

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 126 «Інформаційні системи та технології»
(освітня програма «Системи бізнес-аналітики»)

на тему:

**«Інформаційна система обліку товарів для
продуктового магазину»**

Виконала студентка групи ІСТ-20-1
ДАНИЛКО Єлизавета Сергіївна

Керівник роботи:
СВІНЦИЦЬКА Олександра Миколаївна

Рецензент:
ФУРІХАТА Денис Вікторович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Марина ГРАФ

«25» січня 2024 р.

ЗАВДАННЯ

за спеціальністю 126 «Інформаційні системи та технології»
(освітня програма «Системи бізнес-аналітики»)

Здобувач вищої освіти: **ДАНИЛКО Єлизавета Сергіївна**

Керівник роботи: **СВІНЦИЦЬКА Олександра Миколаївна**

Тема роботи: **Інформаційна система обліку товарів для продуктового магазину**
затверджена наказом закладу вищої освіти від «25» січня 2024 р., №24/с

Термін здачі закінченої роботи: «05» червня 2024 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Об'єктом дослідження є інформаційні технології для управління запасами та обліку товарів у сфері роздрібної торгівлі продуктами харчування. Предметом дослідження є веб-система обліку товарів з використанням сучасних технологій та фреймворків для оптимізації логістики та управління запасами в продуктовому магазині.

Консультанти випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1	СВІНЦИЦЬКА О.М.	05.02.2024	07.02.2024
2	СВІНЦИЦЬКА О.М.	12.03.2024	15.03.2024
3	СВІНЦИЦЬКА О.М.	08.04.2023	30.04.2024

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формування технічного завдання. Опрацювання літературних джерел.	2 лютого 2024 – 5 березня 2024	Виконано
2	Проектування структури системи	6 березня 2024 – 25 березня 2024	Виконано
3	Написання програмного коду	26 березня 2024 – 26 квітня 2024	Виконано
4	Тестування системи	27 квітня 2024 – 14 травня 2024	Виконано
5	Оформлення пояснювальної записки	15 травня 2024 – 1 червня 2024	Виконано
6	Попередній захист	1 червня 2024	Виконано

Здобувач вищої освіти **Єлизавета ДАНИЛКО**
Керівник **Олександра СВІНЦИЦЬКА**

РЕФЕРАТ

Кваліфікаційна робота складається з веб-додатку та пояснювальної записки. Пояснювальна записка до випускної роботи складається з переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури та додатків.

Загальний обсяг роботи становить 90 сторінок, з яких текстова частина викладена на 76 сторінках друкованого тексту. У роботі наведено 37 ілюстрацій, 14 таблиць та 13 сторінок додатків. Список використаних джерел налічує 20 найменувань і займає 2 сторінки.

Метою кваліфікаційної роботи є розробка інформаційної системи обліку товарів продуктового магазину з використанням сучасних веб-орієнтованих технологій для підвищення ефективності управління товарними запасами та продажами.

У роботі визначено основні завдання та функціональні вимоги до системи, проаналізовано наявні рішення на ринку, продемонстровано ключовий функціонал системи. Сформовано комплекс вимог до інформаційної системи, спрямованих на ефективне управління товарними запасами, обробку замовлень клієнтів та облік продажів.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, ОБЛІК ТОВАРІВ, ПРОДУКТОВИЙ МАГАЗИН, PHP, MYSQL, HTML, CSS, JAVASCRIPT, ВЕБ-ТЕХНОЛОГІЇ, ФРЕЙМВОРК, БАЗА ДАНИХ, УПРАВЛІННЯ ЗАПАСАМИ, ЗАМОВЛЕННЯ.

					ІСТ.КР.Б – 126 – 24 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Інформаційна система обліку товарів для продуктового магазину	Літ.	Арк.	Аркушів
Розроб.		Е. С. ДАНИЛКО						
Керівник		О.М. СВИНЦИЦЬКА					3	76
Рецензент		Д.В. ФУРІХАТА				Житомирська політехніка, група ІСТ-20-1		
Н. контр.								
Зав. каф.		М.С. ГРАФ			Пояснювальна записка			

ABSTRACT

The qualification work consists of a web application and an explanatory note. The explanatory note to the graduation thesis consists of a list of conventional abbreviations, an introduction, three chapters, conclusions, a list of references and appendices.

The total volume of the work is 90 pages, of which the text part is presented on 76 pages of printed text. The work contains 37 illustrations, 14 tables and 13 pages of appendices. The list of references includes 20 titles and occupies 2 pages.

The purpose of the qualification work is to develop an information system for accounting for goods in a grocery store using modern web-based technologies to improve the efficiency of inventory and sales management.

The work defines the main tasks and functional requirements for the system, analyses existing solutions on the market, and demonstrates the key functionality of the system. A set of requirements for the information system aimed at efficient inventory management, customer order processing and sales accounting has been formed.

KEYWORDS: INFORMATION SYSTEM, ACCOUNTING OF GOODS, GROCERY STORE, PHP, MYSQL, HTML, CSS, JAVASCRIPT, WEB TECHNOLOGIES, FRAMEWORK, DATABASE, INVENTORY MANAGEMENT, ORDERS.

		Є.С. ДАНИЛКО			ICT.KP.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВІНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
РОЗДІЛ 1. ФОРМУВАННЯ ТА ВИКОРИСТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ ПРОДУКТОВОГО МАГАЗИНУ	10
1.1 Суть інформаційних систем та їх види	10
1.2 Постановка задачі автоматизації бізнес-процесів продуктового магазину ...	14
1.3 Аналіз інформаційних систем для обліку товарів у продуктових магазинах	18
1.4 Обґрунтування вибору інструментальних засобів та вимоги до програмного забезпечення	23
Висновки до першого розділу	29
РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ ПРОДУКТОВОГО МАГАЗИНУ	31
2.1 Визначення варіантів використання системи	31
2.2 Розробка моделі бази даних для обліку товарів	35
2.3 Розробка структури файлів системи	42
Висновки до другого розділу	47
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ ПРОДУКТОВОГО МАГАЗИНУ	49
3.1 Реалізація системи обліку товарів	49
3.2 Структура інтерфейсу системи обліку товарів	57
3.3 Тестування та оцінка ефективності роботи системи	67
Висновки до третього розділу	71
ВИСНОВОК	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	75
ДОДАТКИ	77
Додаток А	78
Додаток Б	81

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІС — Інформаційна система

СУБД — Система управління базами даних

Bootstrap — це популярний безкоштовний фреймворк з відкритим кодом для розробки адаптивних та мобільно-орієнтованих веб-сайтів

SQL — це стандартна мова програмування, призначена для управління реляційними базами даних.

БД – база даних

HTML — це мова розмітки гіпертексту (Hypertext Markup Language), яка використовується для створення структури і контенту веб-сторінок.

CSS — це каскадні таблиці стилів (Cascading Style Sheets), які використовуються для візуального оформлення веб-сторінок

JavaScript — це мова програмування, яка використовується для створення інтерактивності та динамічної поведінки веб-сторінок на стороні клієнта

PHP — це серверна мова програмування, призначена для розробки веб-застосунків

MySQL — це система управління реляційними базами даних з відкритим кодом (My Structured Query Language)

Apache — це HTTP веб-сервер з відкритим кодом

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність теми. Актуальність обумовлена необхідністю ефективного управління товарними запасами, підвищення якості обслуговування клієнтів та оптимізації бізнес-процесів у сучасних ринкових умовах. Автоматизація процесів обліку та управління товарами дозволяє мінімізувати втрати від псування та дефіциту товарів, швидко реагувати на запити покупців, зменшити обсяг ручної праці та зосередити зусилля на покращенні сервісу.

Розробка такої системи відповідає сучасним тенденціям діджиталізації та автоматизації бізнес-процесів, а також є основою для подальшої інтеграції з системами онлайн-замовлень, мобільними додатками та програмами лояльності.

Таким чином, інформаційна система обліку товарів є необхідним інструментом для забезпечення конкурентоспроможності та ефективного розвитку продуктового магазину.

Мета дослідження. Метою дослідження є розробка інформаційної системи обліку товарів для продуктового магазину, яка дозволить підвищити ефективність управління товарними запасами, оптимізувати бізнес-процеси та покращити якість обслуговування клієнтів.

Для досягнення поставленої мети необхідно проаналізувати існуючі процеси обліку товарів у продуктовому магазині, визначити функціональні та нефункціональні вимоги до системи, розробити архітектуру та базу даних, реалізувати основні модулі системи, такі як управління асортиментом, приймання та переміщення товарів, інвентаризація, аналітика продажів тощо.

Об'єкт дослідження. Об'єктом дослідження є інформаційні технології для обліку товарів для продуктового магазину.

Предмет дослідження. Предметом дослідження є методи, моделі та засоби розробки інформаційної системи обліку товарів продуктового магазину з використанням сучасних технологій, таких як бази даних, веб-інтерфейси до створення програмного забезпечення.

Методи дослідження. В кваліфікаційній роботі використано: результати порівняльного аналізу, прогностного аналізу.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		8

Практичне значення отриманих результатів. У результаті виконання кваліфікаційної роботи, буде розроблено повноцінний веб-додаток, який може бути використаний будь-яким закладом вищої освіти та допоможе оптимізувати навчальний процес.

Апробація результатів. Матеріали роботи були представлені на Міжнародній науково-практичній конференції "Інтегровані інтелектуальні робототехнічні комплекси ПРТК-2024", 12-22 травня 2024 року .

- Публікації. За темою кваліфікаційної роботи було опубліковано 1 теза:
1. Данилко Є.С., Вольський Р.А. Розробка структури автоматизованої системи керування процесами роздрібної торгівлі. Тези Інтегровані інтелектуальні робототехнічні комплекси (ПРТК-2024). Сімнадцята міжнародна науково-практична конференція, 12-22 травня 2024 року. — Київ: НАУ, 2024. С.358-360.

Структура та обсяг роботи. Кваліфікаційна робота складається переліку умовних скорочень, вступу, трьох розділів, висновків, списку використаної літератури, який містить 20 найменувань, 2 додатки, та 37 ілюстрації. Загальний обсяг роботи становить 90 сторінок, з них 76 сторінок займає основний текст.

РОЗДІЛ 1. ФОРМУВАННЯ ТА ВИКОРИСТАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ ПРОДУКТОВОГО МАГАЗИНУ

1.1 Суть інформаційних систем та їх види

Інформаційна система (ІС) [1] – це сукупність взаємопов'язаних елементів, призначених для збору, зберігання, обробки, передачі та використання інформації з метою ефективного функціонування певної організації або системи.

Інформаційна система є складною системою, яка об'єднує в собі людські, технічні та програмні ресурси, а також процеси збору, обробки та поширення інформації. Вона складається з декількох взаємопов'язаних компонентів[2]:

1. Апаратне забезпечення (комп'ютери, сервери, мережеве обладнання);
2. Програмне забезпечення (операційні системи, прикладні програми, бази даних);
3. Люди (користувачі, розробники, адміністратори системи);
4. Дані та інформація, які циркулюють в системі;
5. Процеси (процедури збору, обробки та поширення інформації).

Інформаційні системи можна класифікувати за різними критеріями (див. Рис 1.1):

1. Рівні автоматизації:

- Ручні ІС: Усі інформаційні процеси виконуються людиною без застосування будь-яких технічних засобів.
- Автоматизовані ІС: Частину функцій (підсистем) керування або опрацювання даних здійснюють автоматично, а частину — людиною.
- Автоматичні ІС: Усі функції керування й опрацювання даних здійснюють за допомогою технічних засобів без участі людини.

2. Сфері застосування:

- Галузеві ІС: Призначені для автоматизації діяльності в певній галузі, наприклад, у банківській сфері, медицині, освіті тощо.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		10

- Функціональні ІС: Призначені для автоматизації певної функції управління, наприклад, бухгалтерського обліку, управління запасами, управління персоналом тощо.

- Міжгалузеві ІС: Можуть використовуватися в декількох галузях з незначними модифікаціями.

3. Організаційній структурі:

- Централізовані ІС: Усі ресурси ІС зосереджені в одному місці.

- Децентралізовані ІС: Ресурси ІС розподілені між декількома територіально віддаленими підрозділами.

- Розподілені ІС: Компоненти ІС фізично розташовані в різних місцях, але логічно об'єднані в єдину систему.

4. Способі доступу до інформації:

-- Локальні ІС: Інформація зберігається та обробляється на комп'ютерах користувачів.

- Віддалені ІС: Користувачі отримують доступ до інформації та ресурсів ІС через комп'ютерну мережу.

- Хмарні ІС: Інформація та ресурси ІС зберігаються та обробляються на віддалених серверах, до яких користувачі отримують доступ через Інтернет.

5. Інших факторах:

- Масштаб ІС: Можуть бути великими (для обслуговування тисяч користувачів), середніми (для обслуговування сотень користувачів) або малими (для обслуговування декількох користувачів).

- Тип обробки даних: Можуть бути пакетної обробки (для обробки великих обсягів даних за короткий проміжок часу) або транзакційної обробки (для обробки окремих транзакцій у режимі реального часу).

- Тип бази даних: Можуть використовувати реляційні, ієрархічні, мережеві або об'єктно-орієнтовані бази даних.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				11
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис 1.1. Класифікація інформаційних систем

Розуміння різних типів інформаційних систем та їх характеристик є важливим для вибору та впровадження оптимальної системи, яка відповідатиме конкретним потребам організації чи галузі. Правильний вибір інформаційної системи дозволяє підвищити ефективність роботи, оптимізувати бізнес-процеси та забезпечити конкурентні переваги в умовах динамічного розвитку інформаційних технологій.

Інформаційна система обліку товарів продуктового магазину є прикладом автоматизованої, галузевої та функціональної ІС, призначеної для ефективного управління товарними запасами та оптимізації бізнес-процесів у сфері роздрібної торгівлі. Залежно від масштабу та структури продуктового магазину, система може мати централізовану або розподілену організацію, забезпечуючи локальний або віддалений доступ до інформації. Масштаб системи може варіюватися від малого до середнього, відповідно до розміру магазину. Для зберігання та управління даними про товари зазвичай використовується реляційна база даних, яка забезпечує цілісність, безпеку та швидкий доступ до інформації.

Основною метою інформаційної системи є забезпечення потреб організації або системи в релевантній і своєчасній інформації для ефективного прийняття рішень та підтримки операційних процесів. ІС допомагає керівництву та працівникам отримувати, аналізувати та використовувати дані та інформацію для досягнення цілей організації.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВІНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		12

Завданнями інформаційних систем є:

1. Збір та накопичення даних з різних джерел;
2. Забезпечення зберігання та захисту даних;
3. Обробка та аналіз даних для отримання корисної інформації;
4. Своєчасне надання необхідної інформації користувачам системи;
5. Підтримка прийняття обґрунтованих рішень на основі отриманої інформації;

Використання інформаційних систем в організаціях має низку суттєвих переваг. По-перше, вони дозволяють підвищити ефективність роботи за рахунок автоматизації рутинних завдань та процесів, що економить час та зусилля працівників. Крім того, ІС забезпечують швидкий доступ до необхідної інформації та даних, покращують координацію та комунікацію між різними підрозділами організації, а також дають можливість обробляти великі обсяги даних та проводити складні аналітичні розрахунки.

По-друге, інформаційні системи сприяють покращенню процесу прийняття рішень завдяки наданню актуальної, повної та достовірної інформації для прийняття обґрунтованих рішень. ІС дозволяють проводити аналіз та моделювання різних сценаріїв та їх наслідків, що підвищує якість та обґрунтованість рішень за рахунок використання аналітичних інструментів.

Проаналізувавши джерело [4] було визначено, що сфера торгівлі та роздрібно торгівлі має специфічні вимоги до інформаційних систем. Основними особливостями ІС для торгових підприємств є управління великими обсягами товарних запасів та асортиментом товарів, відстеження руху товарів від постачальників до кінцевих споживачів, інтегрування різних бізнес-процесів (закупівля, продаж, облік, логістика), обробка великої кількості щоденних транзакцій продажу, аналіз попиту, підтримка різних каналів продажів (роздрібні магазини, онлайн-продажі), а також управління маркетинговими кампаніями та програмами лояльності.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				13
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Постановка задачі автоматизації бізнес-процесів продуктового магазину

Продуктовий магазин є роздрібним торговельним підприємством, яке займається закупівлею та продажем широкого асортименту продовольчих товарів. Основними бізнес-процесами в магазині є закупівля товарів у постачальників, облік наявних товарних запасів на складі, реалізація товарів кінцевим споживачам через торгову залу.

Закупівля товарів здійснюється шляхом формування замовлень для постачальників на основі аналізу поточних залишків та попиту на товари. Після надходження товарів від постачальників проводиться їх оприбуткування - облік кількості та вартості товарів, що надійшли на склад. Надалі товари зберігаються на складі до моменту їх реалізації.

Продаж товарів здійснюється через торгову залу магазину, де касири обслуговують покупців, оформляючи покупки та проводячи розрахунки. Після продажу товару відбувається його списання зі складських залишків. Регулярно проводиться інвентаризація для звірки фактичної наявності товарів зі складськими залишками.

Наразі існуючі бізнес-процеси закупівлі, обліку та продажу товарів в магазині здійснюються переважно за допомогою застарілих і неефективних методів, що створює низку проблем:

1. Значні затрати часу на формування замовлень, підрахунок залишків, складання звітності вручну.
2. Ризик виникнення помилок в обліку товарів через людський фактор.
3. Складність аналізу продажів, залишків, потреб в замовленнях через відсутність автоматизованих аналітичних інструментів.
4. Неможливість оперативного контролю поставок, запасів, прострочених товарів.

Тому виникає потреба в автоматизації цих бізнес-процесів шляхом впровадження спеціалізованої інформаційної системи товарного обліку. Це дозволить оптимізувати роботу, підвищити точність даних, спростити аналіз та контроль товарних запасів і замовлень.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		14

Основна мета створення системи товарного обліку - автоматизація та оптимізація процесів управління товарними запасами, замовленнями та поставками в продуктовому магазині для підвищення ефективності роботи та забезпечення точності і актуальності інформації.

Функціональні вимоги [5] до інформаційної системи обліку товарів продуктового магазину, розділимо на модулі:

Модуль "Товари":

- Можливість додавання, редагування та видалення товарів
- Функція категоризації товарів за групами та підгрупами
- Зберігання детальної інформації про кожен товар (назва, опис, ціна, торгова марка, одиниця виміру тощо)

Модуль "Склади":

- Можливість додавання, редагування та видалення складів
- Зберігання інформації про розташування
- Відображення кількості товарів на кожному складі
- Функція переміщення товарів між складами
- Формування звітів про запаси товарів на складах

Модуль "Користувачі":

- Можливість додавання, редагування та видалення облікових записів користувачів
- Функція розподілу прав доступу та ролей користувачів
- Автентифікація та авторизація користувачів при вході в систему

Модуль "Постачальники":

- Можливість додавання, редагування та видалення постачальників
- Зберігання контактної інформації та умов співпраці з кожним постачальником
- Функція формування замовлень на поставку товарів
- Формування звітів про роботу з постачальниками

Модуль "Інвентаризація":

- Функція реєстрації фактичної кількості товарів під час інвентаризації

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		15

- Порівняння фактичної кількості з даними в системі та виявлення розбіжностей
- Формування актів інвентаризації та відповідних документів

Модуль "Списання":

- Можливість реєстрації списання товарів з різних причин (брак, закінчення терміну придатності тощо)
- Оновлення кількості товарів на складах після списання
- Формування актів списання та відповідних документів

Модуль "Замовлення у постачальника":

- Можливість формування замовлень на поставку товарів у постачальників
- Формування звітів про замовлення та їх виконання

Модуль "Надходження товару":

- Можливість реєстрації надходження товарів від постачальників
- Функція перевірки відповідності отриманих товарів замовленням
- Оприбуткування товарів на склади та оновлення кількості в системі
- Формування документів про надходження товарів (накладні, акти прийому тощо)

Модуль "Продажі":

- Можливість реєстрації продажів товарів магазину
- Оновлення кількості товарів на складах після продажу
- Функція формування чеків та звітів про продажі
- Аналіз даних про продажі для виявлення трендів та прийняття управлінських рішень

Ці функціональні вимоги визначають основні можливості та операції, які повинна забезпечувати інформаційна система обліку товарів продуктового магазину в рамках кожного модуля. Реалізація цих вимог дозволить ефективно управляти товарними запасами, взаємодіяти з постачальниками, проводити інвентаризацію, списання, реєструвати надходження товарів та продажі, а також формувати необхідну звітність для прийняття управлінських рішень.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Нефункціональні вимоги до інформаційної системи обліку товарів продуктового магазину:

1. Продуктивність:

Інформаційна система повинна забезпечувати швидку реакцію на запити користувачів та ефективно використовувати апаратні ресурси. Час відгуку системи має бути мінімальним, щоб не створювати затримок у роботі користувачів. Необхідно оптимізувати запити до бази даних та мінімізувати час завантаження сторінок.

2. Надійність:

Система повинна функціонувати стабільно та безперебійно, забезпечуючи високу доступність для користувачів. Необхідно мінімізувати ризики збоїв та помилок у роботі системи. Регулярне тестування та моніторинг системи допоможуть виявити та усунути потенційні проблеми до того, як вони вплинуть на роботу користувачів.

3. Масштабованість:

Інформаційна система повинна бути розроблена з урахуванням можливості майбутнього розширення функціональності та збільшення обсягів даних. Система повинна ефективно обробляти зростаючі обсяги даних та кількість користувачів без втрати продуктивності.

4. Зручність використання:

Користувацький інтерфейс системи повинен бути інтуїтивно зрозумілим та зручним у використанні. Навігація по системі має бути логічною та послідовною. Необхідно забезпечити швидкий доступ до основних функцій та мінімізувати кількість кроків, необхідних для виконання типових операцій. Система повинна надавати чіткі та інформативні повідомлення про помилки та підказки для користувачів. Наявність довідкової інформації та навчальних матеріалів полегшить освоєння системи новими користувачами.

5. Крос-платформеність:

Інформаційна система повинна бути доступною з різних пристроїв та операційних систем. Веб-інтерфейс системи має коректно відображатися та

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		17

функціонувати на різних браузерях та пристроях, включаючи настільні комп'ютери, ноутбуки, планшети та смартфони. Адаптивний дизайн забезпечить зручність використання системи на екранах різних розмірів.

Дотримання цих нефункціональних вимог забезпечить надійну, безпечну та ефективну роботу інформаційної системи обліку товарів продуктового магазину. Врахування аспектів безпеки, продуктивності, надійності, масштабованості, зручності використання, крос-платформності та інтеграції дозволить створити систему, яка відповідатиме потребам користувачів та бізнесу в цілому.

Основна задача полягає в автоматизації та оптимізації процесів управління товарними запасами, замовленнями та поставками в продуктовому магазині для підвищення ефективності роботи та забезпечення точності й актуальності інформації. Це буде досягнуто шляхом впровадження спеціалізованої інформаційної системи товарного обліку, яка буде відповідати як функціональним, так і нефункціональним вимогам, забезпечуючи швидку, надійну та зручну роботу для користувачів.

1.3 Аналіз інформаційних систем для обліку товарів у продуктових магазинах

В сучасному світі, де конкуренція на ринку надзвичайно висока, ефективне управління торговельними процесами є ключовим фактором успіху для будь-якого продуктового магазину. Розвиток інформаційних технологій надає можливості вдосконалити цей процес шляхом впровадження інформаційних систем обліку товарів. Одним із етапів розробки такої системи є аналіз аналогів програмного продукту, що дозволяє краще зрозуміти функціональні можливості, переваги та недоліки існуючих рішень на ринку.

У даному пункті буде ретельно проаналізовано три програмних продукти, які вже знаходяться на ринку та використовуються в сфері роздрібної торгівлі.

ZarMoney - це хмарне бухгалтерське програмне забезпечення, розроблене для задоволення потреб бізнесу будь-якого розміру. Вона пропонує комплексний набір функцій, включаючи виставлення рахунків, виставлення

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		18

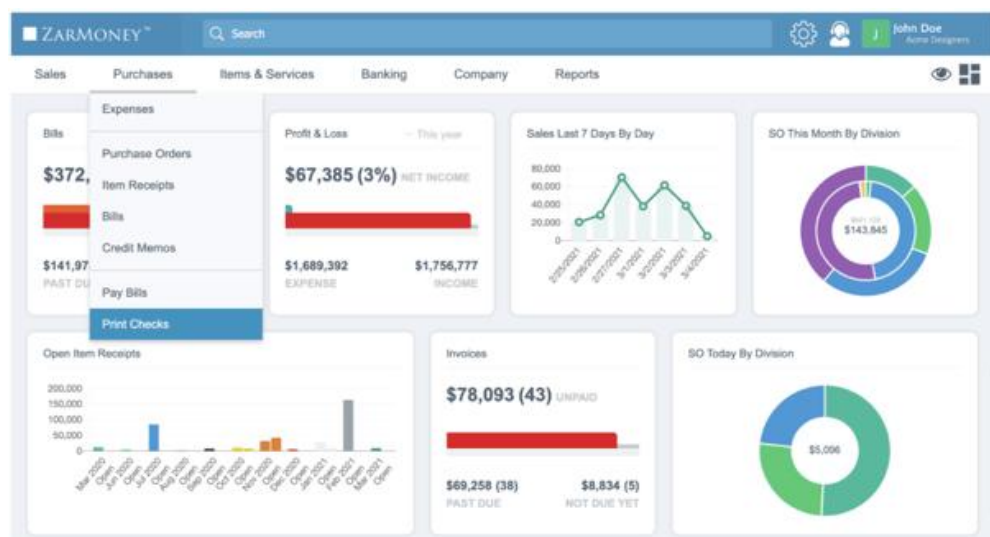
рахунків, відстеження доходів і витрат, фінансову звітність та управління запасами. Користувачі цінують простоту використання, доступність і можливість автоматизації різних фінансових завдань. Крім того, ZarMoney інтегрується з такими популярними платформами, як Shopify, Stripe та Mailchimp, покращуючи можливості синхронізації та обміну даними.

Переваги:

1. Простота використання: Інтуїтивно зрозумілий інтерфейс спрощує завдання.
2. Доступність: Економічно вигідна для підприємств будь-якого розміру.
3. Комплексні функції: Включає управління запасами та звітність.
4. Відмінне обслуговування клієнтів: Чуйна та корисна підтримка.
5. Наявність дашбордів з аналітикою та звітністю (див. Рис. 1.2).

Недоліки:

1. Обмеження в управлінні запасами: Неможливо призначити палетам місця розташування.
2. Варіанти звітності: Деякі обмеження, особливо для звітності за касовим методом.
3. Крива навчання: Нові користувачі можуть зіткнутися з певними труднощами на початковому етапі.
4. Налаштування інформаційної панелі: Обмежені можливості для налаштування відображення.



		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арх.
		О.М. СВИНЦІЦЬКА				
Змн.	Арх.	№ докум.	Підпис	Дата		19

Рис. 1.2 Інтерфейс ZarMoney

Dilovod - це інформаційна система, спрямована на автоматизацію бізнес-процесів, зокрема управління та обліком товарів у продуктовому магазині. Він пропонує широкий спектр можливостей, включаючи ведення обліку товарів на складі, контроль руху грошових коштів, звітність, а також інші аспекти, необхідні для успішної діяльності магазину.

Переваги:

1. Об'єднання управлінського і бухгалтерського обліку в одній програмі, що спрощує звітність.
2. Висока безпека даних.
3. Автоматичні безкоштовні оновлення та регулярні архівні копії для захисту даних.

Недоліки:

1. Вказано слабкий функціонал, що може обмежувати можливості користувачів.
2. Можливі проблеми з юзабіліті, що може ускладнювати використання програми (див. Рис. 1.3).

Товар	Основний склад	Магазин 234	Магазин № 528	Разом
«Таргана кисла» крупа в овочному молоці Грещі, 500г	132.00			132.00
«Таргана солодка» крупа в овочному молоці Грещі, 500г	20.00			20.00
«Таргана солодка» крупа в овочному молоці Грещі, 500г	135.00			135.00
Вершки кукурудзи - Тайланд, 340 г	34.00	12.00		46.00
Горіх Нут Грещі, 1кг	26.00			26.00
Горіх Нут Грещі, 1кг	36.00	5.00		41.00
Горіх нут Грещі, 500г	35.00			35.00
Каперси «БЕЛОН» лег банку бруто 3 кг, 1.7кг нетто	8.00			8.00
Каперси «МАКІДОНІ» ГТ- жб бруто 4кг, 2.2кг нетто	35.00		2.00	37.00
Кукурудза Грещі, 500г	148.00	4.00		152.00
Листя вишні бруто (длин) Грещі бруто 2.5кг, 1.5 кг нетто	14.00	1.00		15.00
Листя вишні бруто (длин) Грещі, 450г	38.00			38.00
Маслина боск		5.00		5.00
Паста з тертих маслин	22.00	4.00		26.00
Сочевиця (очищена) Грещі, 1кг	37.00			37.00
Сочевиця (очищена) Грещі, 500г	36.00			36.00
Таленда (паста з тертих маслин Калімак) Грещі, 185г	36.00			36.00
Таленда (паста з тертих маслин) Грещі, 185г	92.00			92.00
Часник консервованій - Грещі, 240 г	11.00			11.00

Рис. 1.3 Інтерфейс Dilovod

"BAS Управління торгівлею" (BAS UT) - програма для автоматизації торговельних організацій. Дозволяє провести швидку, комплексну

автоматизацію бізнес-процесів управлінського і оперативного обліку, маркетингових стратегій, планування та подальшого аналізу торгових операцій.

Переваги:

1. Комплексність функціоналу, що включає управління замовленнями, контроль запасів та оптимізацію роботи з партнерами.
2. Простий інтерфейс, що полегшує роботу з системою та зменшує час на навчання персоналу (див. Рис. 1.4).
3. Можливість інтеграції з основним сайтом компанії для швидкого оновлення інформації.
4. Надійний захист даних та можливість їх відновлення.

Недоліки:

1. Потреба в налаштуванні системи для оптимальної роботи.
2. Високі витрати на налаштування та консультації фахівців.
3. Необхідність додаткових витрат на програмістів для якісної роботи системи.

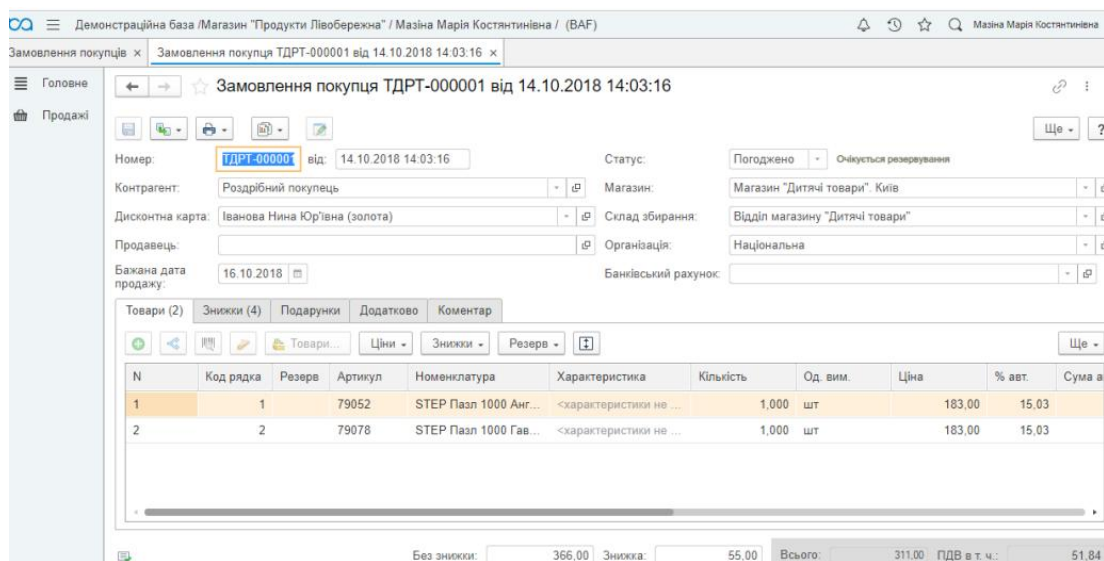


Рис. 1.4 Інтерфейс BAS

Під час розгляду аналогів програмного продукту для розробки інформаційної системи обліку товарів продуктового магазину було виявлено декілька ключових аспектів. Перш за все, було звернуто увагу на функціональні можливості програм, такі як облік товарів, замовлення, інвентаризація, а також звітність і аналітика. Детальний аналіз допоміг виявити сильні та слабкі

сторони кожного аналогу, а також їхню відповідність вимогам і потребам конкретного магазину. Додатково, була оцінена доступність технічної підтримки, можливості розширення та інтеграції з іншими системами (див. табл. 1.1).

Таблиця 1.1 - Таблиця порівняння аналогів

Характеристика	ZarMoney	Dilovod	BAS
Можливість підключати торгове обладнання	Ні	Так	Так
Зручність інтерфейсу	Лаконічний, приємний та зрозумілий інтерфейс	Поганий UX, не зрозумілий	Застарілий дизайн, але інтуїтивно зрозумілий
Інтеграція з іншими сервісами	Забезпечена	Забезпечена	Забезпечена
Ціна	Підписка на місяць за рівнем бізнесу	Підписка на місяць	Одноразова плата
Бухгалтерська звітність	Базові звіти та dashboard	Розширена бухгалтерські звіти різних видів	Розширена бухгалтерські звіти різних видів
Робота з постачальниками	Відсутня	Наявна	Наявна

ZarMoney пропонує широкий функціонал для управління фінансами бізнесу будь-якого розміру, з інтуїтивно зрозумілим інтерфейсом та доступними цінами. Dilovod, з іншого боку, спеціалізується на управлінні товарів у продуктовому магазині та об'єднує управлінський та бухгалтерський облік. Він відзначається високою безпекою даних, але може мати обмежений функціонал та проблеми з юзабіліті. BAS УТ відрізняється комплексним функціоналом, простим інтерфейсом та можливістю інтеграції з веб-сайтом компанії. Проте, для оптимальної роботи вимагає налаштування та може бути високими витратами на консультації фахівців та програмістів.

Під час розробки власної інформаційної системи для обліку товарів у продуктовому магазині, важливо врахувати кілька ключових аспектів, які виявлені під час аналізу аналогів. Серед них - комплексність функціоналу, що включає управління запасами, фінансову звітність та можливість інтеграції з іншими платформами. Також слід звернути увагу на простоту використання інтерфейсу, доступність програми для бізнесу будь-якого розміру, а також якість підтримки клієнтів. Необхідно також враховувати недоліки аналогів, зокрема обмеження управління запасами та проблеми з юзабіліті.

1.4 Обґрунтування вибору інструментальних засобів та вимоги до програмного забезпечення

1. Інструменти розробки

1.1 Мови програмування та технології:

Клієнтська частина

Для розробки клієнтської частини інформаційної системи обліку товарів було обрано наступні мови програмування та технології:

JavaScript - популярна мова програмування, що використовується для створення інтерактивних та динамічних веб-сторінок. За допомогою JavaScript можна реалізувати різноманітні функції, такі як валідація форм, обробка подій, анімація елементів інтерфейсу тощо. JavaScript дозволяє значно покращити досвід користувача та зробити інтерфейс більш зручним та реактивним[7].

jQuery - потужна бібліотека JavaScript, яка спрощує взаємодію з DOM (Document Object Model), обробку подій, анімацію та виконання AJAX-запитів. jQuery надає зручний та стислий синтаксис для виконання поширених завдань, що значно прискорює розробку клієнтської частини.

Bootstrap - популярний фреймворк для розробки адаптивних та responsive веб-інтерфейсів. Bootstrap надає набір готових компонентів та стилів, таких як сітки, кнопки, форми, навігаційні панелі тощо, які дозволяють швидко створювати привабливий та функціональний інтерфейс користувача[6]. Завдяки використанню Bootstrap, веб-додаток буде коректно відображатися на різних пристроях з різними розмірами екрану.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		23

AJAX (Asynchronous JavaScript and XML) - технологія, яка дозволяє веб-сторінкам асинхронно обмінюватися даними з сервером без необхідності повного перезавантаження сторінки. За допомогою AJAX можна оновлювати лише певні частини сторінки, що робить інтерфейс більш динамічним та реактивним. Це особливо корисно для реалізації функцій пошуку, фільтрації, сортування даних тощо.

Використання цих мов програмування та технологій дозволить створити сучасний, інтерактивний та зручний інтерфейс користувача для інформаційної системи обліку товарів. Застосування JavaScript та jQuery забезпечить динамічну поведінку елементів інтерфейсу та реакцію на дії користувача, а Bootstrap дозволить розробити адаптивний та привабливий дизайн. Завдяки AJAX, взаємодія з сервером буде більш ефективною та дозволить уникнути зайвих перезавантажень сторінки.

Серверна частина

Для розробки серверної частини інформаційної системи обліку товарів важливо обрати мову програмування, яка відповідає вимогам проекту та забезпечує ефективну та надійну роботу системи. Розглянемо порівняльну характеристику чотирьох популярних серверних мов програмування: PHP[8], C#, Python та Node.js.(див. табл. 1.2).

Таблиця 1.2 - Таблиця порівняння популярних серверних мов

Характеристика	PHP	C#	Python	Node.js
Парадигма	Процедурна, об'єктно-орієнтована	Об'єктно-орієнтована, функціональна	Об'єктно-орієнтована, процедурна, функціональна	Подійно-орієнтована, функціональна
Типізація	Динамічна, слабка	Статична, строга	Динамічна, строга	Динамічна, слабка
Продуктивність	Середня	Висока	Середня	Висока
Управління пам'яттю	Автоматичне	Автоматичне	Автоматичне	Автоматичне

Продовження таблиці 1.2

Характеристика	PHP	C#	Python	Node.js
Бібліотеки та фреймворки	Laravel, Symfony, CodeIgniter, Yii	ASP.NET, .NET Core	Django, Flask, Pyramid	Express.js, Koa.js, Nest.js
Розповсюдженість	Широко розповсюджена для веб-розробки	Широко розповсюджена в корпоративному секторі	Широко розповсюджена для веб-розробки, аналізу даних, машинного навчання	Широко розповсюджена для веб-розробки та створення серверних додатків
Швидкість розробки	Висока	Середня	Висока	Висока
Асинхронне програмування	Підтримується (з PHP 7.0)	Підтримується (async/await)	Підтримується (asyncio)	Підтримується (на основі подій)
Підтримка баз даних	MySQL, PostgreSQL, MS SQL Server, Oracle	MySQL Server, Oracle, MySQL, PostgreSQL	MySQL, PostgreSQL, MS SQL Server, Oracle, MongoDB	MySQL, PostgreSQL, MS SQL Server, Oracle, MongoDB
Платформи	Кросплатформна	Windows, Linux (3 .NET Core)	Кросплатформна	Кросплатформна
Спільнота та екосистема	Велика та активна спільнота, багато ресурсів	Велика спільнота, потужна екосистема Microsoft	Велика та активна спільнота, численні бібліотеки та інструменти	Велика та активна спільнота, розвинена екосистема npm

Після аналізу характеристик та потреб проекту, для розробки серверної частини інформаційної системи обліку товарів було обрано мову

програмування PHP. PHP має ряд переваг, які роблять її оптимальним вибором для даного проекту:

- PHP є широко розповсюдженою мовою для веб-розробки, що забезпечує багату екосистему бібліотек, фреймворків та інструментів для ефективної розробки серверної частини.
- PHP має вбудовану підтримку взаємодії з різними системами управління базами даних, зокрема з MySQL, яка буде використовуватися в проекті.
- Синтаксис PHP є простим та зрозумілим, що дозволяє швидко розробляти та підтримувати код.
- PHP має високу швидкість розробки та дозволяє створювати динамічні веб-сторінки з меншими затратами часу та ресурсів порівняно з іншими мовами.
- Велика та активна спільнота PHP-розробників забезпечує доступність великої кількості ресурсів, документації та підтримки для вирішення можливих проблем під час розробки.

Таким чином, використання PHP для серверної частини інформаційної системи обліку товарів дозволить ефективно реалізувати всі необхідні функції, забезпечити взаємодію з базою даних MySQL та створити надійну та масштабовану систему.

1.2 Інтегроване середовище розробки:

Для розробки інформаційної системи обліку товарів було обрано інтегроване середовище розробки (IDE) PhpStorm від компанії JetBrains, після дослідження [9]. PhpStorm є потужним та багатофункціональним інструментом, спеціально створеним для розробки PHP-проектів.

Середовище розробки надає зручну навігацію по файлах, класах та функціях проекту, а також дозволяє виконувати пошук використань та визначень символів.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		26

Середовище розробки підтримує системи управління залежностями, такі як Composer, та дозволяє використовувати препроцесори CSS та JavaScript для підвищення продуктивності розробки.

Використання PhpStorm в розробці інформаційної системи обліку товарів дозволить підвищити ефективність та якість процесу розробки, зменшити кількість помилок та пришвидшити написання коду. Потужні інструменти для рефакторингу, зневадження, тестування та роботи з базами даних допоможуть створити надійну та масштабовану систему. Інтеграція з популярними фреймворками та інструментами PHP-екосистеми забезпечить швидку та зручну розробку з використанням перевірених рішень та практик.

1.3 База даних

Для зберігання та управління даними в інформаційній системі обліку товарів було обрано реляційну систему управління базами даних (СУБД) MySQL[10]. MySQL є однією з найпопулярніших та найпотужніших СУБД з відкритим кодом, яка широко використовується для розробки веб-додатків та управління даними в різних галузях[19].

Вибір MySQL обумовлений рядом переваг та особливостей, які роблять її оптимальним рішенням для даного проекту[20]:

- Функціональність: MySQL надає повний набір функцій, необхідних для ефективного управління структурованими даними в рамках інформаційної системи обліку товарів. Вона підтримує реляційну модель даних, забезпечує можливість створення таблиць, встановлення зв'язків між ними, виконання складних запитів та забезпечення цілісності даних за допомогою обмежень та індексів.
- Продуктивність: MySQL відома своєю високою продуктивністю та оптимізацією для виконання типових операцій вставки, оновлення та вибірки даних.
- Масштабованість: MySQL пропонує різні механізми масштабування, такі як партиціонування таблиць, реплікація та кластеризація. Це

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		27

дозволяє системі обробляти зростаючі обсяги даних та підтримувати високе навантаження при збільшенні кількості користувачів та транзакцій.

- Сумісність та інтеграція: MySQL має широку підтримку з боку різних мов програмування та фреймворків, зокрема PHP, що використовується для розробки серверної частини системи. Наявність вбудованих драйверів та бібліотек для взаємодії з MySQL спрощує інтеграцію бази даних з іншими компонентами системи.

- Вартість володіння: MySQL розповсюджується під відкритою ліцензією GNU General Public License (GPL)[18], що дозволяє використовувати її безкоштовно для більшості проектів. Це знижує загальну вартість розробки та впровадження інформаційної системи обліку товарів.

Таким чином, вибір MySQL в якості бази даних для інформаційної системи обліку товарів забезпечує надійне, ефективне та масштабоване зберігання та управління даними. Її функціональність, продуктивність та сумісність з обраним стеком технологій роблять MySQL оптимальним рішенням для реалізації даного проекту.

2. Вимоги до апаратних засобів

2.1 Веб сервер

Веб-сервер є ключовим компонентом інформаційної системи обліку товарів, який відповідає за обробку HTTP-запитів від клієнтів, виконання серверної логіки та взаємодію з базою даних. Для розгортання та функціонування системи було обрано веб-сервер Apache[11].

Apache - це надійний та широко використовуваний веб-сервер з відкритим вихідним кодом, який забезпечує високу продуктивність, безпеку та гнучкість у налаштуваннях.

2.2 Апаратні засоби сервера

Для коректної роботи з інформаційною системою обліку товарів клієнтські пристрої повинні відповідати певним мінімальним вимогам. Це

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		28

забезпечить плавну та ефективну взаємодію користувачів з системою через веб-інтерфейс.

Мінімальні вимоги до апаратних засобів клієнтів:

- Процесор: Intel Core i3 або AMD Ryzen 3 чи новіший, не менше 2,0 ГГц.
- Оперативна пам'ять: не менше 8 ГБ
- Місце на диску: Не менше 128 ГБ на твердотілому накопичувачі (SSD) або жорсткому диску (HDD)
- Мережеве підключення: Ethernet-адаптер або Wi-Fi-модуль зі швидкістю не менше 100 Мбіт/с
- Дисплей: Роздільна здатність: не менше 1366x768 пікселів
- Периферійні пристрої: Клавіатура та миша для введення даних та навігації
- Операційна система: Windows 10, macOS 10.15 або новіші версії
- Останні версії веб-браузерів (Google Chrome, Mozilla Firefox, Microsoft Edge, Safari)

Забезпечення відповідності клієнтських пристроїв мінімальним апаратним вимогам є ключовим аспектом для ефективного функціонування інформаційної системи обліку товарів у продуктовому магазині. Дотримання цих вимог гарантує плавну та безперебійну взаємодію користувачів із системою через веб-інтерфейс, забезпечуючи швидкий відгук і зручність використання. Це, в свою чергу, сприяє підвищенню продуктивності та задоволеності працівників, що позитивно впливає на загальну ефективність роботи магазину.

Висновки до першого розділу

У межах даного розділу було розглянуто теоретичні засади інформаційних систем, здійснено постановку задачі для створення системи обліку товарів продуктового магазину. Визначено мету, завдання, функціональні та нефункціональні вимоги до майбутньої системи. Проаналізовано існуючі програмні рішення-аналоги, виявлено їх переваги та недоліки для врахування під час розробки.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		29

Обґрунтовано вибір інструментальних засобів: MySQL як СУБД, стек PHP/JavaScript/HTML/CSS/Apache для розробки, допоміжні бібліотеки та фреймворки для інтерфейсу, звітності тощо. Визначено апаратні вимоги до серверного та клієнтського обладнання. Розділ заклав теоретичну та практичну основу для успішної реалізації системи товарного обліку.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВІНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		30

РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Визначення варіантів використання системи

Проаналізувавши джерела [12-13], можемо зробити висновок, що чітке визначення варіантів використання системи є надзвичайно важливим етапом у процесі розробки програмного забезпечення. Це дозволяє розробникам краще зрозуміти потреби користувачів, визначити функціональні вимоги до системи та забезпечити її ефективність і зручність використання.

Варіанти використання системи описують взаємодію між користувачами (акторами) та системою, визначаючи послідовність дій, які користувачі можуть виконувати для досягнення певних цілей. Цей опис допомагає визначити межі системи, її функціональність та обмеження.

У контексті інформаційної системи обліку товарів продуктового магазину, визначення варіантів використання є особливо важливим, оскільки система повинна забезпечувати ефективне управління товарами, контроль запасів, взаємодію з постачальниками та відстеження продажів. Правильне розуміння ролей та можливостей адміністратора та менеджера допоможе створити систему, яка відповідає реальним потребам бізнесу та підвищує ефективність роботи магазину.

Для кращого розуміння та візуалізації варіантів використання системи, розглянемо Use Case діаграму, яка ілюструє взаємодію між двома основними ролями користувачів (адміністратором та менеджером) та функціями системи.

На Use Case діаграмах ми можемо побачити дві основні ролі:

- Адміністратор: представлений актором "Адміністратор" на діаграмі. Адміністратор має повний доступ до всіх функцій системи, включаючи керування товарами, категоріями, знижками, постачальниками, складами та продажами. (див. Рис. 2.1)
- Менеджер: представлений актором "Менеджер" на діаграмі. Менеджер має обмежений доступ до функцій системи, зосереджуючись на роботі з продажами, надходженнями, операціями на складах та перегляді товарів з терміном придатності. (див. Рис. 2.2)

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		31

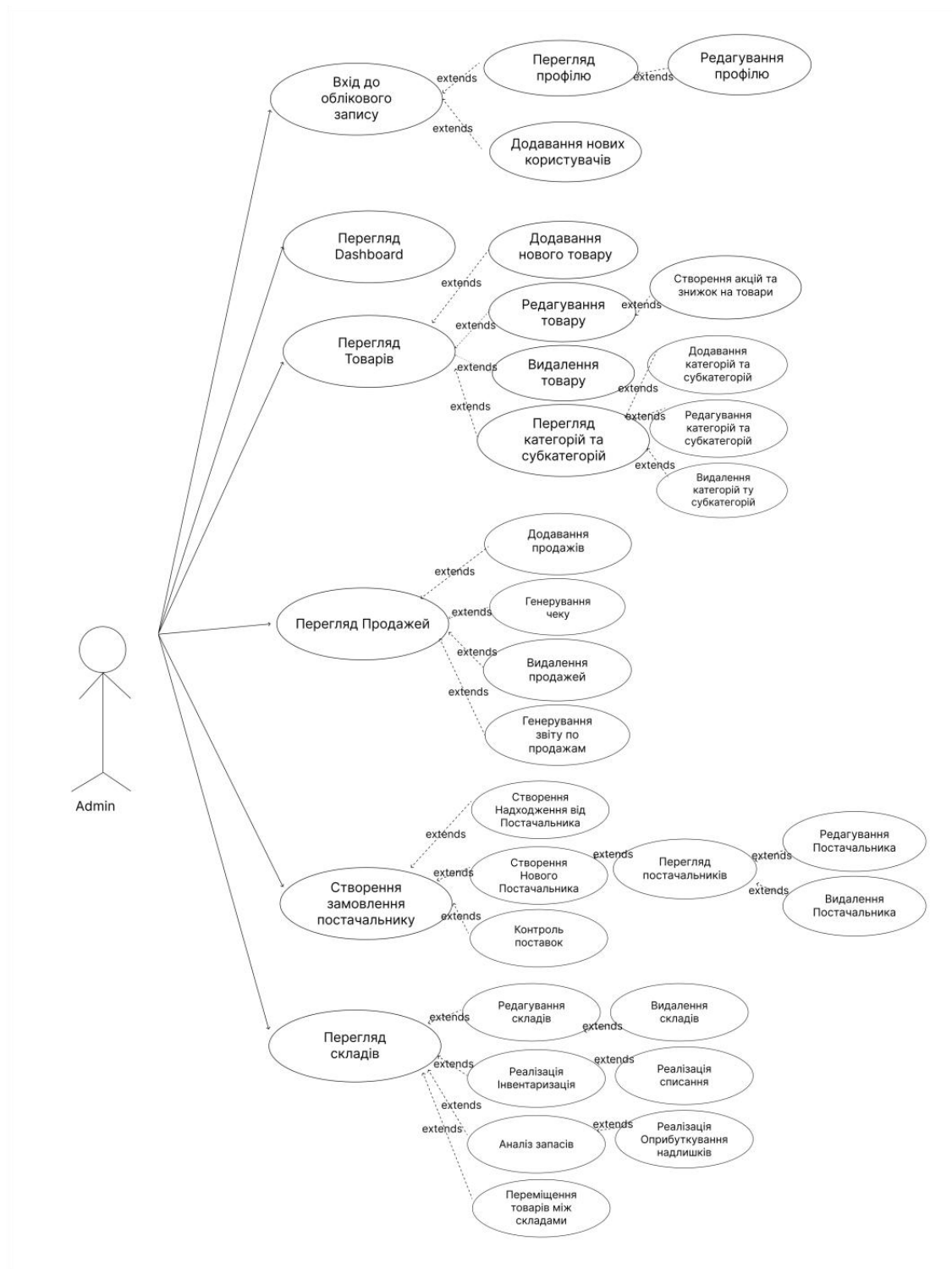


Рис. 2.1 – Use Case діаграма користувача з роллю «Адмін»

Адміністратор має повний контроль над системою та може виконувати наступні дії:

1. Керування товарами:

- Додавання нових товарів до системи
- Редагування інформації про існуючі товари

- Видалення товарів з системи
2. Керування категоріями та субкатегоріями:
- Створення нових категорій та субкатегорій товарів
 - Редагування існуючих категорій та субкатегорій
 - Видалення категорій та субкатегорій
3. Управління знижками:
- Додавання знижок на товари
 - Редагування існуючих знижок
 - Видалення знижок
4. Перегляд зіпсованих товарів:
- Отримання списку товарів, які стали зіпсованими
5. Робота з постачальниками:
- Додавання нових постачальників до системи
 - Редагування інформації про існуючих постачальників
 - Створення замовлень постачальникам
 - Оформлення надходжень від постачальників
 - Генерування звіту "Контроль поставок"
6. Управління складами:
- Додавання нових складів
 - Редагування інформації про існуючі склади
 - Видалення складів
 - Створення списань товарів
 - Проведення інвентаризації на складах
 - Генерування звіту "Аналіз запасів"
7. Робота з продажами:
- Створення продажів
 - Генерування звіту по продажам
8. Переміщення товарів:
- Переміщення товарів з одного складу на інший

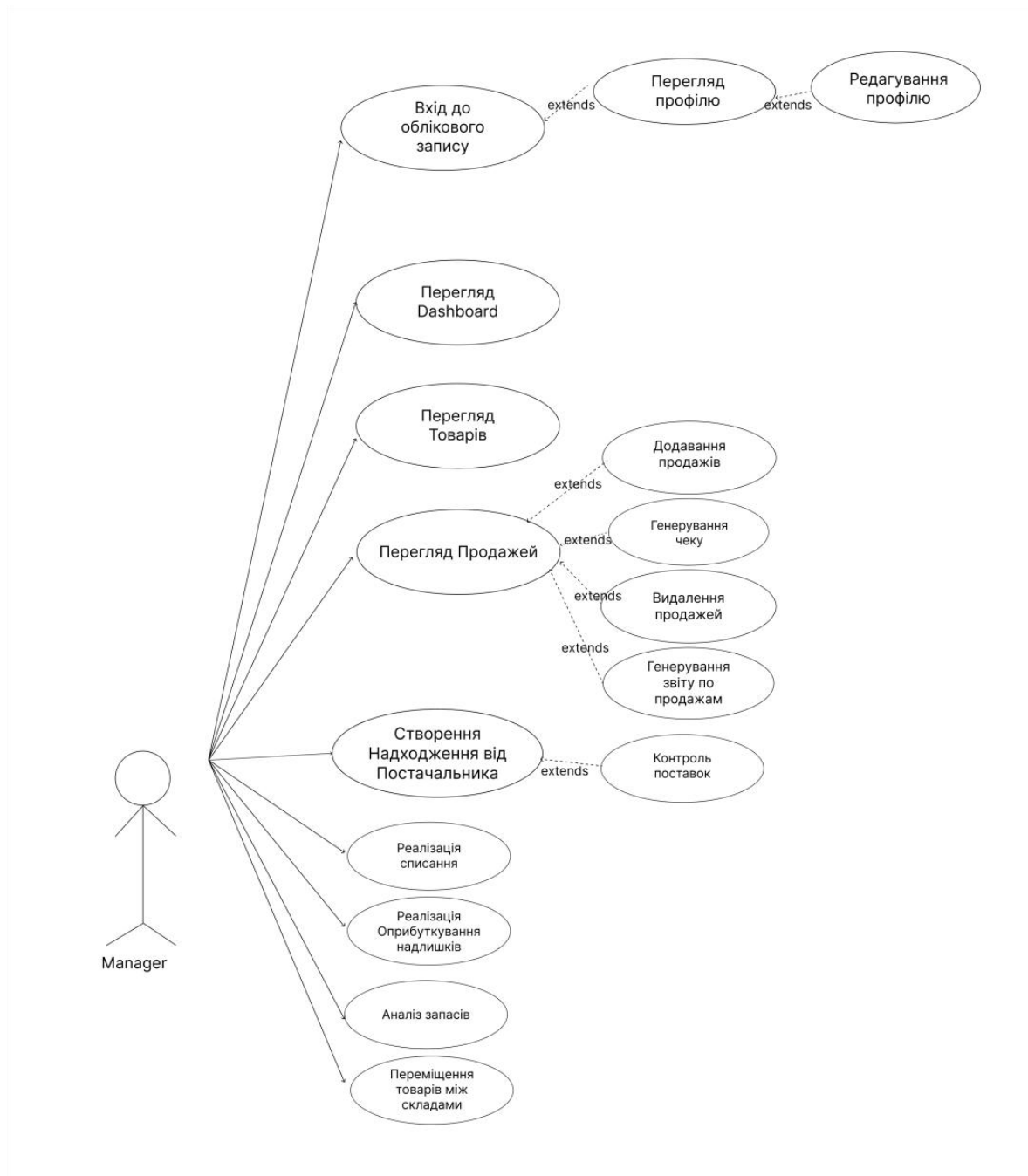


Рис. 2.2 – Use Case діаграма користувача з роллю «Менеджер»

Менеджер має обмежений доступ до системи та може виконувати наступні дії:

1. Робота з продажами:

- Створення продажі
- Генерування звіту по продажам

2. Робота з надходженнями:

- Створення надходжень від постачальників
- Генерування звіту "Контроль поставок"

3. Операції на складах:

- Виконання списання товарів
- Оприбуткування залишків товарів
- Переміщення товарів між складами
- Аналіз запасів на складах

4. Перегляд товарів з терміном придатності:

- Отримання списку товарів з терміном придатності

Таким чином, адміністратор має повний контроль над системою та може виконувати всі операції, пов'язані з товарами, категоріями, знижками, постачальниками, складами та продажами. Менеджер, у свою чергу, має обмежений доступ і може працювати з продажами, надходженнями, виконувати операції на складах та переглядати товари з терміном придатності.

2.2 Розробка моделі бази даних для обліку товарів

База даних є невід'ємною частиною будь-якої інформаційної системи, і система обліку товарів продуктового магазину не є винятком. Правильно спроектована та розроблена база даних забезпечує ефективне зберігання, організацію та доступ до даних, що є ключовими факторами для успішного функціонування системи.

Проектування структури бази даних - це процес визначення сутностей, атрибутів та зв'язків між ними, які відображають предметну область системи. Цей етап є надзвичайно важливим, оскільки від нього залежить ефективність роботи системи, цілісність та узгодженість даних, а також можливість подальшого розширення та модифікації системи[14].

Розробка бази даних на основі спроектованої структури передбачає створення таблиць, встановлення зв'язків між ними, визначення обмежень та індексів. Правильно розроблена база даних забезпечує оптимальну продуктивність системи, мінімізує надмірність даних та гарантує їх цілісність[15].

У контексті інформаційної системи обліку товарів продуктового магазину, база даних повинна зберігати інформацію про товари, категорії, субкатегорії,

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		35

постачальників, склади, продажі та інші пов'язані дані. Ефективна організація цих даних дозволяє системі швидко та точно виконувати операції, такі як пошук товарів, формування звітів, аналіз запасів та відстеження продажів.

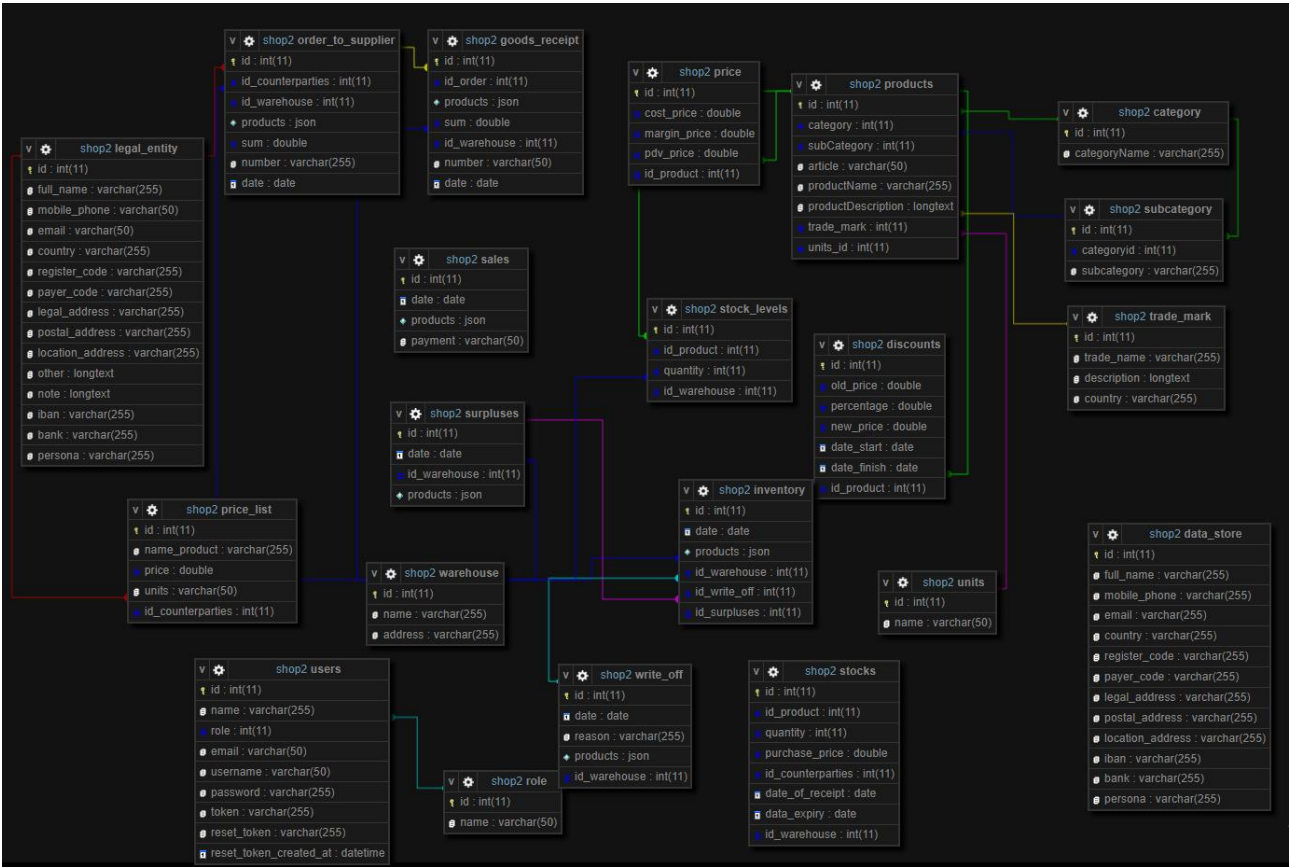


Рис 2.3 - Схема бази-даних

Для ефективного функціонування інформаційної системи було спроектовано та розроблено базу даних, яка складається з 21 взаємопов'язаних таблиць (див. Рис. 2.3). Кожна таблиця відповідає за зберігання специфічної інформації та має певні зв'язки з іншими таблицями.

База даних є невід'ємною частиною будь-якої інформаційної системи, і система обліку товарів продуктового магазину не є винятком. У першому розділі була обрана система керування базами даних MySQL для розробки бази даних цієї системи.

Розглянемо детально призначення кожної таблиці:

Таблиця 2.1 - Таблиця product

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код

Продовження таблиці 2.1

Назва	Тип даних	ПК	ЗК	Опис поля
category	int	-	+	Категорія
subCategory	int	-	+	Субкатегорія
article	varchar	-	-	Артикул
productName	varchar	-	-	Назва товару
productDescription	longtext	-	-	Опис товару
trade_mark	int	-	+	Торгова марка
units_id	int	-	+	Одиниця вимірювання

Таблиця “product” (див. табл. 2.1) призначена для зберігання вичерпної інформації про товари, що продаються в продуктовому магазині, включаючи їх категорії, підкатегорії, назви, описи, бренди та одиниці виміру.

Таблиця 2.2 - Таблиця stocks

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
id_product	int	-	+	Товар
quantity	int	-	-	Кількість
purchase_price	double	-	-	Закупівельна ціна
id_counterparties	int	-	+	Постачальник
date_of_receipt	date	-	-	Дата надходження
data_expiry	date	-	-	Дата терміну придатності
id_warehouse	int	-	+	Склад

Ця таблиця stocks (див. табл. 2.2) дозволяє відстежувати наявність товарів на складі, їх кількість, закупівельні ціни, постачальників, терміни придатності та розміщення на конкретних складах.

Таблиця 2.3 - Таблиця users

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
name	varchar	-	-	Ім'я користувача
role	int	-	+	Роль

Продовження таблиці 2.3

Назва	Тип даних	ПК	ЗК	Опис поля
email	varchar	-	-	Email
username	varchar	-	-	Логін
password	varchar	-	-	Пароль
token	varchar	-	-	Токен для додавання нового користувача
Reset_token	varchar	-	-	Токен для зміни паролю
Reset_token_created_at	datetime	-	-	Дата створення токену

Ця таблиця (див. табл. 2.3) зберігає облікові дані користувачів, їхні ролі та привілеї доступу до системи. Вона забезпечує можливість автентифікації користувачів, керування їхніми правами доступу та відновлення паролів у разі потреби.

Таблиця 2.4 - Таблиця write_off

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
date	date	-	-	Дата
reason	varchar	-	-	Причина списання
products	json	-	-	Продукти
Id_warehouse	int	-	+	Склад

Ця таблиця (див. табл. 2.4) дозволяє відстежувати та документувати випадки списання продуктів у магазині. Вона забезпечує зберігання дати списання, причини, деталей про списані продукти та їх кількість, а також зв'язок з конкретним складом, на якому відбулося списання.

Таблиця 2.5 - Таблиця inventory

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
date	date	-	-	Дата
Id_write_off	varchar	-	+	Списання
products	json	-	-	Продукти
Id_warehouse	int	-	+	Склад

Продовження таблиці 2.5

Назва	Тип даних	ПК	ЗК	Опис поля
Id_surpluses	int	-	+	Оприбуткування надлишків

Ця таблиця (див. табл. 2.5) слугує для документування процесу інвентаризації запасів на складі магазину. Вона зберігає дату проведення інвентаризації, зв'язок із записами про списання продуктів (якщо такі мали місце), деталі про перевірені продукти та їх кількість, а також інформацію про склад і виявлені надлишки. Ця інформація допомагає відстежувати фактичні запаси, виявляти розбіжності та забезпечувати точність даних в інформаційній системі магазину.

Таблиця 2.6 - Таблиця surpluses

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
date	date	-	-	Дата
products	json	-	-	Продукти
Id_warehouse	int	-	+	Склад

Ця таблиця (див. табл. 2.6) допомагає відстежувати випадки, коли фактична наявність товарів на складі перевищує записану в системі кількість. Надлишки можуть виникати через помилки в обліку або інші причини. Записуючи деталі надлишків у цю таблицю, система може коригувати залишки товарів на відповідних складах.

Таблиця 2.7 - Таблиця price

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
cost_price	double	-	-	Закупівельна ціна
margin_price	double	-	-	Ціна з націнкою
pdv_price	double	-	-	Ціна з ПДВ
id_product	int	-	+	Товар

Ця таблиця (див. табл. 2.7) дозволяє зберігати різні цінові компоненти для кожного товару, включаючи його закупівельну ціну, націнку магазину та ціну з ПДВ для кінцевого споживача.

Таблиця 2.8 - Таблиця order_to_supplier

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		39

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
Id_counterparties	Int	-	+	Постачальник
Id_warehouse	Int	-	+	Склад
products	json	-	-	Товари
sum	double	-	-	Сума
number	varchar	-	-	Номер замовлення
date	date	-	-	Дата

Ця таблиця (див. табл. 2.8) дозволяє відстежувати замовлення товарів у різних постачальників для потреб продуктового магазину. Коли товари від постачальника надходять на склад, інформація з цієї таблиці може бути використана для перевірки комплектності замовлення та внесення необхідних коректив у систему обліку запасів.

Таблиця 2.9 - Таблиця goods_receipt

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
Id_order	Int	-	+	Замовлення
products	json	-	-	Товари
sum	double	-	-	Сума
Id_warehouse	int	-	+	Склад
number	varchar	-	-	Номер надходження
date	date	-	-	Дата

Ця таблиця (див. табл. 2.9) допомагає відстежувати виконання замовлень постачальникам. Коли товари надходять, створюється новий запис із деталями про їх кількість, вартість та склад призначення. Ця інформація зіставляється із замовленням для підтвердження правильності постачання та оновлення даних про запаси на відповідному складі.

Таблиця 2.10 - Таблиця legal_entity

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
Full_name	varchar	-	-	Повна назва
Mobile_phone	varchar	-	-	Мобільний телефон

Продовження таблиці 2.10

Назва	Тип даних	ПК	ЗК	Опис поля
email	varchar	-	-	Пошта
country	varchar	-	-	Країна
Register_code	varchar	-	-	Код ЄРДПОУ
Payer_code	varchar	-	-	Код платника ПДВ
Legal_address	varchar	-	-	Юридична адреса
Postal_address	varchar	-	-	Поштова адреса
Location_address	varchar	-	-	Адреса місцезнаходження
other	longtext	-	-	Інше
note	longtext	-	-	Нотатки
iban	varchar	-	-	IBAN
bank	varchar	-	-	Банк
persona	varchar	-	-	Відповідальна особа

Ця таблиця (див. табл. 2.10) зберігає важливі реквізити, контактні дані та іншу ділову інформацію про юридичних осіб, з якими взаємодіє продуктовий магазин. Вона допомагає систематизувати та полегшити роботу з різними контрагентами.

Таблиця 2.11 - Таблиця discounts

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
Old_price	double	-	-	Стара ціна
percantage	double	-	-	Відсоток знижки
New_price	double	-	-	Нова Ціна
Date_start	date	-	-	Дата початку
Date_finish	date	-	-	Дата кінця
Id_product	int	-	+	Товар

Ця таблиця (див. табл. 2.11) зберігає інформацію про знижки на товари в магазині, дозволяючи відстежувати старі та нові ціни, відсоток знижки, період дії знижки та прив'язувати знижки до конкретних товарів за їх кодом (Id_product)

Таблиця 2.12 - Таблиця sales

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
date	date	-	-	Дата
products	json	-	-	Товари
payment	varchar	-	-	Вид оплати

Ця таблиця (див. табл. 2.12) призначена для зберігання інформації про здійснені продажі в магазині. Кожен запис відповідає окремій покупці, містить дату продажу, перелік придбаних товарів (у JSON форматі) та спосіб оплати.

Окрім детально описаних таблиць, база даних інформаційної системи містить й інші таблиці, їх детальний опис та структура будуть представлені у Додатку А до даної роботи.

2.3 Розробка структури файлів системи

Правильна організація файлів та каталогів є ключовим фактором для забезпечення ефективної розробки, підтримки та масштабування системи. Добре продумана архітектура файлової структури дозволяє розробникам легко знаходити та модифікувати відповідні частини коду, полегшує інтеграцію нових компонентів та сприяє кращій співпраці в команді.

Добре продумана архітектура файлової структури, яка відображає загальну архітектуру системи, відіграє ключову роль у досягненні цієї мети. Архітектура системи визначає принципи та підходи, які потім впроваджуються у структурі файлів та каталогів. Існує кілька популярних архітектурних підходів, які визначають різні способи організації файлової структури.

Розглянемо три популярні архітектури програмного забезпечення та їх вплив на структуру файлів системи:

1. Клієнт-серверна архітектура:

Ця архітектура передбачає розділення системи на два основні компоненти: клієнтський інтерфейс та серверну частину. Файлова структура відображає це розділення, де клієнтська частина (наприклад, веб-додаток або мобільний додаток) та серверна частина (наприклад, REST API або сервер додатків) розміщуються у різних каталогах або

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		42

репозиторіях. Ця архітектура забезпечує чітке розмежування відповідальностей та полегшує розробку та тестування різних компонентів паралельно.

2. Мікросервісна архітектура:

Ця архітектура передбачає розбиття системи на набір невеликих, автономних та незалежно розгортаних сервісів. Кожен сервіс має свій власний каталог або репозиторій, що містить усі необхідні файли та залежності для його функціонування. Структура файлів відображає цю модульність, де кожен сервіс має власну ізольовану структуру файлів. Ця архітектура забезпечує гнучкість, масштабованість та незалежність розробки та розгортання різних сервісів.

3. Монолітна архітектура:

Монолітна архітектура є одним із традиційних підходів до розробки програмного забезпечення, і для реалізації проекту інформаційної системи обліку товарів продуктового магазину була обрана саме ця архітектура. У монолітній архітектурі вся система розробляється як єдиний функціональний блок, де всі компоненти, такі як інтерфейс користувача, обробка даних, доступ до бази даних та інші функції, тісно інтегровані одна з одною і працюють як єдине ціле. Це означає, що весь код системи знаходиться в одному репозиторії або каталозі і розгортається як єдиний модуль.

Для реалізації проекту було обрано монолітну архітектуру. Монолітна архітектура - це традиційний підхід до розробки програмного забезпечення, при якому вся система будується як єдиний функціональний модуль. Всі компоненти системи тісно зв'язані між собою і працюють як єдине ціле.[16]

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		43

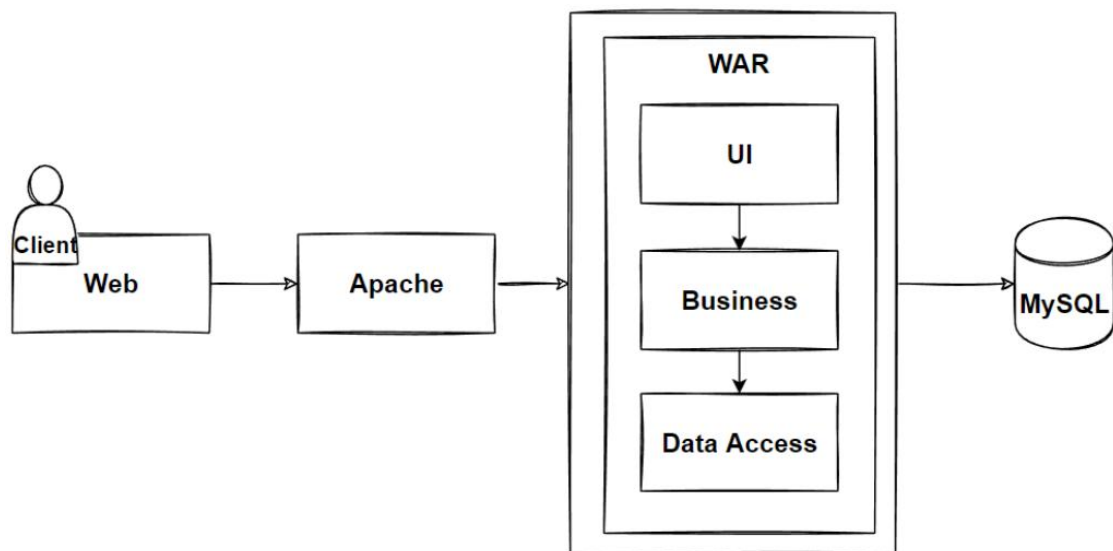


Рис. 2.4. Монолітна архітектура

Всі компоненти системи тісно зв'язані між собою і працюють як єдине ціле. Зазвичай, монолітна система складається з трьох основних частин (див. Рис 2.4):

1. Користувацький інтерфейс (UI) - відповідає за взаємодію з користувачем.
2. Бізнес-логіка (BL) - містить основну функціональність системи і алгоритми обробки даних.
3. Доступ до даних (DAL) - забезпечує зв'язок з базою даних для збереження і отримання інформації.

При виборі архітектури для проекту "Інформаційна система обліку товарів продуктового магазину" важливо враховувати ряд факторів, таких як складність системи, необхідність масштабування, досвід команди розробників і часові обмеження. Після ретельного аналізу вимог до проекту, монолітна архітектура була обрана як найбільш підходящий варіант.

Давайте розглянемо основні переваги монолітної архітектури для даного проекту[17]:

3 Швидкість розробки:

Монолітна архітектура дозволяє швидко розробляти додаток завдяки простоті роботи з єдиною кодовою базою. Це особливо важливо для невеликих команд і проектів з обмеженим часом на реалізацію.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		44

4 Зручність розгортання:

Монолітний додаток складається з одного виконуваного файлу або директорії, що значно спрощує процес розгортання. Немає необхідності в складній координації між різними сервісами і їх версіями.

5 Легкість розробки:

Коли додаток будується на основі єдиної кодової бази, його простіше розробляти. Розробники можуть швидко зрозуміти структуру проекту і вносити зміни без необхідності розбиратися у взаємодії між різними сервісами.

6 Висока продуктивність:

У монолітній архітектурі з централізованою кодовою базою і репозиторієм один API часто може виконувати ту ж функцію, що і кілька API в мікросервісній архітектурі. Це дозволяє уникнути накладних витрат на міжсервісну комунікацію і підвищити продуктивність системи.

7 Спрощене тестування:

Оскільки монолітний додаток є єдиним централізованим модулем, наскрізне тестування (end-to-end testing) може бути виконане швидше, ніж у випадку з розподіленим додатком. Немає необхідності налаштовувати і координувати тестування для кожного окремого сервісу.

8 Легкість відлагодження:

При монолітній архітектурі весь код знаходиться в одному місці, що полегшує відстеження запитів і пошук проблем. Можливо швидко знайти і виправити помилки без необхідності аналізувати взаємодію між різними сервісами.

З огляду на ці переваги, для проекту інформаційної системи обліку товарів продуктового магазину була обрана монолітна архітектура. Відповідно, структура файлів системи відображає цей підхід, де всі компоненти та модулі системи розміщені в єдиному каталозі.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		45

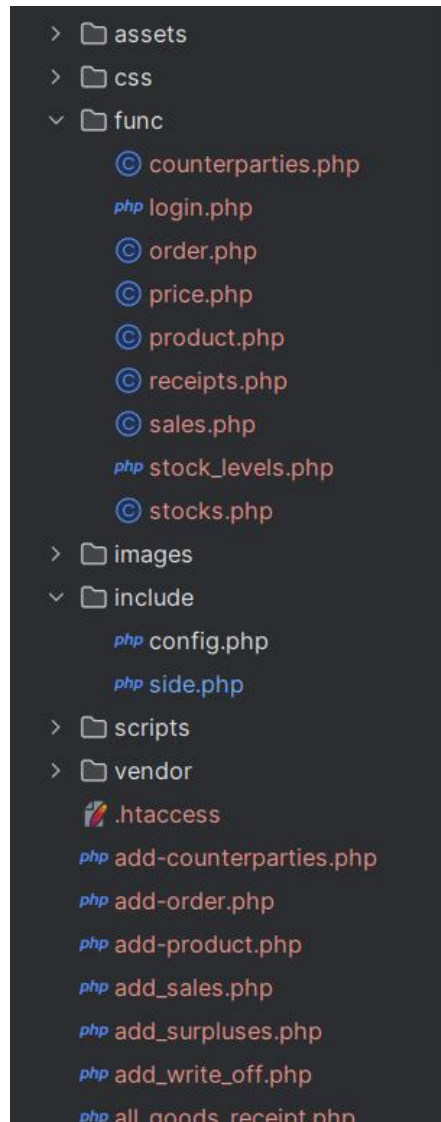


Рис. 2.5 Структура проекту

Структура файлів проекту організована таким чином, щоб забезпечити чіткість та зручність розробки. Кореневий каталог проекту містить основні директорії, такі як "assets", "css", "func", "images", "include", "scripts" та "vendor", які відповідають за різні аспекти системи(див. Рис. 2.5).

Директорія "assets" призначена для зберігання додаткових ресурсів, таких як зображення, шрифти або інші файли, необхідні для коректного функціонування та оформлення системи. Директорія "css" містить файли каскадних таблиць стилів (CSS), які визначають зовнішній вигляд і стилізацію елементів інтерфейсу.

У директорії "func" знаходиться основна функціональність системи. Вона містить ключові файли, такі як "counterparties.php", "login.php", "order.php",

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		46

"price.php", "product.php", "receipts.php", "sales.php", "stock_levels.php" та "stocks.php". Кожен з цих файлів відповідає за певну частину функціональності системи, наприклад, управління контрагентами, авторизацію користувачів, обробку замовлень, ціноутворення, управління товарами, облік надходжень та продажів, контроль рівнів запасів тощо.

Директорія "images" призначена для зберігання зображень, які використовуються в системі. Вона може містити логотипи, іконки та інші графічні елементи, необхідні для візуальної частини інтерфейсу.

Директорія "include" містить файли, які включені в інші частини системи. Зазвичай тут розміщується файли конфігурації, такі як "config.php", а також допоміжні файли, наприклад, "side.php", який може відповідати за бічну панель.

Директорія "scripts" призначена для зберігання скриптів, таких як JavaScript-файли, які відповідають за інтерактивність та динамічну поведінку елементів інтерфейсу на стороні клієнта.

Директорія "vendor" зазвичай використовується для розміщення сторонніх бібліотек або фреймворків, які можуть бути використані в проекті. Вона містить файли, які не є безпосередньо частиною основного коду проекту, але необхідні для його коректного функціонування.

Крім того, в корені проекту знаходиться директорія ".htaccess", яка містить файли конфігурації веб-сервера Apache. Ці файли дозволяють налаштовувати певні аспекти роботи веб-сервера, такі як перенаправлення URL, обробку помилок, налаштування безпеки тощо.

Така структура файлів забезпечує чітку організацію коду, розділення функціональності та зручність розробки та супроводу проекту.

Висновки до другого розділу

У другому розділі було детально розглянуто процес проектування та розробки інформаційної системи обліку товарів продуктового магазину. Було визначено варіанти використання системи, розроблено модель бази даних та структуру файлів системи.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		47

Визначення варіантів використання системи дозволило чітко окреслити функціональні вимоги та можливості системи для двох основних ролей користувачів: адміністратора та менеджера. Use Case діаграми наочно продемонстрували взаємодію користувачів з системою та основні операції, які вони можуть виконувати.

Розробка моделі бази даних була ключовим етапом у створенні ефективної та надійної системи обліку товарів. Було спроектовано та розроблено базу даних, яка складається з 22 взаємопов'язаних таблиць, кожна з яких відповідає за зберігання специфічної інформації. Детальний опис таблиць та їх полів дозволив зрозуміти структуру та призначення кожного елемента бази даних.

Для реалізації проекту було обрано монолітну архітектуру, яка найкраще відповідає вимогам та обмеженням проекту. Було розглянуто основні переваги монолітної архітектури, такі як швидкість розробки, зручність розгортання, легкість розробки, висока продуктивність, спрощене тестування та легкість відлагодження.

Структура файлів системи була розроблена з урахуванням принципів чіткості, модульності та зручності розробки. Кореневий каталог проекту містить основні директорії, такі як "assets", "css", "func", "images", "include", "scripts" та "vendor", кожна з яких відповідає за певний аспект системи. Детальний опис призначення кожної директорії дозволив зрозуміти організацію коду та розділення функціональності в системі.

Таким чином, другий розділ охоплює ключові аспекти проектування та розробки інформаційної системи обліку товарів продуктового магазину. Визначення варіантів використання, розробка моделі бази даних та вибір монолітної архітектури закладають міцний фундамент для подальшої реалізації та впровадження системи.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВІНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		48

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ ТОВАРІВ ПРОДУКТОВОГО МАГАЗИНУ

3.1 Реалізація системи обліку товарів

Після ретельного проектування структури бази даних, архітектури системи, наступним ключовим етапом розробки інформаційної системи обліку товарів продуктового магазину є безпосередня реалізація її основних модулів та компонентів.

На цьому етапі всі попередні проектні рішення та специфікації втілюються у функціонуючу програмну систему. Відповідно до спроектованої архітектури, система розбивається на окремі модулі, кожен з яких відповідає за певну бізнес-логіку чи набір функцій.

Розглянемо ключові етапи роботи та функціональні можливості застосунку.

Процес входу відбувається таким чином:

Спочатку перевіряється, чи користувач уже авторизований, за допомогою функції `isLoggedIn()`, якщо користувач уже авторизований, він буде перенаправлений на `dashboard.php`. Далі, система отримує ім'я користувача та пароль з форми входу і виконується функція `login($username, $password)`

```
public function login($username, $password)
{
    $password = md5($password); // Хешування пароля за допомогою MD5
    $query = mysqli_query($this->con, "SELECT users.*, role.name AS role_name
                                      FROM users
                                      JOIN role ON users.role = role.id
                                      WHERE users.username = '$username' AND users.password = '$password'");

    $user = mysqli_fetch_array($query);

    if ($user) {
        $_SESSION['user_name'] = $user['username'];
        $_SESSION['user_id'] = $user['id'];
        $_SESSION['user_role'] = $user['role_name'];
        return true;
    } else {
        return false;
    }
}
```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		49

Ця функція хешує пароль користувача за допомогою алгоритму MD5, виконує запит до бази даних для пошуку користувача з введеними ім'ям користувача та захешованим паролем, а також отримує назву ролі користувача. Якщо користувач знайдено, функція зберігає ім'я користувача, ідентифікатор користувача та назву ролі в сесії і повертає true.

Після виконання функції login(\$username, \$password), перевіряється результат, Якщо вхід успішний, користувач перенаправляється на dashboard.php. В іншому випадку, встановлюється повідомлення про помилку в \$_SESSION['errmsg'], і користувач перенаправляється назад на index.php (сторінку входу).

Процес створення замовлення для постачальника в цій системі реалізований у функції createOrder класу Order (див. Додаток Б). Функція приймає такі параметри: номер замовлення, дату замовлення, ідентифікатор контрагента (постачальника), ідентифікатор складу, на який буде поставлено товари, та масив з даними про вибрані товари, що включає ідентифікатор товару, кількість та ціну.

```
public function createOrder($number, $data, $counterparties, $warehouse, $selectedProducts) {
    if (!empty($selectedProducts)) {
        $products = array();
        $totalSum = 0;

        foreach ($selectedProducts as $product) {
            $productId = $product['id'];
            $quantity = $product['quantity'];
            $price = $product['price'];

            $products[] = array(
                'id' => $productId,
                'quantity' => $quantity,
                'price' => $price
            );

            $totalSum += $quantity * $price;
        }

        $productsJson = json_encode($products);

        $stmt = mysqli_prepare($this->con, "INSERT INTO order_to_supplier (number, date, id_counterparties,
id_warehouse, products, sum) VALUES (?, ?, ?, ?, ?, ?)");
        mysqli_stmt_bind_param($stmt, "ississ", $number, $data, $counterparties, $warehouse, $productsJson,
$totalSum);

        if (mysqli_stmt_execute($stmt)) {
            return true;
        } else {

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

        return false;
    }

    mysqli_stmt_close($stmt);
} else {
    return false;
}
}

```

Спочатку функція перевіряє, чи вибрано хоча б один товар. Якщо вибрано, вона створює масив \$products, який буде містити інформацію про вибрані товари та їх ціни. Далі обчислюється загальна сума замовлення. Потім масив \$products конвертується в JSON-рядок. Після цього підготовлюється SQL-запит для вставки даних у таблицю order_to_supplier з використанням prepared statements для захисту від SQL-ін'єкцій.

Процес додавання товарів до замовлення за допомогою AJAX реалізований в наступному уривку коду:

```

$(document).ready(function() {
    $('#select-products').click(function() {
        $('#product-modal').modal('show');
    });

    $('.add-product').click(function() {
        var productId = $(this).data('id');
        var productName = $(this).data('name');

        var existingProduct = $('#selected-product[data-id="' + productId + '"]');

        if (existingProduct.length > 0) {
            var quantityInput = existingProduct.find('.product-quantity');
            var currentQuantity = parseInt(quantityInput.val());
            quantityInput.val(currentQuantity + 1);
        } else {
            var productHtml = '<tr class="selected-product" data-id="' + productId + '">' +
                '<td><input type="hidden" name="selected_products[' + productId + '][id]" value="' +
                productId + '">' + productName + '</td>' +
                '<td><input type="number" name="selected_products[' + productId + '][quantity]"' +
                'class="product-quantity" value="1" min="1"></td>' +
                '<td><input type="number" name="selected_products[' + productId + '][price]" class="product-' +
                'price" min="0" step="0.01"></td>' +
                '<td><span class="remove-product" data-id="' + productId + '"><i class="fa fa-' +
                'trash"></i></span></td>' +
                '</tr>';

            $('#selected-products-table tbody').append(productHtml);
        }

        $('#product-modal').modal('hide');
    });

    $(document).on('click', '.remove-product', function() {
        var productId = $(this).data('id');
        $('#selected-product[data-id="' + productId + '"]').remove();
    });
});

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		51

});

Цей код дозволяє вибирати товари для замовлення через модальне вікно та динамічно додавати їх до таблиці вибраних товарів на тій самій сторінці без перезавантаження. При виборі товару код перевіряє його наявність у таблиці та збільшує кількість або додає новий рядок з полями для ідентифікатора, назви, кількості, ціни та видалення товару. Після додавання або збільшення кількості модальне вікно закривається. При натисканні на "Видалити" відповідний рядок видаляється з таблиці. Користувач може вибирати та видаляти товари без перезавантаження сторінки.

Використання AJAX дозволяє забезпечити плавний та зручний інтерфейс для вибору товарів, підвищуючи зручність використання системи та роблячи процес створення замовлення більш інтерактивним та ефективним.

Для обрахування ціни на товар від надходження в системі обліку товарів використовуються функції з класу Price (див. Додаток Б). Розглянемо детальніше процес обрахування ціни та функції, які задіяні в цьому процесі.

```
public function addReceipt($date, $orderId, $warehouseId, $counterpartyId,
$products)
{
    // ...
    foreach ($products as $product) {
        $this->updateStocks($product, $date, $warehouseId, $counterpartyId);
        $this->price->updatePrice($product['id'], $product['price']);
    }
    // ...
}
```

При надходженні нового товару на склад, викликається функція addReceipt() з класу Receipts (див. Додаток Б). Ця функція приймає параметри, серед яких є масив \$products, що містить інформацію про кожен товар, включаючи його ідентифікатор, кількість, ціну та термін придатності.

Для кожного товару в масиві \$products викликається функція updatePrice() з класу Price, передаючи ідентифікатор товару (\$product['id']) та ціну закупівлі (\$product['price']).

```
public function updatePrice($productId, $purchasePrice) {
    // Отримання залишків на складі для вказаного id_product
    $stockQuery = "SELECT quantity, purchase_price FROM stocks WHERE id_product =
'$productId'";
    $stockResult = mysqli_query($this->con, $stockQuery);
```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		52

```

$totalQuantity = 0;
$totalPurchasePrice = 0;

while ($row = mysqli_fetch_assoc($stockResult)) {
    $quantity = $row['quantity'];
    $purchasePriceFromStock = $row['purchase_price'];

    $totalQuantity += $quantity;
    $totalPurchasePrice += $quantity * $purchasePriceFromStock;
}

// Додавання поточної ціни надходження до розрахунку середньої вартості
$totalQuantity++;
$totalPurchasePrice += $purchasePrice;

// Розрахунок середньої вартості (costPrice)
$costPrice = ($totalQuantity != 0) ? $totalPurchasePrice / $totalQuantity :
$purchasePrice;

$marginPrice = $this->calculateMarginPrice($costPrice);
$pdvPrice = $this->calculatePdvPrice($marginPrice);

// Перевірка наявності запису з вказаним id_product
$checkQuery = "SELECT COUNT(*) FROM price WHERE id_product = '$productId'";
$result = mysqli_query($this->con, $checkQuery);
$count = mysqli_fetch_row($result)[0];

if ($count > 0) {
    // Запис існує, виконуємо оновлення
    $updateQuery = "UPDATE price SET
        cost_price = '$costPrice',
        margin_price = '$marginPrice',
        pdv_price = '$pdvPrice'
        WHERE id_product = '$productId'";
    mysqli_query($this->con, $updateQuery);
} else {
    // Запис не існує, виконуємо вставку нового запису
    $insertQuery = "INSERT INTO price (id_product, cost_price, margin_price,
pdv_price)
        VALUES ('$productId', '$costPrice', '$marginPrice', '$pdvPrice')";
    mysqli_query($this->con, $insertQuery);
}
}

```

Функція updatePrice() відіграє ключову роль у процесі обрахування ціни на товар від надходження. Вона приймає ідентифікатор товару (\$productId) та ціну закупівлі (\$purchasePrice) як параметри. Спочатку функція отримує залишки на складі для вказаного id_product з таблиці stocks. Потім вона розраховує загальну кількість товару (\$totalQuantity) та загальну вартість закупівлі (\$totalPurchasePrice) на основі залишків на складі. Далі функція додає поточну ціну надходження до розрахунку середньої вартості та розраховує середню вартість товару (\$costPrice) шляхом ділення загальної вартості

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		53

закупівлі на загальну кількість товару. Після цього викликаються функції `calculateMarginPrice()` та `calculatePdvPrice()` для розрахунку ціни з урахуванням маржі (`$marginPrice`) та ПДВ (`$pdvPrice`) відповідно. Нарешті, функція перевіряє наявність запису для вказаного `id_product` в таблиці `price`. Якщо запис існує, виконується оновлення ціни в таблиці `price` з новими значеннями `$costPrice`, `$marginPrice` та `$pdvPrice`. Якщо запис не існує, виконується вставка нового запису в таблицю `price` з відповідними значеннями.

Функції `calculateMarginPrice()` та `calculatePdvPrice()` використовуються для розрахунку ціни з урахуванням маржі та ПДВ відповідно.

```
private function calculateMarginPrice($costPrice, $margin = 0.2) {
    return $costPrice * (1 + $margin);
}

private function calculatePdvPrice($marginPrice, $pdv = 0.2) {
    return $marginPrice * (1 + $pdv);
}
```

Таким чином, при надходженні нового товару на склад, система обліку товарів автоматично розраховує середню вартість товару на основі попередніх залишків та поточної ціни закупівлі, а також обраховує ціну з урахуванням маржі та ПДВ. Ці дані зберігаються в таблиці `price` для подальшого використання в системі.

Важливим елементом системи обліку товарів є документи та звіти. Одним із таких звітів є "Звіт по продажам", який дозволяє отримати детальну інформацію про продажі товарів за певний період часу. Розглянемо процес генерування цього звіту та функції, які використовуються для його створення.

Для генерування "Звіту по продажам" використовується метод `getSalesReport()` з класу `Sales` (див. Додаток Б). Цей метод приймає параметр `$date`, який вказує на дату, за яку потрібно отримати звіт.

```
public function getSalesReport($date) {
    $query = "SELECT p.article, p.productName, p.units_id, s.products, s.payment
    FROM sales s
    INNER JOIN products p ON JSON_EXTRACT(s.products, '$[*].id_product') LIKE CONCAT('%', p.id, '%')
    WHERE DATE(s.date) = '$date'";
    $result = mysqli_query($this->con, $query);
    return $result;
}
```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		54

Метод `getSalesReport()` виконує SQL-запит до бази даних, який отримує інформацію про продажі за вказану дату. Запит використовує з'єднання таблиць `sales` та `products` за допомогою оператора `INNER JOIN`. Умова з'єднання порівнює `id_product` з таблиці `products` з `id_product` з таблиці `sales`, які зберігаються у форматі JSON. Результат запиту повертається у вигляді об'єкта `mysqli_result`.

Після отримання результату запиту, дані обробляються та групуються за певними критеріями. Для цього використовується цикл `while`, який проходить по кожному рядку результату та групує товари за їхніми ідентифікаторами.

```
while ($row = mysqli_fetch_assoc($result)) {
    $article = $row['article'];
    $productName = $row['productName'];
    $unitsId = $row['units_id'];
    $productsJson = $row['products'];
    $products = json_decode($productsJson, true);

    $unitsName = $productObj->getUnitName($unitsId);

    foreach ($products as $product) {
        $productId = $product['id_product'];
        $productQuantity = $product['quantity'];
        $price = $product['price'];
        $discountPrice = $product['discount_price'];
        $sale = $product['sale'];

        if (!isset($groupedProducts[$productId])) {
            $groupedProducts[$productId] = array(
                'article' => $article,
                'productName' => $productName,
                'unitsName' => $unitsName,
                'quantity' => 0,
                'totalAmount' => 0,
                'totalDiscount' => 0,
                'totalAmountWithDiscount' => 0,
                'totalCost' => 0
            );
        }

        $groupedProducts[$productId]['quantity'] += $productQuantity;

        if ($discountPrice !== "") {
            $groupedProducts[$productId]['totalAmount'] += $price * $productQuantity;
            $groupedProducts[$productId]['totalDiscount'] += $sale * $productQuantity;
            $groupedProducts[$productId]['totalAmountWithDiscount'] += $discountPrice * $productQuantity;
        } else {
            $groupedProducts[$productId]['totalAmount'] += $price * $productQuantity;
            $groupedProducts[$productId]['totalAmountWithDiscount'] += $price * $productQuantity;
        }

        $costData = $priceObj->getProductCostPrice($productId);
        $costPrice = $costData['cost_price'];
        $marginPrice = $costData['margin_price'];
    }
}
```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		55

```

    $groupedProducts[$productId]['totalCost'] += $costPrice * $productQuantity;
    $groupedProducts[$productId]['marginPrice'] = $marginPrice;
}
}

```

Для кожного продажу товару збільшується загальна кількість проданих одиниць, обчислюється сума продажу, сума знижки (якщо є) та сума продажу зі знижкою. Також для кожного товару отримується його собівартість за допомогою методу getProductCostPrice() з класу Price, і обчислюється загальна собівартість проданих товарів.

Після групування даних, формується HTML-таблиця зі звітом по продажам. Кожен рядок таблиці відображає інформацію про проданий товар, включаючи артикул, назву товару, одиницю виміру, продану кількість, суму продажу, надану знижку, суму продажу з урахуванням знижки, собівартість проданого товару, маржинальний дохід та маржинальний дохід з ПДВ.

Для кожного проданого товару розраховується маржинальний дохід та маржинальний дохід з ПДВ. Маржинальний дохід обчислюється як різниця між сумою продажу з урахуванням знижки та собівартістю проданого товару, а маржинальний дохід з ПДВ - як різниця між сумою продажу з урахуванням знижки та собівартістю проданого товару з додаванням ПДВ.

Сформована HTML-таблиця зберігається в змінній \$html, яка потім використовується для генерації PDF-файлу за допомогою бібліотеки MPDF.

У системі реалізовано гнучкий механізм ціноутворення, який дозволяє автоматично розраховувати ціну товару при надходженні на основі середньозваженої собівартості, маржі та ПДВ. Також передбачено можливість встановлення знижок на товари та відображення актуальних цін з урахуванням знижок.

Для зручності користувачів система надає можливість генерування звітів у форматі PDF, що дозволяє легко переглядати та роздруковувати звіти за потреби.

Під час реалізації системи обліку товарів було приділено увагу безпеці даних та запобіганню несанкціонованого доступу. Використано механізми

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		56

аутентифікації користувачів та обмеження доступу до певних функцій системи залежно від ролі користувача.

Розроблена система обліку товарів забезпечує ефективне управління товарами, облік надходжень та продажів, а також надає можливість генерування детальних звітів для аналізу та прийняття управлінських рішень. Завдяки модульній структурі та використанню сучасних веб-технологій, система є гнучкою та легко розширюваною для подальшого вдосконалення та додавання нових функціональних можливостей.

3.2 Структура інтерфейсу системи обліку товарів

Зручний та інтуїтивно зрозумілий користувацький інтерфейс відіграє ключову роль у забезпеченні ефективності та продуктивності роботи з інформаційною системою обліку товарів продуктового магазину. Правильно спроектована структура інтерфейсу дозволяє користувачам легко знаходити необхідні функції, швидко виконувати основні операції та отримувати потрібну інформацію в зручному форматі. У цьому розділі буде детально розглянуто структуру інтерфейсу системи, включаючи опис різних екранів, панелей, форм та елементів керування.

У цьому розділі ми детально розглянемо структуру інтерфейсу інформаційної системи обліку товарів продуктового магазину з точки зору адміністратора. Адміністратор є ключовою роллю в системі, яка має повний доступ до всіх функцій та можливостей управління. Тому структура інтерфейсу адміністратора повинна забезпечувати зручний і всеосяжний контроль над різними аспектами системи.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		57

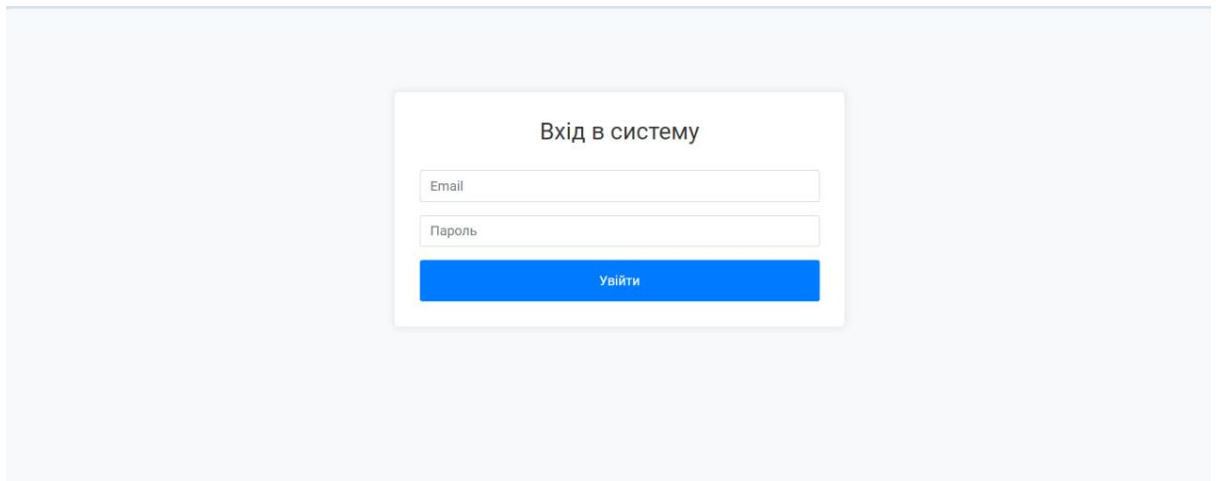


Рис. 3.1 - Сторінка Входу

Сторінка входу (рис. 3.1) є першим інтерфейсом, з яким взаємодіє користувач при доступі до онлайн-платформи управління проєктами. Вона містить поля для введення електронної пошти та пароля, що дозволяє користувачу ідентифікувати себе в системі. Після успішної автентифікації, користувач отримує доступ до основних функціональних можливостей платформи.

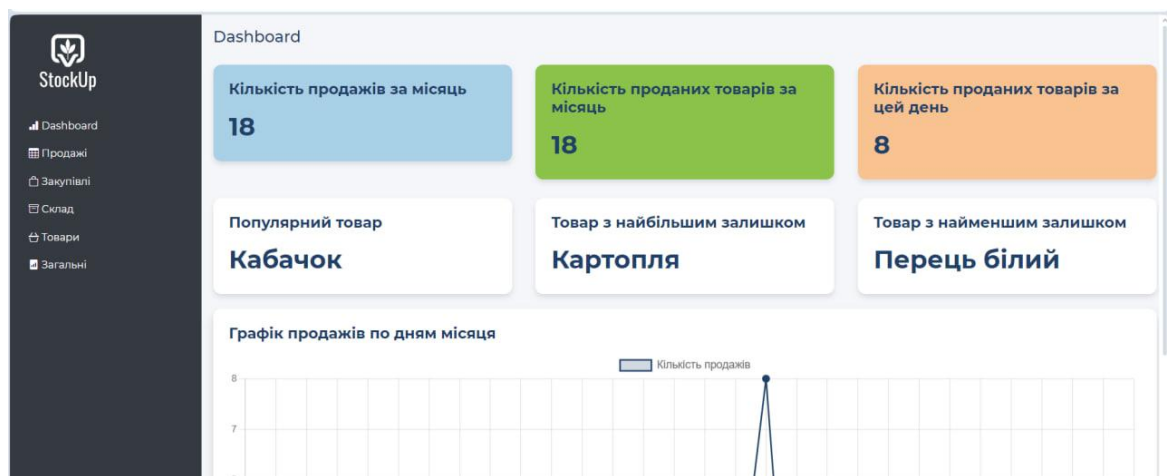


Рис. 3.2 - Dashboard

Після успішного входу в систему, користувач потрапляє на панель управління (Dashboard) (рис. 3.2). Це головний інтерфейс платформи, який надає загальний огляд основних показників та статистики щодо управління проєктами.

На Dashboard відображається наступна інформація: кількість продажів за місяць, кількість проданих товарів за місяць, кількість проданих товарів за день,

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		58

популярний товар, товар з найбільшим залишком, товар з найменшим залишком.

Панель управління (Dashboard) забезпечує зручний і наочний огляд ключових показників, що допомагає керівництву приймати обґрунтовані рішення щодо управління товарними запасами та оптимізації процесів продажів.

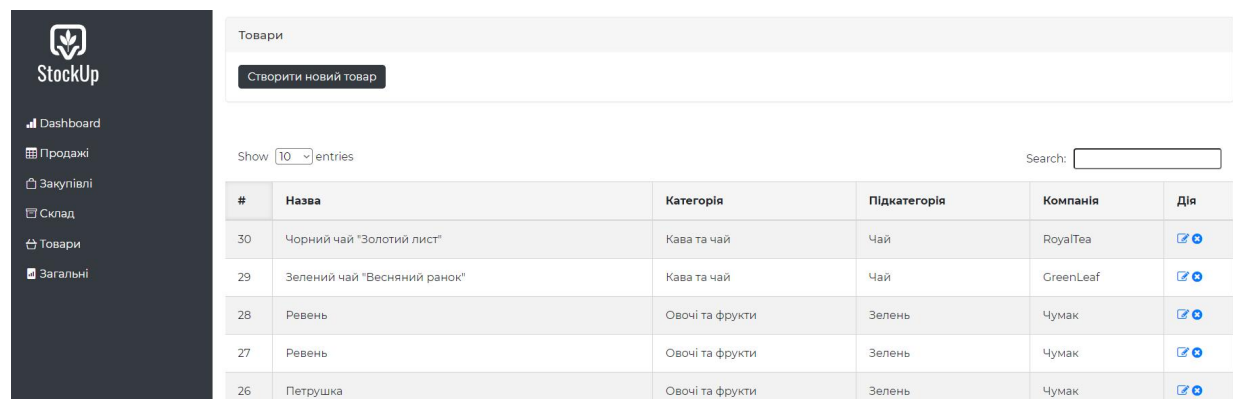


Рис.3.3 - Сторінка товарів

Дана сторінка (див. Рис. 3.3) відображає перелік всіх товарів, доступних на платформі. Вона дозволяє здійснювати пошук, перегляд, додавання, редагування та видалення товарів.

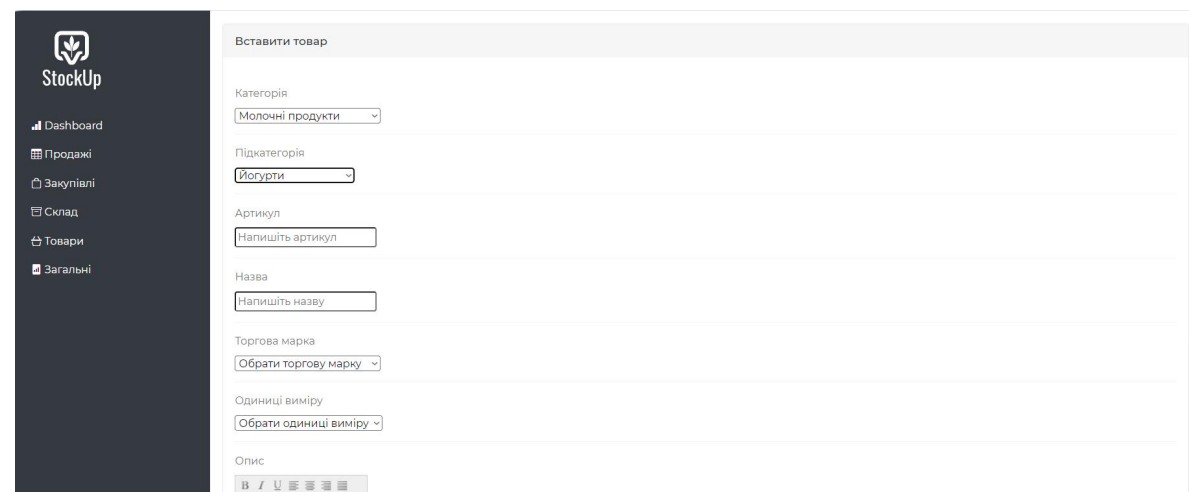
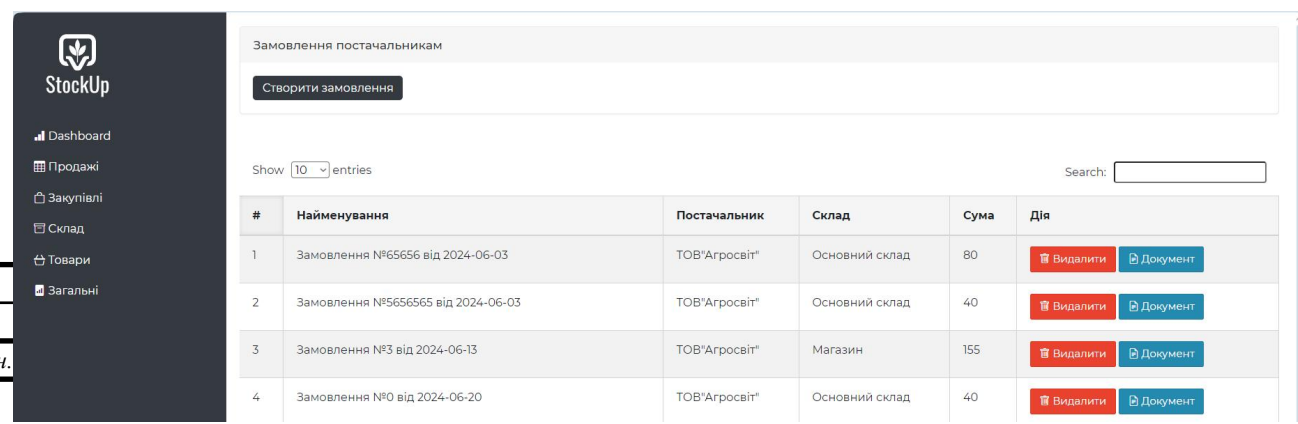


Рис. 3.4 - Сторінка додавання товару

Дана сторінка (див. Рис. 3.4) призначена для створення нового товару в системі. Вона містить форму із полями для введення всієї необхідної інформації



про товар, такої як назва, категорія, постачальник, ціна, опис тощо.

Рис. 3.5 - Сторінка “Замовлення до постачальника”

StockUp

Dashboard

Продажі

Закупівлі

Склад

Товари

Загальні

Замовлення постачальнику

№

000009

Дата

18.06.2024

Постачальник

ТОВ"Агросвіт"

Склад

Магазин

Вибрати товари

Назва товару	Кількість	Ціна
Зелений чай	1	55
Чорний чай	1	60

Зберегти

Рис. 3.6 - Сторінка “Створення замовлення до постачальника”

На рис. 3.5 зображено сторінку "Замовлення до постачальника" в інформаційній системі обліку товарів продуктового магазину. Ця сторінка призначена для перегляду списку замовлень на постачання товарів. В таблиці відображається наступна інформація по кожному замовленню: його найменування, постачальник, склад, на який має бути доставлено товар, сума замовлення та дії, які можна виконати (перегляд деталей та друк замовлення).

Для створення нового замовлення постачальнику користувач повинен перейти на сторінку "Створення замовлення до постачальника", що представлена на рис. 3.6. На цій сторінці потрібно ввести номер замовлення, вибрати дату його створення, постачальника та склад, на який буде доставлено товар. В таблиці нижче вводиться інформація по кожній позиції замовлення: найменування товару, кількість та ціна. При додаванні нових позицій автоматично розраховується загальна сума замовлення. Після введення всіх необхідних даних користувач натискає кнопку "Зберегти" для збереження замовлення в системі.

- Dashboard
- Продажі
- Закупівлі
- Склад
- Товари
- Загальні

Надходження

Додати надходження

Show 10 entries

Search:

#	Номер надходження	Замовлення	Склад	Дата	Дія
1	Надходження №GR49270	Замовлення №65656	Основний склад	2024-06-03	Видалити Документ
2	Надходження №GR48932	Замовлення №5656565	Основний склад	2024-06-14	Видалити Документ
3	Надходження №GR29800	Замовлення №5656565	Основний склад	2024-06-07	Видалити Документ
4	Надходження №GR19557	Замовлення №5656565	Основний склад	2024-06-07	Видалити Документ

Рис. 3.7 - Сторінка “Надходження замовлення”

Сторінка "Надходження замовлення" (див. Рис. 3.7) відображає список надходжень товарів від постачальників за різними замовленнями. Для кожного надходження вказано його номер, замовлення, до якого воно відноситься, склад, на який надійшов товар, дата надходження. Присутні кнопки для перегляду деталей та друку документа про надходження.

Сторінка "Надходження товару" (див. Рис. 3.8) призначена для реєстрації в системі факту надходження конкретного товару. На ній зазначається дата надходження, вибирається замовлення, до якого відноситься даний товар, зі списку раніше створених замовлень постачальникам. В таблиці нижче вводиться інформація безпосередньо по кожній позиції товару, що надійшов: його найменування, кількість, ціна та термін придатності. Після збереження цих даних кількість доступного товару на складі збільшується.

- Dashboard
- Продажі
- Закупівлі
- Склад
- Товари
- Загальні

Надходження товару

Дата: 18.06.2024

Замовлення: Замовлення №15 від 2024-06-18

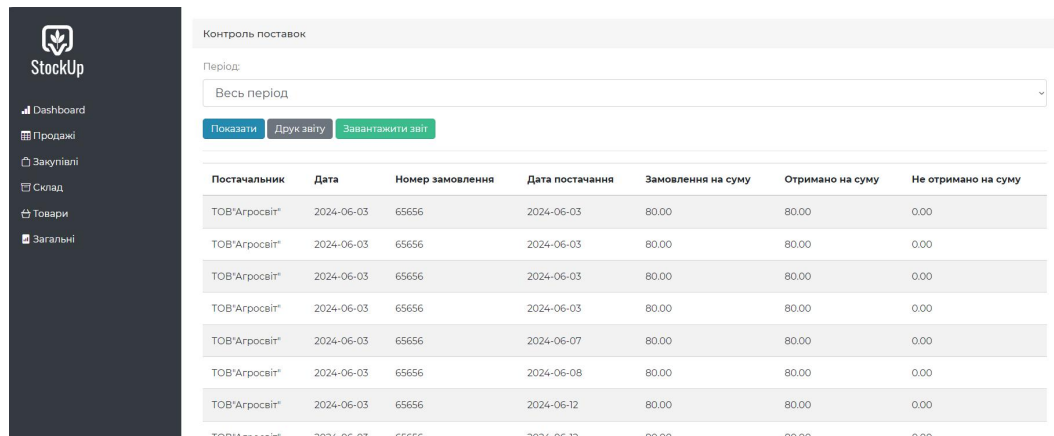
Постачальник: ТОВ

Склад: Магазин

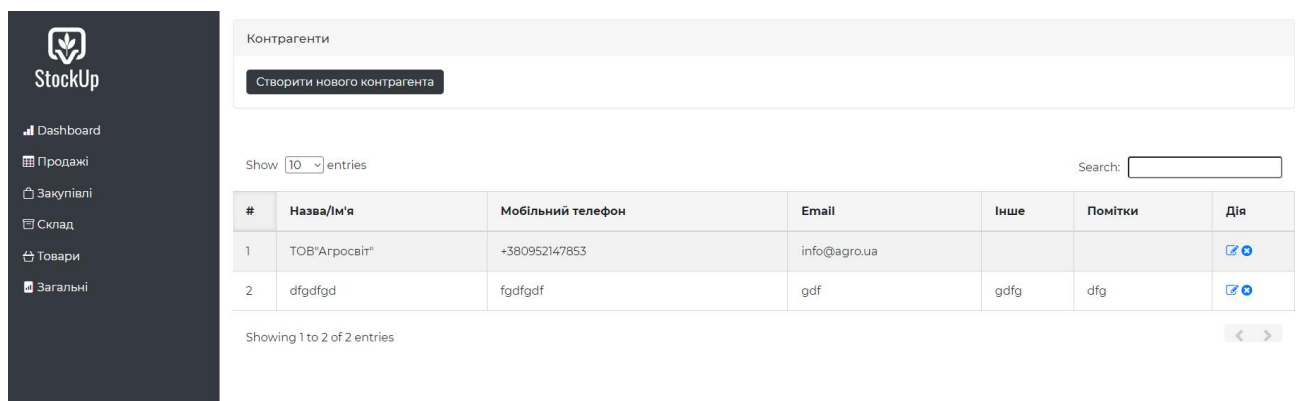
Товар	Кількість	Ціна	Термін придатності
Зелений чай "Весняний ранок"	1	55	14.06.2025
Чорний чай "Золотий лист"	1	65	07.06.2025

Зберегти

Рис. 3.8 - Сторінка “Надходження товару”



На рис. 3.9 представлено сторінку "Контроль поставок", яка представляє детальну інформацію про постачання товарів до продуктового магазину. Вона містить дані про постачальника, дати поставок, номери замовлень, дати отримання, суми замовлених та отриманих товарів, а також суми неотриманих товарів.



На рис. 3.10 представлено сторінку "Постачальники" цієї ж інформаційної системи. Вона містить довідник всіх постачальників, з якими співпрацює продуктовий магазин. По кожному з них вказані наступні дані: назва компанії-постачальника, контактний мобільний телефон, email-адреса, ім'я контактної особи та посилання на сайт компанії. Передбачена можливість додавання нових постачальників та редагування даних по вже існуючим.

Dashboard

Продажі

Закупівлі

Склад

Товари

Загальні

Інвентаризації

Склад

Магазин

Категорії

Овочі та фрукти

М'ясо та риба

Сири

М'ясо-ковбасні вироби

Хліб

Молочні продукти

Бакалія

Солодощі

Кава та чай

Напої

Спеції

sdzsdasz

yyyy

Зелений чай "Весняний ранок"

Партія	Облікова кількість	Фактична кількість	Ціна	Сума	Од. виміру
Партія 18.06.2024	1	2	55	55	уп

Чорний чай "Золотий лист"

Партія	Облікова кількість	Фактична кількість	Ціна	Сума	Од. виміру
Партія 18.06.2024	1	1	65	65	уп

Зберегти

Списання

Оприбуткування

Рис. 3.11 - Сторінка “Інвентаризації”

Сторінка (див. Рис. 3.11) містить форму для введення інформації про інвентаризацію товарів. Форма включає поля для вказівки партії, обігової кількості, фактичної кількості і, одиниці вимірювання та суми.

Звіт (див. Рис. 3.12) надає підсумкову інформацію про результати інвентаризації. Для кожної позиції товару наведено дані про партію, облікову кількість, фактичну кількість, одиницю вимірювання та ціну.

Звіт з інвентаризації

Відомість інвентаризації №51 від 2024-06-18

Склад: Магазин - вул. Київська 27

№	Товар	Партія	Облікова кількість	Фактична кількість	Одиниця вимірювання	Ціна
1	Зелений чай "Весняний ранок"	Партія від 18.06.2024	1	2	уп	55
2	Чорний чай "Золотий лист"	Партія від 18.06.2024	1	1	уп	65

Директор

Члени комісії

Рис. 3.12 - Згенерований звіт з інвентаризації

- Dashboard
- Продажі
- Закупівлі
- Склад
- Товари
- Загальні

Аналіз запасів складу

Склад:

Магазин

Показати

Друк звіту

Завантажити звіт

Товар (Назва)	Артикул	Залишок (кількість)	Ціна	Собівартість
Цукіні жовтий		1	20.00	20.00
Картопля		9	55.00	495.00
Зелений чай "Весняний ранок"	000007	2	55.00	110.00
Чорний чай "Золотий лист"	000008	1	65.00	65.00

Загальна сума залишків товарів: 1

Загальна сума собівартості запасів: 690.00

Рис. 3.13 - Сторінка “Аналіз запасів”

Сторінка "Аналіз запасів" (див. Рис. 3.13) в межах інформаційної системи надає огляд поточного стану товарних запасів. На ній представлено детальну інформацію про асортимент товарів, включаючи артикул, наявну кількість, ціну та собівартість кожної позиції. Ця функціональність є ключовою для ефективного управління товарними запасами, оскільки дозволяє керівництву приймати обґрунтовані рішення щодо поповнення запасів, ціноутворення та оптимізації складських операцій.

- Dashboard
- Продажі
- Закупівлі
- Склад
- Товари
- Загальні

Переміщення товару

З якого складу:

Магазин

В який склад:

Основний склад

Додати товар

Назва товару	Кількість	Дата надходження	Дія
Цукіні жовтий	1	2024-06-20	Видалити
Зелений чай "Весняний ранок"	1	2024-06-18	Видалити

Зберегти

Рис. 3.14 - Сторінка “Переміщення”

Сторінка "Переміщення"(див. Рис. 3.14) в межах інформаційної системи обліку товарів продуктового магазину забезпечує можливість переміщення товарів між різними складами. Вона містить форму, де користувач може вибрати магазин, з якого здійснюється переміщення, та основний склад, на який буде проведено переміщення. Нижче розміщено таблицю, яка деталізує інформацію про товари, що підлягають переміщенню, зокрема назву товару, кількість та дату надходження.

- Dashboard
- Продажі
- Закупівлі
- Склад
- Товари
- Загальні

Товари де закінчується термін придатності

Термін придатності:

Завтра

Показати

№	Товар	Категорія	Склад	Термін придатності
1	Імбир	Овочі та фрукти	Склад 2	19.06.2024
2	Цукіні жовтий	Овочі та фрукти	Основний склад	19.06.2024
3	Перець червоний	Овочі та фрукти	Основний склад	19.06.2024

Рис. 3.15 - Сторінка “Товари, де закінчується термін придатності”

Сторінка "Товари, де закінчується термін придатності" (Рис. 3.15) дозволяє відслідковувати товари, в яких закінчується термін придатності, щоб вчасно їх реалізувати або списати.

На сторінці "Створення акції" (див. Рис. 3.16) можна створити акцію для товарів. Є поля для вводу дати початку та закінчення акції, а також відсотку знижки. Внизу сторінки показана поточна ціна товару (стара ціна) та нова ціна зі знижкою.

- Dashboard
- Продажі
- Закупівлі
- Склад
- Товари
- Загальні

Створити акцію

Дата початку:

01.06.2024

Дата завершення:

09.06.2024

Відсоток знижки:

20%

Додати товар

Зелений чай "Весняний ранок"

Стара ціна з ПДВ: 79.5

Нова ціна: 63.55

Створити акцію

Рис. 3.16 - Сторінка “Створення акції”

На цій сторінці (Рис. 3.16) можна створити нову акцію для стимулювання продажів товарів. У полі "Дата початку" потрібно вказати дату, коли акція починається. У полі "Дата завершення" вказується дата закінчення акції. У полі "Відсоток знижки" можна встановити розмір знижки у відсотках, яка буде застосована до товарів, що беруть участь в акції. Нижче показана таблиця з поточною ціною товару (стара ціна) та новою ціною, яка буде застосована після знижки. Це дозволяє відразу бачити, як зміниться ціна товару під час акції.

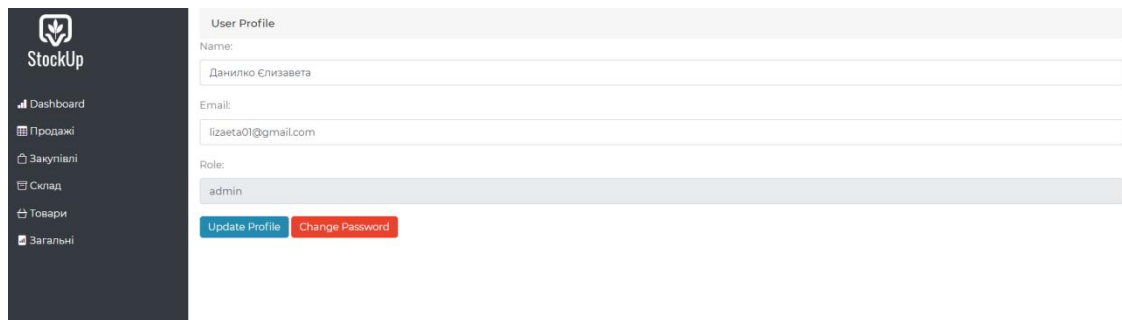
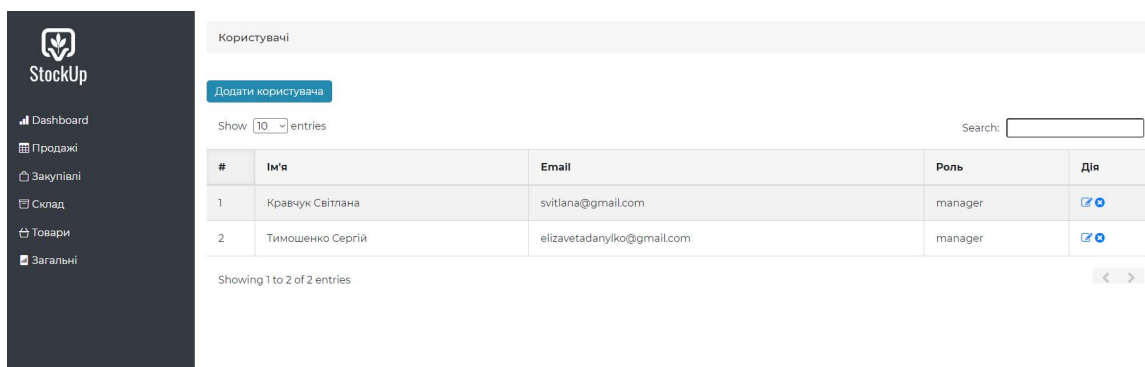


Рис. 3.17 - Сторінка "Профіль"

На сторінці "Профіль" (див. Рис. 3.17) відображається профіль користувача системи. Також, можна переглянути та редагувати особисті дані, такі як ім'я, email та роль користувача. На сторінці є кнопка "Change Password", яка дозволяє змінити пароль користувача для входу в систему.



#	Ім'я	Email	Роль	Дія
1	Кравчук Світлана	svitlana@gmail.com	manager	✎ ✕
2	Тимошенко Сергій	elizavetadanyloko@gmail.com	manager	✎ ✕

Рис. 3.18 - Сторінка "Користувачі"

Сторінка "Користувачі" (Рис. 3.18) відображає список всіх користувачів системи. Для кожного користувача вказано його номер, ім'я, email та роль в системі (адміністратор, менеджер тощо). Адміністратор системи може редагувати ролі користувачів, надаючи їм різні права доступу. Для додавання нового користувача потрібно ввести його email, тоді користувачу прийде лист з посиланням на реєстраційну форму.

Отже, наведені скріншоти та описи демонструють основні функціональні можливості та інтерфейси інформаційної системи обліку товарів продуктового магазину. Система забезпечує всі необхідні інструменти для ефективного управління товарними запасами, відстеження поставок, контролю термінів придатності, створення акцій та знижок, а також керування користувачами та їхніми правами доступу. Завдяки зручному та інтуїтивно зрозумілому

інтерфейсу, ця система допомагає керівництву продуктового магазину приймати обґрунтовані рішення, оптимізувати процеси закупівлі та продажів.

3.3 Тестування та оцінка ефективності роботи системи

Тестування є невід'ємною частиною процесу розробки програмного забезпечення, і воно відіграє ключову роль у забезпеченні якості та надійності інформаційної системи обліку товарів для продуктового магазину. Основна мета тестування полягає у виявленні потенційних помилок, недоліків та проблем у системі перед її впровадженням у реальному середовищі. Існує кілька видів тестування, серед яких: модульне тестування (перевірка окремих компонентів системи), інтеграційне тестування (перевірка взаємодії між компонентами), системне тестування (перевірка системи в цілому) та приймальне тестування (перевірка відповідності системи вимогам замовника).

Для перевірки якості та правильного функціонування інформаційної системи обліку товарів продуктового магазину ручне тестування відіграє важливу роль. Цей вид тестування передбачає безпосередню взаємодію тестувальника з системою, імітуючи дії потенційних користувачів.

Першим кроком буде тестування входу з некоректним паролем. Ми введемо правильний login, але неправильний пароль, і переконаємося, що система відображає відповідне повідомлення про помилку та не надає доступу до системи(див. Рис. 3.19). Також, ми перевіримо спробу входу без введення даних. Ми залишимо поля email та пароль порожніми і натиснемо кнопку "Увійти". У цьому випадку система має відобразити повідомлення про необхідність заповнення обов'язкових полів і не допустити користувача до системи.(див. Рис. 3.20)

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		67

Рис. 3.19 - Попередження про некоректні дані

Ми протестуємо додавання нового користувача з унікальними даними. Ми заповнимо всі необхідні поля, такі як email, роль, і перевіримо, що система успішно створює нового користувача в базі даних(див. Рис. 3.20-3.21). Протестувавши додавання нового користувача з унікальними даними, ми отримали позитивний результат. Система успішно створила нового користувача в базі даних.

Рис. 3.20 - Додавання унікального користувача

Користувачі				
Користувача успішно додано та надіслано реєстраційний email.				
Додати користувача				
Show 10 entries		Search:		
#	Ім'я	Email	Роль	Дія
1		ist201_des@student.ztu.edu.ua	admin	✎ ✕
2	Кравчук Світлана	svitlana@gmail.com	manager	✎ ✕
3	Тимошенко Сергій	elizavetadanylko@gmail.com	manager	✎ ✕
Showing 1 to 3 of 3 entries				

Рис. 3.21 - Створений користувач

Після додавання нового користувача, на email має прийти лист з посиланням на реєстрацію (див. Рис.3.22).

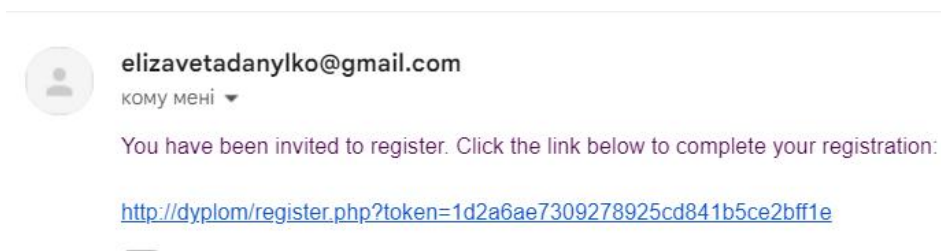


Рис. 3.22 - Надіслений email з посиланням на реєстрацію

Натиснувши на відповідне посилання, ми відкриємо форму реєстрації, де зможемо ввести необхідну інформацію, таку як ім'я, електронну пошту та пароль.

Після заповнення усіх полів, ми спробуємо зареєструвати користувача, але навмисно допустимо помилку в повторенні пароля. У цьому випадку система має відобразити відповідне повідомлення про невідповідність паролів, не дозволяючи додати користувача до бази даних (див. Рис. 3.23).

Переконавшись, що система коректно реагує на некоректні дані, ми введемо всю інформацію правильно і спробуємо зареєструвати користувача ще раз. Цього разу система має успішно додати нового користувача до списку облікових записів. (див. Рис. 3.24)

Рис. 3.23 - Форма реєстрації

Додати користувача				
Show 10 entries			Search:	
#	Ім'я	Email	Роль	Дія
1	Шинкаренко Володимир	ist201_des@student.ztu.edu.ua	admin	✎ ✕
2	Кравчук Світлана	svitlana@gmail.com	manager	✎ ✕
3	Тимошенко Сергій	elizavetadanylo@gmail.com	manager	✎ ✕
Showing 1 to 3 of 3 entries				< >

Рис. 3.24 - Успішно зареєстрований користувач

Далі, ми перевіримо сценарій, коли користувач намагається зареєструватися з email-адресою, яка вже існує в системі. У цьому випадку система має розпізнати дублікат і відобразити відповідне повідомлення про помилку, не дозволяючи створити нового користувача з такими ж даними.(див. Рис. 3.25).

Користувачі				
Користувач з такою ж адресою вже існує.				
Додати користувача				
Show 10 entries			Search:	
#	Ім'я	Email	Роль	Дія
1	Кравчук Світлана	svitlana@gmail.com	manager	✎
2	Тимошенко Сергій	elizavetadanylo@gmail.com	manager	✎
Showing 1 to 2 of 2 entries				

Рис. 3.25 - Додавання користувача, який вже є у системі

Перейдемо до розділу "Інвентаризації" та оберемо зі списку "Магазин" як поточний склад. Далі виберемо необхідну категорію товарів, наприклад "Овочі та фрукти". Після вибору категорії, система має відобразити перелік всіх товарів, що наявні на обраному складі в рамках цієї категорії. Перевіримо, що список товарів, їхні облікові та фактичні кількості, ціни та загальні суми відображаються коректно(див. Рис. 3.26).

Для одного з товарів ми введемо фактичну кількість, відмінну від облікової. Натиснемо кнопку "Зберегти", щоб оновити дані в системі.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		70

Інвентаризації

Склад

Магазин

Категорії

☒ Овочі та фрукти
 ☐ М'ясо та риба
 ☐ Сири
 ☐ М'ясо-ковбасні вироби
 ☐ Хліб
 ☐ Молочні продукти
 ☐ Бакалія
 ☐ Солодощі
 ☐ Кава та чай
 ☐ Напої
 ☐ Спеції

Картопля

Партія	Облікова кількість	Фактична кількість	Ціна	Сума	Од. виміру
Партія 19.06.2024	8	7	55	440	кг

Цукіні жовтий

Партія	Облікова кількість	Фактична кількість	Ціна	Сума	Од. виміру
Партія 20.06.2024	1	1	20	20	кг

Зберегти

Списання

Оприбуткування

Рис. 3.26 - Створення інвентаризації

Після збереження, система має оновити фактичну кількість товару відповідно до введених нами даних. Перевіримо, що значення "Фактична кількість" для цього товару змінилося на вказане нами. (див. Рис. 3.26)

Інвентаризації

Склад

Магазин

Категорії

☒ Овочі та фрукти
 ☐ М'ясо та риба
 ☐ Сири
 ☐ М'ясо-ковбасні вироби
 ☐ Хліб
 ☐ Молочні продукти
 ☐ Бакалія
 ☐ Солодощі
 ☐ Кава та чай
 ☐ Напої
 ☐ Спеції

Картопля

Партія	Облікова кількість	Фактична кількість	Ціна	Сума	Од. виміру
Партія 19.06.2024	7		55	385	кг

Цукіні жовтий

Партія	Облікова кількість	Фактична кількість	Ціна	Сума	Од. виміру
Партія 20.06.2024	1		20	20	кг

Зберегти

Списання

Оприбуткування

Рис. 3.27 - Результат після збереження

Висновки до третього розділу

У розділі 3 було детально описано процес реалізації та тестування системи. Були розглянуті основні модулі системи та структура її користувацького інтерфейсу. Продемонстровано ключові сторінки системи, такі як сторінка входу, панель управління, сторінки для управління товарами, замовленнями, надходженнями, постачальниками, інвентаризацією, аналізом

запасів та створенням акцій. Зазначені функціональні можливості кожної з цих сторінок.

Окремий акцент було зроблено на тестуванні системи. Визначено, що ручне тестування є найбільш доцільним підходом для перевірки функціональності, зручності використання та загальної ефективності системи. Детально описано сценарії тестування входу в систему з некоректними даними, реєстрації нових користувачів, включаючи випадки дублювання облікових записів, та процесу інвентаризації товарів.

Проведене тестування підтвердило, що "Інформаційна система обліку товарів для продуктового магазину" коректно реагує на різні ситуації, забезпечує цілісність даних та відповідає потребам користувачів. Це дозволяє зробити висновок про готовність системи до впровадження та подальшого використання в реальних умовах роботи продуктового магазину.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		72

ВИСНОВОК

У ході виконання кваліфікаційної роботи було розроблено інформаційну систему обліку товарів для продуктового магазину, яка дозволяє ефективно управляти товарними запасами, оптимізувати бізнес-процеси та покращити якість обслуговування клієнтів.

У першому розділі була розглянута сутність інформаційних систем, їх види та класифікація. Проведено аналіз існуючих рішень для обліку товарів у продуктових магазинах. Також були сформульовані вимоги до розроблюваної системи та обґрунтовано вибір інструментальних засобів для її реалізації. Ця теоретична база забезпечує необхідні передумови для успішного проектування та розробки ефективної інформаційної системи обліку товарів.

У другому розділі було детально розглянуто процес проектування інформаційної системи обліку товарів продуктового магазину. Визначено варіанти використання та загальну структуру системи. Розроблено модель бази даних, що відображає всі необхідні сутності та зв'язки між ними. Також була спроектована структура файлів системи відповідно до обраної архітектури. Ретельно спроектована система забезпечує надійну основу для подальшої реалізації та гарантує ефективну роботу майбутньої інформаційної системи.

Третій розділ присвячений безпосередній реалізації та тестуванню розробленої інформаційної системи обліку товарів продуктового магазину. Було описано процес реалізації основних модулів системи відповідно до попередньо розробленого проекту. Детально розглянуто структуру інтерфейсу онлайн-платформи та її основні компоненти. Наприкінці проведено тестування системи та оцінку її ефективності в реальних умовах експлуатації. Успішне впровадження цієї системи дозволить підвищити ефективність управління товарними запасами продуктового магазину.

Процес розробки інформаційної системи обліку товарів для продуктового магазину був комплексним та включав у себе декілька етапів. Спочатку було проведено ґрунтовний аналіз предметної області, визначено вимоги до системи та обрано відповідні інструментальні засоби. Далі було здійснено проектування

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВІНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		73

системи, розроблено модель бази даних та структуру файлів. На етапі реалізації було створено основні модулі системи, такі як управління асортиментом, приймання та переміщення товарів, інвентаризація, аналітика продажів тощо. Також було розроблено зручний та інтуїтивно зрозумілий інтерфейс користувача. Процес розробки супроводжувався постійним тестуванням та налагодженням системи для забезпечення її коректної роботи та відповідності поставленим вимогам.

У результаті проведеної роботи було успішно розроблено інформаційну систему обліку товарів для продуктового магазину. Система дозволяє ефективно управляти товарними запасами, автоматизувати процеси обліку та мінімізувати ризики, пов'язані з людським фактором. Завдяки впровадженню системи, продуктовий магазин отримує потужний інструмент для оптимізації своєї діяльності, підвищення ефективності роботи персоналу та покращення якості обслуговування клієнтів. Розроблена система є масштабованою та гнучкою, що дозволяє легко адаптувати її до майбутніх потреб та змін у бізнес-процесах магазину. Крім того, система забезпечує надійне збереження та обробку даних, генерацію звітів та аналітику продажів, що є цінним джерелом інформації для прийняття управлінських рішень.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		74

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Грицунов О. В. Інформаційні системи та технології. Навчальний посібник. — Х.: ХНАМГ, 2010. — 222 с.
2. Types of Information Systems - Components and Classification of Information Systems [Електронний ресурс].— Режим доступу : <http://www.managementstudyguide.com/types-of-information-systems.htm>
3. Інформаційні системи [Електронний ресурс]. – Режим доступу : <http://informatiia.com/informatsijni-sistemistruktura-i-klasifikatsiya-informatsijnih-sistem>
4. Організація обліку : навч. посіб. / Я. С. Карп'як, В. І. Воськало, В. С. Мохняк ; М-во освіти і науки України, Нац. ун-т "Львів. політехніка". – Львів : Вид-во Львів. політехніки, 2014. – 368 с.
5. Алістер Коберн. Сучасні методи опису функціональних вимог до систем. М.: Лорі, 2002 – 266 с.
6. Uzaŷr S. b. The Bootstrap Framework. Bootstrap. Boca Raton, 2022. С. 19–70.
7. JavaScript - краща мова для програмування [Електронний ресурс] – Режим доступу до ресурсу: <https://apeps.kpi.ua/javascript-krashcha-mova-dlia-programuvania>.
8. Документація PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/docs.php>
9. MySQL 8.4 Reference Manual [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/8.4/en/>.
10. PhpStorm Empowering PHP Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/phpstorm/>.
11. Apache HTTP Server Project [Електронний ресурс] – Режим доступу до ресурсу: <https://httpd.apache.org/dev/devnotes.html>.
12. Ambler Scott. System Use Cases: An Agile Introduction[Електронний ресурс].– 2004. Режим доступу до ресурсу: agilemodeling.com
13. Larman Craig. Applying UML and patterns. Prentice Hall, 2005. – 63–64 с.

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		75

14. PhpStorm Empowering PHP Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/phpstorm/>.
15. C. J. Date with Hugh Darwen: A Guide to the SQL standard : a users guide to the standard database language SQL, 4th ed., Addison Wesley, USA 1997.
16. Fundamentals of Software Architecture: An Engineering Approach. O'Reilly Media. 2020.
17. Мартін Р. Чиста архітектура / Роберт Мартін., 2019. – 512 с.
18. Guadamuz Gonzalez A. GNU General Public License v3: A Legal Analysis. SCRIPT-ed. 2006. P. 154–163.
19. Widenius M., Ahlmark D., AB M. MySQL Reference Manual. O'Reilly Media, Inc., 2002. 712 p.
20. 8 Major Advantages of Using MySQL [Електронний ресурс] - Режим доступу до ресурсу: <https://www.datamation.com/storage/8-major-advantages-of-using-mysql/>

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		76

ДОДАТКИ

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О,М, СВІНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		77

Таблиця А.1 -Таблиця role

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
name	varchar	-	-	Назва ролі

Таблиця А.2 -Таблиця category

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
categoryName	varchar	-	-	Назва категорії

Таблиця А.3 -Таблиця subcategory

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
categoryid	int	-	+	Категорія
subcategory	varchar	-	-	Назва підкатегорії

Таблиця А.4 -Таблиця warehouse

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
name	varchar	-	-	Назва складу
address	varchar	-	-	Адреса

Таблиця А.5 -Таблиця units

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
name	varchar	-	-	Назва одиниці виміру

Таблиця А.6 -Таблиця trade_mark

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
trade_name	varchar	-	-	Назва торгової марки
description	longtext	-	-	Опис
country	varchar	-	-	Країна походження

Таблиця А.7 -Таблиця stock_levels

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
Id_product	Int	-	+	Товар
quantity	Int	-	-	Допустима кількість
Id_warehouse	Int	-	+	Склад

Таблиця А.8 -Таблиця price_list

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
Name_product	varchar	-	-	Назва товару
price	double	-	-	Ціна
units	varchar	-	-	Одиниця виміру
Id_counterparties	int	-	+	Постачальник

Таблиця А.9 -Таблиця data_store

Назва	Тип даних	ПК	ЗК	Опис поля
id	Int	+	-	Унікальний код
Full_name	varchar	-	-	Назва магазину
Mobile_phone	varchar	-	-	Мобільний телефон
email	varchar	-	-	Електронна пошта
Country	varchar	-	-	Країна

Продовження таблиці А.9

Назва	Тип даних	ПК	ЗК	Опис поля
register_code	varchar	-	-	Код ЄРДПОУ
Payer_code	varchar	-	-	Код платника ПДВ
Legal_address	varchar	-	-	Юридична адреса
Postal_address	varchar	-	-	Поштова адреса
Location_address	varchar	-	-	Адреса місцезнаходження
iban	varchar	-	-	IBAN
bank	varchar	-	-	Банк
persona	varchar	-	-	Відповідальна особа

Клас User

```

<?php
session_start();
error_reporting(0);
include("include/config.php");

class User
{
    private $con;

    public function __construct($con)
    {
        if (!$con) {
            die("Database connection is required.");
        }
        $this->con = $con;
    }

    public function login($username, $password)
    {
        $password = md5($password);
        $query = mysqli_query($this->con, "SELECT users.*, role.name AS role_name
                                          FROM users
                                          JOIN role ON users.role = role.id
                                          WHERE users.username = '$username' AND
users.password = '$password'");

        $user = mysqli_fetch_array($query);

        if ($user) {
            $_SESSION['user_name'] = $user['username'];
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['user_role'] = $user['role_name'];
            return true;
        } else {
            return false;
        }
    }

    public function isLoggedIn()
    {
        return isset($_SESSION['user_name']);
    }

    public function getRole()
    {
        return $_SESSION['user_role'];
    }

    public function hasRole($role)
    {
        return $_SESSION['user_role'] === $role;
    }

    public function logout()
    {
        $_SESSION['login'] = "";
    }
}

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВІНЦІЦЬКА				81
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $ldate = date('d-m-Y h:i:s A', time());

        session_unset();
        $_SESSION['errmsg'] = "You have successfully logged out";

        $this->redirectToIndex();
    }

    private function redirectToIndex()
    {
        header("Location: index.php");
        exit();
    }

    public function getUserById($userId)
    {
        $query = mysqli_query($this->con, "SELECT users.*, role.name AS role_name
                                         FROM users
                                         JOIN role ON users.role = role.id
                                         WHERE users.id = $userId");
        return mysqli_fetch_array($query);
    }

    public function getUserInfo($userId) {
        $sql = "SELECT users.name, users.email, role.name AS role_name
               FROM users
               JOIN role ON users.role = role.id
               WHERE users.id = $userId";
        $result = mysqli_query($this->con, $sql);
        return mysqli_fetch_assoc($result);
    }

    public function updateProfile($userId, $name, $email) {
        $sql = "UPDATE users SET name = '$name', email = '$email' WHERE id = $userId";
        return mysqli_query($this->con, $sql);
    }

    public function getAllUsers($currentUserId)
    {
        $query = mysqli_query($this->con, "SELECT users.id, users.name, users.email,
                                         role.name AS role_name
                                         FROM users
                                         JOIN role ON users.role = role.id
                                         WHERE users.id != $currentUserId");

        return $query;
    }

    public function deleteUser($userId)
    {
        $query = mysqli_query($this->con, "DELETE FROM users WHERE id = $userId");
        return $query;
    }

    public function updateUserRole($userId, $newRole)
    {
        $query = mysqli_query($this->con, "UPDATE users SET role = $newRole WHERE id = $userId");
        return $query;
    }

    public function addUser($email, $role, $token)
    {

```

```

        $query = mysqli_query($this->con, "INSERT INTO users (email, role, token)
VALUES ('$email', '$role', '$token')");
        return $query;
    }

    public function checkExistingUser($email)
    {
        $checkQuery = mysqli_query($this->con, "SELECT * FROM users WHERE email =
'$email'");
        return mysqli_num_rows($checkQuery) > 0;
    }

    public function getAllRoles() {
        $query = mysqli_query($this->con, "SELECT * FROM role");
        return $query;
    }

    public function getUserByToken($token)
    {
        $query = mysqli_query($this->con, "SELECT * FROM users WHERE token = '$token'");
        return mysqli_fetch_assoc($query);
    }

    public function updateUserRegistration($email, $name, $username, $password)
    {
        $hashedPassword = md5($password);
        $query = mysqli_query($this->con, "UPDATE users SET name = '$name', username =
'$username', password = '$hashedPassword', token = NULL WHERE email = '$email'");
        return $query;
    }

    public function storeResetToken($email, $token)
    {
        $sql = "UPDATE users SET reset_token = '$token', reset_token_created_at = NOW()
WHERE email = '$email'";
        return mysqli_query($this->con, $sql);
    }

    public function getUserByResetToken($token)
    {
        $sql = "SELECT * FROM users WHERE reset_token = '$token' AND
reset_token_created_at >= DATE_SUB(NOW(), INTERVAL 24 HOUR)";
        $result = mysqli_query($this->con, $sql);
        return mysqli_fetch_assoc($result);
    }

    public function updatePassword($email, $newPassword)
    {
        $hashedPassword = md5($newPassword);
        $sql = "UPDATE users SET password = '$hashedPassword', reset_token = NULL,
reset_token_created_at = NULL WHERE email = '$email'";
        return mysqli_query($this->con, $sql);
    }
}

?>

```

Клас Order

```

<?php
class Order {
    private $con;

    public function __construct($con) {
        $this->con = $con;
    }
}

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

    }
    public function getAllOrders() {
        $query = mysqli_query($this->con, "SELECT * FROM order_to_supplier");
        return $query;
    }
    public function getOrderById($orderId) {
        $query = mysqli_query($this->con, "SELECT order_to_supplier.*, legal_entity.full_name AS
        counterparty_name, warehouse.address AS warehouse_address, warehouse.name AS warehouse_name
        FROM order_to_supplier
        JOIN legal_entity ON legal_entity.id =
        order_to_supplier.id_counterparties
        JOIN warehouse ON warehouse.id = order_to_supplier.id_warehouse
        WHERE order_to_supplier.id = $orderId");
        return mysqli_fetch_assoc($query);
    }
    public function getOrders($period, $year, $month) {
        $query = "SELECT o.id, o.id_counterparties, le.full_name AS supplier_name, o.number AS order_number,
        o.date AS order_date, o.sum AS order_sum,
        gr.date AS delivery_date, gr.sum AS delivery_sum
        FROM order_to_supplier o
        LEFT JOIN goods_receipt gr ON o.id = gr.id_order
        LEFT JOIN legal_entity le ON o.id_counterparties = le.id";

        if ($period === 'year' && $year !== null) {
            $query .= " WHERE YEAR(o.date) = $year";
        } elseif ($period === 'month' && $year !== null && $month !== null) {
            $query .= " WHERE YEAR(o.date) = $year AND MONTH(o.date) = $month";
        } elseif ($period === 'week') {
            $query .= " WHERE o.date >= DATE_SUB(CURDATE(), INTERVAL 1 WEEK)";
        } elseif ($period === 'day') {
            $query .= " WHERE DATE(o.date) = CURDATE()";
        }

        $result = mysqli_query($this->con, $query);
        $orders = array();
        while ($row = mysqli_fetch_assoc($result)) {
            $orders[] = $row;
        }
        return $orders;
    }
    public function deleteOrder($orderId) {
        $query = mysqli_query($this->con, "DELETE FROM order_to_supplier WHERE id = $orderId");
        return $query;
    }

    public function getCounterpartyNameById($counterpartyId) {
        $query = mysqli_query($this->con, "SELECT full_name FROM legal_entity WHERE id = $counterpartyId");
        $row = mysqli_fetch_assoc($query);
        return $row['full_name'];
    }

    public function getWarehouseNameById($warehouseId) {
        $query = mysqli_query($this->con, "SELECT name FROM warehouse WHERE id = $warehouseId");
        $row = mysqli_fetch_assoc($query);
        return $row['name'];
    }
    public function generateOrdersTable($orders) {
        $tableHtml = '<table class="table table-striped">
        <thead>
        <tr>
        <th>Постачальник</th>
        <th>Дата</th>
        <th>Номер замовлення</th>

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		84

```

        <th>Дата постачання</th>
        <th>Замовлення на суму</th>
        <th>Отримано на суму</th>
        <th>Не отримано на суму</th>
    </tr>
</thead>
<tbody>';

$totalOrderSum = 0;
$totalDeliverySum = 0;
$totalNotDeliveredSum = 0;

foreach ($orders as $row) {
    $orderSum = $row['order_sum'];
    $deliverySum = $row['delivery_sum'] !== null ? $row['delivery_sum'] : 0;
    $notDeliveredSum = $orderSum - $deliverySum;

    $tableHtml .= "<tr>
        <td>" . $row['supplier_name'] . "</td>
        <td>" . $row['order_date'] . "</td>
        <td>" . $row['order_number'] . "</td>
        <td>" . ($row['delivery_date'] !== null ? $row['delivery_date'] : '-') . "</td>
        <td>" . number_format($orderSum, 2) . "</td>
        <td>" . number_format($deliverySum, 2) . "</td>
        <td>" . number_format($notDeliveredSum, 2) . "</td>
    </tr>";

    $totalOrderSum += $orderSum;
    $totalDeliverySum += $deliverySum;
    $totalNotDeliveredSum += $notDeliveredSum;
}

$tableHtml .= '</tbody></table>';

return $tableHtml;
}

public function generateTotalHtml($orders) {
    $totalOrderSum = 0;
    $totalDeliverySum = 0;
    $totalNotDeliveredSum = 0;

    foreach ($orders as $row) {
        $orderSum = $row['order_sum'];
        $deliverySum = $row['delivery_sum'] !== null ? $row['delivery_sum'] : 0;
        $notDeliveredSum = $orderSum - $deliverySum;

        $totalOrderSum += $orderSum;
        $totalDeliverySum += $deliverySum;
        $totalNotDeliveredSum += $notDeliveredSum;
    }

    $totalHtml = '<p><strong>Разом по замовленням:</strong> ' . number_format($totalOrderSum, 2) . '</p>';
    $totalHtml .= '<p><strong>Разом отримано:</strong> ' . number_format($totalDeliverySum, 2) . '</p>';
    $totalHtml .= '<p><strong>Разом не отримано:</strong> ' . number_format($totalNotDeliveredSum, 2) .
'</p>';

    return $totalHtml;
}

public function generatePdfContent($orders) {

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		85

```

$html = '<html>
    <head>
        <style>
            body { font-family: Arial, sans-serif; }
            table { border-collapse: collapse; width: 100%; }
            th, td { border: 1px solid black; padding: 8px; text-align: left; }
            th { background-color: #f2f2f2; }
        </style>
    </head>
    <body>
        <h1 style="text-align: center;">Контроль поставок</h1>
        <table>
            <thead>
                <tr>
                    <th>Постачальник</th>
                    <th>Дата</th>
                    <th>Номер замовлення</th>
                    <th>Дата поставання</th>
                    <th>Замовлення на суму</th>
                    <th>Отримано на суму</th>
                    <th>Не отримано на суму</th>
                </tr>
            </thead>
            <tbody>';

$totalOrderSum = 0;
$totalDeliverySum = 0;
$totalNotDeliveredSum = 0;

foreach ($orders as $row) {
    $orderSum = $row['order_sum'];
    $deliverySum = $row['delivery_sum'] !== null ? $row['delivery_sum'] : 0;
    $notDeliveredSum = $orderSum - $deliverySum;

    $html .= "<tr>
        <td>" . $row['supplier_name'] . "</td>
        <td>" . $row['order_date'] . "</td>
        <td>" . $row['order_number'] . "</td>
        <td>" . ($row['delivery_date'] !== null ? $row['delivery_date'] : '-') . "</td>
        <td>" . number_format($orderSum, 2) . "</td>
        <td>" . number_format($deliverySum, 2) . "</td>
        <td>" . number_format($notDeliveredSum, 2) . "</td>
    </tr>";

    $totalOrderSum += $orderSum;
    $totalDeliverySum += $deliverySum;
    $totalNotDeliveredSum += $notDeliveredSum;
}

$html .= '</tbody>
</table>
<p><strong>Разом по замовленням:</strong> ' . number_format($totalOrderSum, 2) . '</p>
<p><strong>Разом отримано:</strong> ' . number_format($totalDeliverySum, 2) . '</p>
<p><strong>Разом не отримано:</strong> ' . number_format($totalNotDeliveredSum, 2) . '</p>
</body>
</html>';

return $html;
}

public function createOrder($number, $data, $counterparties, $warehouse, $selectedProducts) {
    if (!empty($selectedProducts)) {
        $products = array();
    }
}

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		86

```

$totalSum = 0;

foreach ($selectedProducts as $product) {
    $productId = $product['id'];
    $quantity = $product['quantity'];
    $price = $product['price'];

    $products[] = array(
        'id' => $productId,
        'quantity' => $quantity,
        'price' => $price
    );

    $totalSum += $quantity * $price;
}

$productsJson = json_encode($products);

$stmt = mysqli_prepare($this->con, "INSERT INTO order_to_supplier (number, date,
id_counterparties, id_warehouse, products, sum) VALUES (?, ?, ?, ?, ?, ?)");
mysqli_stmt_bind_param($stmt, "ississ", $number, $data, $counterparties, $warehouse,
$productsJson, $totalSum);

if (mysqli_stmt_execute($stmt)) {
    return true;
} else {
    return false;
}

mysqli_stmt_close($stmt);
} else {
    return false;
}
}
}
?>

```

Клас Price

```

<?php
class Price
{
    private $con;

    public function __construct($con)
    {
        $this->con = $con;
    }

    public function getOldPrice($productId)
    {
        $query = "SELECT margin_price, pdv_price FROM price WHERE id_product = $productId";
        $result = mysqli_query($this->con, $query);
        $priceData = mysqli_fetch_assoc($result);
        return $priceData;
    }

    public function createDiscount($startDate, $endDate, $percentage, $productId)
    {
        // Отримати стару ціну з таблиці price для вибраного продукту
        $priceData = $this->getOldPrice($productId);
        $oldPrice = $priceData['margin_price'];
    }
}

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦИЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		87

```

$pdvPrice = $priceData['pdv_price'];

// Обчислити нову ціну зі знижкою та ПДВ
$newPrice = $oldPrice * (1 - $percentage / 100) * 1.2;

// Вставити дані в таблицю discounts
$insertQuery = "INSERT INTO discounts (old_price, percentage, new_price, date_start, date_finish,
id_product) VALUES ($oldPrice, $percentage, $newPrice, '$startDate', '$endDate', $productId)";
mysqli_query($this->con, $insertQuery);
}
public function isDiscountExists($productId)
{
    $query = "SELECT COUNT(*) AS count FROM discounts WHERE id_product = $productId AND CURDATE() BETWEEN
date_start AND date_finish";
    $result = mysqli_query($this->con, $query);
    $row = mysqli_fetch_assoc($result);
    return $row['count'] > 0;
}
public function getAllDiscounts()
{
    $query = "SELECT p.productName, d.old_price, d.new_price, d.percentage, d.date_start, d.date_finish
FROM discounts d
INNER JOIN products p ON d.id_product = p.id
WHERE CURDATE() BETWEEN d.date_start AND d.date_finish";
    $result = mysqli_query($this->con, $query);
    $discounts = array();
    while ($row = mysqli_fetch_assoc($result)) {
        $discounts[] = $row;
    }
    return $discounts;
}

public function getProductPriceData($productId)
{
    $query = "SELECT pr.pdv_price AS price, pr.margin_price, d.new_price AS discount_price, d.percentage
FROM products p
INNER JOIN price pr ON p.id = pr.id_product
LEFT JOIN discounts d ON p.id = d.id_product AND CURDATE() BETWEEN d.date_start AND
d.date_finish
WHERE p.id = '$productId'";
    $result = mysqli_query($this->con, $query);
    $priceData = mysqli_fetch_assoc($result);
    return $priceData;
}

public function getProductPrice($productId)
{
    $priceData = $this->getProductPriceData($productId);
    return $priceData['discount_price'] !== null ? $priceData['discount_price'] : $priceData['price'];
}

public function deleteExpiredDiscounts()
{
    $currentDate = date('Y-m-d');
    mysqli_query($this->con, "DELETE FROM discounts WHERE date_finish < '$currentDate'");
}

public function getProductCostPrice($productId)
{
    $costQuery = "SELECT cost_price, margin_price FROM price WHERE id_product = '$productId'";
    $costResult = mysqli_query($this->con, $costQuery);
    $costRow = mysqli_fetch_assoc($costResult);

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		88

```

        return $costRow;
    }
    public function checkPriceExistence($productName, $unit, $counterpartyId)
    {
        $checkQuery = "SELECT COUNT(*) FROM price_list WHERE name_product = '$productName' AND units =
'$unit' AND id_counterparties = '$counterpartyId'";
        $result = mysqli_query($this->con, $checkQuery);
        $count = mysqli_fetch_row($result)[0];
        return $count > 0;
    }

    public function addPriceListItem($productName, $price, $unit, $counterpartyId)
    {
        $insertQuery = "INSERT INTO price_list (name_product, price, units, id_counterparties) VALUES
('$productName', '$price', '$unit', '$counterpartyId')";
        mysqli_query($this->con, $insertQuery);
    }
    public function getPriceListByCounterparty($counterpartyId)
    {
        $query = "SELECT pl.id, pl.name_product, pl.price, pl.units, le.full_name AS counterparty_name
FROM price_list pl
JOIN legal_entity le ON pl.id_counterparties = le.id
WHERE pl.id_counterparties = '$counterpartyId'";

        $result = mysqli_query($this->con, $query);
        $pricelist = array();
        while ($row = mysqli_fetch_assoc($result)) {
            $pricelist[] = $row;
        }
        return $pricelist;
    }

    public function processPriceListData($data, $counterpartyId)
    {
        $duplicateErrors = array();
        foreach ($data as $item) {
            $productName = mysqli_real_escape_string($this->con, $item['product_name']);
            $unit = mysqli_real_escape_string($this->con, $item['unit']);
            $price = mysqli_real_escape_string($this->con, $item['price']);

            if (!$this->checkPriceExistence($productName, $unit, $counterpartyId)) {
                $this->addPriceListItem($productName, $price, $unit, $counterpartyId);
            } else {
                $duplicateErrors[] = "Товар '$productName' з ціною $price і одиницею виміру '$unit' вже
наявний у системі для цього постачальника.";
            }
        }
        return $duplicateErrors;
    }
    public function updatePrice($productId, $purchasePrice) {

        $stockQuery = "SELECT quantity, purchase_price FROM stocks WHERE id_product = '$productId'";
        $stockResult = mysqli_query($this->con, $stockQuery);

        $totalQuantity = 0;
        $totalPurchasePrice = 0;

        while ($row = mysqli_fetch_assoc($stockResult)) {
            $quantity = $row['quantity'];
            $purchasePriceFromStock = $row['purchase_price'];

            $totalQuantity += $quantity;

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				
Змн.	Арк.	№ докум.	Підпис	Дата		89

```

        $totalPurchasePrice += $quantity * $purchasePriceFromStock;
    }

    $totalQuantity++;
    $totalPurchasePrice += $purchasePrice;

    $costPrice = ($totalQuantity != 0) ? $totalPurchasePrice / $totalQuantity : $purchasePrice;

    $marginPrice = $this->calculateMarginPrice($costPrice);
    $pdvPrice = $this->calculatePdvPrice($marginPrice);

    $checkQuery = "SELECT COUNT(*) FROM price WHERE id_product = '$productId'";
    $result = mysqli_query($this->con, $checkQuery);
    $count = mysqli_fetch_row($result)[0];

    if ($count > 0) {

        $updateQuery = "UPDATE price SET
            cost_price = '$costPrice',
            margin_price = '$marginPrice',
            pdv_price = '$pdvPrice'
            WHERE id_product = '$productId'";
        mysqli_query($this->con, $updateQuery);
    } else {

        $insertQuery = "INSERT INTO price (id_product, cost_price, margin_price, pdv_price)
            VALUES ('$productId', '$costPrice', '$marginPrice', '$pdvPrice')";
        mysqli_query($this->con, $insertQuery);
    }
}

private function calculateMarginPrice($costPrice, $margin = 0.2) {
    return $costPrice * (1 + $margin);
}

private function calculatePdvPrice($marginPrice, $pdv = 0.2) {
    return $marginPrice * (1 + $pdv);
}
}
?>

```

		Є.С. ДАНИЛКО			ІСТ.КР.Б – 126 – 24 – ПЗ	Арк.
		О.М. СВИНЦІЦЬКА				90
Змн.	Арк.	№ докум.	Підпис	Дата		