обліку.	21
РОЗДІЛ 2. ПРЕДМЕТНА ОБЛАСТЬ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ	27
2.1 Опис предметної області	27
2.2 Огляд існуючих програмних систем управління складом в Україні.....	31
2.3 Моделювання процесу складського обліку	40
РОЗДІЛ 3. АВТОМАТИЗОВАНА СИСТЕМА ОЦІНЮВАННЯ РІВНЯ СОЦІАЛЬНО-ЕКОНОМІЧНОГО РОЗВИТКУ РЕГІОНІВ	43
3.1 Функціональна модель складського обліку.....	43
3.2 Архітектура Web-додатку	48
3.3 Програмна реалізація	49
3.4 Функціональні можливості Web-додатку.....	55
ВИСНОВКИ	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	64
ДОДАТКИ	66

ВСТУП

Актуальність дослідження: Впровадження інформаційної системи на підприємстві відіграє надзвичайно важливу роль у забезпеченні успішного функціонування та конкурентоспроможності. Для досягнення успіху необхідні не лише якісні товари, але й чіткий контроль процесів, облік продажів та поставок. Впровадження інформаційної системи допомагає побудувати цей процес належним чином.

Завдяки автоматизації складського обліку, підприємство може покращити якість та швидкість виконання основних процесів на складі. Це відбувається шляхом використання сучасного програмного забезпечення та комп'ютерного обладнання. Автоматизація складського обліку є ефективним методом оптимізації бізнес-процесів, який сприяє впровадженню спеціалізованих програмних продуктів та обладнання. Контроль матеріальних цінностей є необхідною складовою прибуткового торговельного бізнесу. Навіть при надійних співробітниках, без належного контролю можуть виникати спокуси вкрати або несвідомо порушити обов'язки. Знання залишків товару дозволяє точно оцінити потреби в наступних поставках.

З метою забезпечення конкурентоспроможності, підприємствам потрібно постійно прогресувати, шукати нові методи оптимізації та автоматизувати управління. Впровадження інформаційних систем сприяє удосконаленню систем управління та регулюванням матеріальних та інформаційних потоків на складах.

Мета і завдання дослідження: Метою даної роботи є вивчення проблем та тенденцій розвитку інформаційних систем в області складського обліку, а також розробка програмної системи автоматизації для оптимізації процесів складського обліку. Для досягнення поставленої мети були сформульовані наступні **завдання:**

- Аналізувати проблеми, пов'язані з впровадженням систем управління складськими процесами.
- Визначити актуальність автоматизації в області складського обліку.
- Дослідити тенденції розвитку інформаційних систем в області складського обліку.
- Описати предметну область складського обліку та провести огляд існуючих програмних систем управління складом в Україні.
- Розробити модель процесу складського обліку для подальшої автоматизації.
- Реалізувати програмну систему автоматизації на основі розробленої моделі.
- Дослідити функціональні можливості розробленого Web-додатку для складського обліку.

Об'єкт дослідження: Об'єктом дослідження є інформаційні системи в області складського обліку та їх вплив на ефективність управління складськими процесами.

Предмет дослідження: Предметом дослідження є проблеми впровадження систем управління складськими процесами, актуальність автоматизації в області складського обліку та тенденції розвитку інформаційних систем у цій галузі. Також розглядається предметна область складського обліку та моделювання процесу автоматизації.

Методи дослідження: У даній роботі використовуються такі методи дослідження:

- Аналіз літературних джерел та наукових публікацій, що стосуються проблем впровадження систем управління складськими процесами та автоматизації складського обліку.

- Огляд існуючих програмних систем управління складом в Україні для вивчення їх функціональності та можливостей.
- Моделювання процесу складського обліку для розробки програмної системи автоматизації.
- Програмна реалізація розробленої системи на основі обраної архітектури та технологій.

Практичне значення. Результати цього дослідження мають важливе практичне значення для підприємств, що займаються складським обліком. Розроблена програмна система автоматизації може сприяти покращенню ефективності та точності складського обліку, зменшенню часу на виконання процесів та підвищенню рівня контролю над запасами. Додатково, отримані результати дослідження можуть стати основою для подальшого вдосконалення та розвитку інформаційних систем у сфері складського обліку.

Структура та обсяг випускної кваліфікаційної роботи. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел із 38 найменувань, додатків і містить 65 сторінок основного тексту, 12 рисунків і 4 таблиці.

РОЗДІЛ 1.

ПРОБЛЕМИ І ТЕНДЕНЦІЇ РОЗВИТКУ ІНФОРМАЦІЙНИХ СИСТЕМ В ОБЛАСТІ СКЛАДСЬКОГО ОБЛІКУ

1.1 Проблеми впровадження систем управління складськими процесами

Складський облік - це систематичний процес збору, реєстрації, контролю та аналізу інформації про всі аспекти складської діяльності та запасів, який дозволяє ефективно керувати складськими процесами. Його основна мета полягає в забезпеченні точності, достовірності та актуальності даних про запаси, рух товарів, фінансову інформацію та інші показники, що відносяться до складської діяльності.

Складський облік включає в себе такі елементи:

- Ідентифікація та класифікація товарів: кожен товар на складі має бути однозначно ідентифікований та класифікований, щоб забезпечити його відстеження та контроль.
- Приймання товарів: процедура приймання включає перевірку достовірності поставки, порівняння отриманих товарів з документами, оцінку їх стану та якості. Результати цього процесу реєструються в системі складського обліку.
- Зберігання товарів: відповідне розташування товарів на складі є важливою складовою складського обліку. Це включає визначення оптимальних методів зберігання, розподіл товарів за зонами та ідентифікацію місць зберігання.
- Відвантаження товарів: процес відвантаження включає вибір товарів для відправки, оформлення відповідних документів, підготовку до транспортування та контроль за відправкою товарів.

- **Інвентаризація:** це процес перевірки та порівняння фактичних запасів з записами в системі складського обліку. Інвентаризація дозволяє виявити розбіжності та помилки в обліку та забезпечує актуальність даних про запаси.
- **Звітність та аналіз:** складський облік передбачає генерацію звітів та аналітичних даних, що відображають різні показники складської діяльності, такі як рівень запасів, оборотність товарів, затрати на складську діяльність та інші. Ці дані допомагають приймати обґрунтовані рішення щодо оптимізації складських процесів та управління запасами.

Протягом багатьох десятиліть представники малого та середнього бізнесу вели облік товарів на складі вручну. Для цих цілей залучалося велике число спеціалістів з вузьким профілем, обов'язки яких включали:

- оформлення документів щодо отримання, витрат та внутрішнього переміщення товарів у журналі обліку;
- внесення даних у реєстри обліку складських операцій;
- складання щомісячного звіту про рух товарів на складі для бухгалтерії; проведення інвентаризації товарів, що зберігаються на складі, для визначення залишків, виявлення недостач або перевищень товарів та матеріалів.

У сучасному бізнес-середовищі, де ефективність та оптимізація процесів є ключовими факторами успіху, системи управління складськими процесами відіграють важливу роль у покращенні продуктивності та забезпеченні якості управління запасами. Однак, впровадження таких систем не є безпроблемним завданням і зустрічається з рядом викликів та труднощів.

Кожна компанія або підприємство має свій підхід до вирішення задачі

автоматизації складських процесів. Проте, незалежно від вибраного підходу, всі вони стикаються з рядом проблем у процесі впровадження:

- Недостатня відкритість та сприйняття нововведень

Однією з найпоширеніших проблем є недостатня відкритість та сприйняття нововведень з боку працівників складу. Упровадження нової системи управління може супроводжуватися опором, неприйняттям змін та страхом перед невідомим. Це може призвести до незадоволення персоналу, низької мотивації та недостатньої участі в процесі впровадження. Для подолання цієї проблеми важливо провести ретельну підготовку персоналу, пояснити переваги нової системи та забезпечити необхідну підтримку та навчання працівників.

- Технічні складнощі та інтеграція з існуючими системами

Інтеграція нової системи управління складськими процесами з вже існуючими технологічними рішеннями та програмним забезпеченням може бути викликом. Різниця у форматі даних, нездатність до обміну інформацією та конфлікти між системами можуть призвести до зниження ефективності та неспроможності отримати повну картину управління складськими процесами. Для подолання цієї проблеми важливо провести технічний аналіз, визначити можливі точки інтеграції та забезпечити взаємодію між системами шляхом встановлення стандартів та протоколів обміну даними.

- Необхідність перерозподілу ресурсів:

Впровадження системи управління складськими процесами може вимагати перерозподілу ресурсів, таких як персонал, простір складу та обладнання. Це може бути проблемою, особливо якщо підприємство має обмежені ресурси або вже працює на повну потужність. Необхідно здійснити оцінку ресурсів та ретельне планування для забезпечення успішного впровадження системи

- Фінансові обмеження та ресурси

Впровадження системи управління складськими процесами може вимагати значних фінансових вкладень. Недостатні фінансові можливості організації або недооцінка витрат можуть обмежити можливості реалізації проекту та призвести до недостатньої функціональності системи. Для подолання цієї проблеми важливо провести детальний аналіз фінансових ресурсів, визначити ефективність витрат та знайти оптимальні шляхи фінансування проекту.

- Проблеми зі сумісністю обладнання:

Впровадження WMS (Warehouse Management System) може вимагати заміни або модернізації обладнання на складі. Це може бути складним процесом, особливо якщо існуюче обладнання несумісне або застаріле. Проблеми зі сумісністю обладнання можуть призвести до збоїв у роботі системи та непередбачуваних затримок.

- Недостатня підтримка та участь керівництва

Ефективне впровадження системи управління складськими процесами потребує активної підтримки та участі керівництва організації. Недостатня увага та інтерес з боку керівництва може призвести до недооцінки важливості проекту та відсутності необхідної підтримки. Для подолання цієї проблеми важливо залучити керівництво до процесу планування та впровадження системи, пояснити потенційні переваги та результати, а також забезпечити підтримку на рівні ресурсів, часу та експертної експертизи.

- Проблеми з безпекою даних:

Впровадження системи управління складськими процесами передбачає обробку великого обсягу конфіденційної інформації, такої як дані про клієнтів, запаси та фінансові дані. Проблеми з безпекою даних можуть виникнути, якщо система не належним чином захищена від

несанкціонованого доступу, злому або втрати даних. Необхідно забезпечити належні заходи безпеки, такі як шифрування даних, механізми автентифікації та забезпечення резервного копіювання даних.

Внедрення інформаційної системи на підприємстві є складним і багатограним процесом. Його успішне завершення вимагає ретельного планування, аналізу потреб, розробки відповідного програмного забезпечення та ефективного управління змінами. Для розв'язання проблем, що виникають у процесі впровадження інформаційної системи, існують різні підходи.

Кожен проект автоматизації складу представляє собою комплексну систему етапів та завдань, які вирішуються спільними зусиллями проектною командою, до якої входять фахівці з різних галузей. Час виконання проекту може варіюватись, а його цілі можуть бути різними. Крім того, програмний продукт, який використовується для автоматизації, може мати різноманітну функціональність.

Однак, незважаючи на різноманітність проектів автоматизації складу, їх сутність залишається майже незмінною, змінюється лише форма їх реалізації. Головною метою автоматизації складських процесів є забезпечення ефективного управління запасами, підвищення швидкості та точності операцій, зменшення витрат та покращення обслуговування клієнтів.

1.2. Актуальність автоматизації

Станом на сьогодні підвищення ефективності діяльності будь-якої компанії чи підприємства є одним із найважливіших завдань, які сприяють розвитку ринкових відносин в умовах конкуренції. Це пояснюється тим, що компанії стикаються з необхідністю вдосконалення власних економічних

структур у відповідь на зміни ринкових відносин. Автоматизація якомога більшої кількості бізнес-процесів, у тому числі процесів складської логістики, відіграє важливу роль в економічному розвитку компаній. На даний момент зростає суспільний інтерес до систем управління складськими процесами та логістикою в цілому.

Автоматизація спрямована на поліпшення ефективності та точності операцій, пов'язаних з управлінням складськими процесами. Сучасні технології дозволяють автоматизувати рутинні завдання, такі як ідентифікація та відстеження товарів, управління запасами, планування маршрутів доставки і багато інших. Це дозволяє знизити час і затрати, пов'язані з виконанням цих завдань, а також уникнути людських помилок, що часто виникають при ручному виконанні.

Одним з регіонів, де особливий інтерес до логістики проявляється, є Львівська область в Україні (рис. 1.2). З її вигідним географічним розташуванням, яке забезпечує зручний зв'язок з країнами Європи, Львівська область стає привабливим логістичним вузлом для багатьох підприємств. Наявність сучасних систем управління складськими процесами та автоматизованих рішень в цьому регіоні може значно сприяти підвищенню ефективності логістичних операцій та забезпеченню конкурентоспроможності компаній [14].

Таким чином, автоматизація складської логістики є актуальною і важливою задачею для сучасних компаній. Вона допомагає підвищити ефективність, знизити витрати і покращити якість обслуговування. Ураховуючи рост інтересу до цієї теми, є необхідність у вдосконаленні і розробці нових автоматизованих рішень, що відповідатимуть потребам сучасного ринку і допоможуть компаніям досягти успіху.

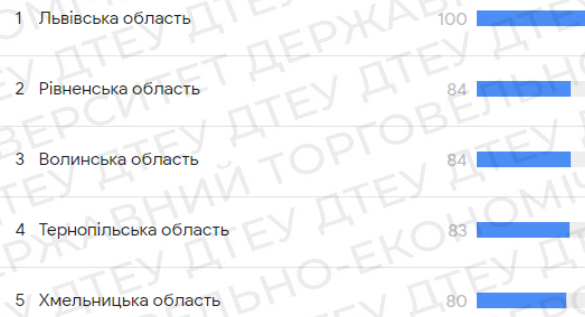
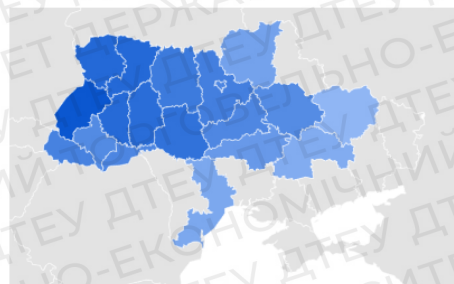


Рис. 1.2 Популярність запиту «Логістика» в Україні за останні 12 місяців

В сучасних ринкових умовах зростання вимог до конкурентоспроможності продукції призвело до необхідності підвищення ефективності діяльності промислового підприємства. Зокрема, керівники підприємств прагнуть досягти цього за допомогою сучасних методів моделювання та впровадження автоматизованих систем управління

При автоматизації бізнес-процесів людина не виключається повністю з ланцюжка створення додаткової вартості, вона зберігає присутність у більшості функціональних областей діяльності підприємства. Автоматизація швидше означає раціональний розподіл навантаження між людиною і машиною, баланс якого залежить від конкретного підприємства і цілей автоматизації [1]

Складський облік є важливою складовою бухгалтерського обліку підприємства. Він включає в себе систематичне збирання, аналіз і звітування про інформацію, що стосується матеріальних цінностей, зберіганих на складах. Однак, у зв'язку зі зростанням обсягів бізнесу, складський облік стає все більш складним і часо- та ресурсомістким процесом. Тому актуальність автоматизації складського обліку стає необхідною для забезпечення ефективності та точності цього процесу.

Ось кілька переваг автоматизації складського обліку:

1. Зниження помилок: Використання спеціалізованого програмного забезпечення для складського обліку дозволяє автоматизувати процеси збирання та обробки даних, що знижує ризик людських помилок.
2. Швидкість та точність: Автоматизація дозволяє прискорити процеси складського обліку і забезпечує швидке та точне звітування. Це дає можливість керівникам підприємства отримувати актуальну інформацію для прийняття рішень у реальному часі.
3. Ефективне управління запасами: Автоматизована система складського обліку дозволяє вести контроль за запасами, їх рухом та використанням. Це сприяє оптимізації запасів, попередженню зайвих закупівель та зменшенню втрат.
4. Інтеграція з іншими системами: Автоматизована система складського обліку може бути інтегрована з іншими інформаційними системами підприємства, такими як система управління виробництвом або система управління взаємовідносинами з клієнтами. Це дозволяє отримати комплексну картину діяльності підприємства і покращує координацію різних процесів.
5. Зменшення витрат: Автоматизація складського обліку дозволяє зменшити витрати на ручну працю, збільшити продуктивність працівників та знизити втрати через покращення керування запасами.
6. Аналітика та прогнозування: Автоматизована система складського обліку надає можливість проводити аналіз даних та прогнозування попиту. Це дозволяє підприємствам планувати закупівлі та розподіляти ресурси ефективно.
7. Відстеження виконання нормативів: Автоматизована система складського обліку дозволяє встановлювати стандарти та нормативи для кожного типу матеріальних цінностей. Це дозволяє контролювати

дотримання встановлених параметрів, що забезпечує якість зберігання та уникнення порушень.

8. **Покращення зв'язку з постачальниками:** Автоматизована система складського обліку може бути інтегрована з системою постачальників, що спрощує процеси замовлення та отримання матеріальних цінностей. Це сприяє зменшенню часу на пошук та комунікацію з постачальниками, покращує планування поставок і допомагає забезпечити належний рівень запасів.

9. **Моніторинг термінів придатності:** Автоматизована система складського обліку може включати функціонал для відстеження термінів придатності продукції на складі. Це допомагає попереджати спливання термінів придатності, забезпечує своєчасне вилучення застарілих товарів і допомагає знизити втрати.

10. **Забезпечення безпеки та контролю:** Автоматизована система складського обліку дозволяє встановлювати системи контролю доступу та відстеження переміщення матеріальних цінностей на складі. Це сприяє підвищенню безпеки та запобігає крадіжкам або недостачам.

11. **Аналіз продуктивності:** Завдяки автоматизованій системі складського обліку можна проводити аналіз продуктивності складських процесів. Це дозволяє виявляти ефективність роботи складського підрозділу, виявляти можливі проблемні аспекти та приймати заходи для покращення продуктивності.

12. **Збереження історії:** Автоматизована система складського обліку зберігає детальну історію процесів, пов'язаних з матеріальними цінностями на складі. Це створює можливість аналізувати минулі події, виявляти тенденції та здійснювати стратегічне планування на основі накопичених даних.

13. Підвищення клієнтського задоволення: Швидкість та точність, забезпечені автоматизацією складського обліку, дозволяють забезпечити своєчасну обробку та доставку замовлень. Це призводить до задоволення клієнтів, збільшення їх довіри та покращення репутації підприємства.

Автоматизація складського обліку стає необхідним інструментом для сучасних підприємств, оскільки вона сприяє ефективному управлінню запасами, зменшенню помилок, оптимізації процесів та підвищенню продуктивності. Використання автоматизованої системи допоможе підприємствам досягти конкурентної переваги та успішно протистояти викликам сучасного бізнесу.

1.3 Тенденції розвитку інформаційних систем в області складського обліку

Розвиток інформаційних систем в галузі складського обліку відбувається у відповідності з деякими важливими тенденціями. Складський облік, який є необхідною частиною логістичних процесів, вимагає точного та ефективного контролю для багатьох підприємств. Використання сучасних технологій у розвитку інформаційних систем, які займаються складським обліком, має ключове значення для поліпшення ефективності, точності та автоматизації складських процесів.

Давайте розглянемо основні тенденції в цій сфері, зосередившись на використанні сучасних технологій:

Штучний інтелект (Artificial Intelligence, AI) і машинне навчання:

Один з головних аспектів використання штучного інтелекту (ШІ) і машинного навчання (МН) у складському обліку полягає у передбаченні

попиту. За допомогою машинного навчання можна аналізувати великі обсяги даних про продажі, попит і зовнішні фактори, що впливають на попит. Складні алгоритми, глибинне навчання та статистичні методи дозволяють штучному інтелекту передбачати майбутній попит з високою точністю. Це дозволяє підприємствам ефективніше планувати запаси і ресурси.

Також ШІ і МН можуть бути використані для автоматизації процесів управління складом, наприклад, ідентифікації, класифікації і сортування товарів. З використанням візуального розпізнавання об'єктів і нейронних мереж системи штучного інтелекту можуть автоматично розпізнавати й відстежувати товари на складі, спрощуючи тим самим процеси інвентаризації і пошуку.

Ще одним важливим аспектом є виявлення аномалій і попередження проблем на складі. ШІ може аналізувати дані про температуру, вологість, рух та інші параметри, щоб виявляти незвичні або підозрілі зміни. Це дозволяє операторам складу реагувати своєчасно на проблеми, такі як погіршення умов зберігання або крадіжки.

Використання ШІ і МН в сфері складського обліку може принести значні переваги, такі як зменшення витрат на запаси, підвищення ефективності та точності процесів і зниження ризику помилок. Однак, необхідно враховувати етичні аспекти, забезпечувати конфіденційність даних і забезпечувати належну підготовку персоналу для використання цих технологій.

Інтернет речей (Internet of Things, IoT):

IoT відкриває нові можливості для збору даних про стан та рух товарів у реальному часі. Сенсори та пристрої IoT можуть бути встановлені на складах, у контейнерах та на транспортних засобах, що дозволяє отримувати актуальну інформацію про розташування товарів, температуру, вологість та

інші параметри. Це сприяє поліпшенню відстежування запасів, управлінню якістю та уникненню затримок у поставках. Інтеграція даних з IoT в інформаційні системи дозволяє отримати повну картину про стан складських запасів [8].

Хмарні технології та обробка даних у реальному часі:

Хмарні технології надають можливість зберігати та обробляти великі обсяги даних у віддалених центрах обробки. Це дозволяє швидко та ефективно аналізувати дані про запаси, рух товарів та виконання замовлень. Застосування обробки даних у реальному часі дозволяє оперативно реагувати на зміни в попиті, запасах та умовах поставок, що забезпечує більш гнучке та точне управління складською діяльністю.

Розширена реальність (Augmented Reality, AR) та віртуальна реальність (Virtual Reality, VR):

Розширена реальність (Augmented Reality, AR) та віртуальна реальність (Virtual Reality, VR) - це дві технології, які можуть знайти застосування в складському обліку, поліпшуючи ефективність та точність операцій. Відтак, давайте розглянемо, як AR та VR можуть бути використані у цій галузі:

Розширена реальність (AR) в складському обліку:

Візуалізація та ідентифікація товарів:

За допомогою AR-технологій, персонал складу може сканувати або спостерігати товари за допомогою пристроїв, таких як смартфони або AR-окуляри, і отримувати додаткову інформацію про них. Наприклад, можна відобразити інформацію про назву товару, його характеристики, номери партій, дати прибуття тощо. Це спрощує процес ідентифікації та контролю товарів на складі.

Підготовка замовлень:

AR може допомагати персоналу складу збирати замовлення, відображаючи наочні покажчики для розміщення товарів у контейнерах або на полицях. Це спрощує та прискорює процес збору замовлень, зменшує кількість помилок.

Навчання та навчання персоналу:

AR можна використовувати для навчання нових співробітників або навчання нових процесів. Вони можуть отримувати інтерактивні інструкції та демонстрації на своїх пристроях, що скорочує час навчання та кількість помилок.

Віртуальна реальність (VR) у складському обліку:

Симуляція роботи складу:

Віртуальна реальність може допомогти створити віртуальне середовище для навчання та імітації різних складських операцій. Співробітники можуть віртуально взаємодіяти зі стелажми, контейнерами, обладнанням та іншими складськими елементами, щоб набути реальних навичок і ефективності.

Планування та оптимізація складських приміщень:

Віртуальну реальність можна використовувати для моделювання та візуалізації розташування товарів та оптимізації використання складського простору. Віртуальне середовище дозволяє співробітникам випробувати різні сценарії продакт-плейсменту та оцінити їх ефективність перед фактичним впровадженням.

Контроль стану обладнання та обслуговування:

Віртуальна реальність може допомогти персоналу складу відстежувати стан і підтримувати обладнання на складі. Це може включати віртуальну перевірку технічних характеристик, діагностику проблем і перегляд інформації щодо ремонту або технічного обслуговування обладнання.

розширення електронної комерції:

Зростання популярності онлайн-шопінгу призводить до збільшення обсягів замовлень та вимагає швидкого та ефективного виконання доставки. Системи складського обліку повинні бути готовими впоратися зі збільшеним потоком замовлень та оптимізувати процеси підготовки та відправлення товарів, щоб забезпечити високу якість обслуговування для клієнтів.

Блокчейн:

Використання технології блокчейн може усунути необхідність проведення інвентаризації майна та звірки розрахунків за зобов'язаннями. Завдяки блокчейну, складський облік може бути проведений у режимі реального часу, дозволяючи точно відстежувати запаси, реєструвати прихід і відпуск товарів, а також моніторити рух запасів усією ланкою постачання, від початкового оприбуткування до випуску або вибуття з виробництва. Кожна транзакція, яка стосується прихід або відпуску майна, підтверджується одержувачем і записується одноразово, що унеможливорює потребу у проведенні інвентаризації. Блокчейн також надає можливість спростити взаємодію з контрагентами економічного суб'єкта, оскільки дозволяє формувати або списувати заборгованість обох сторін угоди в момент здійснення операції [3].

Суть блокчейну полягає у тому, що він сам по собі є своєрідним бухгалтером[7]. Особливість цієї системи полягає у постійній перевірці даних у блоках, сповіщенні про їх зміну та одночасному запису нової інформації в новий блок. Блокчейн є ідеальним інструментом для бухгалтерського обліку, оскільки він запобігає помилкам та зловживанням, завдяки можливості відстеження всіх фінансових операцій та отримання повідомлень про зміни в режимі реального часу. Це підтверджує необхідність впровадження блокчейну в практику бухгалтерського обліку. Ця технологія є трендом, який

поступово, але впевнено набирає популярність. Доводом її значимості є можливість використання її державними регуляторами США для контролю фінансових операцій. Таким чином, фахівці, здатні складати звітність через систему блокчейн, незабаром будуть цінуватися вище, ніж бухгалтери, що використовують традиційні програми обліку.

Розвиток інформаційних систем для складського обліку з використанням передових технологій відкриває нові перспективи для підвищення ефективності, точності та автоматизації процесів у сфері складського управління. Сучасні технології, такі як штучний інтелект, машинне навчання, Інтернет речей (IoT), хмарні технології, а також доповнена та віртуальна реальність (AR та VR), відіграють ключову роль у керуванні запасами, оптимізації планування поставок та поліпшенні якості обслуговування.

Застосування цих технологій дозволяє компаніям отримати конкурентну перевагу та ефективно реагувати на змінні вимоги ринку. У сучасному світі, де конкуренція стає все більш жорсткою, а умови бізнесу змінюються зі швидкістю світла, ефективне управління складською діяльністю стає критичним фактором для успіху в логістиці та постачанні товарів.

РОЗДІЛ 2.

ПРЕДМЕТНА ОБЛАСТЬ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ

2.1 Опис предметної області

Перш ніж вести дослідження слід знати саме визначення, що ж таке «Предметна область».

Предметна область - це матеріальна система або система, що характеризує елементи матеріального світу, інформація про яку зберігається і обробляється. Предметна область розглядається, як деяка сукупність реальних об'єктів і зв'язків між ними [4].

Аналіз предметної області складського обліку включає детальне вивчення основних аспектів та важливих факторів, які впливають на цю область діяльності. Цей опис допомагає зрозуміти особливості складського обліку, виявити потенційні проблеми та знайти шляхи для їх вирішення. Основні елементи аналізу предметної області складського обліку включають:

1. Структура складської системи: дослідження структури складської системи включає аналіз типів складів, їх розташування, розміщення та організацію простору. Вивчення структури допомагає визначити оптимальний розмір та конфігурацію складських приміщень.
2. Технологічні процеси: аналіз технологічних процесів на складі дозволяє визначити послідовність операцій, їх ефективність та можливість автоматизації. Це включає вивчення процедур приймання, зберігання, відвантаження, інвентаризації та інших операцій, що здійснюються на складі.
3. Інформаційна система: оцінка існуючої інформаційної системи складського обліку включає аналіз наявних програмних засобів,

технологій збору та обробки даних, системи ідентифікації товарів та методів звітності. Визначення потреб у вдосконаленні інформаційної системи може допомогти покращити точність, доступність та швидкість обробки інформації.

4. Законодавчі та регуляторні вимоги: аналіз законодавчих та регуляторних вимог, що стосуються складського обліку, дозволяє забезпечити дотримання вимог щодо безпеки, охорони праці, зберігання товарів та обліку фінансових операцій.

5. Проблеми та виклики: аналіз предметної області також включає виявлення і опис проблем та викликів, з якими можуть стикатися в сучасному складському обліку. Це можуть бути проблеми з організацією простору, ефективністю процесів, недостатня точність обліку запасів та інші.

Для максимально точного аналізу предметної області, потрібно чітко розмежувати досліджувані об'єкти між собою. Головним об'єктом на складі є товар, а для аналізу предметної області визначено такі сутності: користувач, продукт, виробник, замовлення та клієнти.

Сутність користувача має ряд можливостей, зокрема:

- Перегляд наявних товарів.
- Зміна ціни та кількості товарів.
- Додавання та видалення запчастин.
- Перегляд клієнтів.
- Редагування, додавання та видалення даних про клієнтів.
- Перегляд замовлень.
- Створення нових замовлень.
- Редагування вже створених замовлень.
- Зміна стану замовлень.

Сутність продукту містить інформацію про товари:

- Ціна.
- Виробник.
- Назва.
- Артикул.
- Кількість.

Аналізуючи вимоги до програмного забезпечення для обліку товарів, слід виявити функціональні вимоги. Функціональні вимоги – це вимоги до програмного забезпечення, для опису поведінки системи, її особливості та функції [5].

Функціональні вимоги до програмного забезпечення для обліку товарів включають:

- Можливість перегляду виробників.
- Можливість додавання та видалення виробників.
- Можливість перегляду клієнтів.
- Можливість редагування, додавання та видалення клієнтів.
- Можливість перегляду товарів.
- Можливість зміни ціни та кількості товарів.
- Можливість додавання та видалення товарів.
- Можливість перегляду замовлень.
- Можливість додавання та редагування замовлень.
- Можливість зміни стану замовлення.

На основі цих даних, виділимо варіанти використання системи. Перелік варіантів використання та їх короткий опис показано в таблиці (табл 2.2).

Таблиця 2.2 Використання системи

Варіант використання	Опис
Перегляд наявних товарів	Користувач може переглядати список товарів, доступних на складі.
Зміна ціни та кількості	Користувач може змінювати ціну та кількість товару.
Додавання та видалення	Користувач може додавати та видаляти запчастини.
Перегляд клієнтів	Користувач може переглядати інформацію про клієнтів.
Редагування клієнтів	Користувач може редагувати, додавати та видаляти дані про клієнта.
Перегляд замовлень	Користувач може переглядати список замовлень.
Створення замовлень	Користувач може створювати нові замовлення.
Редагування замовлень	Користувач може редагувати вже створені замовлення.
Зміна стану замовлень	Користувач може змінювати стан замовлень.
Перегляд виробників	Користувач може переглядати список виробників.
Додавання та видалення	Користувач може додавати та видаляти виробників.
Можливість зміни ціни	Користувач може змінювати ціну товару.

Можливість додавання та видалення товарів	Користувач може додавати та видаляти товари.
---	--

Розуміння предметної області складського обліку є важливим кроком при розробці стратегії управління складською діяльністю та вдосконаленні складських процесів. Він допомагає зрозуміти потреби, визначити пріоритети та знайти оптимальні рішення для покращення ефективності та продуктивності складської діяльності.

2.2 Огляд існуючих програмних систем управління складом в Україні

Одним зі способів впровадження автоматизованої системи управління складом є використання готових комерційних програмних продуктів. Ці рішення вже мають певну функціональність, яка може задовольнити потреби більшості малих і середніх підприємств. Однак такі програми також можуть бути налаштовані під конкретні потреби підприємства.

Іншим підходом є розробка власної інформаційної системи, спеціально адаптованої до потреб конкретного підприємства. Хоча цей підхід може бути більш часо- і ресурсомістким, він надає більшу гнучкість і можливість налаштування системи під індивідуальні потреби підприємства.

Ми будемо досліджувати перший спосіб. [9-11]

Україна є країною, де постійно зростає важливість ефективного управління складськими процесами для підприємств різних галузей. На сучасному ринку існує кілька програмних систем, спеціалізованих на управлінні складом, які пропонують широкий спектр функціональних можливостей для оптимізації цих процесів.

Розглянемо кілька з них:

1. Перша програмна система, яку варто відзначити, - "1С: Підприємство".

Ця система є однією з найпопулярніших в Україні і надає широкий функціонал для управління складом. Основним призначенням "1С: Підприємство" є автоматизація бухгалтерського обліку, фінансового управління, звітності та аналітики для підприємств. Програма дозволяє вести облік фінансових операцій, управляти складським обліком, контролювати потоки грошей, формувати фінансову звітність, розраховувати заробітну плату, вести документообіг та інше.

"1С: Підприємство" має модульну структуру, що дозволяє користувачам вибирати лише ті функції, які необхідні для конкретного підприємства або галузі. Програма має різні конфігурації, спеціально розроблені для різних видів діяльності, таких як торгівля, виробництво, послуги, громадські організації та багато інших.

"1С: Підприємство" також підтримує велику кількість податкових і бухгалтерських стандартів, що дозволяє використовувати програму в різних країнах і відповідати вимогам законодавства.

Програма має зручний інтерфейс, широкі можливості з налаштування та інтеграцію з іншими системами. Вона здатна працювати як в локальній мережі підприємства, так і в хмарному середовищі, що дозволяє більш гнучко використовувати програму залежно від потреб користувача.

2. Друга система, яку варто вказати, - "ERPNext". ERPNext пропонує інтегрований підхід до управління, об'єднуючи різні функції, такі як бухгалтерія, фінанси, керування продажами, управління складом, управління проектами, управління виробництвом, керування кадрами та

інше. Вона дає змогу підприємствам вести комплексний облік, спрощувати процеси та поліпшувати ефективність роботи[6].

Основними принципами ERPNext є модульність, гнучкість та доступність. Вона має модульну структуру, яка дозволяє користувачам вибирати тільки необхідні функціональність і розширювати систему за потребою. Крім того, ERPNext базується на відкритих стандартах і має відкритий вихідний код, що дає можливість користувачам вносити зміни, розробляти додаткові функції та адаптувати систему до своїх потреб.

Основні функціональні можливості ERPNext включають:

- Бухгалтерію та фінансове управління: ведення бухгалтерського обліку, контроль фінансових операцій, формування звітності.
- Управління продажами та закупівлями: обробка замовлень клієнтів, керування постачальниками, управління запасами.
- Управління складом: контроль запасів, складського обліку, управління постачанням.
- Управління проектами: планування проектів, відстеження витрат, управління завданнями.
- Управління виробництвом: планування виробництва, контроль якості, відстеження операцій.
- Управління кадрами: ведення персональних даних співробітників, облік заробітної плати, керування відпустками.

3. "SAP Business One". Ця популярна програмна система має широкий функціонал для управління складом. SAP Business One є інтегрованою системою управління підприємством, розробленою компанією SAP SE. Вона спрямована на задоволення потреб малого та середнього бізнесу, надаючи рішення для автоматизації бухгалтерського обліку, фінансів, управління складом, продажів, закупівель, виробництва, керування

взаєминами з клієнтами та багато інших аспектів діяльності підприємства.

Основними перевагами SAP Business One є:

- Інтегрована функціональність: SAP Business One об'єднує різні аспекти бізнесу в одній системі, що дозволяє забезпечити цілісність даних та ефективність роботи.

Бухгалтерський облік та фінансове управління: програма надає інструменти для ведення бухгалтерського обліку, формування фінансових звітів, керування бюджетом, управління грошовим потоком тощо.

- Управління продажами та клієнтськими взаєминами: SAP Business One дозволяє керувати замовленнями клієнтів, контролювати продажі, відстежувати стан запасів, управляти ціновою політикою та стимулювати збут.
- Управління закупівлями та складським обліком: програма дозволяє керувати процесом закупівель, оптимізувати запаси, виконувати планування потреб, вести облік поставчань та контролювати витрати.
- Управління виробництвом: SAP Business One надає функціонал для планування виробництва, контролю якості, відстеження операцій, керування матеріалами та складським обліком.
- Аналітика та звітність: система забезпечує можливості аналізу даних, створення звітів та панелей управління для прийняття обґрунтованих рішень.
- Гнучкість та розширюваність: SAP Business One може бути налаштована під потреби конкретного підприємства і розширена за допомогою додаткових модулів та інтеграцій.

4. "Warehouse Management System" (WMS). Warehouse Management System (WMS) є програмним забезпеченням, спеціально розробленим для ефективного управління складською логістикою та оптимізації операцій в складському просторі. WMS дозволяє автоматизувати процеси, пов'язані з прийманням, зберіганням, переміщенням та відвантаженням товарів на складі.

5. Основна мета WMS полягає в досягненні ефективності та точності управління запасами, зниженні витрат та підвищенні продуктивності на складі. Він надає розширені функції та інструменти для керування складськими операціями, такими як ідентифікація та зберігання товарів, виконання замовлень, інвентаризація, відстеження руху товарів та інше. Деякі основні функції та можливості WMS включають:

- Планування та розміщення запасів: WMS допомагає оптимізувати розташування товарів на складі, враховуючи їх характеристики та рівень попиту. Він дозволяє визначати оптимальні місця для зберігання товарів з метою забезпечення ефективності використання простору та швидкого доступу до товарів.
- Приймання товарів: WMS дозволяє автоматизувати процеси приймання товарів на склад, включаючи перевірку кількості, якості та ідентифікацію товарів. Він надає можливість сканувати штрих-коди або використовувати RFID-технології для швидкого та точного визначення товарів.
- Управління переміщенням товарів: WMS забезпечує ефективне переміщення товарів на складі. Він автоматизує процеси внутрішнього переміщення, розміщення товарів на полицях, консолідацію та розконсолідацію товарів, що дозволяє знизити час та зусилля, пов'язані з переміщенням товарів.

- **Управління замовленнями та відвантаженням:** WMS надає можливість ефективно обробляти та виконувати замовлення клієнтів. Він автоматизує процеси підготовки замовлень, підбору товарів, упаковки та відвантаження, що сприяє підвищенню швидкості та точності виконання замовлень.
- **Відстеження запасів та інвентаризація:** WMS надає засоби для точного відстеження руху та кількості товарів на складі. Він дозволяє проводити регулярну інвентаризацію, оновлювати інформацію про запаси в режимі реального часу та забезпечувати актуальну інформацію про наявність товарів.
- **Звітність та аналітика:** WMS надає розширені можливості для створення звітів, аналізу даних та моніторингу продуктивності складських операцій. Він дозволяє отримувати аналітичну інформацію про запаси, виконання замовлень, ефективність простору та інші ключові показники.

6. **"BAS Комплексне управління підприємством".** "BAS Комплексне управління підприємством" може бути налаштована під потреби конкретного підприємства, враховуючи його особливості та галузеві вимоги. Вона також надає інтеграцію з іншими системами та модулями для забезпечення цілісності даних та ефективної взаємодії між різними функціональними областями підприємства.

Основні характеристики та можливості BAS Комплексного управління підприємством включають:

- **Фінансове управління:** система дозволяє вести облік фінансових операцій, складати бюджет, створювати фінансові звіти,

виконувати аналіз фінансових показників та контролювати грошові потоки.

- Бухгалтерський облік: BAS Комплексне управління підприємством надає можливості вести облік доходів і витрат, опрацьовувати податкову звітність, формувати звіти про фінансовий стан підприємства.
- Управління продажами та клієнтськими взаєминами: система дозволяє керувати замовленнями клієнтів, обробляти запити на пропозиції, відстежувати стан реалізації замовлень, забезпечувати зв'язок з клієнтами та керувати взаємодією з ними.
- Управління закупівлями та постачаннями: система BAS дозволяє планувати та контролювати процес закупівель, вести облік постачальників, контролювати запаси та виконувати аналіз ефективності закупівель.
- Управління виробництвом: програмне рішення BAS надає функціонал для планування та керування виробництвом, контролю за виробничими процесами, управління ресурсами, планування використання обладнання та оптимізації виробничих операцій.
- Управління персоналом: система BAS допомагає керувати персоналом, включаючи процеси найму, обліку робочого часу, оплати праці, управління навчанням та розвитком співробітників.

- Аналітика та звітність: BAS Комплексне управління підприємством надає можливості створення звітів, аналізу даних та моніторингу ключових показників продуктивності та ефективності підприємства.

За цими критеріями проаналізуємо кожен з систем і складемо таблицю:

Таблиця 2.1 Автоматизовані системи управління складом

Система	Переваги	Недоліки
"1С: Підприємство"	- Широкий функціонал для управління складом	- Потребує адаптації до конкретних потреб підприємства
	- Можливість адаптації до різних галузей та розмірів підприємств	- Обмежені можливості звітності та аналітики
		- Обмежена інтеграція з іншими системами
"ERPNext"	- Відкрите програмне забезпечення з можливостями управління складом	- Потребує налагодження та налаштування для використання
	- Модулі звітності і аналітики для стратегічного управління складом	- Менша популярність та менший розповсюджений функціонал порівняно з іншими системами

		- Менша інтеграція з іншими системами
"SAP Business One"	- Широкий функціонал для управління складом	- Висока вартість та складність впровадження
	- Інтеграція з іншими системами для забезпечення потоку даних	- Високі вимоги до апаратних ресурсів та інфраструктури
	- Глобальна популярність та підтримка	
"Warehouse Management System" (WMS)	- Спеціалізація на управлінні складом	- Може бути менш адаптована до інших аспектів управління підприємством
	- Оптимізація процесів у складському господарстві	- Вимагає високої експертизи та налаштування для ефективного використання
	- Висока точність і швидкість обробки замовлень	
"BAS Комплексне управління"	- Комплексний підхід до управління різними аспектами	- Може бути перенавантаженою зайвим функціоналом

підприємством"	підприємства, включаючи управління складом	
	- Функціонал для контролю залишків, планування закупівель, керування постачанням тощо	- Обмежені можливості звітності та аналітики
	- Інтеграція з іншими системами	

Зверніть увагу, що це лише загальний аналіз, і вибір системи для конкретного підприємства повинен ґрунтуватися на його конкретних потребах та умовах. Ці системи є лише кількома з численних пропозицій на ринку програмного забезпечення для управління складом.

2.3 Моделювання процесу складського обліку

Моделювання процесу складського обліку - це процес створення віртуальної або концептуальної моделі, яка описує структуру, процедури та потоки руху товарів на складі з метою аналізу, оптимізації та управління складською діяльністю.

Основна мета моделювання процесу складського обліку полягає в тому, щоб дослідити та оптимізувати різні аспекти роботи складу. Це дозволяє аналізувати ефективність розміщення товарів на складі, планувати і прогнозувати запаси, визначати оптимальні розміри складських приміщень,

оптимізувати процеси прийому та відвантаження товарів, а також виявляти й усувати можливі проблеми та ризики, пов'язані з управлінням складською діяльністю.

Концепція моделювання процесу складського обліку базується на використанні інструментів і методів математичного моделювання, комп'ютерного моделювання та симуляції. Це дозволяє створити абстрактну модель складського процесу, яка допомагає аналізувати його різні аспекти та робити прогнози результатів в різних сценаріях. Концепція також включає в себе використання аналітичних інструментів, систем управління складськими запасами та алгоритмів оптимізації для підтримки процесу прийняття рішень та оптимального управління складською діяльністю.

Концептуальні засади:

Концептуальні засади моделювання процесу складського обліку включають в себе:

1. Визначення цілей: Перший крок у моделюванні процесу складського обліку - визначення чітких цілей. Цілі можуть включати зменшення витрат, підвищення продуктивності, оптимізацію запасів, поліпшення обслуговування клієнтів та інше. Цілі мають бути конкретними, вимірюваними та досяжними.
2. Виявлення потреб: Аналізуючи складську діяльність, необхідно виявити потреби та проблеми, з якими стикається компанія. Це можуть бути надмірні запаси, проблеми з розміщенням товарів, затримки в обробці замовлень або недостатня ефективність операцій. Розуміння цих потреб допоможе спрямувати моделювання на конкретні аспекти, що потребують удосконалення.
3. Вибір методів моделювання: На основі потреб і обсягу дослідження слід вибрати найбільш підходящі методи моделювання.

Математичне моделювання використовує математичні рівняння та алгоритми для опису різних аспектів складського обліку, таких як оптимальне розміщення товарів або розклад замовлень. Комп'ютерне моделювання передбачає використання спеціального програмного забезпечення для побудови віртуальної моделі складської системи та симуляції різних сценаріїв.

4. Збір та аналіз даних: Для побудови моделі потрібно зібрати відповідні дані про складську діяльність. Це можуть бути дані про запаси, транзакції, обсяги продажів, витрати, розміщення товарів тощо. Збирання даних може вимагати використання спеціальних систем складського обліку або автоматизованих систем управління запасами.

5. Розробка та тестування моделі: На основі зібраних даних розробляється концептуальна модель складського обліку. Це може бути математична модель з використанням різних алгоритмів, комп'ютерна модель з віртуальним представленням складської системи або симуляційна модель для проведення експериментів та тестування різних сценаріїв. Модель повинна бути перевірена та протестована для переконання в її ефективності та точності.

6. Аналіз результатів та прийняття рішень: Отримані результати моделювання аналізуються та інтерпретуються з метою прийняття обґрунтованих рішень. Наприклад, на основі моделі можна розробити оптимальні стратегії управління запасами, визначити оптимальні місця для розміщення товарів або планувати процеси прийому та відвантаження. Результати моделювання допомагають підвищити ефективність та продуктивність складської діяльності, знизити витрати та покращити обслуговування клієнтів.

Ці етапи моделювання процесу складського обліку допомагають підприємствам аналізувати та оптимізувати різні аспекти складської діяльності. Вони сприяють прийняттю обґрунтованих рішень та досягненню покращень у управлінні запасами, оптимізації процесів та поліпшенні ефективності складського обліку.

РОЗДІЛ 3.

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АВТОМАТИЗАЦІЇ

3.1 Функціональна модель складського обліку

Функціональна модель складського обліку - це модель, яка описує функції та процеси, що відбуваються в системі управління складом підприємства. Вона визначає основні функції, взаємозв'язки між ними та послідовність дій, які потрібно виконати для ефективного управління складськими процесами.

Суть функціональної моделі складського обліку полягає в тому, щоб систематизувати та уявити комплексність складської діяльності підприємства через ідентифікацію, класифікацію та послідовне розташування функцій та процесів, які відбуваються на складі. Вона допомагає зрозуміти, які конкретні функції необхідні для ефективного управління запасами, обліку, зберігання та руху товарів.

Концепція функціональної моделі складського обліку базується на розумінні основних процесів та взаємозв'язків між ними, а також враховує специфіку та потреби конкретного підприємства. Вона дозволяє відобразити

потік матеріальних та інформаційних ресурсів на складі, сприяє забезпеченню контролю, ефективності та оптимізації складських процесів.

Функціональна модель складського обліку включає (рис. 3):

1. Приймання товарів:

- Приймання поставок: Отримання товарів від постачальників.
- Перевірка якості: Перевірка якості та відповідності товарів замовленню.
- Внесення до системи: Реєстрація отриманих товарів в системі складського обліку.

2. Зберігання товарів:

- Розміщення на складі: Визначення місць розміщення товарів на складі.
- Ідентифікація та маркування: Присвоєння ідентифікаторів та маркування товарів для їхнього легкого виявлення та ідентифікації.
- Контроль за запасами: Ведення обліку залишків товарів на складі.

3. Комплектація та відбірка товарів:

- Комплектація замовлень: Збір товарів для формування замовлень клієнтів.
- Відбірка товарів для внутрішніх потреб: Збір товарів для внутрішніх потреб підприємства (наприклад, для виробництва).
- Контроль якості: Перевірка якості товарів перед їхнім відправленням.

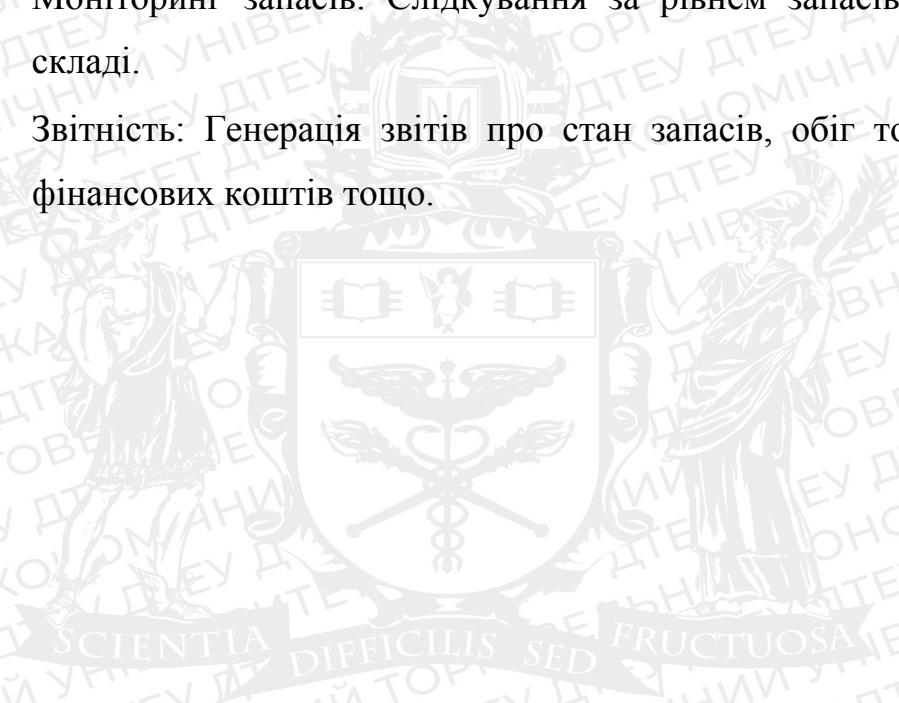
4. Відвантаження товарів:

- Підготовка до відвантаження: Підготовка товарів для відправлення клієнтам.

- Оформлення документів: Оформлення необхідних документів (накладних, рахунків-фактур тощо).
- Відправлення: Відправлення товарів клієнтам або до інших підрозділів підприємства.

5. Моніторинг та звітність:

- Моніторинг запасів: Слідкування за рівнем запасів товарів на складі.
- Звітність: Генерація звітів про стан запасів, обіг товарів, рухи фінансових коштів тощо.



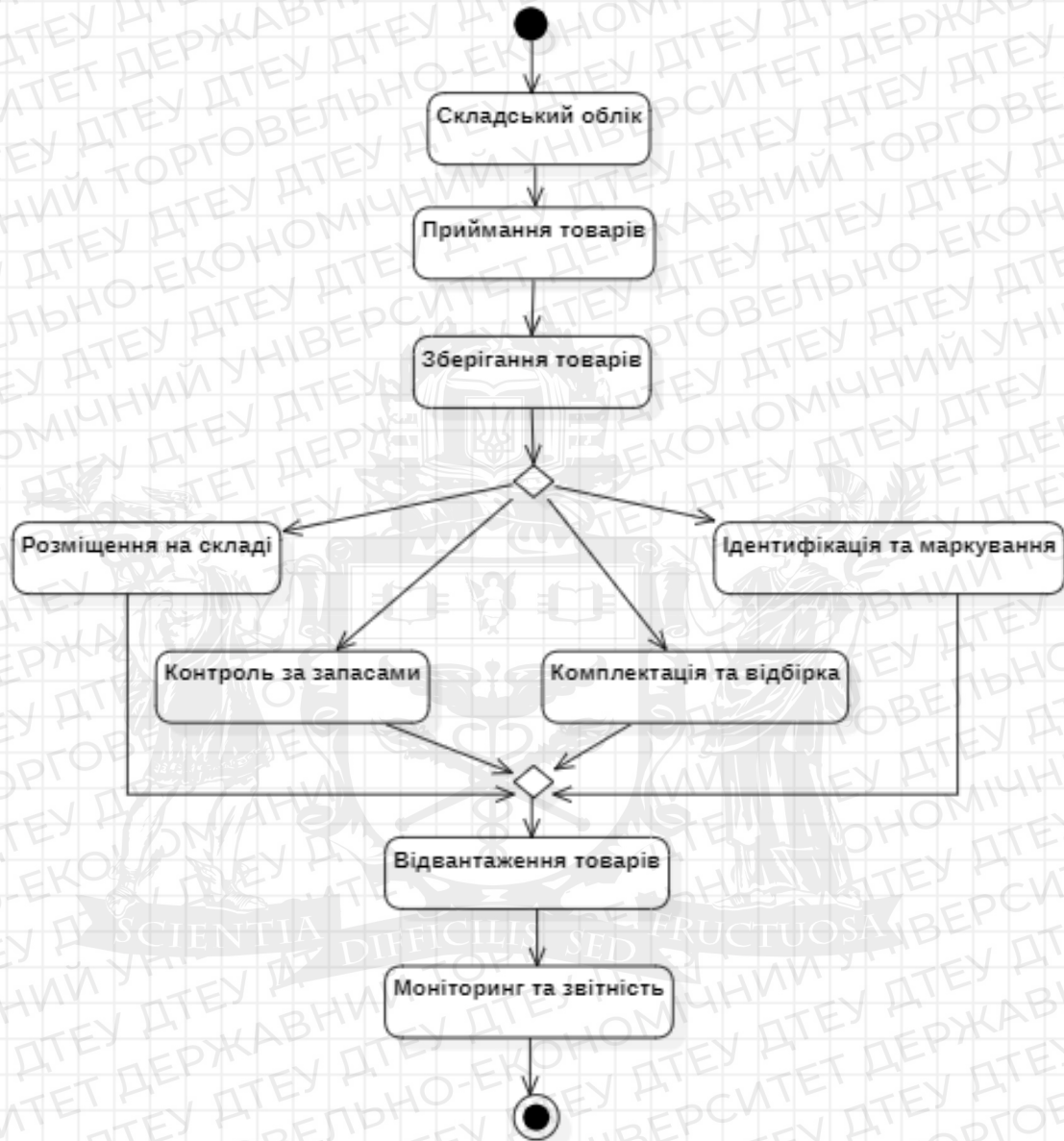


Рис. 3. Функціональна модель

Для побудови функціональної моделі, я буду використовував моделювання UML. В середовищі StarUML.

StarUML - це популярний інструмент для моделювання, що підтримує роботу з мовою UML. Він надає зручне графічне середовище для створення

та редагування UML-моделей, яке дозволяє вам візуалізувати структуру, поведінку та взаємодію програмних систем.

UML (Unified Modeling Language) - це стандартизована мова моделювання, яка використовується для візуалізації, специфікації, конструювання та документування структури та поведінки програмних систем. Вона дозволяє розробникам, аналітикам та інженерам моделювати системи за допомогою стандартних графічних символів та нотацій. [2]

UML була створена з метою стандартизації процесу моделювання програмних систем та забезпечення уніфікованого способу вираження концепцій, структур, взаємодій та поведінки системи. Вона використовується для розробки програмного забезпечення, проектування баз даних, моделювання бізнес-процесів та багатьох інших областей.

UML надає широкий набір графічних елементів та нотацій для моделювання різних аспектів системи, включаючи структуру (класи, об'єкти, компоненти), поведінку (діаграми послідовностей, діаграми активностей), взаємодію (діаграми прецедентів, діаграми комунікацій) та багато іншого.

Основні переваги використання UML включають:

- Стандартизована нотація для моделювання систем.
- Зрозумілість та легкість сприйняття графічних моделей.
- Можливість комунікації між різними зацікавленими сторонами проекту за допомогою єдиного мовного засобу.
- Підтримка аналізу, проектування та документування програмних систем на різних етапах розробки.

3.2 Архітектура Web-додатку

Створення архітектури додатку - це комплекс заходів, спрямованих на чітке визначення того, як буде побудована система.

Архітектура веб-додатку - це план, який описує всі програмні компоненти (базу даних, додатки, проміжне програмне забезпечення тощо) і те, як вони взаємодіють. Архітектура визначає, як дані передаються через HTTP, щоб клієнтські та внутрішні сервери могли їх розуміти. Вона гарантує, що всі запити користувачів підтримуються достовірними даними, що записи створюються та управляються, а доступ та автентифікація базуються на авторизації.

Загалом, архітектура веб-додатку включає в себе три основні компоненти:

- Веб - Браузер. Клієнтський компонент або фронт-енд компонент - це основний компонент, який взаємодіє з користувачем, отримує вхідні дані, керує логікою представлення і контролює взаємодію між користувачем і додатком. Він також перевіряє дані, введені користувачем, коли це необхідно.
- Веб сервер. внутрішній або серверний компонент, який обробляє бізнес-логіку та запити користувачів, надсилає запити до відповідних компонентів і керує всіма операціями додатку. Він може ініціювати та контролювати запити від різних клієнтів.
- Сервер База даних. Надає дані, необхідні додатку. Виконує завдання, пов'язані з даними. В ієрархічній архітектурі сервер бази даних може використовувати збережені процедури для управління бізнес-логікою.

Архітектуру Web-додатку на Django показано на Рис 3.1.

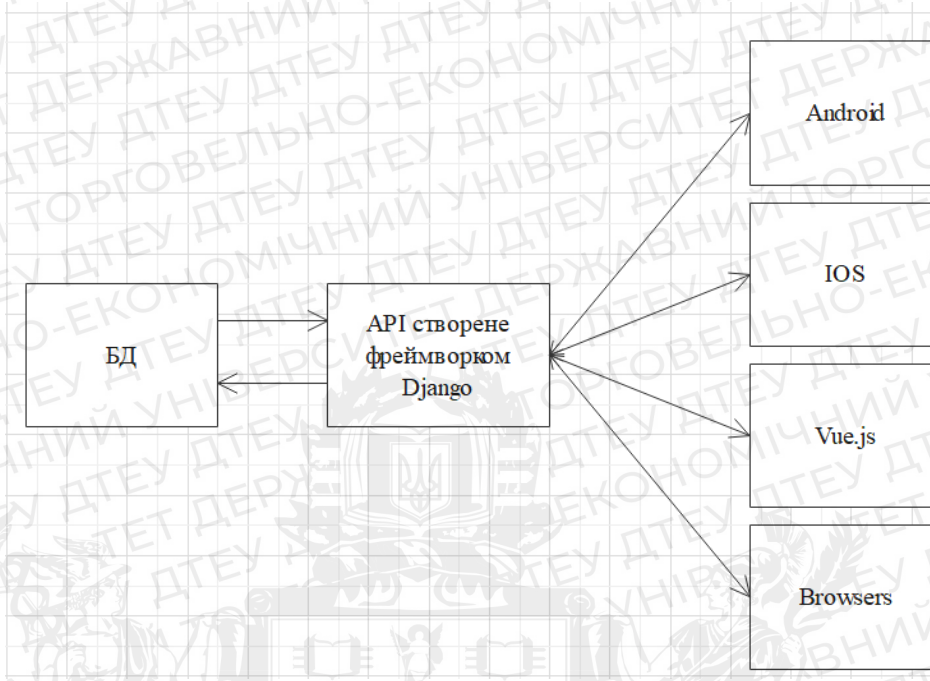


Рис. 3.1. Архітектура Web-додатку на Django

Структура Web-додатку, що використовує Django є схожою на звичайну трьохрівневу структуру, але має деяку відмінність. Завдяки DRF, або Django REST Framework Один сервер може працювати з декількома оболонками, наприклад з пристроями на різних операційних системах, браузерах а також надавати послуги іншим серверам. Також така структура додає більшої захищеності бази даних, тому, що користувач не взаємодіє з нею напряму [13].

3.3 Програмна реалізація

Для реалізації всіх функцій складу підприємства по виробництву вікон було вирішено використовувати HTML, CSS, Python і його фреймворк Django у середовищі розробки PyCharm. Для зв'язку з базою даних

використовувались вбудовані можливості Django, чого достатньо для реалізації даного програмного застосунку, а також використовувався SQLiteStudio для вирішення помилок міграції при розробці. Така реалізація дозволяє нам доволі легко працювати над реалізацією Web-додатка.

Надалі я опишу переваги і недоліки мови програмування Python:

Python - одна з найбільш динамічніших мов програмування, що за індексом PYPL є найпопулярнішою мовою по кількості пошуків туторіалів по ній, а саме 31%, що в 2 рази більше, ніж у 2-го місця, мови програмування Java;

Python - є доволі лаконічною мовою з простим синтаксисом, що схожий на природну мову. Це забезпечує читаємість коду, що в свою чергу позитивно впливає на можливості підтримки і інспекції проєкту;

Кількість і різноманітність бібліотек, що використовуються для реалізації різних програмних продуктів забезпечує Python використання у всіх відомих сферах діяльності і розробки програмних застосунків;

Розробка на Python є дуже швидкою, що дозволяє виводити на ринок конкурентоздатні продукти швидше за інших;

Python підтримує асинхронність, що дозволяє працювати швидше і обробляти інші запити під час очікування відповіді від сервера;

Python може бути інтегрованим з іншими мовами, такими як JavaScript, C++, PHP, Java, та інші;

Також до переваг можна віднести те, що це динамічна мова програмування.

Хоч Python має стільки переваг, він має певні недоліки, що є незначними для реалізації нашого проєкту, а саме:

Python є повільнішим, якщо його порівнювати з іншими мовами програмування, як C++ і Java;

У динамічних мовах програмування є проблема з важким пошуком помилок, хоч в Python 3.5 ця проблема частково вирішується, але все ж таки на великих проєктах можуть бути певні проблеми.

Як ми бачимо проблеми, що є у мови програмування Python є несуттєвими для нас, тому вибір можна вважати доречним. Також далі я пропоную розглянути переваги і недоліки фреймворка Django.

Переваги:

Швидка розробка від ідеї до готового продукту;

Універсальність і гнучкість;

Адмін панель Django є однією із головних переваг, що забезпечує його використання не тільки для розробки Web-застосунків;

Модульна система;

Архітектура, що реалізує паттерн MVC, що дозволяє розробляти компоненти паралельно і змінювати їх окремо;

Django безпечний;

Django підтримує API;

Докладна документація;

Недоліки:

Django є сильно перегружений функціоналом, тому новачкам може бути важко розібратись а також більшість функцій можуть бути просто непотрібними для легких проєктів [12].

Django використовує для візуального відображення HTML і CSS, що також є перевагою тому, що це є доволі зручно. Всі сторінки і таблиці баз даних реалізовувались вже під час розробки.

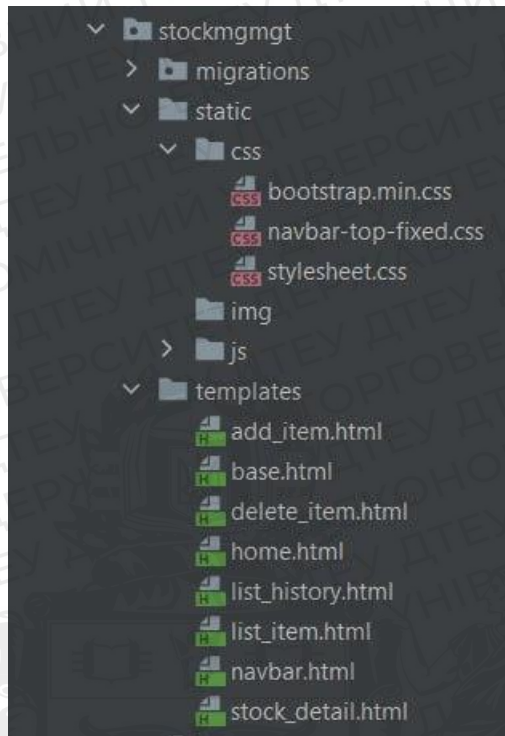


Рис. 3.2 Структура файлів візуальної частини проєкту

Web-додаток складається з 6 сторінок і файлу `base.html`, що є шаблоном для всіх інших. Шаблон - це текстовий файл. Він може бути створений у будь-якому текстовому форматі (HTML, XML, CSV тощо). Шаблони містять змінні, які замінюються значеннями при обробці шаблону, і теги, які керують логікою роботи шаблону. Нижче наведено мінімальний шаблон для ілюстрації деякої базової інформації. Кожен елемент описано далі в цьому документі.

З назв зрозуміло, для чого використовується та чи інша сторінка.

```

class Stock(models.Model):
    category = models.ForeignKey(Category, on_delete=models.CASCADE, blank=False)
    item_name = models.CharField(max_length=50, blank=True, null=True)
    quantity = models.FloatField(default='0.0', blank=True, null=True)
    receive_quantity = models.IntegerField(default='0', blank=True, null=True)
    receive_by = models.CharField(max_length=50, blank=True, null=True)
    issue_quantity = models.IntegerField(default='0', blank=True, null=True)
    issue_by = models.CharField(max_length=50, blank=True, null=True)
    issue_to = models.CharField(max_length=50, blank=True, null=True)
    phone_number = models.CharField(max_length=50, blank=True, null=True)
    created_by = models.CharField(max_length=50, blank=True, null=True)
    reorder_level = models.IntegerField(default='0', blank=True, null=True)
    last_updated = models.DateTimeField(auto_now_add=False, auto_now=True)
    color = models.ForeignKey(Colors, on_delete=models.CASCADE, blank=False)

    def __str__(self):
        return self.item_name

```

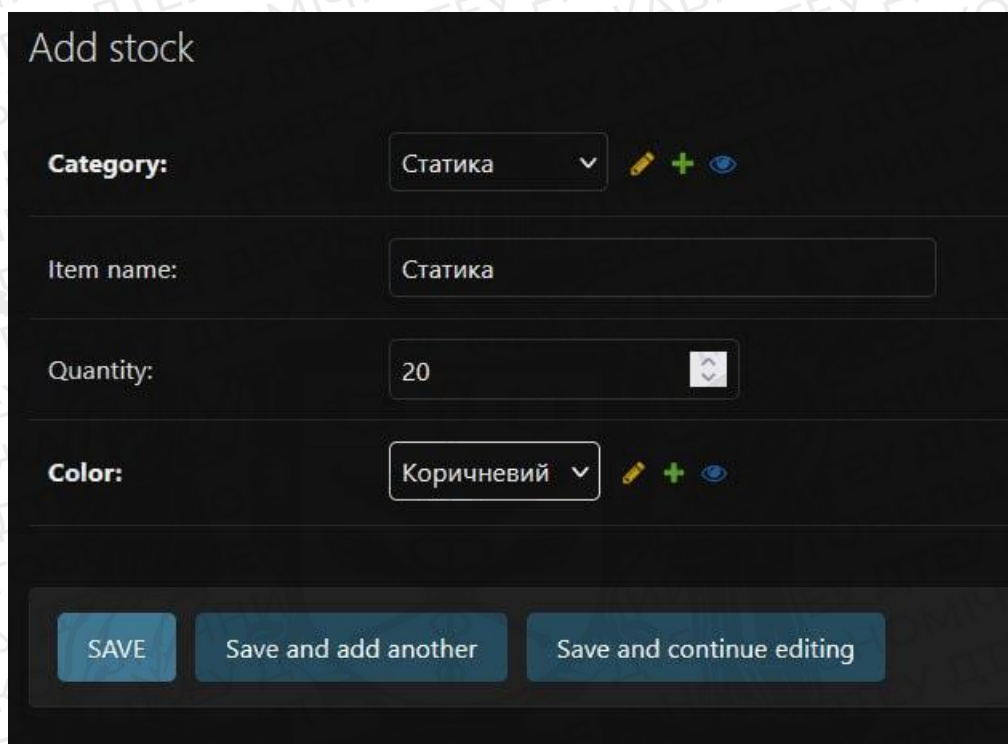
Рис. 3.3. Таблиця Stock в models.py

Доступ до бази даних відбувається через запити, що отримує сервер. На рис. 3.3. можна побачити структуру основної таблиці Stock.

CATEGORY	ITEM NAME	QUANTITY	COLOR
Імпорт	Імпорт	6	Метал
Статика	WDS 7 Series	5	Білий
Дверна рама	WDS 7 Series	12	Чорний
Рама	WDS 7 Series	8	Білий
Створка	WDS 7 Series	8	Білий
Дверна рама	WDS SL 60	2	Коричневий
Ручки	Ручка ракушка	15	Сірий
Ручки	Ручка віконна Clío WHL-022	20	Чорний
Ручки	Ручка віконна Kronos WH	6	Білий
Фурнітура	Запорна пластина KBE 58/13	8	Метал
Фурнітура	Запорна пластина Rehau	6	Метал

Рис 3.4. Вигляд таблиці Stock через адмінську панель

Для рядків, що будуть приймати в себе текстові значення було прийнято рішення використовувати тип даних «varchar» з обмеженням кількості введених символів.



The image shows a dark-themed admin panel form titled "Add stock". It contains the following fields and controls:

- Category:** A dropdown menu with "Статика" selected, accompanied by edit, add, and view icons.
- Item name:** A text input field containing "Статика".
- Quantity:** A numeric input field with "20" and a spinner control.
- Color:** A dropdown menu with "Коричневий" selected, accompanied by edit, add, and view icons.

At the bottom of the form are three buttons: "SAVE", "Save and add another", and "Save and continue editing".

Рис. 3.5. Додавання нових записів через адмін-панель

Додавати, змінювати і видаляти записи можна як через панель адміна, що показано на рис. 3.5, так і через спеціальні сторінки, що буде показано далі.

3.4 Функціональні можливості Web-додатку

При вході на сайт першим ділом користувач бачить базову сторінку. На ній знаходиться навігація на інші сторінки, але на них не можна потрапити не пройшовши реєстрацію. Головну сторінку відображено на рис. 3.6.

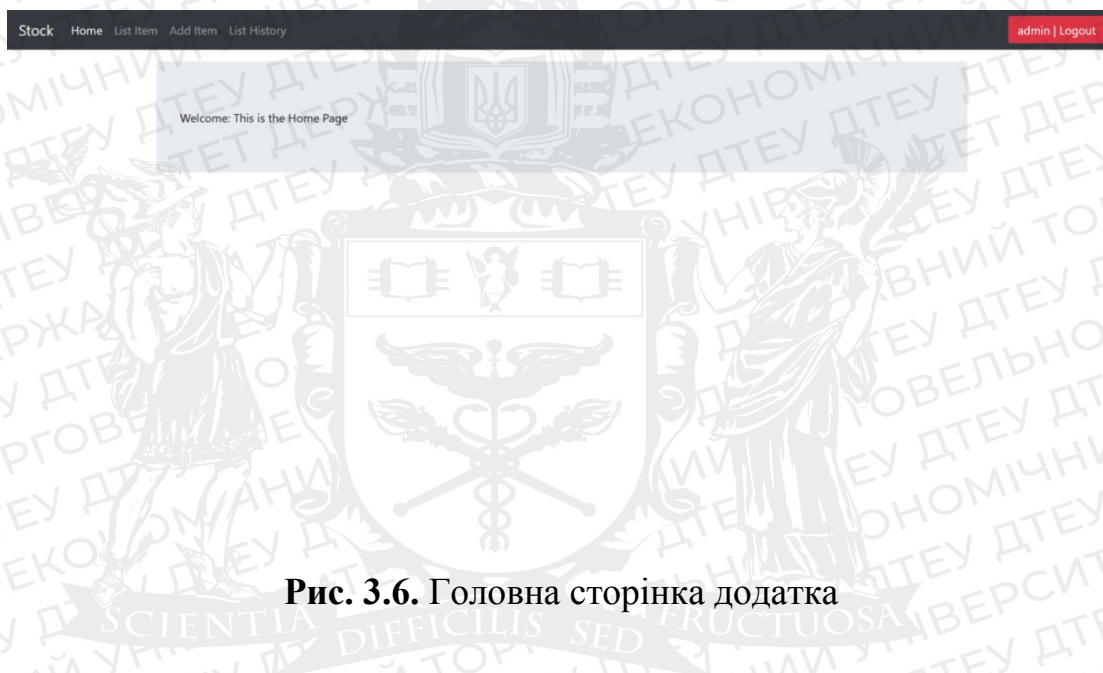


Рис. 3.6. Головна сторінка додатка

Після авторизації користувач може зайти на сторінку із списком всіх наявних на складі товарів і переглянути їх (рис 3.7). Сайт відображає категорію товару, його назву, кількість, що є на складі, кількість товару на складі, після якої потрібно докуповувати певний товар, колір, дату останньої дії з товаром і кнопку видалити.

List of Items

Category:

Item name:

Color:

Export to csv:

COUNT	CATEGORY	ITEM NAME	QUANTITY IN STORE	REORDER LEVEL	COLOR	LAST UPDATED	DELETE
1	Фурнітура	Запорна пластина Rehau	6	0	Метал	May 21, 2023, 1:48 p.m.	Delete
2	Фурнітура	Запорна пластина KBE 58/13	8	0	Метал	May 21, 2023, 1:49 p.m.	Delete
3	Ручки	Ручка віконна Kropos	6	0	Білий	May 21, 2023, 1:50	Delete

Рис. 3.7. Сторінка із списком всіх наявних матеріалів

На цій сторінці реалізований пошук товарів по категорії, назві і кольору, як зображено на рис. 3.8.

List of Items

Category: Ручки

Item name: WH

Color:

Export to csv:

COUNT	CATEGORY	ITEM NAME	QUANTITY IN STORE	REORDER LEVEL	COLOR	LAST UPDATED	DELETE
1	Ручки	Ручка віконна Kropos WH	6	0	Білий	May 21, 2023, 1:50 p.m.	Delete
2	Ручки	Ручка віконна Сіо WH-022	20	0	Чорний	May 21, 2023, 1:51 p.m.	Delete

Рис. 3.8. Пошук по сторінці

Якщо користувач натисне в навігаційній панелі на кнопку Add Item, то його перекине на сторінку додавання нового товару, що по функціоналу є

аналогом подібної форми в адмінській панелі та зображена на рис 3.9. Також таке саме меню, але з можливістю редагування даних про товар можна відкрити, якщо натиснути на назву товару.

Stock Home List Item Add Item List History admin | Logout

Category: Ручки

Item name: Ручка віконна Kronos WH

Quantity: 6

Color: Білий

Save

Рис. 3.9. Сторінка додавання або зміни параметрів товару

Якщо натиснути на кількість товару, то відкривається меню деталей про товар та дій з ним. В цьому меню є 2 кнопки, а саме перемістити товар до і додати товар до складу.

Stock Home List Item Add Item List History admin | Logout

ISSUE THIS ITEM RECEIVE THIS ITEM

ITEM NAME	QUANTITY IN STORE	LAST UPDATED	REORDER LEVEL
Ручка віконна Kronos WH	6	May 21, 2023, 1:50 p.m.	0

Рис. 3.10. Сторінка деталей про товар і дій з ним

Якщо на попередній сторінці натиснути на червону кнопку, то відкриється сторінка, що зображена на рис. 3.11., в якій можна вказати кількість товару і куди він переміщується. Кількість, що ми вказали буде списана із складу, якщо вона там є.

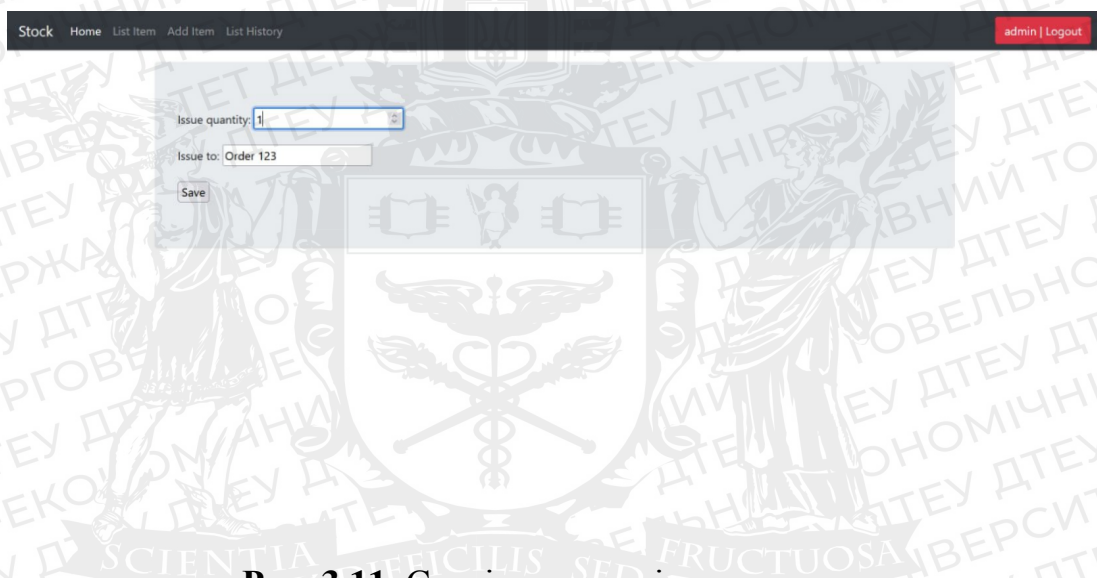


Рис. 3.11. Сторінка переміщення товару

Якщо на сторінці деталей про товар натиснути на синю кнопку, то відкриється сторінка, що зображена на рис. 3.12., в якій можна вказати кількість товару, що була додана на склад. Кількість, що ми вказали буде додана до вже наявної кількості на складі.

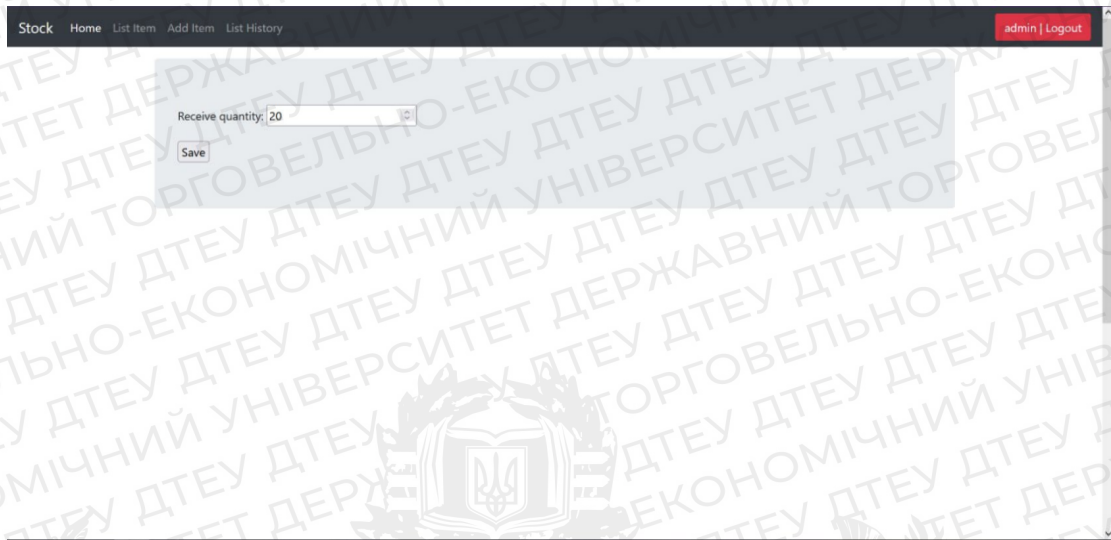


Рис. 3.12. Сторінка додавання товару до вже існуючого

Наступна сторінка відкривається, коли ми натискаємо на показник критичної кількості товарів. На сторінці, що зображена на рис 3.13. можна виставити кількість товару, при якому сайт буде сигналізувати, що його може не вистачити для наступних замовлень.

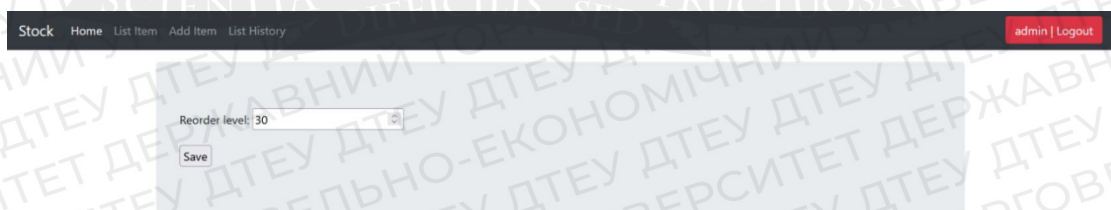


Рис. 3.13. Сторінка зазначення кількості товару для початку закупівель

На рис. 3.14 показано як саме сайт сигналізує про те, що певного товару на складі менше, ніж нам може знадобитись для майбутніх замовлень.



Рис. 3.14. Повідомлення про меншу кількість товару на складі

Сторінка, що показана на рис. 3.15, відповідає за видалення товару з списку і викликається за допомогою кнопки delete на сторінці з усіма наявними товарами. Це може бути корисно, коли певний товар більше не продається і його потрібно викреслити.

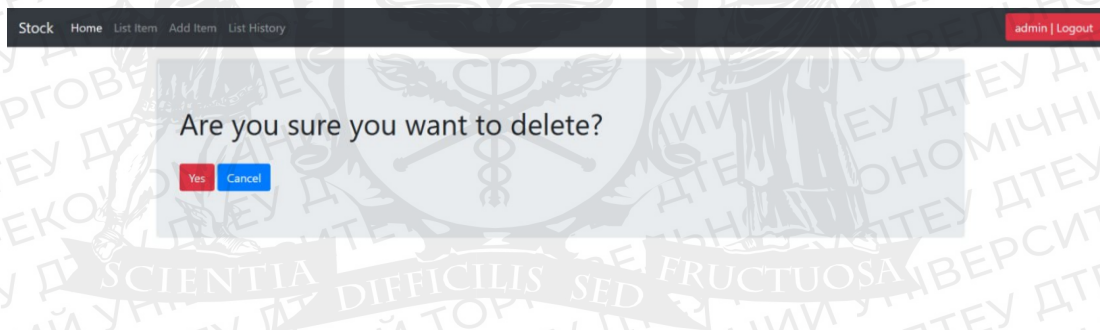


Рис. 3.15. Сторінка видалення товару

Сторінка, що показує всю історію операцій з товарами показана на рис 3.16. Ця сторінка зберігає інформацію про надходження на склад певної продукції і їх відправлення по певним замовленням, або на інші склади.

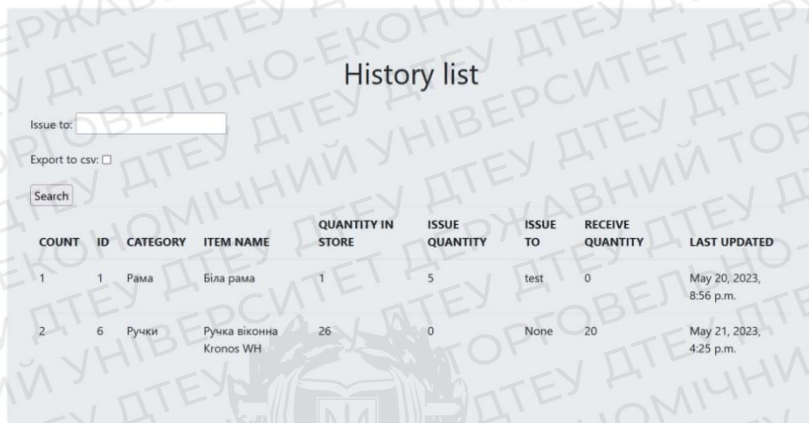


Рис. 3.16. Сторінка історії змін в товарах

Також даний програмний застосунок має можливість експорту в формат CSV даних про товари, що були обрані в списку товарів, або в певному замовленні. Для цього потрібно написати, яке саме замовлення нам потрібно, натиснути галочку Export to csv і кнопку Search. Після цього буде завантажено файл, який матиме інформацію про те, які товари входять в обране замовлення. Сам файл показаний в рис 3.17.

CATEGORY	ITEM NAME	QUANTITY	ISSUE QUANTITY	RECEIVE QUANTITY	RECEIVE BY	ISSUE BY	LAST UPDATED	COLOR
Фурнітура	Запорна пластина KBE 58/13	6	2	0	admin		2023-05-21 17:26:18.901873+00:00	Метал
Ручки	Ручка ракушка	13	2	0	admin		2023-05-21 17:26:48.555239+00:00	Сірий
Створка	WDS 7 Series	6	2	0	admin		2023-05-21 17:27:08.564480+00:00	Білий
Рама	WDS 7 Series	6	2	0	admin		2023-05-21 17:27:31.706501+00:00	Білий
Статика	WDS 7 Series	4	1	0	admin		2023-05-21 17:27:49.921918+00:00	Білий
Імпост	Імпост	5	1	0	admin		2023-05-21 17:28:10.604419+00:00	Метал

Рис 3.17. Вивід у файл CSV ВИСНОВКИ

ВИСНОВКИ

У випускній кваліфікаційній роботі були досліджені проблеми і тенденції розвитку інформаційних систем в області складського обліку. Була встановлена актуальність автоматизації цих систем у зв'язку зі зростаючими вимогами до ефективності та точності управління складськими процесами.

Результати дослідження, проведеного в рамках даної роботи, можна узагальнити наступним чином:

Виявлено проблеми впровадження систем управління складськими процесами. Однією з основних проблем є складність інтеграції нової системи з існуючими інформаційними системами підприємства. Це може призводити до неповної або некоректної передачі даних, що впливає на точність та надійність управління запасами.

Актуальність автоматизації в області складського обліку: Дослідження підтвердило необхідність автоматизації складського обліку для підвищення ефективності та точності управління запасами. Автоматизація дозволяє знизити час на виконання операцій, зменшити помилки та покращити контроль над запасами, що є критичними факторами для підприємств, особливо в умовах постійно зростаючої конкуренції.

Дослідження виявило кілька важливих тенденцій у розвитку інформаційних систем в галузі складського обліку. Одна з них - зростання використання хмарних технологій для забезпечення доступу до даних з будь-якого пристрою та полегшення обміну інформацією між різними складськими пунктами. Також спостерігається зростання використання аналітики даних для покращення стратегій управління запасами та прийняття обґрунтованих рішень.

Здійснений огляд існуючих програмних систем управління складом в Україні показав, що на ринку присутні різні рішення з різними

функціональними можливостями. Однак, була виявлена необхідність розробки власної програмної системи автоматизації, яка б враховувала специфіку складського обліку на конкретному підприємстві.

Була розроблена програмна система автоматизації складського обліку на основі Web-додатку. Розроблена система може значно полегшити роботу з управління запасами, покращити точність обліку та забезпечити швидкий доступ до необхідної інформації.

Результати дослідження мають практичне значення для підприємств, що займаються складським обліком. Впровадження розробленої системи автоматизації може допомогти підприємствам оптимізувати складські процеси, зменшити час на виконання операцій та підвищити точність обліку. Крім того, результати дослідження можуть стати основою для подальшого вдосконалення та розвитку інформаційних систем у галузі складського обліку.

Отримані результати дослідження мають практичне значення для підприємств, які займаються складським обліком. Впровадження розробленої системи автоматизації може сприяти покращенню ефективності та точності складського обліку, зниженню витрат та підвищенню рівня контролю над запасами. Крім того, результати дослідження можуть бути використані для подальшого вдосконалення та розвитку інформаційних систем в галузі складського обліку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гарнов А. П. Економіка підприємства // підручник для бакалаврів 2019 - 303 с
2. СТАНКЕВИЧ І. В., ТІГАРСВА В. А ЕКОНОМІКО-МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ, СТАНКЕВИЧ І. В., ТІГАРСВА В. А. // Наука й економіка 2014, № 4 (36) с.207-210
3. Ярошук О. Блокчейн у системі бухгалтерського обліку // Перспективи розвитку освіти, науки і бізнесу в глобальному середовищі : матеріали VIII Міжнародної науково-практичної конференції, 2020. 292 с., С. 99-100.
4. Проектування інформаційної системи обліку руху матеріалів на складі [Електронний ресурс] – Режим доступу до ресурсу: <https://ukrbukva.net/page,2,100724-Proektirovanie-informacionnoiy-sistemy-ucheta-dvizheniya-materialov-na-sklade.html>
5. Functional and Nonfunctional Requirements: Specification and Types [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://www.altexsoft.com/blog/business/functional-and-non-functionalrequirements-specification-and-types>
6. ERPNext[Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/ERPNext>
7. Блокчейн, штучний інтелект і бухгалтерія [Електронний ресурс]. – Режим доступу <https://core.ac.uk/download/pdf/268453007.pdf>
8. Internet of Things, IoT [Електронний ресурс] – Режим доступу до ресурсу: <https://www.it.ua/knowledge-base/technology-innovation/internet-veschej-internet-of-things-iot>

9. Functional and Nonfunctional Requirements: Specification and Types

[Електронний ресурс]. – 2018. – Режим доступу до ресурсу:

<https://www.altexsoft.com/blog/business/functional-and-non-functionalrequirements-specification-and-types/>

10.ТОП-10 програм для обліку товару та складу [Електронний ресурс].

– Режим доступу до ресурсу: [https://a4.com.ua/10-program-obliku-](https://a4.com.ua/10-program-obliku-tovaru-ta-skladu/)

[tovaru-ta-skladu/](https://a4.com.ua/10-program-obliku-tovaru-ta-skladu/)

11.11 найкращих програм для керування проектами з відкритим кодом

[Електронний ресурс] – Режим доступу до

ресурсу:<https://techukraine.net/11->

[%D0%BD%D0%B0%D0%B9%D0%BA%D1%80%D0%B0%D1%89%](https://techukraine.net/11-%D0%BD%D0%B0%D0%B9%D0%BA%D1%80%D0%B0%D1%89-%D0%B8%D1%85-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC-%D0%B4%D0%BB%D1%8F-%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5/)

[D0%B8%D1%85-](https://techukraine.net/11-%D0%BD%D0%B0%D0%B9%D0%BA%D1%80%D0%B0%D1%89-%D0%B8%D1%85-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC-%D0%B4%D0%BB%D1%8F-%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5/)

[%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC-](https://techukraine.net/11-%D0%BD%D0%B0%D0%B9%D0%BA%D1%80%D0%B0%D1%89-%D0%B8%D1%85-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC-%D0%B4%D0%BB%D1%8F-%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5/)

[%D0%B4%D0%BB%D1%8F-](https://techukraine.net/11-%D0%BD%D0%B0%D0%B9%D0%BA%D1%80%D0%B0%D1%89-%D0%B8%D1%85-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC-%D0%B4%D0%BB%D1%8F-%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5/)

[%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%](https://techukraine.net/11-%D0%BD%D0%B0%D0%B9%D0%BA%D1%80%D0%B0%D1%89-%D0%B8%D1%85-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC-%D0%B4%D0%BB%D1%8F-%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5/)

[D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5/](https://techukraine.net/11-%D0%BD%D0%B0%D0%B9%D0%BA%D1%80%D0%B0%D1%89-%D0%B8%D1%85-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC-%D0%B4%D0%BB%D1%8F-%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5/)

12.The Django template language [Електронний ресурс] – Режим доступу

до ресурсу:<https://docs.djangoproject.com/en/4.2/ref/templates/language/>

13.РОЗРОБКА ВЕБ ДОДАТКІВ З ВИКОРИСТАННЯМ PYTHON І

DJANGO [Електронний ресурс] – Режим доступу до ресурсу:

[https://webcase.com.ua/uk/blog/razrobotka-veb-prilozhenij-s-](https://webcase.com.ua/uk/blog/razrobotka-veb-prilozhenij-s-ispolzovaniem-python-i-django/)

[ispolzovaniem-python-i-django/](https://webcase.com.ua/uk/blog/razrobotka-veb-prilozhenij-s-ispolzovaniem-python-i-django/)

14.Google Trends. [Електронний ресурс] – Режим доступу до ресурсу:

<https://trends.google.com/trends/>

ДОДАТКИ

ДОДАТОК А

Програмний код реалізації web-додатку (реалізація моделей)

```
from django.db import models
```

```
# Create your models here.
```

```
class Category(models.Model):
```

```
    name = models.CharField(max_length=50, blank=False, null=True)
```

```
    def __str__(self):
```

```
        return self.name
```

```
class Colors(models.Model):
```

```
    name = models.CharField(max_length=50, blank=False, null=True)
```

```
    def __str__(self):
```

```
        return self.name
```

```
class Stock(models.Model):
```

```
    category = models.ForeignKey(Category, on_delete=models.CASCADE,  
    blank=False)
```

```
item_name = models.CharField(max_length=50, blank=True, null=True)
quantity = models.FloatField(default='0.0', blank=True, null=True)
receive_quantity = models.IntegerField(default='0', blank=True, null=True)
receive_by = models.CharField(max_length=50, blank=True, null=True)
issue_quantity = models.IntegerField(default='0', blank=True, null=True)
issue_by = models.CharField(max_length=50, blank=True, null=True)
issue_to = models.CharField(max_length=50, blank=True, null=True)
phone_number = models.CharField(max_length=50, blank=True, null=True)
created_by = models.CharField(max_length=50, blank=True, null=True)
reorder_level = models.IntegerField(default='0', blank=True, null=True)
last_updated = models.DateTimeField(auto_now_add=False, auto_now=True)
color = models.ForeignKey(Colors, on_delete=models.CASCADE, blank=False)

def __str__(self):
    return self.item_name
```

```
class StockHistory(models.Model):
```

```
    category = models.ForeignKey(Category, on_delete=models.CASCADE,
blank=False)
    item_name = models.CharField(max_length=50, blank=True, null=True)
    quantity = models.FloatField(default='0.0', blank=True, null=True)
    receive_quantity = models.IntegerField(default='0', blank=True, null=True)
    receive_by = models.CharField(max_length=50, blank=True, null=True)
    issue_quantity = models.IntegerField(default='0', blank=True, null=True)
    issue_by = models.CharField(max_length=50, blank=True, null=True)
    issue_to = models.CharField(max_length=50, blank=True, null=True)
```

```
phone_number = models.CharField(max_length=50, blank=True, null=True)
created_by = models.CharField(max_length=50, blank=True, null=True)
reorder_level = models.IntegerField(default='0', blank=True, null=True)
last_updated = models.DateTimeField(auto_now_add=False, auto_now=True)
color = models.ForeignKey(Colors, on_delete=models.CASCADE, blank=False)
```



ДОДАТОК Б

Програмний код реалізації web-додатку (Сторінка списку всіх товарів)

```
{% extends 'base.html' %}

{% block title %}
    Item List
{% endblock %}

{% block content %}
<h1 align="center">{{title}}</h1><br>
<form method="POST" action=">{% csrf_token %}
    {{form.as_p}}
    <input type="submit" value='Search'/>
</form>
<br>
    <table class='table'>
    <thead>
    <tr>
    <th>COUNT</th>
    <th>CATEGORY</th>
    <th>ITEM NAME</th>
    <th>QUANTITY IN STORE</th>
    <th>REORDER LEVEL</th>
    <th>COLOR</th>
    <th>LAST UPDATED</th>
    </thead>
    </table>
</br>
```

```

<th>DELETE</th>
</tr>
</thead>
{% for instance in queryset %}
<tr>
<td>{{ forloop.counter }}</td>
<td>{{ instance.category }}</td>
<td><a href="{% url 'update_item' instance.id
%}">{{ instance.item_name }}</a></td>
<td>
{% if instance.quantity <= instance.reorder_level %}
<div style="background-color: orange;">
<a href="{% url 'stock_detail' instance.id
%}">{{ instance.quantity }}</a></div>
{% else %}
<a href="{% url 'stock_detail' instance.id
%}">{{ instance.quantity }}</a>
{% endif %}
</td>
<td><a href="{% url 'reorder_level' instance.id
%}">{{ instance.reorder_level }}</a></td>
<td>{{ instance.color }}</td>
<td>{{ instance.last_updated }}</td>
<td><a href="{% url 'delete_item' instance.id %}">Delete</a></td>
</tr>
{% endfor %}
</table>

```

{% endblock %}



ДОДАТОК В

Програмний код реалізації web-додатку (Файл, що відповідає за програмну частину всіх сторінок)

```
from django.shortcuts import render, redirect
from .models import *
from .forms import StockCreateForm, StockSearchForm, StockUpdateForm,
IssueForm, ReceiveForm, ReorderLevelForm, StockHistorySearchForm
from django.http import HttpResponse
from django.contrib.auth.decorators import login_required
import csv

# Create your views here.
def home(request):
    title = 'Welcome: This is the Home Page'
    context = {
        "title": title,
    }
    return render(request, "home.html", context)

@login_required
def list_item(request):
    title = 'List of Items'
    queryset = Stock.objects.all()
```



```
form = StockSearchForm(request.POST or None)
```

```
if request.method == 'POST':
```

```
    queryset = Stock.objects.filter( #  
        category__icontains=form['category'].value(),  
        item_name__icontains=form['item_name'].value()  
    )
```

```
    if form['export_to_csv'].value() == True:
```

```
        response = HttpResponse(content_type='text/csv')
```

```
        response['Content-Disposition'] = 'attachment; filename="List of stock.csv"'
```

```
        writer = csv.writer(response)
```

```
        writer.writerow(['CATEGORY', 'ITEM NAME', 'QUANTITY'])
```

```
        instance = queryset
```

```
        for stock in instance:
```

```
            writer.writerow([stock.category, stock.item_name, stock.quantity])
```

```
        return response
```

```
context = {
```

```
    "form": form,
```

```
    "title": title,
```

```
    "queryset": queryset,
```

```
}
```

```
return render(request, "list_item.html", context)
```

```
@login_required
```

```
def add_item(request):
```

```
form = StockCreateForm(request.POST or None)
if form.is_valid():
    form.save()
    return redirect('/list_item')
context = {
    "form": form,
    "title": "Add Item",
}
return render(request, "add_item.html", context)

def update_item(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = StockUpdateForm(instance=queryset)
    if request.method == 'POST':
        form = StockUpdateForm(request.POST, instance=queryset)
        if form.is_valid():
            form.save()
            return redirect('/list_item')

    context = {
        'form': form
    }
    return render(request, 'add_item.html', context)

def delete_item(request, pk):
```

```
queryset = Stock.objects.get(id=pk)
if request.method == 'POST':
    queryset.delete()
    return redirect('/list_item')
return render(request, 'delete_item.html')
```

```
def stock_detail(request, pk):
    queryset = Stock.objects.get(id=pk)
    context = {
        "title": queryset.item_name,
        "queryset": queryset,
    }
    return render(request, "stock_detail.html", context)
```

```
def issue_items(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = IssueForm(request.POST or None, instance=queryset)
    if form.is_valid():
        instance = form.save(commit=False)
        instance.quantity -= instance.issue_quantity
        instance.issue_by = str(request.user)
        instance.save()
        issue_history = StockHistory(
            id=instance.id,
            last_updated=instance.last_updated,
```

```
category_id=instance.category_id,
item_name=instance.item_name,
quantity=instance.quantity,
issue_to=instance.issue_to,
issue_by=instance.issue_by,
issue_quantity=instance.issue_quantity,
color=instance.color,
)
issue_history.save()

return redirect('/stock_detail/' + str(instance.id))
# return HttpResponseRedirect(instance.get_absolute_url())

context = {
    "title": 'Issue ' + str(queryset.item_name),
    "queryset": queryset,
    "form": form,
    "username": 'Issue By: ' + str(request.user),
}
return render(request, "add_item.html", context)
```

```
def receive_items(request, pk):
    queryset = Stock.objects.get(id=pk)
    form = ReceiveForm(request.POST or None, instance=queryset)
    if form.is_valid():
        instance = form.save(commit=False)
```

```
instance.quantity += instance.receive_quantity
instance.save()
receive_history = StockHistory(
    id=instance.id,
    last_updated=instance.last_updated,
    category_id=instance.category_id,
    item_name=instance.item_name,
    quantity=instance.quantity,
    receive_quantity=instance.receive_quantity,
    receive_by=instance.receive_by,
    color=instance.color,
)
receive_history.save()

return redirect('/stock_detail/' + str(instance.id))
# return HttpResponseRedirect(instance.get_absolute_url())
context = {
    "title": 'Receive ' + str(queryset.item_name),
    "instance": queryset,
    "form": form,
    "username": 'Receive By: ' + str(request.user),
}
return render(request, "add_item.html", context)

def reorder_level(request, pk):
    queryset = Stock.objects.get(id=pk)
```

```
form = ReorderLevelForm(request.POST or None, instance=queryset)
```

```
if form.is_valid():
```

```
    instance = form.save(commit=False)
```

```
    instance.save()
```

```
    return redirect("/list_item")
```

```
context = {
```

```
    "instance": queryset,
```

```
    "form": form,
```

```
}
```

```
return render(request, "add_item.html", context)
```

```
@login_required
```

```
def list_history(request):
```

```
    header = 'History list'
```

```
    queryset = StockHistory.objects.all()
```

```
    form = StockHistorySearchForm(request.POST or None)
```

```
    context = {
```

```
        "form": form,
```

```
        "title": header,
```

```
        "queryset": queryset,
```

```
    }
```

```
    if request.method == 'POST':
```

```
        issue_to = form['issue_to'].value()
```

```
        queryset = StockHistory.objects.filter(
```

```
            issue_to__icontains=form['issue_to'].value(),
```

```
        )
```

```
if (issue_to != ""):
    queryset = queryset.filter(issue_to=issue_to)

if form['export_to_csv'].value() == True:
    response = HttpResponse(content_type='text/csv')
    response['Content-Disposition'] = 'attachment; filename="Stock
History.csv"'
    writer = csv.writer(response)
    writer.writerow(
        ['CATEGORY',
        'ITEM NAME',
        'QUANTITY',
        'ISSUE QUANTITY',
        'RECEIVE QUANTITY',
        'RECEIVE BY',
        'ISSUE BY',
        'LAST UPDATED',
        'COLOR'])
    instance = queryset
    for stock in instance:
        writer.writerow(
            [stock.category,
            stock.item_name,
            stock.quantity,
            stock.issue_quantity,
            stock.receive_quantity,
```

```
stock.receive_by,  
stock.issue_by,  
stock.last_updated,  
stock.color])  
return response  
  
context = {  
    "form": form,  
    "title": header,  
    "queryset": queryset,  
}  
return render(request, "list_history.html", context)
```

