

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

**Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки**

«До захисту в ЕК»

Директор інституту(декан факультету)

_____ **Андрій ФОРСЮК**
(підпис) (ім'я та прізвище)

«10» червня 2024р.

«До захисту допущено»

Завідувач кафедри

_____ **Сергій ГРИБКОВ**
(підпис) (ім'я та прізвище)

«10» червня 2024р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності 122 «Комп'ютерні науки»

(код та назва спеціальності)

освітньо-професійної програми Інформаційні системи та штучний інтелект

на тему: Розроблення інтернет магазину продажу молочної продукції

Виконав: здобувач 4 курсу, групи КН-4-4

_____ **Холявка Кирило Юрійович** _____
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник Ліманська Наталія Володимирівна _____
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____
(ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

Рецензент _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ - 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь бакалавр
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма Інформаційних технологій, штучного інтелекту і кібербезпеки

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інформаційних технологій,
штучного інтелекту і кібербезпеки
Сергій ГРИБКОВ

15 квітня 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Холявка Кирило Юрійович

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інтернет магазину продажу молочної продукції
керівник роботи Ліманська Н.В. ст. викл.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 15 квітня 2024 року № 279-кв

2. Строк подання здобувачем роботи 03.06.2024 р.

3. Вхідні дані до роботи

1) Відомості про продукти

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

1) Системний аналіз об'єкту дослідження

2) Проектування бази даних

3) Проектування інформаційної системи

4) Охорона праці та техніка безпеки

5. Перелік графічного матеріалу

1) Фізична модель бази даних

2) Інтерфейс системи

(підпис)

(прізвище та ініціали)

АНОТАЦІЯ

Мета кваліфікаційної роботи спрямована на демонстрацію набутих навичок протягом всього курсу навчання та вирішення проблеми, з якою зіткнувся відділ продажу молочної продукції. Основною метою є створення веб-сайту збуту продукції для полегшення процесу продажу та залучення нових клієнтів. У роботі будуть розглянуті етапи розробки сайту використовуючи мови python, html, java-script та sqlite, включаючи аналіз вимог, проектування, розробку та впровадження. В результаті цієї роботи очікується покращення ефективності відділу продажів та збільшення обсягів продажів молочної продукції завдяки новому каналу збуту через Інтернет.

Кваліфікаційна робота обсягом у 79 сторінок, 10 таблиць, 30 рисунків, 13 літературних джерел.

Ключові слова: PYTHON, SQLITE, HTML, JAVA-SCRIPT, БАЗА ДАНИХ, ЗАПИТ, ФОРМА, ІНТЕРФЕЙС КОРИСТУВАЧА, ІНТЕРФЕЙС АДМІНІСТРАТОРА.

ANNOTATION

The purpose of the qualification work is to demonstrate the skills acquired throughout the course of study and to solve a problem faced by the dairy sales department. The main goal is to create a sales website to facilitate the sales process and attract new customers. The work will cover the stages of website development using python, html, java-script and sqlite, including requirements analysis, design, development and implementation. As a result of this work, it is expected to improve the efficiency of the sales department and increase sales of dairy products through a new sales channel via the Internet.

The qualification work consists of 79 pages, 10 tables, 30 figures, 13 references.

Keywords: PYTHON, SQLITE, HTML, JAVA-SCRIPT, DATABASE, QUERY, FORM, USER INTERFACE, ADMINISTRATOR INTERFACE.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ’ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ	9
1.1. Загальна характеристика ТОВ "ФАВОР"	9
1.2. Організаційна структура ТОВ "ФАВОР"	10
1.3. Аналіз нинішнього стану комп’ютеризації ТОВ "ФАВОР"	12
1.4. Дослідження функціональної моделі	13
1.5. Огляд існуючих рішень для розв’язання виявлених проблем	16
1.6. Обґрунтування доцільності проектування й розроблення інтернет магазину продажу молочної продукції	18
1.7. Розрахунок економічного ефекту від впровадження системи	18
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ	28
2.1. Загальні положення.	28
2.2. Призначення і цілі створення системи.	28
2.3. Характеристика об’єкта автоматизації.....	29
2.4. Вимоги до системи	29
2.5. Склад і зміст робіт по створенню системи	39
2.6. Порядок контролю і приймання системи.....	39
2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію .	40
2.8. Вимоги до документації.....	40
2.9. Джерела розробки	40
РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ	41
3.1. Інформаційне забезпечення системи.....	41
3.2. Алгоритмізація та реалізація комплексу задач автоматизації	43
3.3. Інструкція користувача	77

3.4. Технічне та системне забезпечення розробки	79
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	82
ВИСНОВКИ.....	85
Основні результатами роботи.....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
ДОДАТКИ	89

ВСТУП

Протягом останніх десятиліть спостерігається значна трансформація у споживчих звичках, причому Інтернет став вирішальним каналом для пошуку та купівлі товарів. Отже, підхоплення тренду інтернет-покупок є важливим для підприємства, щоб надавати споживачам зручну та доступну можливість купівлі продукції онлайн. Створення веб-сайту для продажу молочної продукції відповідає сучасним вимогам ринку та сприяє розширенню аудиторії клієнтів шляхом залучення тих, хто віддає перевагу онлайн-покупкам.

Подальший розвиток інтернет-торгівлі у сфері харчової промисловості, зокрема у сегменті молочної продукції, відображає тенденцію до цифровізації бізнес-процесів. Впровадження веб-сайту для продажу молочної продукції сприятиме не лише підвищенню зручності для клієнтів, а й забезпечувати можливість ефективного управління запасами, аналізу продажів та взаємодії з клієнтами через інтеграцію з різними інформаційними системами. Такий підхід до розвитку бізнесу покликаний забезпечити конкурентоспроможність підприємства в умовах швидко змінного ринкового середовища.

Завдання даної кваліфікаційної роботи полягає в розробці веб-сайту для продажу молочної продукції з метою покращення ефективності відділу продажу та збільшення обсягів продажів через новий канал збуту в Інтернеті. Для досягнення цієї задачі будуть використані мови програмування Python, HTML, JavaScript та SQLite. Робота включатиме аналіз вимог, проектування, розробку та впровадження веб-сайту з метою створення зручного та привабливого інтернет-магазину для клієнтів.

Очікується, що в результаті даної роботи буде досягнуто значне покращення в роботі відділу продажу молочної продукції, забезпечено зростання обсягів продажів та залучення нових клієнтів завдяки ефективному використанню інтернет-технологій.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ

1.1. Загальна характеристика ТОВ "ФАВОР"

ТОВ "Фавор" є провідним молокопереробним заводом у місті Києві, який славиться якістю та інноваційністю своєї продукції. Заснований понад 28 років тому, завод здобув визнання як один із перших в Україні, що випускає якісний йогурт та інші молочні продукти під торговими марками ТМ "АМА" і "А-МАМ", орієнтованих на дітей від 2 років.

Відправною точкою "Фавор" сягають 1993 року, коли невелика команда молодих та амбіційних фахівців з молочного виробництва вирішила створити щось унікальне. За понад чверть століття розвитку, завод перетворився на сучасний виробничий комплекс з високотехнологічним обладнанням та сертифікованими системами управління якістю.

Головною метою "Фавор" є забезпечення споживачів якісною та безпечною молочною продукцією, що відповідає найвищим стандартам. Компанія має на меті підтримувати здоров'я та добробут споживачів, особливо молодших поколінь, надаючи їм смачні та корисні продукти.

Основною сферою діяльності "Фавор" є виробництво молочних продуктів з високим вмістом корисних речовин для дітей від 2 років. Також компанія спеціалізується на виробництві дієтичних та вітамінізованих продуктів, спрямованих на зміцнення здоров'я та підтримку імунітету.

"Фавор" відома своєю натуральністю, якістю та безпекою продукції, що підтверджується сертифікатами та високими відзнаками. Завдяки ретельному вибору сировини та сучасним технологіям, компанія гарантує бездоганний смак та корисні властивості своїх молочних продуктів.

1.2. Організаційна структура ТОВ "ФАВОР"

Роль і взаємодія підрозділів:

1. Роль виробничого відділу полягає в виробництві молочної продукції згідно з попитом та стандартами якості. Його Взаємодія з відділом збуту полягає в обміні інформацією про обсяги продукції. Також, він отримує від збутового відділу замовлення на виробництво.
2. Головною роллю відділ збуту є здійснення продажу та збуту продукції. Взаємодія з виробничим відділом полягає в передачі замовлень на виробництво, а також в обміні інформацією про вимоги та відгуки клієнтів. Він також передає фінансовому відділу інформацію про реалізовані обсяги продукції та укладені угоди з клієнтами.
3. Роль фінансового відділу, забезпечення фінансової стабільності підприємства. Взаємодія з відділами виробництва та збуту полягає в контролі за фінансовою звітністю та визначенні цінової політики на підставі обсягів виробництва та реалізації продукції.
4. Роль відділу обліку у компанії ТОВ "ФАВОР" полягає в забезпеченні точного обліку фінансових операцій, а також в управлінні фінансовими ресурсами підприємства. Взаємодія з відділами виробництва, збуту і фінансів полягає в забезпеченні їх необхідною фінансовою інформацією для прийняття стратегічних рішень. Також, відділ обліку відповідає за складання фінансової звітності і дотримання встановлених стандартів бухгалтерського обліку.

1.2.1. Загальна схема організаційної структури рис. 1.1

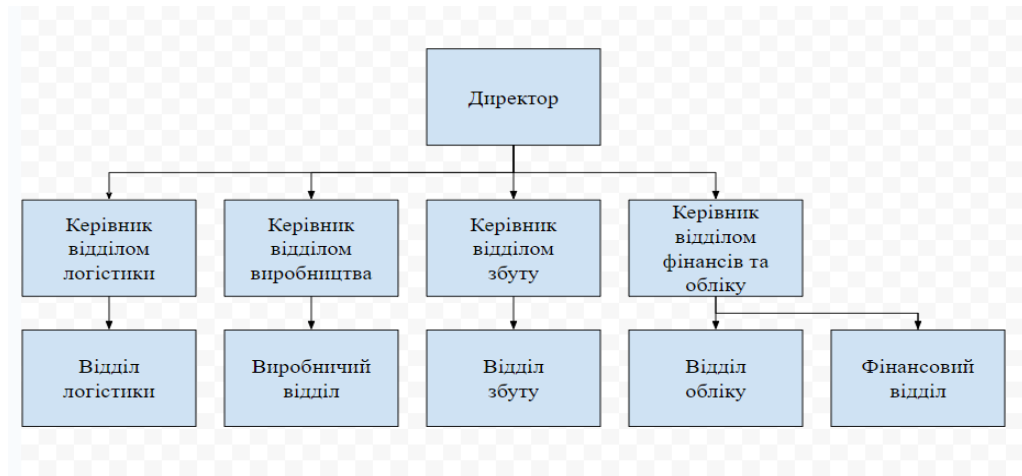


Рисунок 1.1 – Загальна схема організаційної структури

Управління на підприємстві здійснюється через ієрархічну систему, де директор визначає загальні стратегічні напрямки діяльності. Він координує роботу п'яти основних підрозділів: виробництва, збуту, фінансів, обліку та логістики. Кожен з цих підрозділів має свого керівника, який відповідає за виконання поставлених завдань і цілей.

1.2.2. Структура відділу збуту

Призначення: Організація продажу та збуту продукції компанії на ринку.

Особливості роботи: Розробка стратегій продажу, укладання контрактів з клієнтами, взаємодія з дистриб'юторами та рекламними агентствами.

Функції:

- пошук нових клієнтів та ринків збуту;
- укладання та виконання контрактів;
- аналіз ринкової конкуренції та розробка цінової політики;
- планування та проведення рекламних кампаній;
- забезпечення взаємодії з відділами виробництва та фінансів.
- Посадові обов'язки працівників:
 - менеджери з продажу – пошук та залучення клієнтів, укладання угод;
 - маркетологи – аналіз ринку, розробка стратегій продажу;
 - комерційні агенти – переговори з потенційними клієнтами та партнерами.

- Взаємодія з іншими відділами:
- з відділом виробництва – планування виробництва з урахуванням попиту на продукцію;
- з фінансовим відділом – контроль за фінансовою звітністю, визначення цінової політики.

Таблиця 1.1. Взаємодія відділу збуту з іншими відділами

Відділ	Отримує	Передає
Відділ виробництва	Плани виробництва, обсяги продукції, технічні вимоги до продукції	Замовлення на виробництво, відгуки клієнтів, вимоги клієнтів
Відділ Фінансів	Фінансові звіти, бюджетні плани, фінансові результати	Інформація про реалізовані обсяги продукції, угоди з клієнтами
Відділ Логістики	Статус доставки	Адреса замовлення, номер телефону клієнта

1.3. Аналіз нинішнього стану комп'ютеризації ТОВ "ФАВОР"

Недоліки у сфері комп'ютеризації в ТОВ "ФАВОР" виходять за межі лише відсутності веб-сайту для продажу товарів. Ця проблема, хоч і ключова, не єдине обмеження, з яким стикається підрозділ збуту молочної продукції. На сьогоднішній день, коли майже кожен аспект бізнесу обертається навколо цифрових технологій, відсутність ефективного використання Інтернету може серйозно зашкодити підприємству.

Відсутність онлайн-присутності обмежує доступ клієнтів до асортименту продукції. У сучасному світі, коли покупці все частіше обирають зручність та швидкість онлайн-покупок, відсутність можливості ознайомлення з товаром через Інтернет може відлякати потенційних покупців. Більше того, конкуренти,

які мають веб-присутність, автоматично стають більш привабливими для клієнтів.

Відсутність онлайн-системи продажу може призвести до ускладнення процесів замовлення та управління запасами. Замість автоматизованої системи, яка б дозволяла клієнтам замовляти продукцію в будь-який час та з будь-якого місця, відділ збуту може бути змушений використовувати більш складні часо-та ресурсозатратні методи.

Впровадження системи онлайн-продажу стає не лише важливим кроком для підвищення конкурентоспроможності ТОВ "ФАВОР", але й необхідністю в умовах сучасного ринкового середовища. Це дозволить підприємству оптимізувати процеси, збільшити обсяги продажів та зберегти свою позицію на ринку.

1.4. Дослідження функціональної моделі

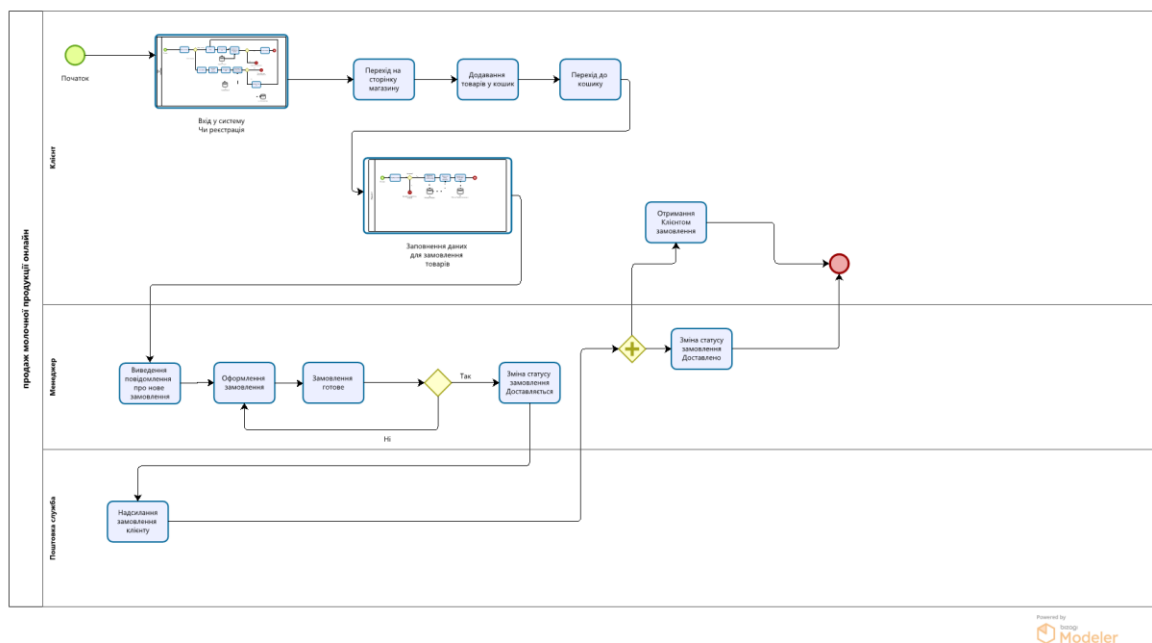


Рисунок 1.2 – Функціональна модель продажу молочної продукції

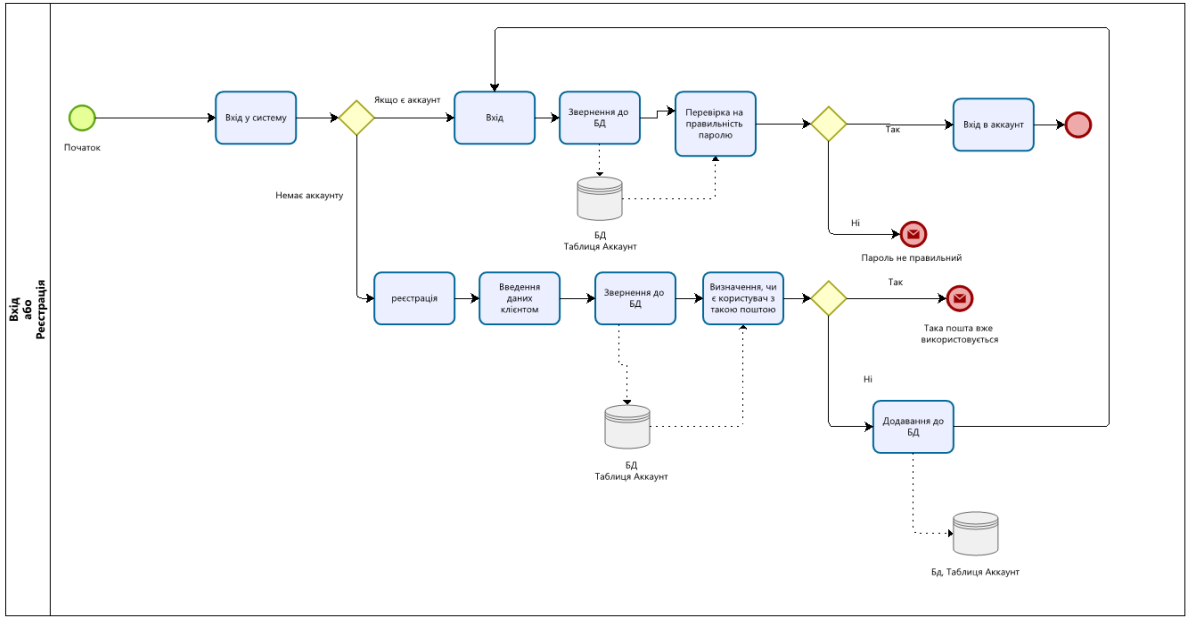


Рисунок 1.3 – Під-процес Вхід та реєстрація

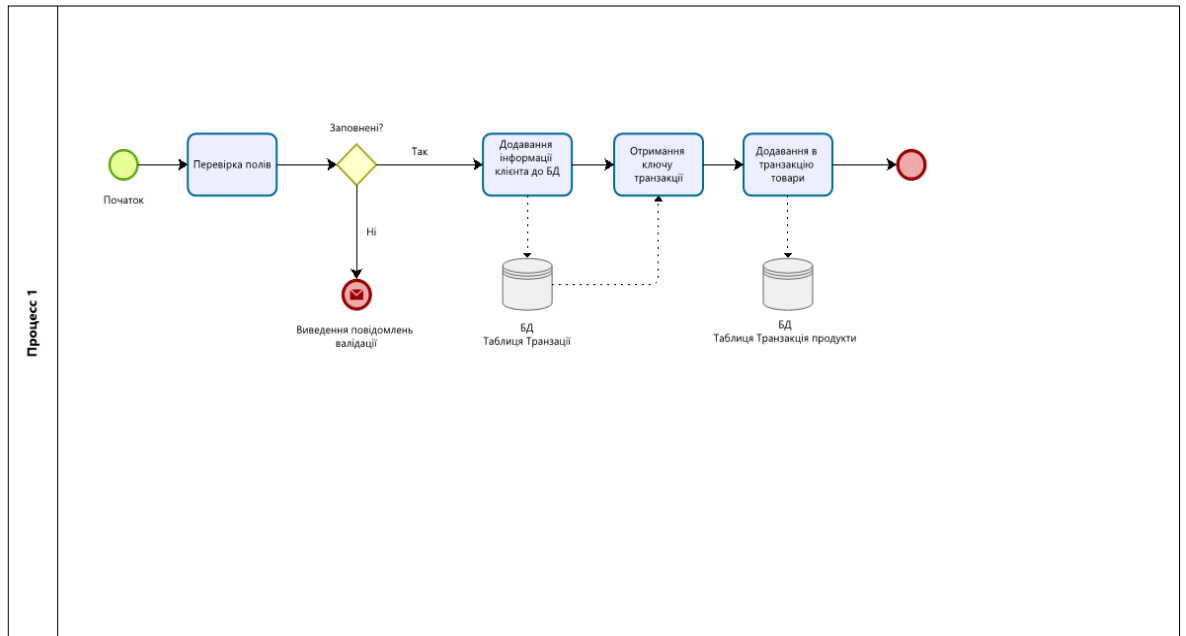


Рисунок 1.4 – Під-процес Кошик

При створенні моделі на рівні визначень об'єкти бази даних зображують у вигляді прямокутників, всередині яких, записують визначення збережуваних об'єктів, вказують зв'язки між ними і отриману схему узгоджують з клієнтом.

Зв'язки між об'єктами встановлюють у відповідності з бізнес-логікою функціонування організації.

1.4.1. В процесі аналізу бізнес-процесів у ТОВ "ФАВОР" було виявлено кілька проблем

Найбільш очевидною проблемою є відсутність веб-сайту для продажу товарів та системи онлайн-продажу. Це обмежує доступ клієнтів до асортименту продукції, ускладнює процес замовлення та управління запасами, та призводить до втрати можливих продажів та зниження конкурентоспроможності.

Відділ збуту молочної продукції не використовує потенціал Інтернету для привертання нових клієнтів та збільшення обсягів продажів. Відсутність онлайн-інструментів обмежує можливості просування продукції та комунікації з клієнтами.

Відсутність автоматизованої системи онлайн-продажу призводить до ускладнення процесів замовлення та управління запасами. Це може призводити до помилок в обробці замовлень, затримок у доставці та недоліків у запасах.

У результаті вищезгаданих проблем можливі втрати потенційних продажів та зниження конкурентоспроможності підприємства на ринку. Конкуренти, які вже мають онлайн-присутність, можуть бути більш привабливими для клієнтів через більшу доступність та зручність в обслуговуванні.

1.4.2. Задачі автоматизації

Для усунення наявних проблем і покращення ефективності бізнес-процесів у ТОВ "ФАВОР" необхідно впровадити наступні заходи з автоматизації:

Розробка і запуск веб-сайту, що буде включати каталог продукції, інформацію про компанію, контактні дані та можливість онлайн-замовлень. Це зробить продукцію доступною для більшої аудиторії і полегшить процес замовлення.

Розробка та впровадження системи онлайн-продажу, яка автоматизує прийом замовлень, обробку платежів та керування запасами. Це дозволить оптимізувати процеси та полегшить взаємодію з клієнтами.

Активне використання електронної пошти для залучення нових клієнтів. Це дозволить підприємству підвищити свою видимість та привернути увагу цільової аудиторії.

Проведення навчання та підготовки персоналу з використання нових систем та технологій, щоб забезпечити ефективне впровадження та використання автоматизованих процесів.

Встановлення механізмів моніторингу та звітності, щоб виявляти можливості для подальшого вдосконалення системи та адаптації до змін у ринкових умовах та потребах клієнтів.

1.5. Огляд існуючих рішень для розв'язання виявлених проблем

Для огляду і порівняння інформаційних систем, доступних на вітчизняному ринку, можна розглянути декілька популярних платформ:

1. bitrix24: ця система включає в себе crm, внутрішній портал, спільну роботу, звіти та аналітику, інтеграцію з поштою та інші функції. особливості включають можливість керування завданнями, календарем, документами, зв'язками з клієнтами та інші;
2. amocrm: ця система спрямована на автоматизацію процесів продажу. вона надає можливості керування лідами, угодами, контактами, робочим часом та іншими аспектами продажів;
3. salesforce: це одна з найбільш відомих та розповсюджених crm-систем у світі. salesforce надає широкий спектр функцій для управління клієнтами, включаючи збору даних, аналітику, маркетинг та інше;
4. flask: це фреймворк для створення веб-додатків на мові програмування python. він часто використовується для розробки веб-сайтів, веб-додатків та api.

Таблиця 1.2. Порівняння систем за функціональністю для можливого впровадження

Система	Вартість грн/корис тувач	Функціональність	Швидко дія	Сумісніс ть	Мова інтерфейс у
Bitrix24	Від 740	Висока	Середня	Широкий	Українська , англійська, інші
АmoCRM	Від 580	Висока	Висока	Широкий	Українська , англійська, інші
Salesforce	Від 975	Дуже висока	Висока	Широкий	Українська , англійська, інші
Flask	Безкоштов но	Залежить від розробника	Залежит ь від реалізаці ї	Python, широкий спектр	Українська , англійська, інші

Flask був обраний з урахуванням його безкоштовної вартості, високих можливостей у розробці веб-сайтів та простоти використання. Це фреймворк Python, який надає широкі можливості для створення різноманітних веб-додатків та сайтів із використанням простого та зрозумілого синтаксису.

1.6. Обґрунтування доцільності проектування й розроблення інтернет магазину продажу молочної продукції

Після аналізу поточного рівня комп'ютеризації виявлено ряд проблем, з яких основна полягає в відсутності веб-сайту для продажу товарів і системи онлайн-продажів. Ця проблема обмежує доступ клієнтів до асортименту продукції, ускладнює процес замовлення та контролю над запасами, а також веде до втрати можливостей для потенційних продажів та погіршення конкурентоспроможності підприємства.

У зв'язку з цим, розробка та впровадження веб-сайту та системи онлайн-продажу є вкрай доцільними кроками для підприємства. Ці заходи дозволять оптимізувати процеси замовлення, збільшити доступність продукції для клієнтів, покращити взаємодію з ними та збільшити обсяги продажів. Відповідно, це допоможе підтримувати та розвивати позиції підприємства на ринку та підвищити його конкурентоспроможність.

Крім того, розглянуті альтернативи в формі інформаційних систем, таких як Bitrix24, AmoCRM, Salesforce та Flask, підтверджують важливість розвитку веб-присутності та автоматизації бізнес-процесів. Вони надають широкий спектр функціоналу для управління клієнтами, продажами, внутрішньою комунікацією та іншими аспектами, що дозволить підприємству ефективніше працювати та досягати поставлених цілей.

Отже, проектування та розроблення веб-сайту та системи онлайн-продажу для ТОВ "ФАВОР" є необхідним та обґрунтованим кроком для подальшого розвитку підприємства і збільшення його ефективності на ринку.

1.7. Розрахунок економічного ефекту від впровадження системи

Визначення економічного ефекту від впровадження системи є дуже важливим моментом, адже в його основі лежить техніко-економічне обґрунтування розробки автоматизованої системи.

1.7.1. Визначення розміру оплати праці

Джерелами прибутку від впровадження системи можуть бути такі фактори:

Зменшення витрат на оплату праці. За допомогою автоматизованої системи можливе зменшення кількості працівників, необхідних для виконання певних завдань, або зменшення часу, витраченого на виконання рутинних операцій.

Підвищення продуктивності праці. Автоматизована система може допомогти працівникам працювати ефективніше, зменшуючи час, витрачений на виконання завдань, і підвищуючи загальну продуктивність.

Зменшення помилок. Автоматизована система може допомогти уникнути людських помилок або зменшити їх кількість, що може призвести до заощадження коштів на виправлення помилок та уникнення втрат.

Збільшення обсягів продажів. Завдяки покращенню обслуговування клієнтів та оптимізації процесів продажу, автоматизована система може допомогти збільшити обсяги продажів та підвищити прибуток.

Економія часу. Зменшення часу, витраченого на виконання операцій, може призвести до збільшення ефективності та зменшення витрат на робочу силу. Визначаємо ознаку – онлайн магазин молочної продукції.

1.7.2. Визначення ступеня новизни розроблюваних задач

Ступінь новизни розроблюваних задач для онлайн магазину молочної продукції відноситься до класу В. Це передбачає розробку проекту з використанням типових проектних рішень за умови їх адаптації до специфіки продажу молочних продуктів. Розробка інтернет-магазину може базуватися на типових рішеннях для електронної комерції, але з урахуванням особливостей та вимог даної галузі, зокрема, що стосується зберігання та доставки товарів.

1.7.3. Визначення групи складності алгоритму

Група складності алгоритму для магазину молочної продукції може визначатися як група 2. Це передбачає алгоритми обліку, звітності, статистики, пошуку.

1.7.4. Визначення виду інформації, що використовується

Вид інформації, що використовується, визначається на основі аналізу вхідної та вихідної інформації, що буде циркулювати в інформаційній системі, згідно таблиці 1.3.

Таблиця 1.3. Визначення виду інформації

Вид інформації	Позначення	Кількість наборів даних
Кількість видів змінної інформації	ЗІ	$m = 2$
Кількість видів нормативно-довідкової інформації	НДІ	$n = 5$
Кількість баз даних	БД	$p = 1$
Обробка в режимі реального часу	РЧ	так

1.7.5. Визначення витрати часу на розробку ескізного проекту і технічного завдання за даними

Витрати часу на розробку ескізного проекту Т1 і технічного завдання Т2 визначаються за даними з таблиці 4.

Таблиця 1.4. Визначення витрат часу

Вид системи	Стадія розробки системи			
	Ескізний проект (Т1)		Технічне завдання (Т2)	
	В	Г	В	Г
Управління науково-технічною інформацією	67	46	24	15

1.7.6. Визначення витрати часу на стадіях "технічний проект", "робочий проект" і "впровадження"

Вхідними даними для визначення є:

- кількість форм вхідної інформації 2;
- кількість форм вихідної інформації 5;
- базове значення витрат часу для стадії «Технічний проект» ТБ3=61
- базове значення витрат часу для стадії «Робочий проект» ТБ4=114
- базове значення витрат часу для стадії «Впровадження» ТБ5=32

Базове значення витрат часу ТБ коригується за допомогою поправочних коефіцієнтів для всіх стадій розробки автоматизованої системи.

1.7.6.1. Визначення витрат часу для стадії «Технічний проект» (T_3) за (1.1).

$$T_3 = T_{Б3} * k_{п} * k_0 \quad (1.1)$$

$k_{п}$ – коефіцієнт трудомісткості робіт на стадії “технічний проект” розраховується за формулою 1.2.

$$k_{п} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} \quad (1.2)$$

Значення коефіцієнтів обирається з таблиць 5 та 6.

Таблиця 1.5. Коефіцієнти k_1, k_2, k_3 для стадії «Технічний проект»

Вид використаної інформації	Ступінь новизни
	В
k^1 (ЗІ)	1.0
k^2 (НДІ)	0.72
k^3 (БД)	2.08

Таблиця 1.6. Коефіцієнт ступеню новизни проекту (k_0)

Стадія розробки системи	Вид обробки	Ступінь новизни
		В
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

$$k_{\pi} = \frac{1 * 2 + 0.72 * 5 + 2.08 * 1}{2 + 5 + 1} = 0,96 \quad (1.2)$$

$$T_3 = 61 * 0,96 * 1.26 = 74 \quad (1.1)$$

1.7.6.2. Визначення витрат часу на стадії «Робочий проект» (T_4) за (1.3).

Таблиця 1.7. Коефіцієнти k_1 , k_2 , k_3 для стадії «Робочий проект»

Вид використаної інформації	Ступінь новизни
	В
k^1 (ЗІ)	1.1
k^2 (НДІ)	0.58
k^3 (БД)	0.48

$$k_{\pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} \quad (1.2)$$

$$k_{\pi} = \frac{1,1 * 2 + 0,58 * 5 + 0,48 * 1}{2 + 5 + 1} = 0,7 \quad (1.2)$$

Для знаходження k_c для формули необхідно ідентифікувати складність контролю вхідної та вихідної інформації.

Тобто $k_c = 1.0$

$$T_4 = T_{Б4} * k_n * k_o * k_c \quad (1.3)$$

$$T_4 = 114 * 0,7 * 1,32 * 1,0 = 105 \quad (1.3)$$

1.8.6.3. Визначення витрат часу на стадії «впровадження» (T_5) за (1.4).

$$k_n = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} \quad (1.2)$$

$$k_n = \frac{1,1 * 2 + 0,58 * 5 + 0,48 * 1}{2 + 5 + 1} = 0,7 \quad (1.2)$$

$$T_5 = 32 * 0,7 * 1,21 * 1,0 = 27 \quad (1.4)$$

Отже, загальні витрати людської праці складають (1.5):

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_4 + T_5 = 61 + 24 + 74 + 105 + 27 = 291 \quad (1.5)$$

Визначимо чисельність виконавців Ч за формулою (1.6).

$$\text{Ч} = \frac{T_{\Sigma}}{\Phi} \quad (1.6)$$

Якщо для виконання роботи припустимо кількість робочих годин складає 530 із 7-годинним робочим днем, тому на розробку проекту виділено Φ , днів:

$$\Phi = 530/7 = 75 \text{ днів}$$

Для роботи $\Phi = 75$ днів. Тоді визначаємо кількість місяців із розрахунку 25 робочих днів.

$$\text{Кількість місяців на розробку, М: } \mathbf{M = \Phi/25 = 75/25 = 3 \text{ місяці}}$$

Отже, для виконання такого проекту потрібно така чисельність виконавців Ч, яка обраховується за формулою: $\text{Ч} = 291/75 = 3,88$ виконавця.

Прийmemo розмір заробітної плати програміста - 25000 грн, тоді загальна сума заробітних плат програмістів за формулою (1.7).

$$V'_1 = Ч * М * ЗП = 5 * 3 * 25000 = 375000 \text{ грн} \quad (1.7)$$

1.7.7. Витрати, пов'язані з розробкою програми на ПК

1.7.7.1. Розрахунок річного фонду часу роботи ПК

Дійсний річний фонд часу ПК у годинах дорівнює числу робочих годин у році для оператора, за винятком часу на технічне обслуговування і ремонт ПК (в середньому 5 год/міс + 6 роб.днів/рік) за формулою (1.8).

$$T_{ПК} = 2000 - (6 * 8 + 5 * 12) = 1892 \text{ год.} \quad (1.8)$$

Оскільки під час виконання курсової роботи здобувач в середньому витрачає 450 год. машинного часу, то величина фонду часу ПК дорівнює за формулою (1.9).

$$T_{ПК} = 1892 * (450/2000) = 425.7 \text{ год.} \quad (1.9)$$

Поточні витрати на експлуатацію V

Балансована вартість ПК, де Ц_р - ринкова вартість ПК, орієнтовно складає 30000 грн, кун – коефіцієнт, що враховує витрати на установку ПК . кун=0,12 за формулою (1.10)

$$Ц_{ПК} = Ц_{р} * (1 + k_{ун}) = 30000 * (1 + 0,12) = 33600 \text{ грн} \quad (1.10)$$

Амортизаційні відрахування використання ПК, З_{ам}, обчислюються за формулою (1.11)

$$З_{ам} = \frac{Ц_{ПК}}{H_A} = 33600/5 = 6720 \text{ грн} \quad (1.11)$$

Витрати на електроенергію (З_{ел}), споживану ПК, обчислюються за формулою (1.12)

$$З_{ел} = P_{ПК} * T_{ПК} * Ц_{ел} * A \quad (1.12)$$

де потужність ПК, = 0.5 кВт; фонд корисного часу роботи ПК, год, вартість 1 кВт електроенергії для підприємств, = 1,86 грн/кВт, коефіцієнт інтенсивного використання ПК, $A = 0.9$.

$$Z_{\text{ЕЛ}} = 0,5 * 425.7 * 1.68 * 0.9 = 321 \text{ грн} \quad (1.12)$$

Витрати на поточний ремонт і технічне обслуговування ПК (ЗР) визначаються як 6% від балансової вартості ПК, ЦПК за формулою (1.13).

$$Z_{\text{Р}} = C_{\text{ПК}} * 0.06 = 33600 * 0.06 = 2016 \text{ грн} \quad (1.13)$$

Непрямі витрати, пов'язані з експлуатацією ПК, визначаються як 5% від балансової вартості ПК ЦПК за формулою (1.14).

$$Z_{\text{МАТ}} = C_{\text{ПК}} * 0.05 = 33600 * 0.05 = 1680 \text{ грн} \quad (1.14)$$

Поточні витрати на експлуатацію V''_1 обчислюється (1.15)

$$V''_1 = Z_{\text{ОП}} + Z_{\text{АМ}} + Z_{\text{ЕЛ}} + Z_{\text{Р}} + Z_{\text{МАТ}} \quad (1.15)$$

Заробітна плата обслуговуючого персоналу складає в середньому - 10000 тож, поточні витрати на експлуатацію, V''_1 , грн, складають:

$$V''_1 = 10000 + 6720 + 321 + 2016 + 1680 = 20737 \text{ грн} \quad (1.15)$$

А, загальні витрати на розробку програмного забезпечення комп'ютерної системи обчислюються за формулою (1.16)

+

$$V_1 = V'_1 + V''_1 = 375000 + 20737 = 395737 \text{ грн} \quad (1.16)$$

1.8.7.2. Розрахунок витрат на придбання і установку ПК за формулою (1.17)

$$V_2 = C_{\text{ПК}} = 33600 \text{ грн} \quad (1.17)$$

1.7.7.3. Розрахунок витрат на підготовку приміщення і навчання персоналу розраховується за формулою (1.18).

Витрати на підготовку приміщення $V_3 = 0$, так як приміщення є в наявності.
Витрати на навчання персоналу V_4 .

В середньому навчання персоналу триватиме 1 місяць, тому можна вважати, що $V_4 = 4500$ грн;

Загальна вартість розробки і впровадження системи вираховується за формуло:

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 \quad (1.18)$$

$$V_{\Sigma} = 395737 + 33600 + 0 + 4500 = 433837 \text{ грн} \quad (1.18)$$

Оскільки норма амортизаційних втрат для комп'ютерних систем $HA = 5$, то для обрахунку річного економічного ефекту за формулою (1.19).

$$V_p = \frac{433837}{5} = 86767 \text{ грн} \quad (1.19)$$

Термін окупності розробки визначається $T_{ок}$ за формулою (1.20):

$$T_{ок} = \frac{1}{K_{ЕФ}} \quad (1.20)$$

де коефіцієнт економічної ефективності $K_{ЕФ} = \frac{\Pi_p}{K_p}$, де річний прибуток Π_p .

Від впровадження системи буде досягнуто за рахунок збільшення кількості продажів на день, якщо в середньому буде продаватись на 500 грн. і як результат – можливість виконувати більше замовлень та більше продавати молочної продукції, і орієнтовно складатиме 182500 грн на рік.

$$K_{ЕФ} = \frac{182500}{86767} = 2.1$$

Отже, термін окупності ІС складатиме:

$$T_{ок} = \frac{1}{2.1} = 5 \text{ місяців}$$

1.7.8. Обґрунтування техніко-економічного ефекту на інформаційну систему онлайн-магазину.

Впровадження інформаційної системи для онлайн-магазину молочної продукції матиме значний техніко-економічний ефект.

Збільшення обсягу продажів. Інтернет-магазин дозволить розширити аудиторію, оскільки покупці матимуть можливість здійснювати покупки онлайн з будь-якого місця і в будь-який час. Це призведе до збільшення кількості замовлень та загального обороту.

Зменшення витрат на оренду та утримання торгових площ. Оскільки інтернет-магазин не потребує фізичного приміщення для торгівлі, витрати на оренду та утримання магазину будуть мінімізовані.

Інформаційна система дозволяє автоматизувати багато рутинних операцій, таких як обробка замовлень, управління запасами та відстеження витрат. Це дозволить зменшити ризик помилок.

Підвищення клієнтського задоволення. Онлайн-продажі надають клієнтам більшу зручність та швидкість у виборі та придбанні товарів. Це підвищить задоволення клієнтів і збільшить їх лояльність до бренду.

Зниження витрат на маркетинг. Інтернет-магазин відкриває доступ до цифрових маркетингових інструментів, таких як реклама в соціальних медіа та електронна розсилка. Це дозволить знизити витрати на маркетинг та збільшити ефективність рекламних кампаній.

Аналіз та прогнозування. Інформаційна система забезпечує доступ до аналітичних звітів та інструментів прогнозування, що дозволить краще розуміти потреби клієнтів та реагувати на них швидше та ефективніше.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ

2.1. Загальні положення.

2.1.1. Найменування системи: «Інтернет магазин продажу молочної продукції»

2.1.2. Результати робіт зі створення системи оформляються згідно з вимогами ДСТУ на відповідні етапи розробки. Порядок оформлення і передачі результатів у даному випадку визначається змістом і календарним планом виконання розробки.

2.1.3. У випадку необхідності на наступних стадіях робіт по створенню системи окремі положення можуть уточнюватись і розвиватись.

2.2. Призначення і цілі створення системи.

2.2.1. Призначення систем

Система призначена для забезпечення продажу молочної продукції. Система автоматизує замовлення клієнтів, створення звітів. Також система містить дані клієнтів та внутрішні дані та комплекс додаткових функцій, які дозволяють виконувати зручний пошук за різними властивостями.

2.2.2. Цілі створення системи

Основною метою створення системи є розробка сайту для забезпечення реалізації молочної продукції через інтернет та автоматизація обліку замовленої продукції. Це забезпечить створення умов для поліпшення діяльності.

Задачі організаційного управління роботою характеризуються високою складністю, комплексністю й не можуть повністю розв'язуватись ізольовано. Тому в умовах функціонування автоматизованої системи з'являється можливість виконання всіх завдань на сучасному рівні із забезпеченням точності, оперативності та достовірності інформації.

2.3. Характеристика об'єкта автоматизації

2.3.1. Короткі відомості про об'єкт автоматизації

Об'єктом автоматизації є діяльність молокозаводу ТОВ "ФАВОР".

2.4. Вимоги до системи

2.4.1. Вимоги до системи в цілому

2.4.1.1. Вимоги до структури і функціонування системи.

2.4.1.1.1. Система повинна мати клієнт-серверну архітектуру, що використовує єдину базу даних (надалі — БД).

Згідно з функціональною структурою, система повинна бути пов'язана в мережі з автоматизованими робочими місцями:

керівника (власника);

клієнт;

менеджера.

2.4.1.1.2. Діагностування функціонування системи в мережі відділу продажу має передбачати виявлення відхилень від нормального процесу розв'язання задач і порушень у роботі комп'ютерно-технічних засобів, а також програмних помилок, забезпечуючи користувачів відповідними діагностичними повідомленнями.

Взаємозв'язок між підсистемами має здійснюватися на інформаційному рівні через загальну БД із використанням технічних засобів локальних комп'ютерних мереж.

2.4.1.1.3. Розвиток і модернізація системи повинні проводитися шляхом уточнення, нарощування чи заміни виконуваних функцій, модернізації технічних і програмних засобів по мірі розробки і впровадження нових поколінь комп'ютерів. Структура і технологія програмного забезпечення системи повинні забезпечити простоту їх модернізації та розвитку, з можливістю збільшення розмірності задач і масивів інформації, а також можливості реалізації їх на нових ПК.

Програмно-технічні засоби функціонування системи повинні мати програми з економіко-математичними та статистичними методами, методами моделювання, а також засоби табличного, текстового, графічного відображення даних. Програмна та інформаційна сумісність має забезпечуватися загальносистемним протоколом обміну, використанням проблемно-орієнтованих пакетів прикладних програм міжмашинних зв'язків і єдиною системою класифікації і кодування.

2.4.1.1.4. Функціонування системи має забезпечувати діалогову та мережну (розподілену) обробку даних.

2.4.1.2. Вимоги до чисельності і кваліфікації персоналу.

2.4.1.2.1. Персонал, який використовує автоматизовану систему, повинен дотримуватися наступних вимог:

- Пройти навчання і отримати навички роботи на ПК;
- Дотримуватись технологічних інструкцій при роботі з системою в діалоговому режимі;
- Дотримуватись умов експлуатації ПК у відповідності з інструкціями по експлуатації;
- Дотримуватись правил зберігання інформації і організації резервних копій БД;
- Дотримуватись правил техніки безпеки при роботі на ПК.

2.4.1.2.2. Користувачами системи може виступати адміністратор, клієнт. Вхід у систему повинен здійснюватися через пароль, який відображає рівень користувача: із правом коригування БД і без права. Залежно від рівня користувач отримує повний чи обмежений доступ до системи.

2.4.1.3. Показники призначення.

2.4.1.3.1. Відповідно до п. 2.2.1, показники призначення повинні характеризувати ступінь та якість автоматизації планової, інформаційно-облікової і управлінської діяльності відділу продаж молокозаводу для його оптимального функціонування. Перелік і допустимі значення показників, при

яких зберігається цільове призначення системи, повинні бути визначені на стадії технічного проектування.

2.4.1.3.2. Система повинна мати можливість налаштування на параметри об'єкта управління та периферійного обладнання при її модернізації та розвитку, а також зміні процесів та методів організаційного управління.

2.4.1.4. Вимоги до надійності.

2.4.1.4.1. Система є багатофункціональною і призначена для використання протягом робочого дня. Всі функції системи виконуються дискретно. У відповідності з ДСТУ 2226-93 оцінка надійності проводиться по кожній функції окремо. Враховуючи особливості функціонування системи, показники її надійності є показниками надійності СУБД, на якій вона реалізована, та технічних засобів, на яких вона експлуатується. Основними показниками надійності є:

L_i — ймовірність безвідмовного виконання задачі в заданий термін (ймовірність того, що i -тий запит буде виконаний);

K_r — коефіцієнт готовності ПТК (програмно-технічного комплексу);

T_v — середній час відновлення ПТК;

T_e — мінімальний час між двома відмовами за календарний місяць.

2.4.1.4.2. Комплекс технічних засобів повинен передбачати:

- можливість запуску і розв'язання функціональних задач із різних робочих станцій;

Для забезпечення надійності програмного та інформаційного забезпечення необхідно передбачити використання:

- модульного, структурного і об'єктно-орієнтованого програмування;
- програмних засобів контролю вхідної інформації з видачею користувачу повідомлень про виявлені помилки;
- програмних засобів коригування для виявлення і виправлення помилок у БД;
- засобів захисту від збоїв, несанкціонованого доступу, помилкових дій персоналу і т.д.;

- резервних копій БД.

2.4.1.5. Вимоги до безпеки.

Для забезпечення безпеки при експлуатації, налагодженні, монтажі, обслуговуванні і ремонті технічних засобів системи потрібно дотримуватись вимог ДСТУ: ДСТУ 2293-99, ДСТУ ISO 6309:2007, ДСТУ 12.0.230:2008, ДСТУ 7237:2011, ДСТУ 7238:2011, ДСТУ 7239:2011; по доступним рівням освітленості, вібраційних і шумових навантажень слід дотримуватися вимог відповідно ДСТУ Б А.3.2-15:2011, ДСТУ EN 14253:2018, ДСТУ 2867-94.

2.4.1.6. Вимоги з ергономіки та технічної естетики.

Загальні ергономічні і естетичні вимоги до системи повинні відповідати держстандартам ДСТУ 8604:2015, ДСТУ 7298:2013. Освітленість робочого місця повинна відповідати ДСТУ EN 12464-1:2016, ДБН В.2.5-28-2006.

Засоби відображення повинні розміщуватися таким чином, щоб кут спостереження екрану складав не більше, ніж 45 градусів, мінімальна відстань спостереження екрану — 0,3 м, рекомендована — 0,5 м. При розробленні ПЗ слід створити зручний інтерфейс для запобігання втомлюваності користувача.

2.4.1.7. Вимоги по експлуатації, технічного обслуговування, ремонту і зберігання компонентів системи.

2.4.1.7.1. Види обслуговування системи визначаються у відповідності з ДСТУ EN 13306:2019. Загальні вимоги по експлуатації, технічному обслуговуванню і ремонту повинні відповідати ДСТУ 3576-97.

2.4.1.7.2. Для розміщення технічних засобів системи необхідні площі, визначені в ДБН В.2.2-9-2009. При цьому слід дотримуватися вимог, зазначених в експлуатаційній документації. Напруга живлення технічних засобів системи 220/380 В змінного струму, частотою (50 ± 1) Гц. Допустиме відхилення напруги від +10 до -15%, тривалість перерв у живленні не повинна перевищувати 0,001 с.

2.4.1.7.3. Кількість, кваліфікація і режими роботи обслуговуючого персоналу повинні відповідати рекомендаціям, зазначеним в технічних умовах і інструкціях з експлуатації окремих ТЗ.

2.4.1.7.4. Склад, розміщення і умови зберігання компонентів технічних засобів системи визначається рекомендаціями, зазначеними в експлуатаційній документації на ці елементи.

2.4.1.7.5. Регламент обслуговування повинен відповідати їх рівню і умовам роботи, щоб у випадку відмови системи забезпечити роботу в аварійному режимі.

2.4.1.8. Вимоги до захисту інформації від несанкціонованого доступу.

Для надійності збереження і доступу до інформації необхідно використовувати засоби захисту:

серверних операційних систем Windows;

локальної мережі та програми захисту в мережі Firewall.

клієнт-серверної СУБД:

тригери, представлення;

процедури та функції;

встановлення груп користувачів і ролей використання.

Крім цього, кожен сеанс роботи системи має розпочинатися з введення індивідуального паролю. Система парольного захисту повинна мати власні засоби періодичної зміни паролів або використовувати стандартні засоби середовища розроблення. Для надійного захисту від несанкціонованого доступу кожен із працівників повинен мати персональний пароль. Крім того, деякі таблиці треба захистити від можливого редагування, доповнення чи вилучення інформації.

2.4.1.9. Вимоги щодо збереження інформації при аваріях.

2.4.1.9.1. Необхідно передбачити засоби резервного збереження БД в архіві після коригування і можливість завантажити БД з архіву у випадку її руйнування.

2.4.1.9.2. Резервний архів і БД мають знаходитись на різних машинних носіях чи пристроях.

2.4.1.10. Вимоги по захисту від впливу зовнішніх діянь.

2.4.1.10.1. Електрична складова електромагнітного поля завад в приміщеннях не повинна перевищувати 0,3 В/м² в діапазоні частот від 0,15 до 300 МГц. Для захисту від впливу електромагнітних полів та індустриальних завад слід передбачити різноманітні екрани та фільтри.

2.4.1.10.2. Засоби, які виключають вплив шкідливих факторів на функціонування комплексу технічних засобів, повинні бути запроектовані згідно з ДБН В.2.2-9-2009. Обчислювальні засоби по стійкості до зовнішніх впливів повинні відповідати ДСТУ 2506-94.

2.4.1.11. Вимоги до патентної чистоти.

При створенні даної системи патентні дослідження не проводяться.

2.4.1.12. Вимоги по стандартизації і уніфікації.

У системі кодування інформації необхідно проводити за світовим класифікатором і стандартом.

2.4.2. Вимоги до функцій.

2.4.2.1. Перелік функцій із зазначенням вхідної та вихідної інформації наведено в таблиці 8.

Функції мають забезпечити раціональну організацію роботи користувача на основі безперервної технології: заповнення БД, довідників, формування різнорідних звітів і виконання інших функцій, визначених чинним документом. При цьому пріоритетом є зручність введення та використання інформації користувачем за рахунок формування підказок і меню на екрані монітора.

Таблиця 2.1. Перелік функцій, вхідної та вихідної інформації

№п/п	Найменування функції	Вхідна інформація	Вихідна інформація
1	Формування та виведення каталогу товарів	Таблиці БД «Склад», «Товари»	Форма зі списком категорій та товарів
2	Формування та виведення замовлень клієнтів	Таблиці БД «Акаунт», «Замовлення»	Форма зі списком замовлень клієнтів

Продовження таблиці 2.1.

4	Фільтр товарів	Запит користувача, «Склад», «Товари»	Форма з результатами пошуку по фільтрам
6.	Формування та виведення звіту заробітку з проданих продуктів діапазоні дат	Таблиця БД «Транзакції», «Акаунт», «Товари»	Звіт «Продукти»
7	Формування та виведення звіту з замовлень в діапазоні дат	Таблиця БД «Транзакції», «Акаунт»	Звіт «Транзакції»
8	Формування та виведення звіту з кількості проданих товарів в діапазоні дат	Таблиця БД «Склад», «Транзакції»	Звіт «Склад»

2.4.3. Вимоги до видів забезпечення.

2.4.3.1. У вимогах до математичного забезпечення (МЗ) система не вимагає спеціального математичного забезпечення для реалізації покладених на неї функцій. Достатньо можливостей обраної СУБД.

2.4.3.2. Вимоги до інформаційного забезпечення (ІЗ).

2.4.3.2.1. Інформаційне забезпечення системи повинно містити дані, достатні для виконання всіх покладених на систему функцій. ІЗ повинно гарантувати раціональну організацію зберігання інформації та доступу до неї.

Заповнення БД інформацією покладається на замовника за методиками і формами, створеними розробниками системи.

Склад, структура і спосіб організації інформації представляються у логічній моделі БД і можуть уточнюватись на етапі технічного проектування.

2.4.3.2.2. Слід передбачити захист даних від руйнування при аваріях і порушеннях у енергоживленні системи — використання резервних копій БД.

2.4.3.3. Вимоги до лінгвістичного забезпечення (ЛЗ).

2.4.3.3.1. Для розробки програмних засобів, які реалізують виконання функцій і забезпечують сервіс користувачів повинні використовуватися мови

високого рівня, які забезпечують створення структурних програм, а також мова обраної СУБД для здійснення доступу та маніпулювання даними.

2.4.3.3.2. Організація діалогу користувача до системи має будуватися на наборах меню і підказок, орієнтованих на виконання користувачем функцій. Запити користувача до системи повинні подаватись переважно природною мовою.

2.4.3.4. Вимоги до програмного забезпечення (ПЗ).

2.4.3.4.1. Загальносистемне ПЗ має забезпечувати надійне і якісне виконання функціональних завдань системи. До загально системного ПЗ належить:

- операційна система (далі ОС) — Windows;
- система управління БД (далі СУБД) — SQLite .

2.4.3.4.2. Загальні вимоги до системного ПЗ можна сформулювати так:

- мінімальні вимоги до ресурсів технічних засобів (ТЗ);
- максимальна швидкодія;
- повне задоволення потреб функціональних завдань системи.

2.4.3.4.3. Вимоги до ОС:

● мінімальне використання ресурсів комп'ютера для власних потреб, передусім оперативної і дискової пам'яті;

- максимальна швидкодія при управлінні зовнішніми пристроями;
- ОС сервера — Windows, ОС клієнта — Windows

2.4.3.4.4. Вимоги до СУБД:

- максимальне задоволення потреб функціональних задач;
- надійність;
- ефективне управління потрібного обсягу і структури;
- швидкість виконання запитів користувачів;
- мінімальні вимоги до ТЗ.

2.4.3.4.5. Програмні засоби введення та виведення даних і ведення діалогу повинні забезпечувати:

- виведення необхідних даних на екран у вигляді відповідних відеограм;

- супровід введення даних контролем і сигналізацією користувачу про наявність помилок з можливістю їх виправлення під час введення даних;
- керований комп'ютером діалог при введенні даних;
- виведення даних у відповідному вигляді (формі документа) за запитом користувача.

2.4.3.4.6. При розробленні спеціального ПЗ слід виконати наступні вимоги:

- використані програми мають бути сумісні між собою та із загальносистемним ПЗ;
- ПЗ має розроблятися засобами об'єктно-орієнтованого програмування;
- забезпечити відповідність інтерфейсу користувача стандартам Windows;
- необхідна модульна структура програм;
- повинна бути передбачена можливість розширення складу задач у відповідності з новими функціональними потребами;
- ПЗ не повинно залежати від типу зовнішніх пристроїв (принтерів, дисків, сканерів тощо);
- діалог із користувачем повинен проводитись за допомогою клавіатури або миші з поясненням виконання дій і можливістю отримання підказки.

2.4.3.5. Вимоги до технічного забезпечення.

2.4.3.5.1. Технічні засоби системи (табл. 9) повинні забезпечувати виконання функцій, перерахованих в таблиці 8.

2.4.3.5.2. Засоби обчислювальної техніки повинні забезпечувати обмін інформації в об'ємах, приведених в п. 4.3.2.

Таблиця 2.2. Вимоги до технічного забезпечення системи

№ п/п	Основні характеристики комп'ютера
Технічне забезпечення для сервера	
1	DELL T620 Intel Xeon Quad Core 3,0 GHz\32 Gb\4 TB RAID5\LAN 10 Gbit
Технічне забезпечення для клієнта	
1	Athlon QL-65 Dual Core 2,1 GHz; RAM: 2048 Mb; HDD: 250 Gb;
2	Монітор 15"
3	Миша USB/Wireless
4	Клавіатура USB/Wireless

2.4.3.6. Вимоги до метрологічного забезпечення.

Система не має вимірювальних каналів, вимірювального обладнання і приладів, тому вимоги до даного виду забезпечення не висуваються.

2.4.3.7. Вимоги до організаційного забезпечення.

2.4.3.7.1. Організаційне забезпечення системи розробляється в відповідності з вимогами державного стандарту по АСУП.

2.4.3.7.2. При впровадженні системи не передбачається збільшення штатної чисельності підприємства. Територіальне розміщення робочих місць, на яких буде встановлена система, визначається підприємством.

2.4.3.7.3. До функціонування системи висуваються наступні вимоги:

- наказом директора визначається список співробітників, які мають доступ до системи;

- контроль і прийняття рішень при аварійних ситуаціях при експлуатації системи здійснює відповідальний за систему.

2.5. Склад і зміст робіт по створенню системи

2.5.1. Стадії створення системи і терміни виконання робіт наведені в таблиці 10.

Таблиця 2.3. Найменування робіт при створенні системи

№ п/п	Найменування робіт	Строки виконання робіт
1	Передпроектне дослідження об'єкта автоматизації	15.02.2024
2	Технічне завдання	23.04.2024
3	Технічний проект	25.04.2024
4	Оформлення документації	28.05.2024

2.6. Порядок контролю і приймання системи

2.6.1. Система вводиться на діючому ТОВ «ФАВОР»

При введенні в дію система повинна пройти приймальні випробування згідно з ДСТУ 3974-2000.

2.6.2. Випробування для визначення працездатності і рішення про можливість приймання системи в дослідну експлуатацію проводять розробники разом із замовником. Програму випробувань складає розробник і затверджує замовник.

2.6.3. Здача в дослідну експлуатацію здійснюється на основі технічного завдання та інструкції користувача. За результатами дослідної експлуатації формується перелік доробок і рекомендовані строки їх виконання.

2.6.4. Введення в дію системи оформлюється актом здачі-прийому.

2.7. Вимоги до складу і змісту робіт із підготовки до введення системи в дію

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- проводить укомплектування технічних засобів;
- організовує навчання користувачів системи роботі на ПК і вивчення інструкції з її експлуатації;
- проводить досліdну експлуатацію і вводить систему в дію.

2.8. Вимоги до документації

2.8.1. На систему розробляється комплекс документації у складі: технічне завдання та технічний проект.

2.8.2. Документація на систему розробляється у відповідності з вимогами Державних стандартів серії 19 «Єдина система програмної документації» та серії 24 «Єдина система стандартів автоматизованих систем управління».

2.9. Джерела розробки

2.9.1. При розробленні технічного завдання на систему використано наступні документи:

ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання;

ДСТУ 3973–2000 Система розроблення та поставлення продукції на виробництво;

ДСТУ Б В.2.5–82:2016 Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом.

РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧ АВТОМАТИЗАЦІЇ

3.1. Інформаційне забезпечення системи

Під час розробки системи використано наступні технології:

- Erwin data modeler;
- СКБД SQLite;
- Середовище розробки PyCharm;
- Мова програмування Python та фреймворк Flask;
- Мова HTML, CSS, JS [1] та набір інструментів з відкритим кодом Bootstrap. [2]
- Шаблонізатор Jinja.

Для проектування бази даних, яка буде забезпечувати виконання функцій системи, створено логічну модель рис. 3.1 з використанням додатку Erwin data modeler. [10]

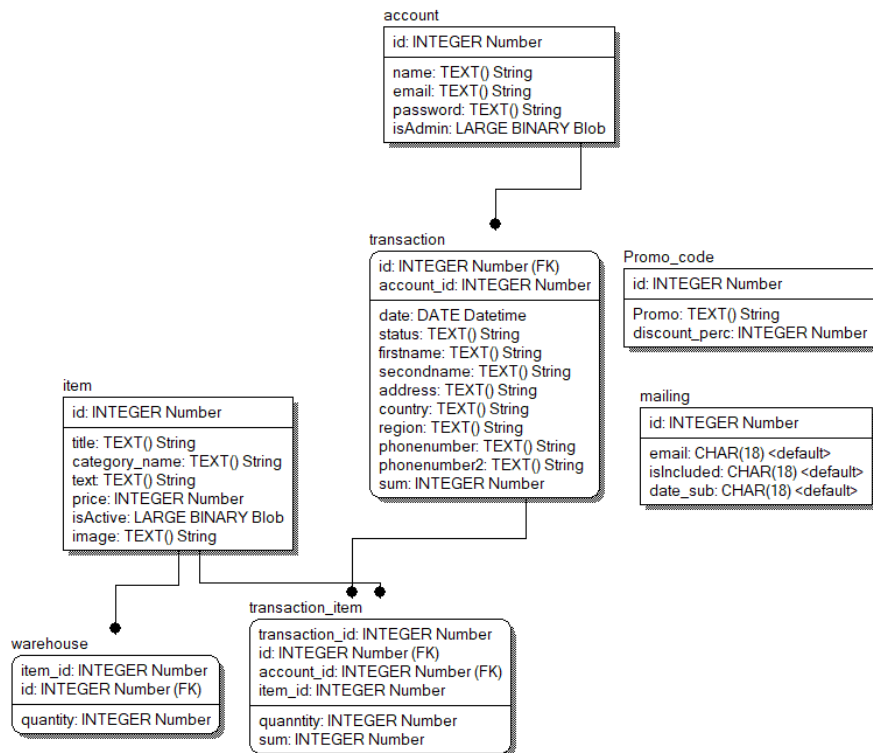


Рисунок 3.1 – логічна модель бази даних

На основі розробленої логічної моделі, розроблено фізичну модель (додаток), яка визначає спосіб зберігання та організацію даних на рівні операційної системи та апаратного забезпечення. В моделі визначено типи даних для кожного атрибута сутності.

Для створення проекту у СКБД, в налаштування erwin генеруємо sql-запит, щоб створити базу даних у SQLite.

Моделі формуються на мові Python за допомогою класів і знаходяться у файлі table.py

Нижче наведено приклад створення моделі Transaction (Тразнакції)
class Transaction(db.Model):

```

    id = db.Column(db.Integer, primary_key=True)
    account_id = db.Column(db.Integer, db.ForeignKey('account.id'),
nullable=False)

    date = db.Column(db.DateTime, nullable=False)
    status = db.Column(db.String(20), nullable=False)
    firstname = db.Column(db.String(32), nullable=False)
    secondname = db.Column(db.String(32), nullable=False)
    address = db.Column(db.String(50), nullable=False)
    country = db.Column(db.String(32), nullable=False)
    region = db.Column(db.String(32), nullable=False)
    phonenumber = db.Column(db.Text, nullable=False)
    phonenumber2 = db.Column(db.Text, nullable=False)
    sum = db.Column(db.Integer, nullable=False)
    items = relationship('TransactionItem', backref='transaction', lazy=True)

```

Аналогічним способом описані всі інші сутності БД.

Для генерування БД на основі моделей, потрібно запустити проект і в терміналі ввести наступні дві команди у відповідній послідовності:

1. python
2. from main import app
3. from table import db

4. db.create_all()

Генерована БД зображена на рис. 3.2.

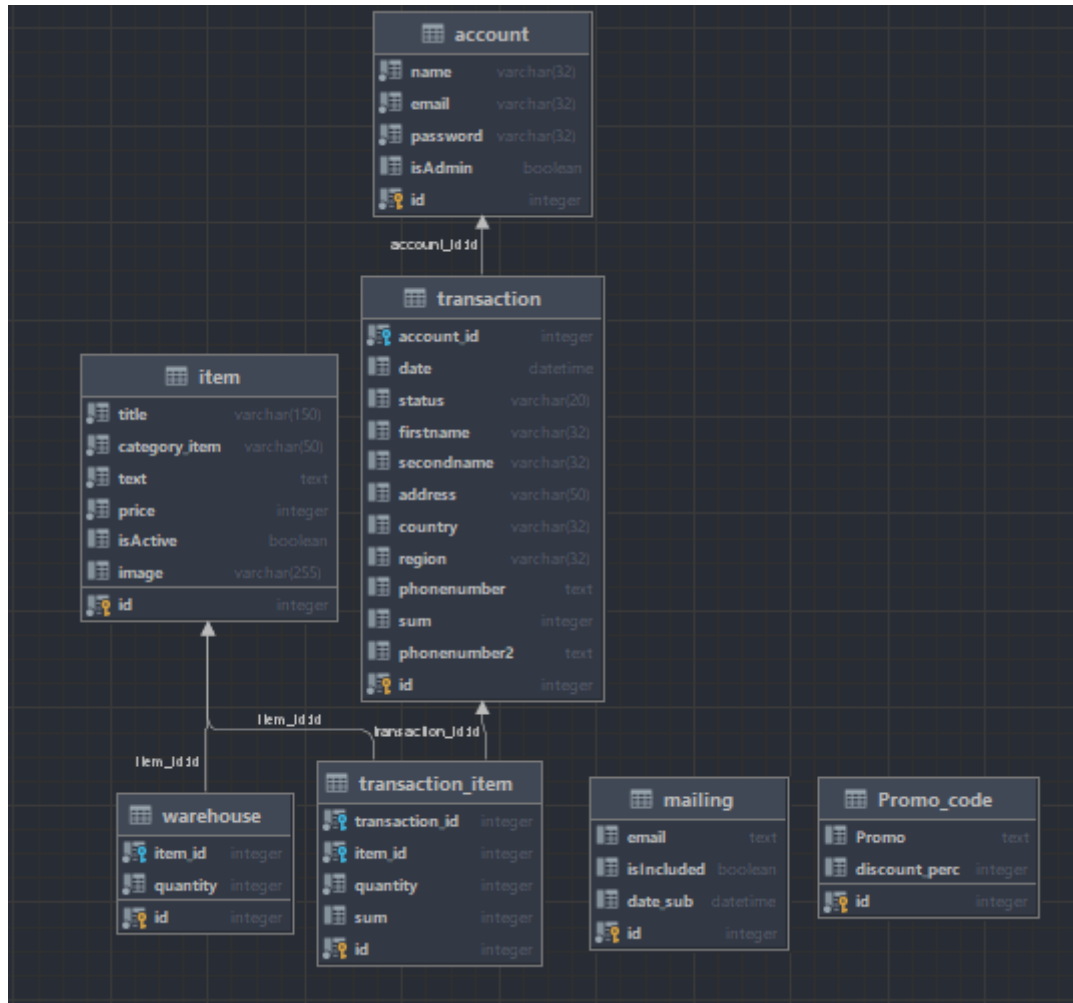


Рисунок 3.2 – Генерована БД

3.2. Алгоритмізація та реалізація комплексу задач автоматизації

Функціональні елементи головної сторінки це кнопки, зображення, кнопки для переходу на сторінки "Про нас" та "Магазин".

Використання іконок та фотографій для візуального представлення інформації.

Описові заголовки та текст, які пояснюють переваги продукції та діяльність компанії.

Ця структура зображена на рис. 3.3. Вона дозволяє користувачам легко орієнтуватися на сайті, дізнаватися про компанію, її переваги, та переглядати продукцію, що сприяє залученню нових клієнтів і підвищенню довіри до бренду.

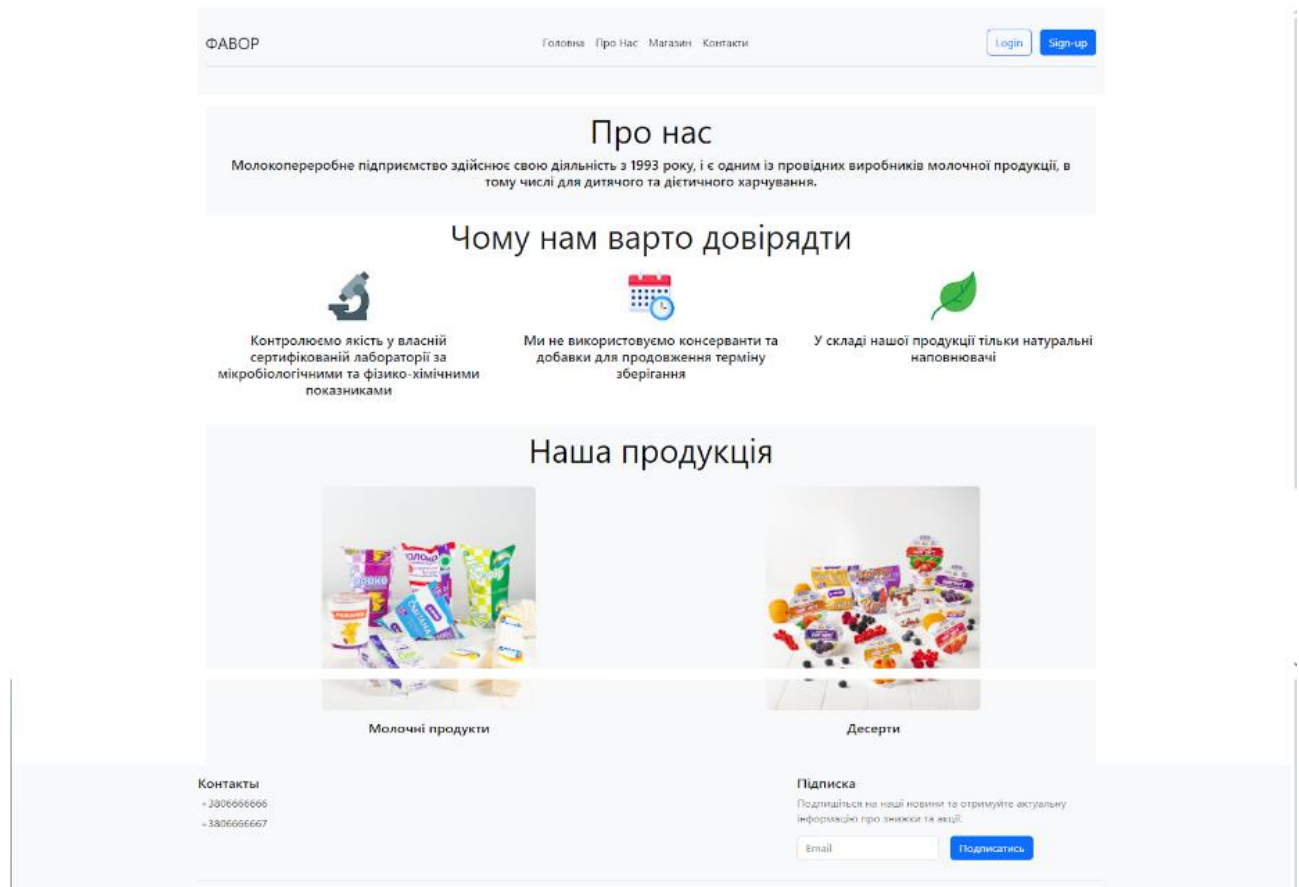


Рисунок 3.3 – Головна сторінка

Перехід між сторінками за допомогою верхнього меню з посиланнями та кнопками. За формування сторінки відповідає `main.py` з використанням функції `index`, нижче наведений код котрий виконується на сервері.

```
@app.route('/', methods=['GET', 'POST'])
def index():
    return render_template('index.html')
```

Ця функцію відтворює головну сторінку `index.html`, котра є шаблоном для `base.html`, таким чином, не потрібно кожен раз повторювати на сторінках верхнє меню та підвал з цим допомагає шаблонізатор Jinja.

```
{% extends 'base.html' %}
{% block title %}
<title>Головна</title>
{% endblock %}
```

{% block main %}

{% endblock %}

Для форми реєстрації та входу в систему було додано дві сторінки, наведені нижче на рис. 3.4 та рис. 3.5.

Рисунок 3.4 – форма для реєстрації

Рисунок 3.5 – форма для входу у систему

Під час реєстрації йде перевірка на наявність у БД пошти, якщо така є, виводиться помилка з повідомленням – “Обліковий запис з цією адресою електронної пошти вже існує”.

Після вдалого введення даних у поля та натискання на конопку запускається функція в `main.py` `signup` код наведений нижче.

```
@app.route('/account/signup', methods=['POST', 'GET'])
```

```
def signup():
```

```

if request.method == "POST":
    name = request.form['nickname']
    email = request.form['email']
    password = request.form['password']

    existing_account = Account.query.filter_by(email=email).first()
    if existing_account:
        return render_template('account/signup.html',
                               message="Обліковий запис з цією адресою електронної
пошти вже існує")

    account = Account(name=name, email=email, password=password)
    try:
        db.session.add(account)
        db.session.commit()
        return redirect(url_for('login'))
    except Exception as e:
        return str(e)

return render_template('account/signup.html', message="")

```

Після вдалої реєстрації система перенаправляє користувача на сторінку Входу, де йому потрібно ввести пошту та пароль. Натискаючи на кнопку запускається функція у main.py login котра і здійснює перевірку та надає права адміністратора, та додає до верхнього меню кнопку Адмін панель, котру звичайний користувач не бачить (якщо в БД це зазначено). Якщо дані введено не вірно, буде виведено помилку – “Невірний пароль чи E-mail.”.

```

@app.route('/account/login', methods=['GET', 'POST'])
def login():
    message = ""
    if request.method == 'POST':

```

```

email = request.form['email']
password = request.form['password']

account = Account.query.filter_by(email=email).first()
if account and account.password == password:
    session['user_id'] = account.id
    session['isAdmin'] = account.isAdmin
    session['cart_count'] = 0
    return redirect(url_for('index'))
else:
    message = "Невірний пароль чи E-mail."

return render_template('account/login.html', message=message)

```

Сторінка магазину створена таким ж чином як і інші сторінки, через шаблонізатор зображено на рис 3.6. На сторінці можна побачити всі продукти, та відфільтрувати по категоріям, якщо ми увійшли у систему, то ми зможемо зробити замовлення, інакше кнопка додавання до кошику буде нам надсилати на реєстрацію.

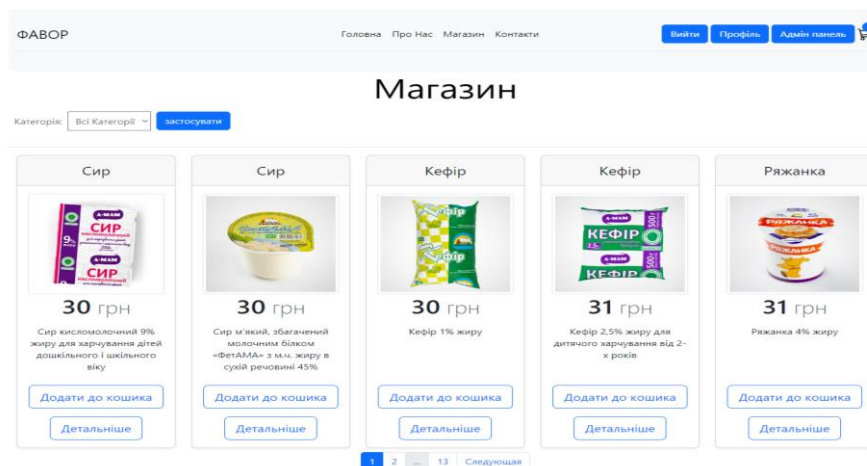


Рисунок 3.6 – Сторінка магазину

На рис 3.7. можна побачити всі категорії, обравши якусь та натиснувши застосувати оновиться сторінка з товарами тієї категорії котру обрали. Оберемо у категорії десерт та застосуємо, на рис.3.8 це зображено.

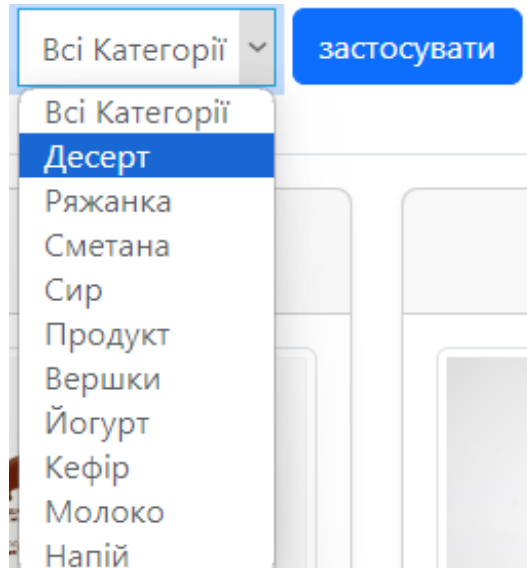


Рисунок 3.7 – Випадаючий список категорій магазину

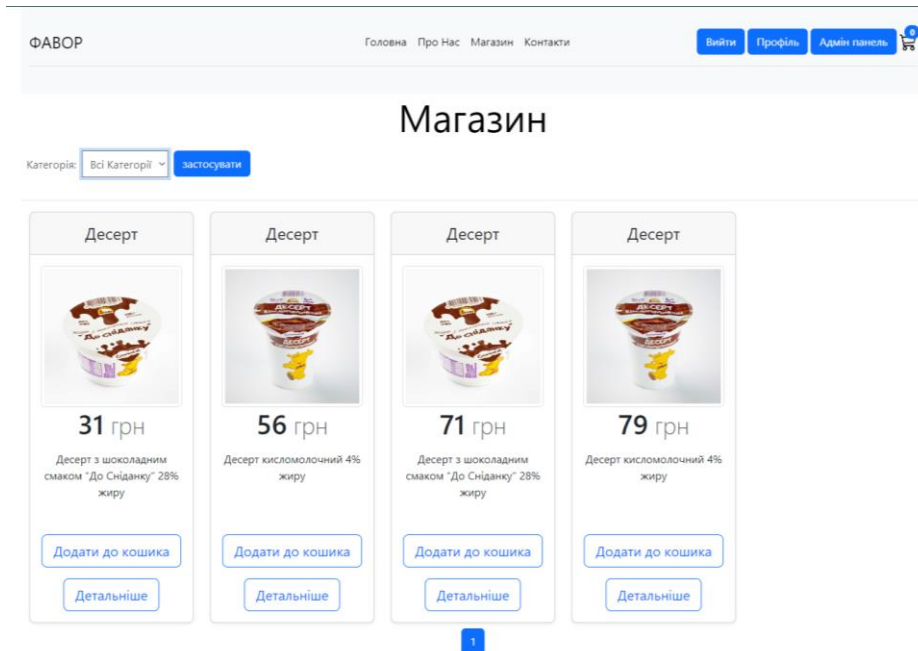


Рисунок 3.8 – Обрана категорія десерт

За сторінку магазину відповідає така функція як shop код якої наведено нижче.

```
@app.route('/shop', methods=['POST', 'GET'])
```

```

def shop():
    page = request.args.get('page', 1, type=int)
    items_per_page = 5
    selected_category = request.form.get('filter')

    if selected_category and selected_category != "0":
        items = Item.query.filter_by(category_item=selected_category).order_by(Item.price).paginate(
            page=page,
            per_page=items_per_page)
    else:
        items = Item.query.order_by(Item.price).paginate(page=page,
            per_page=items_per_page)

    categories = set(item.category_item for item in Item.query.all())

    return render_template('shop.html', items=items, categories=categories)

```

Після натискання на кнопку Додати до кошика, відбувається такий процес, код якого наведено нижче. Ця частина коду відповідає за анімацію натискання та передачу даних на сервер, кількість та який товар треба додати до кошика користувача.

```

function addToCart(itemId, button) {
    const countSelect = document.getElementById('count');
    const itemCount = countSelect ? countSelect.value : 1;

    fetch('/add_to_cart', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',

```

```

    },
    body: JSON.stringify({ itemId: itemId, itemCount: itemCount }),
  })
  .then(response => {
    if (response.ok) {
      updateCartCounter();

      if (button) {
        button.style.backgroundColor = 'green';
        button.style.color = 'white';
        button.innerText = 'Додано';

        setTimeout(function () {
          button.style.backgroundColor = '';
          button.style.color = '';
          button.innerText = 'Додати до кошика';
        }, 500);
      }
      return response.json();
    } else {
      // Обработка ошибки добавления товара в корзину
      throw new Error('При добавлении товара в корзину возникла
ошибка!');
    }
  })
  .catch(error => {
    console.error('Ошибка:', error);
    alert(error.message);
  });

```

```
}
```

Нижче наведений код функції `add_to_cart`, котра відповідає за отримання цих даних та обробку їх.

```
@app.route('/add_to_cart', methods=['POST'])
def add_to_cart():
    if 'user_id' in session:
        try:
            data = request.json
            item_id = data['itemId']
            count = data['itemCount']
            for _ in range(int(count)):
                cart_items = session.get('cart_items', [])
                cart_items.append(item_id)
                session['cart_items'] = cart_items
            return jsonify({'message': 'Товар додано у кошик!'}), 200
        except Exception as e:
            return jsonify({'error': str(e)}), 500
    else:
        return jsonify({'error': 'Користувач не аутентифікований!'}), 401
```

Також можна перейти за кнопкою Детальніше, де можна подивитись більше інформації про товар, та також додати його в кошик, з можливістю обрати кількість товару. Сторінка продукту зображена на рис. 3.9.

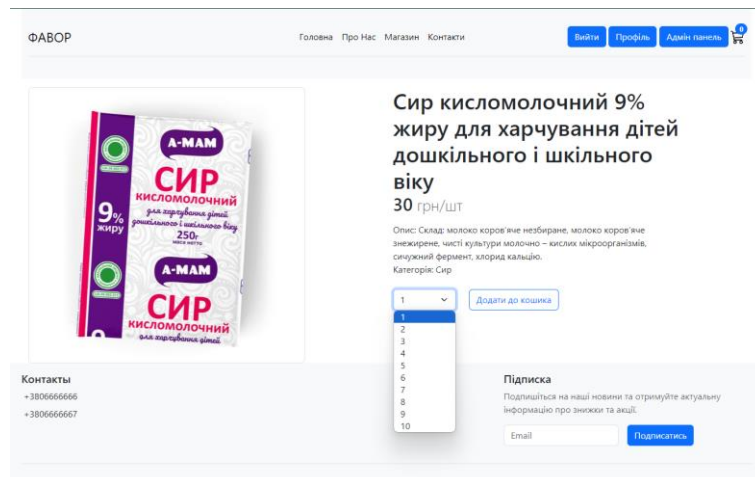


Рисунок 3.9 – Сторінка продукту

Після додавання всіх потрібних продуктів до кошика треба перейти до нього за піктограмою у правому верхньому кутку, там буде зображено кількість усіх товарів доданих до кошику у синьому кружечку.

Натиснувши на кошик з'являється форма, котру треба заповнити для успішного замовлення. Пошта та ім'я користувача заповнюються автоматично.

Також можна змінити кількість товару котрий ми захочемо замовити, для цього треба буде натиснути кнопку оновити.

Є можливість використати промокод котрий надає фірма, вони потрібні для утримання постійних клієнтів, котрих заохотить купляти продукцію повторно.

Рисунок 3.10 – Форма кошику.

Функція відповідальна за кошик наведено нижче.

```
@app.route('/create_order', methods=['POST', 'GET'])
def create_order():
    account_info = list()
    cart_sum = int()

    if request.method == "POST":
        print(request.form["button"])
        if request.form["button"] == 'reload_count':
            form_data = request.form.to_dict()
            print("form_data", form_data)
            session.pop('cart_items', None)
            cart_items = []
            for key, value in form_data.items():
                if key.startswith('count_'):

                    cart_item_id = key.split('_')[1]
                    print("cart_item_id ", cart_item_id)
                    for _ in range(int(value)):
                        cart_items.append(cart_item_id)
            session['cart_items'] = cart_items
            if 'cart_items' in session:
                order = Counter(session['cart_items'])
                cart_items = Item.query.with_entities(Item.id, Item.title, Item.price) \
                    .filter(Item.id.in_(order.keys())).all()
                for i in range(len(cart_items)):
                    cart_sum += cart_items[i][2] * order[str(cart_items[i][0])]

            sorted_order = sorted(order.items())
            formatted_order = [[int(k), v] for k, v in sorted_order]
```

```

if 'user_id' in session:
    user_id = session['user_id']
    account_info = Account.query.filter_by(id=user_id).first()
    return render_template("create_order.html",
                           sum=cart_sum,                cart_items=cart_items,
                           order=formatted_order,
                           account_info=account_info)

```

```

account_id = session['user_id']
date = datetime.now().date()
status = "Обработка"
firstname = request.form['firstname']
secondname = request.form['secondname']
address = request.form['address']
country = request.form['country']
region = request.form['region']
phonenumber = request.form['phonenumber']
phonenumber2 = request.form['phonenumber2']
promo = request.form['promocode']

```

```

order = Counter(session['cart_items'])
cart_items = Item.query.with_entities(Item.id, Item.title, Item.price) \
    .filter(Item.id.in_(order.keys())).all()
for i in range(len(cart_items)):

```

```

    cart_sum += cart_items[i][2] * order[str(cart_items[i][0])]

```

```

if promo:

```

```

    discount

```

```

    PromoCode.query.filter_by(promo=promo).first().discount_perc

```

=

if 'cart_items' in session:

```
order = Counter(session['cart_items'])
```

```
cart_items = Item.query.with_entities(Item.id, Item.title, Item.price) \
    .filter(Item.id.in_(order.keys())).all()
```

for i in range(len(cart_items)):

```
    cart_sum += cart_items[i][2] * order[str(cart_items[i][0])]
```

```
sorted_order = sorted(order.items())
```

```
formatted_order = [[int(k), v] for k, v in sorted_order]
```

if 'user_id' in session:

```
user_id = session['user_id']
```

```
account_info = Account.query.filter_by(id=user_id).first()
```

```
return render_template("create_order.html",
```

```
                        sum=cart_sum,
```

```
                        cart_items=cart_items,
```

```
                        order=formatted_order, account_info=account_info)
```

```
return render_template("create_order.html", account_info=account_info)
```

Після вдалого заповнення даних на доставку, вони передаються у відділ логістики, також ми можемо їх переглянути у профілі зображено на рис.

3.11.

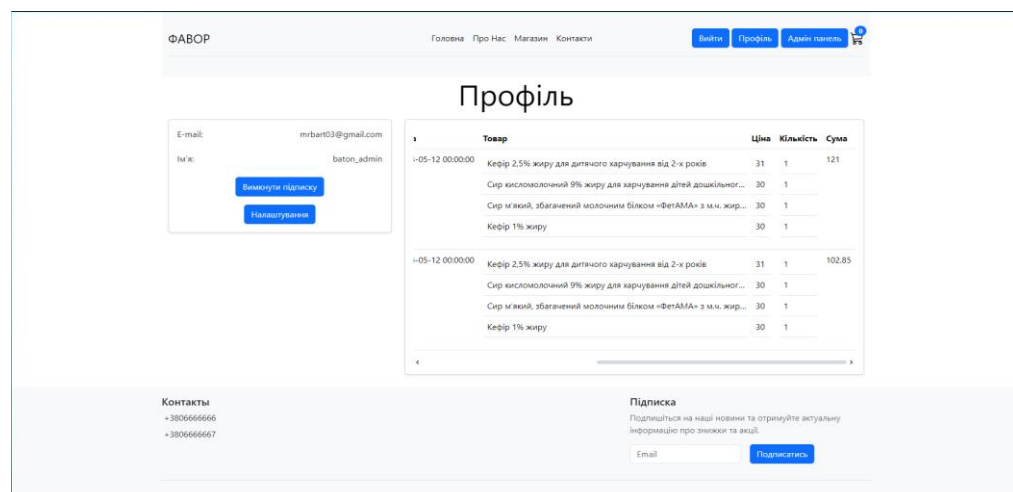


Рисунок 3.11 – Профіль користувача

Також у профілі можна переглянути та змінити дані акаунти, такі як ім'я користувача, пароль, чи пошту зображено на рис. 3.12. Є функція увімкнення чи вимкнення підписки на новини.

Рисунок 3.12 – Налаштування акаунту

Нижче наведені функції `settings_profile` та `profile`, котрі відповідають за сторінку налаштування та профілю.

```
@app.route('/account/profile', methods=['POST', 'GET'])
```

```
def profile():
```

```
    user_id = session.get('user_id')
```

```
    if user_id:
```

```
        account = Account.query.get(user_id)
```

```
        transaction
```

```
        Transaction.query.filter_by(account_id=user_id).order_by(Transaction.status.desc(),
```

```
                                Transaction.date.desc()).all()
```

```
        transaction_item = TransactionItem.query.all()
```

```
        item = Item.query.all()
```

```
        mailing = Mailing.query.filter_by(email=account.email).first()
```

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

```

if request.form["button"] == "offMailing":
    mailing.isIncluded = False
elif request.form["button"] == "onMailing":
    if not mailing:
        new_email = Mailing(email=account.email,
date_sub=datetime.now().date())

        db.session.add(new_email)
    else:
        mailing.isIncluded = True
        send_notification(account.email)

    db.session.commit()

if account:
    return render_template('account/profile.html', account=account,
transaction=transaction,
transaction_item=transaction_item, item=item,
mailing=mailing)

return render_template('index.html')

@app.route('/account/settings', methods=['POST', 'GET'])
def settings_profile():
    user_id = session.get('user_id')
    if user_id:
        account = Account.query.get(user_id)

    if request.method == "POST":

```

```

if request.form['old_password'] == account.password:
    account.password = request.form['new_password']

if request.form['email'] != account.email and \
    "@" in request.form['email'] and \
    (".com" in request.form['email']
     or ".net" in request.form['email']
     or ".ua" in request.form['email']):
    existing_account =
Account.query.filter_by(email=request.form['email']).first()
    if not existing_account:
        account.email = request.form['email']

if request.form['name'] != account.name:
    account.name = request.form['name']

db.session.commit()
return redirect(url_for('profile'))

if account:
    return render_template('account/settings.html', account=account)

return render_template('index.html')

```

У цей час в панелі адміністратора з'являється помітка, що було зроблене замовлення, це виглядає таким чином рис. 3.13. Біля кнопки транзакції починає відображатись цифра з кількістю замовлень котрі на обробці. Натиснувши на неї виводиться таблиця з усіма замовленнями, замовлення котрі вже доставлені відправляються у самий низ, доставляються – у середині. І, таким чином, обробка завжди зверху, для зручного використання таблиці.

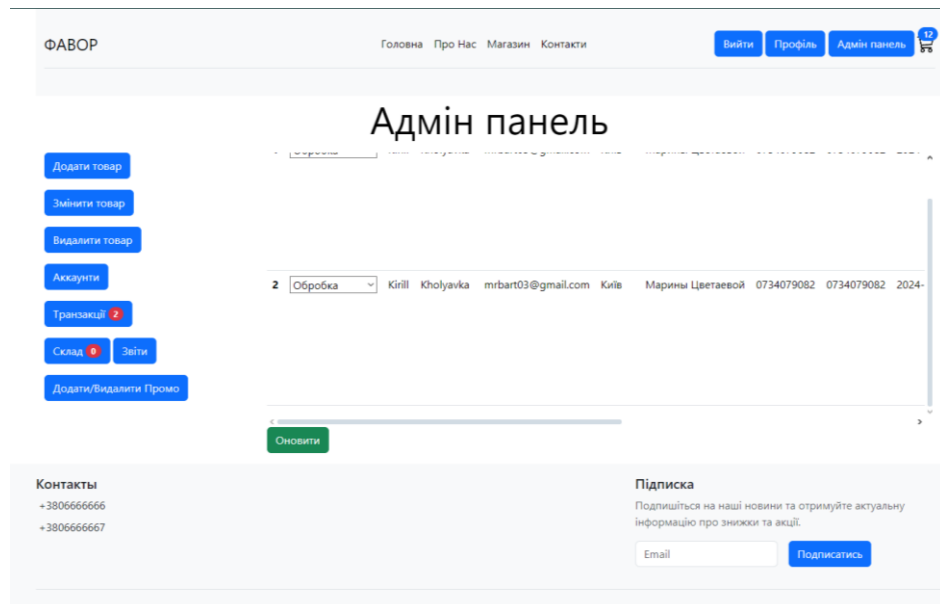


Рисунок 3.13 – Адмін панель з натиснутою кнопкою транзакції.

Функція `admin_panel` до сторінки наведена нижче. У ній відбувається звертання до інших функцій, за для виводу таблиць, звітів, чи форм.

```
@app.route('/admin/adminpanel', methods=['POST', 'GET'])
```

```
def admin_panel():
```

```
    if 'user_id' not in session or not session['isAdmin']:
```

```
        return redirect(url_for('index'))
```

```
    count_trans = Transaction.query.filter_by(status='Обробка').count()
```

```
    count_warehouse = Warehouse.query.filter_by(quantity=0).count()
```

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

```
        if request.form['button'] == 'add_item':
```

```
            return create()
```

```
        if request.form['button'] == 'update_item':
```

```
            return update()
```

```
        if request.form['button'] == 'remove_item':
```

```
            return remove()
```

```
        if request.form['button'] == 'update_data':
```

```
            return table_transactions()
```

```
        if request.form['button'] == 'update_quantity':
```

```

return table_warehouse()
if request.form['button'] == 'csv_transaction' \
    or request.form['button'] == 'csv_items' \
    or request.form['button'] == 'csv_sum_items':
return report_button()
if request.form['button'] == 'add_promo' or request.form['button'] ==
'del_promo':
return create_promo()
return render_template('admin/adminpanel.html', count_trans=count_trans,
count_warehouse=count_warehouse)

```

На сторінці Адмін панель можна побачити багато кнопок, вони всі відповідають за форми, таблиці, та звіти. Нижче буде показано всі на рис.3.14. – рис. 3.20.

Рисунок 3.14 – Форма додавання товару

На формі додавання товару треба заповнити всі поля, після чого цей товар можна буде додати у БД. Нижче код додавання товарів.

```

@app.route('/admin/create', methods=['POST', 'GET'])
def create():

```

```

if 'user_id' not in session or not session['isAdmin']:
    return redirect(url_for('index'))

if request.method == "POST":
    title = request.form['title']
    category_item = request.form['category_item']
    text = request.form['text']
    price = request.form['price']
    count = request.form['count']
    isActive = True if request.form.get('isActive') == 'True' else False

    if 'image' in request.files:
        file = request.files['image']
        if file and allowed_file(file.filename):
            filename = secure_filename(f"{datetime.now()}_{file.filename}")
            file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
        else:
            return render_template('admin/adminpanel.html',
error='Непринятый формат файла')
        else:
            filename = ""

    item = Item(title=title, category_item=category_item, text=text,
                price=price, isActive=isActive, image=filename)

    try:
        db.session.add(item)
        db.session.commit()
        item_id = item.id
        warehouse = Warehouse(item_id=item_id, quantity=count)

```

```

    db.session.add(warehouse)

    db.session.commit()

    return render_template('admin/adminpanel.html')
except Exception as e:
    return print(e)

else:
    return render_template('admin/create.html')

```

Рисунок 3.15 – Форма оновлення товару

На формі оновлення товару можна обрати товар та оновити про нього дані. Нижче код оновлення товарів.

```

@app.route('/admin/update', methods=['POST', 'GET'])
def update():
    if 'user_id' not in session or not session['isAdmin']:
        return redirect(url_for('index'))
    if request.method == "POST":
        product_id = request.form.get('product_id')
        if product_id != "0":

```

```

product = Item.query.get(product_id)
product.title = request.form['title']
product.category_item = request.form['category_item']
product.text = request.form['text']
product.price = request.form['price']
product.isActive = True if request.form.get('isActive') == 'on' else False
new_quantity = int(request.form['count'])
warehouse = Warehouse.query.filter_by(item_id=product_id).first()
warehouse.quantity = new_quantity

db.session.commit()

return render_template('admin/adminpanel.html')

else:

return render_template('admin/adminpanel.html')

else:

products = Item.query.all()
return render_template('admin/update.html', products=products)

```

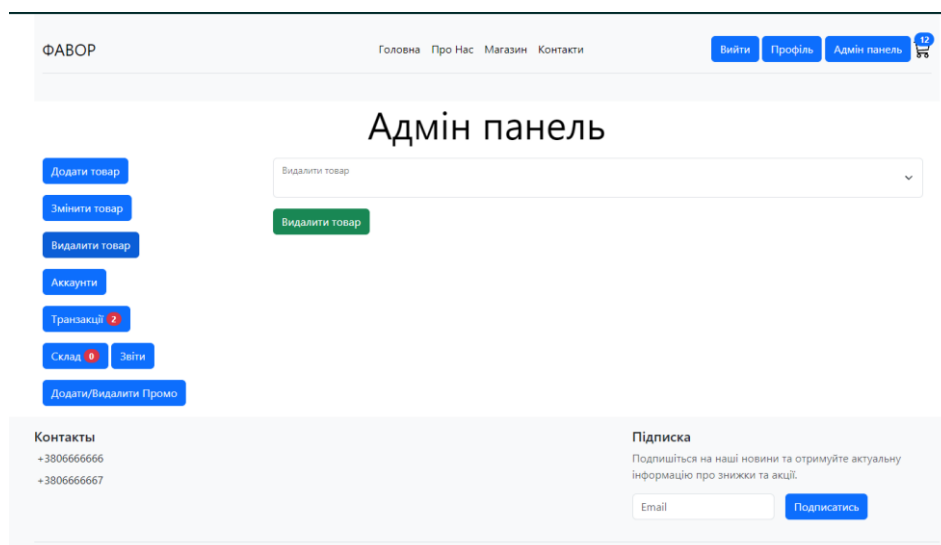


Рисунок 3.16 – Форма видалення товару

На формі видалення товару можна обрати товар та видалити його з БД.

Нижче код видалення товарів.

```

@app.route('/admin/remove', methods=['POST', 'GET'])
def remove():
    if 'user_id' not in session or not session['isAdmin']:
        return redirect(url_for('index'))
    if request.method == "POST":
        product_id = request.form.get('product_id')
        if product_id != "0":
            product = Item.query.get(product_id)
            warehouse_entries = Warehouse.query.filter_by(item_id=product_id).all()
            for entry in warehouse_entries:
                db.session.delete(entry)
            db.session.delete(product)
            db.session.commit()
            return render_template('admin/adminpanel.html')
    else:
        products = Item.query.all()
        return render_template('admin/remove.html', products=products)

```

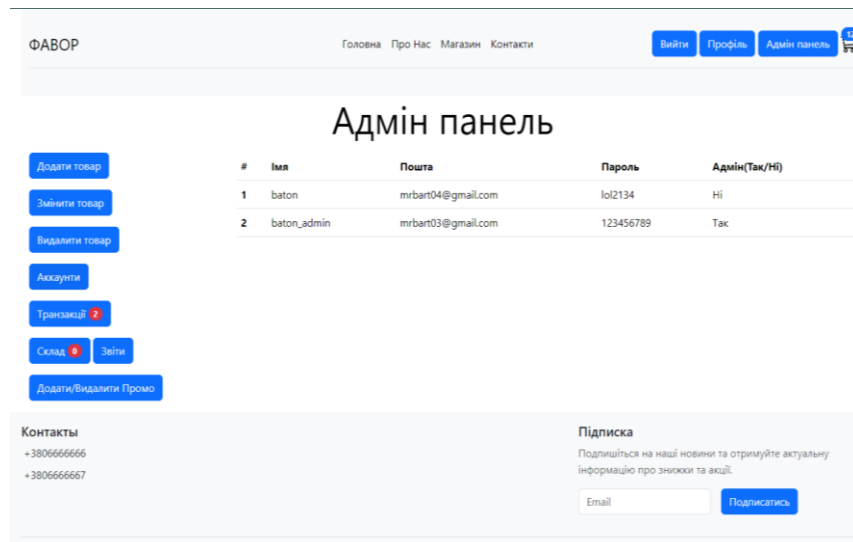


Рисунок 3.17 – Таблиця Акаунтів

В таблиці акаунтів можна побачити всіх користувачів цього сайту, також ми можемо побачити хто є адміном. Нижче код таблиці акаунтів.

```
@app.route('/admin/print_account')
def table_account():
    if 'user_id' not in session or not session['isAdmin']:
        return redirect(url_for('index'))
    account = Account.query.all()
    return render_template('admin/print_account.html', account=account)
```

#	Товар	Кількість
1	Молоко питне пастеризоване 2,5% жиру для харчування дітей від 2-х років	<input type="text" value="41"/>
2	Молоко питне пастеризоване 3,2% жиру для харчування дітей від 2-х років	<input type="text" value="41"/>
3	Молоко питне пастеризоване вітамінізоване 3,2% жиру для харчування дітей від 2-х років	<input type="text" value="45"/>
4	Молоко питне пастеризоване вітамінізоване 2,5% жиру для харчування дітей від 2-х років	<input type="text" value="72"/>
5	Кефір 2,5% жиру для дитячого харчування від 2-х років	<input type="text" value="61"/>
6	Кефір 1% жиру	<input type="text" value="33"/>
7	Сметана 20% жиру	<input type="text" value="30"/>
8	Сметана 20% жиру	<input type="text" value="36"/>

Рисунок 3.18 – Таблиця склад

В таблиці склад ми бачимо всі товари котрі є у наявності, якщо їх кількість стає 0, то біля кнопки склад це буде відобразатись. Нижче код таблиці складу.

```
@app.route('/admin/warehouse', methods=['POST', 'GET'])
def table_warehouse():
    if 'user_id' not in session or not session['isAdmin']:
        return redirect(url_for('index'))
    warehouse = Warehouse.query.all()
    item = Item.query.all()
    if request.method == "POST":
        form_data = request.form.to_dict()
```

```

for key, value in form_data.items():
    if key.startswith('quantity_'):
        warehouse_id = key.split('_')[1]
        new_quantity = value
        warehouse_to_update = Warehouse.query.get(warehouse_id)
        if warehouse_to_update:
            warehouse_to_update.quantity = new_quantity
            db.session.commit()
        else:
            print(f"{warehouse_id} не знайдена")
return redirect(url_for('admin_panel'))
return render_template('admin/warehouse.html', warehouse=warehouse,
item=item)

```

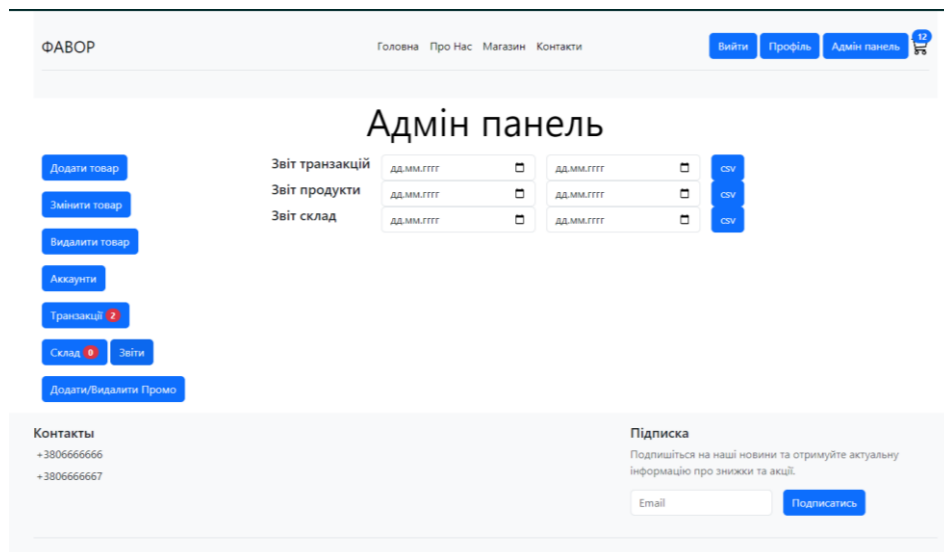


Рисунок 3.19 – Звіти

На сторінці звітів ми можемо обрати діапазон дат, за котрий нам потрібно згенерувати звіт. Нижче код звітів та генерації csv файлу та зображення звітів рис 3.19.1 – 3.19.3.

```

@app.route('/admin/report', methods=['POST', 'GET'])
def report_button():
    if 'user_id' not in session or not session['isAdmin']:
        return redirect(url_for('index'))
    trans_sum = 0
    if request.method == "POST":
        if request.form['button'] == 'csv_transaction':
            date_start = datetime.strptime(request.form['date_start'], '%Y-%m-%d').date()
            if request.form['date_end'] != "":
                date_end = datetime.strptime(request.form['date_end'], '%Y-%m-%d').date()
            else:
                date_end = datetime.now().date().strftime('%Y-%m-%d')
            if date_start == "":
                return render_template('admin/adminpanel.html')

            filename = f'instance/csv/transactions_{date_start} - {date_end}.csv'
            transactions = Transaction.query.filter(Transaction.date.between(date_start, date_end)).all()
            first_line = ["Звіт по транзакціям", "", "", " за ", "", "", f"({date_start} - {date_end})"]
            second_line = ['Дата', 'Статус', 'Ім'я', 'Прізвіще', 'Адреса', 'Країна', 'Область', 'Телефон', 'Телефон магазину', 'Сума']
            table = []
            for transaction in transactions:
                table.append([transaction.date.strftime('%Y-%m-%d'),
                               transaction.status,
                               transaction.firstname,
                               transaction.secondname,
                               transaction.address, transaction.country, transaction.region,

```

```

        transaction.phonenumber,          transaction.phonenumber2,
transaction.sum])
        trans_sum += int(transaction.sum)
        end_line = [",", " ", " ", " ", " ", " ", " ", " ", 'Сума', trans_sum]
        return send_file(file_bilder(table, second_line, filename, first_line,
end_line), as_attachment=True)

if request.form['button'] == 'csv_items':
    items_collection = []
    earning = 0
    date_start = datetime.strptime(request.form['date_start_2'], '%Y-%m-
%d').date()
    if request.form['date_end_2'] != "":
        date_end = datetime.strptime(request.form['date_end_2'], '%Y-%m-
%d').date()
    else:
        date_end = datetime.now().date().strftime('%Y-%m-%d')
    if date_start == "":
        return render_template('admin/adminpanel.html')
    transactions =
Transaction.query.filter(Transaction.date.between(date_start, date_end)).all()
    for transaction in transactions:
        transaction_items =
TransactionItem.query.filter_by(transaction_id=transaction.id).all()
        for item in transaction_items:
            items = Item.query.filter_by(id=item.item_id).first()
            items_collection.append([transaction.id,
transaction.date.strftime('%Y-%m-%d'),
items.title,
item.quantity,

```

```

        items.price
    ]
)

    earning += item.quantity * items.price
    first_line = ["Звіт по проданим товарам", " за ", f"({date_start} -
{date_end})"]
    second_line = ["№ транзакції", "дата", "назва", "кількість", "ціна"]
    filename = f'instance/csv/transaction_items_{date_start} - {date_end}.csv'
    end_line = [",", ",", " ", 'Заробіток', earning]
    return send_file(file_bilder(items_collection, second_line, filename,
first_line, end_line),
                    as_attachment=True)

if request.form['button'] == 'csv_sum_items':
    table = []
    date_start = datetime.strptime(request.form['date_start_3'], '%Y-%m-
%d').date()
    if request.form['date_end_3'] != "":
        date_end = datetime.strptime(request.form['date_end_3'], '%Y-%m-
%d').date()
    else:
        date_end = datetime.now().date().strftime('%Y-%m-%d')
    if date_start == "":
        return render_template('admin/adminpanel.html')
    transactions =
Transaction.query.filter(Transaction.date.between(date_start, date_end)).all()
    item_sales = Counter()

    for transaction in transactions:

```

```

        transaction_items =
TransactionItem.query.filter_by(transaction_id=transaction.id).all()
        for item in transaction_items:
            item_title = Item.query.filter_by(id=item.item_id).first().title
            item_sales[item_title] += item.quantity

        for title, count in item_sales.items():
            table.append([title, count])

        first_line = ["Звіт кількість проданих товарів", f" ({date_start} -
{date_end})"]
        second_line = ["Назва", "Кількість"]
        filename = f'instance/csv/transaction_items_count_{date_start} -
{date_end}.csv'

        return send_file(file_bilder(table=table, second_line=second_line,
            filename=filename, first_line=first_line),
            as_attachment=True)
    return render_template('admin/report.html')

def file_bilder(table, second_line, filename, first_line=None, end_line=None):
    with codecs.open(filename, mode='w', encoding='Windows-1251') as file:
        writer = csv.writer(file)

        if first_line:
            writer.writerow(first_line)

        writer.writerow(second_line)

        for row in table:

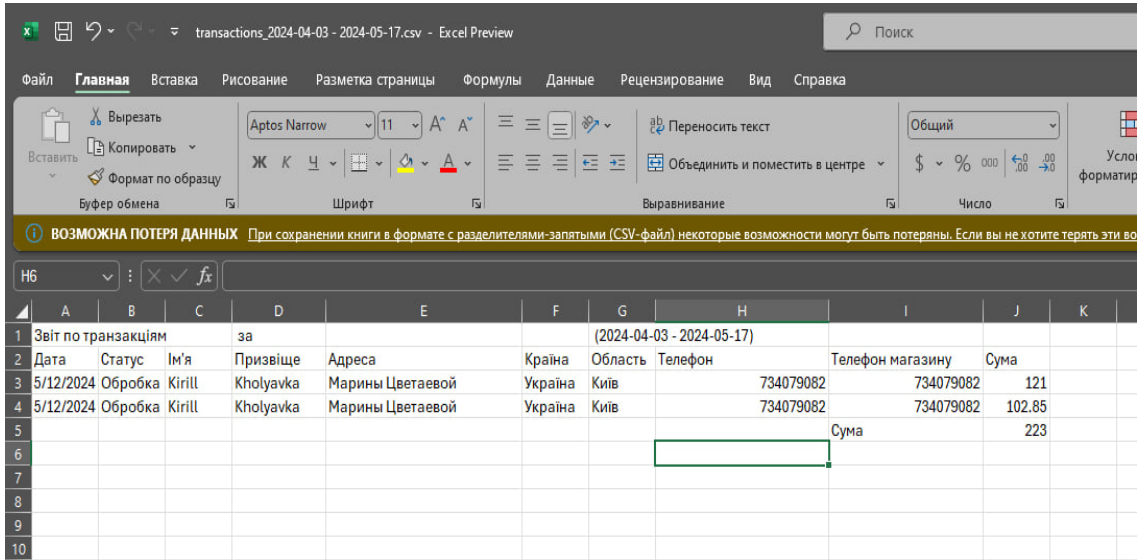
```

writer.writerow(row)

if end_line:

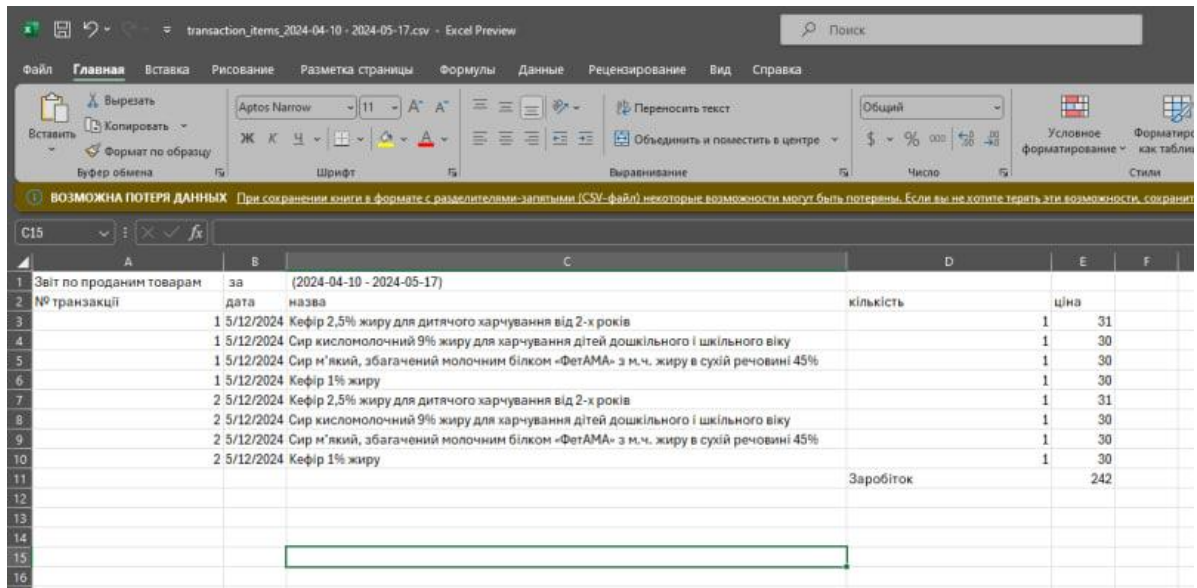
writer.writerow(end_line)

return filename



№	Дата	Статус	Ім'я	Прізвище	Адреса	Країна	Область	Телефон	Телефон магазину	Сума
1	5/12/2024	Обробка	Kirill	Kholyavka	Марины Цветаевой	Україна	Київ	734079082	734079082	121
4	5/12/2024	Обробка	Kirill	Kholyavka	Марины Цветаевой	Україна	Київ	734079082	734079082	102.85
									Сума	223

Рисунок 3.19.1 – Звіт по транзакціям



№ транзакції	дата	назва	кількість	ціна
1	5/12/2024	Кефір 2,5% жиру для дитячого харчування від 2-х років	1	31
2	5/12/2024	Сир кисломолочний 9% жиру для харчування дітей дошкільного і шкільного віку	1	30
3	5/12/2024	Сир н'який, збагачений молочним білком «ФетАМА» з н.ч. жиру в сухій речовині 45%	1	30
4	5/12/2024	Кефір 1% жиру	1	30
5	5/12/2024	Кефір 2,5% жиру для дитячого харчування від 2-х років	1	31
6	5/12/2024	Сир кисломолочний 9% жиру для харчування дітей дошкільного і шкільного віку	1	30
7	5/12/2024	Сир н'який, збагачений молочним білком «ФетАМА» з н.ч. жиру в сухій речовині 45%	1	30
8	5/12/2024	Кефір 1% жиру	1	30
			Заробіток	242

Рисунок 3.19.2 – Звіт по проданим товарам

Звіт кількість проданих товарів		(2024-04-10 - 2024-05-17)		
Назва	Кількість			
Кефір 2,5% жиру для дитячого харчування від 2-х років	2			
Сир кисломолочний 9% жиру для харчування дітей дошкільного і шкільного віку	2			
Сир м'який, збагачений молочним білком «ФетАМА» з м.ч. жиру в сухій речовині 45%	2			
Кефір 1% жиру	2			

Рисунок 3.19.3 – Звіт по кількості проданих товарів

ФАВОР Головна Про Нас Магазин Контакти [Вийти](#) [Профіль](#) [Адмін панель](#)

Адмін панель

[Додати товар](#)

[Змінити товар](#)

[Видалити товар](#)

[Аккаунти](#)

[Транзакції 2](#)

[Склад 0](#) [Звіти](#)

[Додати/Видалити Промо](#)

Промо-код

Знижка %

[Додати промо](#) [Видалити промо](#)

#	Промо	Знижка
1	2024	15
2	2023	10
3	213	12
4	23322	23

Контакти
 +38066666666
 +38066666667

Підписка
 Подпишіться на наші новини та отримуйте актуальну інформацію про знижки та акції.
 Email [Підписатись](#)

Рисунок 3.20 – Форма додавання, видалення промокодів та таблиця

В формі промокодів можна додати, чи видалити промо-код, також можна подивитись промо-коди котрі вже існують. Нижче код для форми промокодів.

```
@app.route('/admin/promo_code', methods=['POST', 'GET'])
```

```

def create_promo():
    if 'user_id' not in session or not session['isAdmin']:
        return redirect(url_for('index'))

    promocode_table = PromoCode.query.all()

    if request.method == "POST":
        if request.form['button'] == 'add_promo':
            promo = request.form['title']
            discount = request.form['discount']
            if promo and discount:
                promocode = PromoCode(promo=promo, discount_perc=discount)
                db.session.add(promocode)
                db.session.commit()
                return redirect(url_for('admin_panel'))
            elif request.form['button'] == 'del_promo':
                promo = request.form['title']
                if promo:
                    promocode = PromoCode.query.filter_by(promo=promo).first()
                    if promocode:
                        db.session.delete(promocode)
                        db.session.commit()
                        return redirect(url_for('admin_panel'))
        return render_template('admin/promo_code.html',
            promo_code=promocode_table)

```

На сторінці профілю, чи у підвалі сайту можна підписатись на новини зображено на рис. 3.21. та рис. 3.22., таким чином ми будемо отримувати нові промо-коди, чи новини.

E-mail: mrbart03@gmail.com

Ім'я: baton_admin

Вимкнути підписку

Рисунок 3.21 – Увімкнення чи вимкнення підписки на сторінці профілю користувача

Підписка

Подпишіться на наші новини та отримуйте актуальну інформацію про знижки та акції.

Email

Підписатись

Рисунок 3.22 – Підписка на новини у підвалі сайту

Функція відповідальна за підписку `subscribe` та `send_notification`

```
@app.route('/subscribe', methods=['POST'])
```

```
def subscribe():
```

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

```
        email = request.json.get('email') # Получаем значение электронной
        почты из запроса
```

```
        if email:
```

```
            existing_email = Mailing.query.filter_by(email=email).first()
```

```
            if not existing_email:
```

```
                new_email = Mailing(email=email, date_sub=datetime.now().date())
```

```
                db.session.add(new_email)
```

```
                db.session.commit()
```

```
                send_notification(email, message='Дякую за підписку! Ви будете
                отримувати наші новини та промокоди.')
```

```
            #todo email
```

```
return jsonify({'message': ''}), 200
```

```
def send_notification(email, message):
    msg = Message('Ви підписались на наші новини',
                  sender='mrbart03@gmail.com',
                  recipients=[email])
```

```
msg.body = f'{message}'
```

```
try:
```

```
    result = mail.send(msg)
```

```
    if result is None:
```

```
        print("200", email)
```

```
    else:
```

```
        print("403", email, result)
```

```
except Exception as e:
```

```
    print("500", email, e)
```

Функція відповідальна за вимкнення та увімкнення підписки знаходиться у Profile, частина коду нижче.

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

```
    if request.form["button"] == "offMailing":
```

```
        mailing.isIncluded = False
```

```
    elif request.form["button"] == "onMailing":
```

```
        if not mailing:
```

```
            new_email = Mailing(email=account.email,
```

```
            date_sub=datetime.now().date())
```

```
            db.session.add(new_email)
```

```
        else:
```

```
            mailing.isIncluded = True
```

send_notification(account.email, message='Дякую за підписку! Ви будете отримувати наші новини та промокоди.')

db.session.commit()

3.3. Інструкція користувача

Ця інструкція призначена для користувачів веб-сайту інтернет-магазину молочної продукції ТОВ "ФАВОР". Вона містить кроки для реєстрації, входу, перегляду продукції, додавання товарів до кошика та оформлення замовлення.

3.3.1. Процес реєстрації користувача:

1. перейдіть на головну сторінку сайту;
2. натисніть кнопку "Реєстрація" у верхньому правому куті сторінки;
3. заповніть форму реєстрації, вказавши необхідну інформацію:
 - a. ім'я;
 - b. електронна пошта;
 - c. пароль.
4. підтвердіть реєстрацію, натиснувши кнопку "Зареєструватися".

3.3.2. Процес входу до системи:

1. на головній сторінці сайту натисніть кнопку "Вхід" у верхньому правому куті;
2. введіть свої облікові дані (електронну пошту та пароль);
3. натисніть кнопку "Увійти".

3.3.3. Процес перегляду магазину продукції:

1. перейдіть на головну сторінку сайту;
2. натисніть на посилання "Магазин" у верхньому меню;
3. оберіть потрібну категорію продукції з меню;
4. перегляньте доступні товари у вибраній категорії.
5. Процес додавання товарів до кошика:
6. знайдіть потрібний товар у магазині;

7. натисніть на додати до кошика або детальніше для переходу на сторінку детального опису;
8. оберіть необхідну кількість товару у відповідному полі;
9. натисніть кнопку "Додати до кошика".

3.3.5. Перегляд кошика:

1. натисніть на іконку кошика у верхньому правому куті сторінки;
2. перегляньте список доданих товарів, їх кількість та загальну суму замовлення;
3. за необхідності змініть кількість товарів або видаліть непотрібні позиції.

3.3.6. Оформлення замовлення:

1. перейдіть до кошика, натиснувши на іконку кошика у верхньому правому куті;
2. Заповніть форму оформлення замовлення, вказавши:
 - a. Ім'я та прізвище;
 - b. Контактний номер телефону;
 - c. номер телефону магазину;
 - d. Адресу доставки;
 - e. Промо-код (за наявністю).
3. Перевірте введену інформацію та натисніть кнопку "Підтвердити замовлення".

3.3.7. Управління обліковим записом:

1. Увійдіть до свого облікового запису;
2. Перейдіть до розділу "Профіль" у верхньому меню;
3. Оновіть особисту інформацію, за потреби змініть пароль або пошту;
4. Перегляньте історію замовлень та стан поточних замовлень.

3.3.8. Підтримка користувачів:

1. Перейдіть до розділу "Контакти" у верхньому меню сторінки.
2. Скористайтеся контактними даними для зв'язку з технічною підтримкою.

3. Отримайте відповідь на ваше питання або допомогу у вирішенні проблеми протягом 24 годин.

3.4. Технічне та системне забезпечення розробки

3.4.1. Обґрунтування вибору технічних засобів

Для забезпечення стабільної та ефективної роботи веб-сайту інтернет-магазину ТОВ "ФАВОР" були обрані такі технічні засоби:

1. Серверне обладнання:

- a. Віртуальний приватний сервер (VPS) або хмарні рішення (наприклад, Amazon Web Services, Google Cloud Platform) для розміщення веб-додатка.
- b. Параметри сервера: CPU – 4 ядра, RAM – 8 GB, SSD – 500 GB, що достатньо для обслуговування середнього рівня трафіку і забезпечення швидкого доступу до даних.

2. Мережеве обладнання:

- a. Роутери та комутатори для забезпечення надійного та швидкого з'єднання між сервером і користувачами.
- b. Файрволи для захисту серверної інфраструктури від зовнішніх загроз.

3. Резервне копіювання:

- a. NAS-сервер (Network Attached Storage) або хмарне сховище для регулярного резервного копіювання даних.

3.4.2. Розрахунок та визначення топології комп'ютерної мережі

Топологія комп'ютерної мережі для підтримки роботи веб-сайту включає:

1. Тип топології: Зіркова топологія, яка забезпечує центральний вузол (сервер) для обслуговування запитів від клієнтських пристроїв.
2. Компоненти мережі:
 - a. Центральний сервер: Розташований в дата-центрі або хмарному середовищі.
 - b. Клієнтські пристрої: Підключаються до сервера через Інтернет.

- с. Мережеве обладнання: Роутери, комутатори, файрволи для забезпечення безпеки та надійності з'єднань.

3. Схема підключення:

- а. Клієнтські пристрої -> Інтернет -> Роутер/файрвол -> Центральний сервер.
- б. Резервне копіювання: Центральний сервер -> NAS/хмарне сховище.

3.4.3. Обґрунтування вибору ОС та протоколу обміну даними

Операційна система:

1. Windows: Вибір обґрунтований надійністю, безпекою, відкритим кодом та широкою підтримкою для веб-серверів.
2. Особливості: Легка налаштуваність, стабільність роботи, наявність великої кількості інструментів для адміністрування та розробки.
3. Протокол обміну даними:
 - а. HTTP/HTTPS: Основні протоколи для передачі даних між клієнтськими браузером та веб-сервером. HTTPS забезпечує шифрування даних для безпеки передачі.
 - б. SSH: Для захищеного віддаленого доступу до сервера адміністраторів.

3.4.4. Розробка і обґрунтування стратегії адміністрування системи

Моніторинг та підтримка:

1. Регулярні оновлення ОС та програмного забезпечення для забезпечення безпеки та стабільності роботи.
2. Резервне копіювання:
 - а. Автоматизовані скрипти резервного копіювання для створення копій бази даних та файлів веб-сайту з регулярністю (щоденно, щотижнево).
 - б. Перевірка цілісності резервних копій та тестування відновлення.
3. Відновлення після збоїв:
 - а. Розробка плану відновлення для швидкого відновлення роботи сервера та веб-сайту у разі збою.

б. Реплікація даних у реальному часі для мінімізації втрат даних.

3.4.5. Заходи захисту від несанкціонованого доступу до системи

1. Фізичний захист:
2. Сервери розміщені в дата-центрах з високим рівнем фізичної безпеки, включаючи контроль доступу, відеоспостереження та охорону.
3. Мережевий захист:
 - а. Використання файрволів для контролю та фільтрації мережевого трафіку.
 - б. VPN для безпечного доступу до внутрішніх ресурсів.
4. Захист даних:
 - а. Шифрування даних на рівні з'єднання (HTTPS) та зберігання (шифрування бази даних).
 - б. Використання складних паролів та двофакторної автентифікації для адміністраторів та користувачів.
5. Програмний захист:
 - а. Регулярне оновлення системи та встановлених програм для усунення вразливостей.
 - б. Сканування на вразливості та аудит безпеки.
6. Політика безпеки:
 - а. Розробка та впровадження політики безпеки для всіх співробітників, що включає правила користування системою та обробки конфіденційних даних.
 - б. Навчання персоналу основам кібербезпеки та правилам роботи з інформаційною системою.

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. В даному розділі розглядаються питання охорони праці та техніки безпеки, які необхідно враховувати при розробці та впровадженні програмного забезпечення для інтернет-магазину молочної продукції. Забезпечення безпечних умов праці є невід'ємною частиною успішної роботи підприємства, оскільки це допомагає запобігти нещасним випадкам та знизити ризики для здоров'я працівників.

4.2. Організація охорони праці на підприємстві

На підприємстві повинна бути організована система охорони праці, яка включає в себе:

- Проведення інструктажів з техніки безпеки для всіх працівників. Інструктажі мають проводитися регулярно та охоплювати всі аспекти безпечної роботи.
- Регулярне проведення навчальних занять та тренінгів з питань охорони праці. Це дозволяє постійно підвищувати рівень обізнаності працівників щодо можливих ризиків та методів їх уникнення.
- Розробка та впровадження заходів для зниження ризиків виробничих травм. Важливо впроваджувати інноваційні рішення та використовувати сучасні засоби захисту.
- Забезпечення працівників необхідними засобами індивідуального захисту. Серед них можуть бути рукавички, захисні окуляри, каски тощо.

4.3. Техніка безпеки при роботі з комп'ютерним обладнанням

- Для працівників, які працюють з комп'ютерами, необхідно забезпечити:
- Зручні робочі місця з ергономічними стільцями та столами. Це допоможе уникнути проблем зі спиною та зменшити стомлюваність.
- Відповідне освітлення робочого місця, яке запобігає перевтомі очей. Оптимальне освітлення повинно бути м'яким та рівномірним.
- Дотримання правил користування комп'ютерною технікою, включаючи регулярні перерви для відпочинку. Рекомендується кожні 1-2 години робити короткі перерви.

- Захист від електромагнітного випромінювання, яке може негативно впливати на здоров'я. Це може включати використання екранів-фільтрів та відповідних налаштувань моніторів.

4.4. Пожежна безпека

З метою забезпечення пожежної безпеки необхідно:

- Оснащення приміщень первинними засобами пожежогасіння (вогнегасниками, пожежними кранами). Вогнегасники повинні бути доступними в кожному приміщенні.
- Забезпечення наявності вільних шляхів евакуації та їх правильне маркування. Важливо, щоб всі працівники знали розташування евакуаційних виходів.
- Проведення регулярних навчань з евакуації у разі пожежі. Це дозволить працівникам діяти швидко та впевнено в надзвичайних ситуаціях.
- Установка пожежної сигналізації та систем автоматичного пожежогасіння. Це значно підвищить рівень безпеки та дозволить швидко реагувати на загрозу.

4.5. Електробезпека

Забезпечення електробезпеки включає:

- Використання електрообладнання відповідно до вимог нормативних документів. Всі електроприлади повинні відповідати стандартам безпеки.
- Регулярну перевірку та технічне обслуговування електричних мереж та пристроїв. Це допоможе уникнути коротких замикань та інших електричних аварій.
- Забезпечення працівників інструкціями з безпечного користування електроприладами. Інструкції повинні бути чіткими та доступними для розуміння.
- Встановлення захисного заземлення та пристроїв захисного вимкнення. Це забезпечить додатковий рівень захисту від ураження електричним струмом.
-

4.6. Психофізіологічний комфорт працівників

- Для підтримання психофізіологічного комфорту працівників необхідно:
- Організація перерв для відпочинку та розслаблення. Це може включати кімнати для відпочинку з комфортними умовами.
- Створення сприятливого психологічного клімату в колективі. Важливо підтримувати позитивну атмосферу та взаємоповагу між працівниками.
- Забезпечення можливості фізичних вправ протягом робочого дня. Працівникам слід надавати можливість займатися фізкультурою або робити прості вправи для розслаблення.

4.7. Впровадження заходів для зменшення стресу

- Зменшення стресу є важливою складовою охорони праці:
- Організація тренінгів з управління стресом. Це допоможе працівникам навчитися справлятися зі стресовими ситуаціями.
- Проведення командоутворюючих заходів. Це сприяє зміцненню колективу та покращенню взаємодії між працівниками.
- Надання психологічної підтримки. Працівники повинні мати доступ до консультацій психологів за потреби.

ВИСНОВКИ

В даній роботі було проведено аналіз та розробку веб-сайту для інтернет-магазину молочної продукції ТОВ "ФАВОР". Основною метою проекту було створення зручної та ефективної платформи для продажу продукції компанії в інтернеті, що дозволить розширити клієнтську базу та збільшити обсяги продажів.

Основні результатами роботи

Проведено детальний аналіз ринку молочної продукції та основних конкурентів, що дозволило визначити основні тенденції та вимоги до функціоналу майбутнього веб-сайту.

Виявлено, що більшість конкурентів вже активно використовують інтернет-торгівлю як канал збуту, що підкреслює необхідність розробки власного онлайн-магазину.

Розроблено детальне технічне завдання, яке включає опис необхідних функцій та вимог до веб-сайту.

Створено структуру сайту, що включає головну сторінку, каталог продукції, сторінку реєстрації та входу користувачів, кошик покупок та сторінку оформлення замовлення.

Особлива увага приділена зручності користування сайтом та його ергономічності.

Розроблено сучасний та привабливий дизайн, який відповідає корпоративному стилю компанії та забезпечує зручну навігацію для користувачів.

Дизайн враховує принципи адаптивності, що дозволяє комфортно використовувати сайт на різних пристроях.

Використано сучасні технології та інструменти для розробки веб-сайту, такі як HTML, CSS, JavaScript, а також фреймворк Flask для бекенд-розробки.

Забезпечено інтеграцію з базою даних для зберігання інформації про продукти, користувачів та замовлення.

Реалізовано основні функції, включаючи реєстрацію та авторизацію користувачів, перегляд каталогу продукції, додавання товарів до кошика та оформлення замовлення.

Проведено комплексне тестування веб-сайту для виявлення та виправлення можливих помилок та недоліків.

На основі проведеного аналізу можна зробити висновок, що впровадження веб-сайту для інтернет-магазину є стратегічно важливим кроком для подальшого розвитку ТОВ "ФАВОР". Успішна реалізація проекту забезпечить зростання обсягів продажів, покращення обслуговування клієнтів та зміцнення позицій компанії на ринку молочної продукції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 8 етапів створення інтернет-магазину Wezom. [Електронний ресурс] – Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/etapy-razrabotki-internet-magazina>.
2. Bootstrap. Bootstrap Най популярніша бібліотека HTML, CSS та JS. [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>.
3. monday.com CRM Solution. monday.com. [Електронний ресурс] – Режим доступу до ресурсу: <https://monday.com>
4. Welcome to Flask – Flask Documentation (3.0.x). Welcome to Flask – Flask Documentation (3.0.x). [Електронний ресурс] – Режим доступу до ресурсу: <https://flask.palletsprojects.com/>
5. Управління ІТ проектами [Електронний ресурс] : методичні рекомендації до самостійної роботи для здобувачів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійних програм «Комп'ютерні науки» та «Інформаційні системи та штучний інтелект» денної та заочної форм навчання / укладачі : С. В. Грибков, О. Л. Сєдих ; Національний університет харчових технологій. – Київ : НУХТ, 2022 – 25 с.– № 51.64.
6. Управління ІТ проектами [Електронний ресурс]: лабораторний практикум для студентів напряму підготовки 6.050101 "Комп'ютерні науки" денної та заочної форм навч. / уклад. О. А. Хлобистова, М. В. Гладка. - К. : НУХТ, 2013. – 108 с. URL: <http://library.nuft.edu.ua/ebook/file/51.07A.pdf>.
7. Управління ІТ проектами [Електронний ресурс]: методичні рекомендації до виконання курсової роботи для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навч. / уклад. М. В. Гладка, О. А. Хлобистова. – К. : НУХТ, 2014.– 91 с. URL: <http://library.nuft.edu.ua/ebook/file/51.13.pdf>.

8. Проектування інформаційних систем [Електронний ресурс]: конспект лекцій для студентів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання. Уклад.: О. М. М'якшило, О. В. Харкянен: НУХТ, 2018. – 47 с.
9. М'якшило О.М. CASE-технології у проектуванні інформаційних систем: електронний навчальний посібник для студентів вищих навчальних закладів / О.М. М'якшило, Л.Г. Загоровська, – К.: НУХТ, 2017. – 190 с.
10. М'якшило О.М. Моделювання баз даних засобами CASE – технології ERWin: конспект лекцій/ О.М. М'якшило – К.:НУХТ, 2007 – 60 с.
11. Проектування інформаційних систем. [Електронний ресурс]: лабораторний практикум для студ. освітнього ступеню "бакалавр" спец. 122 "Комп'ютерні науки" денної і заочної форм навчання. Частина 1 / Уклад.: О.М. М'якшило, О.В. Харкянен – К.: НУХТ, 2017 – 33 с.
12. Проектування інформаційних систем. [Електронний ресурс]: лабораторний практикум для студ. освітнього ступеню "бакалавр" спец. 122 "Комп'ютерні науки" денної і заочної форм навчання. Частина 2 "Проектування клієнтського додатку" / Уклад.: О.М. М'якшило, О.В. Харкянен – К.: НУХТ, 2017 – 33 с.
13. Проектування інформаційних систем [Електронний ресурс]: методичні рекомендації до виконання курсового проекту для студентів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання./Уклад.: О. М. М'якшило, О. В. Харкянен: НУХТ, 2018. – 24 с.

ДОДАТКИ

Додаток А «Схема фізичної моделі»

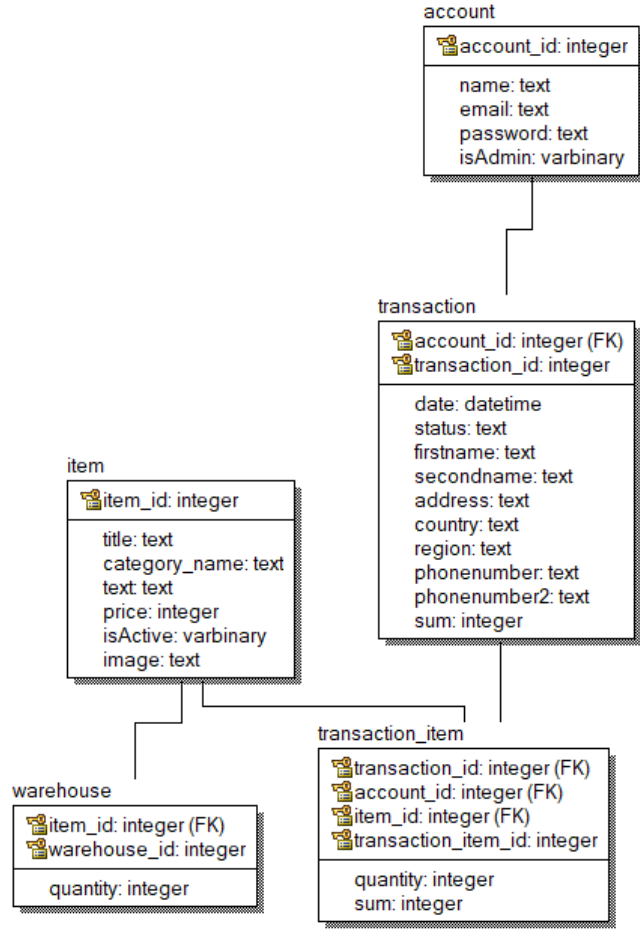


Рисунок А.1 – Фізична модель

Додаток Б «Програмний код»

Додаток Б.1, програмний код відповідаю за рахування кількості товарів у кошику.

```
@app.route('/get_cart_count')
def get_cart_count():
    if 'cart_items' in session:
        session['cart_count'] = len(session['cart_items'])
        return jsonify({'cartCount': session['cart_count']})
    else:
        session['cart_count'] = 0
        return jsonify({'cartCount': session['cart_count']})

@app.route('/product/<int:product_id>')
def product(product_id):
    item = Item.query.get_or_404(product_id)
    return render_template('product.html', item=item)
```

Додаток Б.2, програмний код відповідаю за відображення сторінки «Про Нас».

```
@app.route('/about')
def about():
    return render_template('about.html')

table.py
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy.orm import relationship

db = SQLAlchemy()
```

Додаток Б.3, програмний код, що відповідає за структуру таблиці продуктів.

```
class Item(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(50), nullable=False)
```

```

category_item = db.Column(db.String(50), nullable=False)
text = db.Column(db.Text, nullable=False)
price = db.Column(db.Integer, nullable=False)
isActive = db.Column(db.Boolean, default=True)
image = db.Column(db.String(255), nullable=False)

```

```

def __repr__(self):
    return f"{self.title}, {self.category_item}, {self.text}, {self.price}, {self.image}"

```

Додаток Б.4, програмний код, що відповідає за структуру таблиці акаунтів.

```

class Account(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(32), nullable=False)
    email = db.Column(db.String(32), nullable=False)
    password = db.Column(db.String(32), nullable=False)
    isAdmin = db.Column(db.Boolean, default=False)
    transactions = relationship('Transaction', backref='account', lazy=True)

```

Додаток Б.5, програмний код, що відповідає за структуру таблиці транзакцій.

```

class Transaction(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    account_id = db.Column(db.Integer, db.ForeignKey('account.id'), nullable=False)
    date = db.Column(db.DateTime, nullable=False)
    status = db.Column(db.String(20), nullable=False)
    firstname = db.Column(db.String(32), nullable=False)
    secondname = db.Column(db.String(32), nullable=False)
    address = db.Column(db.String(50), nullable=False)
    country = db.Column(db.String(32), nullable=False)
    region = db.Column(db.String(32), nullable=False)
    phonenummer = db.Column(db.Text, nullable=False)
    phonenummer2 = db.Column(db.Text, nullable=False)
    sum = db.Column(db.Integer, nullable=False)

    items = relationship('TransactionItem', backref='transaction', lazy=True)

```

Додаток Б.6, програмний код, що відповідає за структуру таблиці кількості продуктів у транзакції.

```
class TransactionItem(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    transaction_id = db.Column(db.Integer, db.ForeignKey('transaction.id'), nullable=False)
    item_id = db.Column(db.Integer, db.ForeignKey('item.id'), nullable=False)
    quantity = db.Column(db.Integer, nullable=False)
```

Додаток Б.7, програмний код, що відповідає за структуру таблиці складу.

```
class Warehouse(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    item_id = db.Column(db.Integer, db.ForeignKey('item.id'), nullable=False)
    quantity = db.Column(db.Integer, nullable=False)
    item = db.relationship('Item', backref=db.backref('warehouses', lazy=True))
```

Додаток Б.8, програмний код, що відповідає за структуру таблиці промо-кодів.

```
class PromoCode(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    promo = db.Column(db.Text, nullable=False)
    discount_perc = db.Column(db.Integer, nullable=False)
```

Додаток Б.8, програмний код, що відповідає за структуру таблиці підписки.

```
class Mailing(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.Text, nullable=False)
    isIncluded = db.Column(db.Boolean, default=True)
    date_sub = db.Column(db.DateTime, nullable=False)
```