

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій  
(повне найменування факультету)

Кафедра програмних засобів  
(повне найменування кафедри)

**Пояснювальна записка**

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОБІЛЬНИХ ПРИСТРОЇВ  
SOFTWARE DEVELOPMENT FOR AN ONLINE STORE  
SELLING MOBILE DEVICES

Виконав: студент 4 курсу, групи КНТ-220  
Спеціальності 122 Комп'ютерні науки  
(код і найменування спеціальності)

Освітня програма (спеціалізація)  
Комп'ютерні науки

ЧЕРНІГОВЦЕВ М.О.  
(ПРІЗВИЩЕ та ініціали)

Керівник ПАРХОМЕНКО А.В.  
(ПРІЗВИЩЕ та ініціали)

Рецензент ЗЕЛЕНЬОВА І.Я.  
(ПРІЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»  
(повне найменування закладу вищої освіти)

Факультет комп'ютерних наук та технологій  
Кафедра програмних засобів  
Ступінь вищої освіти бакалавр  
Спеціальність 122 Комп'ютерні науки  
(код і найменування)  
Освітня програма (спеціалізація) Комп'ютерні науки  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
Сергій СУББОТІН  
“ ” 2024 року

**З А В Д А Н Н Я**  
**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА**

ЧЕРНІГОВЦЕВА Михайла Олександровича  
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення інтернет-магазину з продажу мобільних пристроїв. Software Development for an Online Store Selling Mobile Devices

керівник проєкту (роботи) к.т.н., доцент, ПАРХОМЕНКО Анжеліка Володимирівна,  
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року № 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програмного забезпечення. 3. Конструювання програмного забезпечення. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | ПРІЗВИЩЕ, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-4 Основна частина | ПАРХОМЕНКО А.В., доцент                   |                |                           |
| Нормоконтроль       | БЄЛОВА А.В., асистент                     |                |                           |

7. Дата видачі завдання «04» квітня 2024 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                             | Строк виконання етапів проєкту (роботи) | Примітка     |
|-------|----------------------------------------------------------------------|-----------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                          | 1 тиждень                               | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                           | 1 тиждень                               | Розділ 1     |
| 3     | Вибір мови програмування та інших технологій розробки.               | 2 тиждень                               | Розділ 2     |
| 4     | Розробка архітектури програмного забезпечення.                       | 2 тиждень                               | Розділ 2     |
| 5     | Розробка серверної та клієнтської частини програмного забезпечення   | 3-4 тижні                               | Розділ 3     |
| 6     | Тестування та експериментальне дослідження програмного забезпечення. | 5 тиждень                               | Розділ 4     |
| 7     | Оформлення пояснювальної записки та документів до неї.               | 6 тиждень                               | Додатки      |
| 8     | Нормоконтроль та рецензування.                                       | 7 тиждень                               |              |
| 9     | Захист роботи.                                                       | 8 тиждень                               |              |

Студент(ка)

\_\_\_\_\_  
( підпис )      Михайло ЧЕРНІГОВЦЕВ  
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_  
( підпис )      Анжеліка ПАРХОМЕНКО  
(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 104 с., 19 табл., 43 рис., 2 дод., 46 джерел.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ІНТЕРНЕТ-МАГАЗИН,  
ВЕБЗАСТОСУНОК, МОБІЛЬНИЙ ПРИСТРІЙ, ФРЕЙМВОРК DJANGO,  
МОВА ПРОГРАМУВАННЯ PYTHON.

Об'єкт дослідження – процес обліку продажу мобільних пристроїв в інтернет-магазині.

Предмет дослідження – програмні засоби для реалізації інтернет-магазину з продажу мобільних пристроїв.

Мета роботи – розробити прототип інтернет-магазину для обліку продажу мобільних пристроїв.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, технологія розробки вебзастосунків, мова програмування Python, фреймворк Django, шаблон проектування Model View Template, система керування базами даних MySQL.

Результати. Створено кросплатформне програмне забезпечення інтернет-магазину з продажу мобільних пристроїв.

Висновки. Розроблене програмне забезпечення дозволить підвищити ефективність введення обліку продажів мобільних пристроїв в інтернет-магазині.

Галузь використання – електронна комерція.

## **ABSTRACT**

Explanatory note to the diploma qualifying work of the bachelor: 104 pages, 19 tables, 43 figures, 2 appendixes, 46 sources.

SOFTWARE, ONLINE STORE, WEB APPLICATION, MOBILE DEVICE, DJANGO FRAMEWORK, PYTHON PROGRAMMING LANGUAGE.

Object of the study is the process of accounting for sales of mobile devices in an online store.

Subject of the study is software tools for implementing an online store selling of mobile devices.

The purpose of the work is to develop a prototype of an online store for accounting the sales of mobile devices.

Materials, methods, and tools: structural and object-oriented programming, web application development technology, Python programming language, Django framework, Model View Template design template, MySQL database management system.

Results. Cross-platform software for an online store selling mobile devices has been created.

Conclusions. The developed software will make it possible to increase the efficiency of accounting the sales of mobile devices in the online store.

The field of application is e-commerce.

## ЗМІСТ

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
|                                                                                   | С. |
| Перелік скорочень та умовних познач.....                                          | 8  |
| Вступ.....                                                                        | 9  |
| 1 Аналіз предметної області.....                                                  | 10 |
| 1.1 Дослідження предметної області електронної комерції в Україні .....           | 10 |
| 1.2 Аналіз існуючих програмних рішень.....                                        | 13 |
| 1.3 Постановка задач дипломної роботи .....                                       | 20 |
| 2 Проєктування програмного забезпечення .....                                     | 21 |
| 2.1 Аналіз вимог до програмного забезпечення .....                                | 21 |
| 2.2 Вибір та обґрунтування засобів розробки інтернет-магазину.....                | 23 |
| 2.3 Обґрунтування вибору архітектурного шаблону .....                             | 27 |
| 2.4 Обґрунтування вибору програмних засобів.....                                  | 34 |
| 2.4.1 Вибір мови програмування .....                                              | 34 |
| 2.4.2 Вибір середовища розробки.....                                              | 38 |
| 2.4.3 Вибір системи керування базами даних .....                                  | 41 |
| 2.4.4 Огляд використаних фреймворків.....                                         | 45 |
| 2.5 Засоби проєктування макету користувацького інтерфейсу<br>вебзастосунків ..... | 48 |
| 2.6 Висновок до другого розділу .....                                             | 51 |
| 3 Конструювання програмного забезпечення .....                                    | 52 |
| 3.1 Розробка бази даних.....                                                      | 52 |
| 3.2 Розробка класів моделей та представлень .....                                 | 56 |
| 3.3 Схема алгоритму функціонування системи .....                                  | 62 |
| 3.4 Розробка інтерфейсу .....                                                     | 64 |
| 3.5 Висновок до третього розділу.....                                             | 67 |
| 4 Експлуатація, тестування та експериментальне дослідження програми .....         | 68 |
| 4.1 Призначення й умови застосування програми.....                                | 68 |
| 4.2 Вимоги до інтерфейсу.....                                                     | 69 |

|                                                |    |
|------------------------------------------------|----|
| 4.3 Умови виконання програми .....             | 70 |
| 4.4 Розгортання програмного забезпечення ..... | 70 |
| 4.5 Методика та результати тестування.....     | 74 |
| 4.6 Висновок до четвертого розділу.....        | 81 |
| Висновки .....                                 | 83 |
| Перелік джерел посилань .....                  | 84 |
| Додаток А Текст програми.....                  | 88 |
| Додаток Б Слайди презентації .....             | 96 |

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

|          |                                   |
|----------|-----------------------------------|
| ADI      | - Artificial Design Intelligence; |
| Adobe XD | - Adobe Experience Design;        |
| CMS      | - Content Management System;      |
| MVC      | - Model View Controller;          |
| MVT      | - Model View Template;            |
| MVVM     | - Model View View Model;          |
| UML      | - Unified Modelling Language;     |
| VS Code  | - Visual Studio Code;             |
| СКБД     | - системи керування базами даних; |
| TIC      | - торгові інтернет-системи.       |

## ВСТУП

Актуальність розробки. На сьогоднішній день системи інтернет-магазини є невід'ємною складовою електронної комерції. Програмне забезпечення для електронної комерції дозволяє бізнесу здійснювати торгівлю товарами і послугами через мережу Інтернет. Можна навести наступні основні можливості електронної комерції: широкий асортимент товарів та послуг спричиняють зростання онлайн-покупок, а як наслідок збільшення прибутку компаній та зниження витрат на оренду приміщень, мобільний доступ завдяки поширенню смартфонів, здійснення продажу товарів та послуг на міжнародному рівні, персоналізовані пропозиції та послуги, що надаються за допомогою технологій аналізу даних і штучного інтелекту (рекомендаційні системи), впровадження нових технологій, таких як блок чейн для підтримки конфіденційності користувачів (підтримка анонімних транзакцій) тощо. Тому розробка програмного рішення з продажу мобільних пристроїв є актуальною задачею.

Для розробки інтернет-магазину з продажу мобільних пристроїв на сьогоднішній день використовують системи керування вмістом або спеціалізовані фреймворки (каркасні рішення) для розробки вебзастосунків. Використання сучасних інструментів забезпечує можливості керування товарами, обробки платежів та інтеграції з різними сервісами з метою прискорення процесу розробки, а також дозволяють створити індивідуальні рішення, що відповідають вимогам замовників інтернет-магазинів.

Мета даної дипломної кваліфікаційної роботи – розробити прототип інтернет-магазину для обліку продажу мобільних пристроїв.

## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

### **1.1 Дослідження предметної області електронної комерції в Україні**

На сьогоднішній день інтернет-магазини є дуже популярними засобами купівлі-продажу товарів [1]. Покупці та продавці вже оцінили зручність купівлі-продажу товарів через Інтернет, а саме: гнучкість системи, можливість здійснювати покупки та продаж на далеких відстанях, широкий асортимент товарів, вигідні ціни та ін. [2]-[4]. На даний момент не існує єдиного визначення поняття «електронна комерція» [2].

Електронна комерція – це використання електронних комунікацій та технологій обробки цифрової інформації для встановлення та змін відносин створення вартості між організаціями та між організаціями та індивідами [2].

Щодо поняття «електронної торгівлі» існує чимало визначень. За визначенням Світової Організації Торгівлі, електронна торгівля – це організація процесу товарно-грошового обміну у формі купівлі-продажу на базі електронних технологій. Комісія ООН з міжнародного торгового права (UNCITRAL) визначає електронну торгівлю як організацію і технологію купівлі продажу товарів, послуг електронним способом з використанням телекомунікаційних мереж та електронних фінансово-економічних інструментів [2].

Хоча більшість авторів не зосереджуються на відмінностях між поняттями «електронна комерція» та «електронна торгівля», ці терміни не є ідентичними. Електронна торгівля є вузьким поняттям і є складовою частиною електронної комерції. Відмінність між ними полягає в тому, що електронна торгівля охоплює лише процес купівлі-продажу, тоді як електронна комерція включає весь спектр товарно-грошових відносин [2].

Згідно Закону України «Про електронну комерцію», електронна комерція – це відносини, спрямовані на отримання прибутку, що виникають під час вчинення правочинів щодо набуття, зміни або припинення цивільних

прав та обов'язків, здійснені дистанційно з використанням інформаційно-телекомунікаційних систем, внаслідок чого в учасників таких відносин виникають права та обов'язки майнового характеру [2].

Отже, проаналізувавши наведені визначення, можна зробити висновок, що електронна комерція охоплює будь-які бізнес-процеси, які здійснюються за допомогою інформаційних технологій та Інтернету. Основні складові електронної комерції включають електронний обмін інформацією, електронний рух капіталу, електронну торгівлю, електронні гроші, електронний маркетинг, електронний банкінг, електронні страхові послуги тощо [2].

Електронна комерція функціонує на трьох основних рівнях [1]-[4]:

- перший рівень – Інтернет-комерція – включає найпростіші форми взаємодії між учасниками ринку, з використанням традиційних та загальнодоступних баз для обміну інформацією. Цей рівень електронної комерції є найбільш розвиненим та поширеним в Україні;

- другий рівень – електронна комерція – перехід від традиційних методів торгівлі до електронної торгівлі є закономірним процесом, який призведе до ліквідації торговельних посередників. Вони трансформуються в менеджерів, які обслуговують та організовують процеси електронної комерції;

- третій рівень – електронний бізнес – розробка та впровадження нових форм взаємодії між учасниками, нових способів обміну інформацією, а також методів обробки та інтерпретації отриманої інформації.

Одним з програмних рішень введення Інтернет-комерції є інтернет - магазин.

Інтернет-магазини можна класифікувати за наступними критеріями [5]:

- за методом роздрібного продажу товарів у мережі: інтернет-магазини; вебвітрини, торгові системи; торгові ряди; контентні проєкти (споживацькі енциклопедії, системи інтернет -замовлень товарів тощо);

- за бізнес-моделлю, що включає повністю онлайновий магазин та суміщення офлайнового та онлайнового бізнесу(коли інтернет-магазин створюється на основі вже діючої реальної торгової структури);
- за взаємовідносинами з постачальниками: від магазинів з власним складом до тих, що працюють за договорами з постачальниками;
- за ступенем автоматизації серед торгових систем, які включають вебвітрини, інтернет-магазини та торгові інтернет-системи (ТІС).

Вебвітрина - це інтерфейс, який включає товарний каталог, систему навігації та процес оформлення замовлення, що передається менеджеру для обробки. Менеджер забезпечує зв'язок зі складом, організовує доставку товарів покупцеві та відповідає за процес оплати. Одночасно проводиться рекламна діяльність, аналіз попиту та аналітичні роботи [5].

Робота вебвітрини подібна до поштової торгівлі, проте замовлення товарів та ознайомлення з ними здійснюються через мережу Інтернет. У цьому контексті вебвітрина виступає як засіб залучення покупців, інтерфейс для їх взаємодії та засіб проведення маркетингових заходів [5].

Інтернет-магазин відрізняється відмінною характеристикою - повною автоматизацією системи обробки замовлень, що дозволяє надавати індивідуальний підхід до кожного зареєстрованого клієнта [5].

Як загальна особливість між інтернет-магазином і ТІС є їх здатність забезпечувати повний цикл торгівлі у режимі онлайн-підключення. Однак ТІС має додаткову функціональність, оскільки інтегрується в систему внутрішнього документообігу компанії. Системи доставки товарів залишаються неавтоматизованими для інтернет-магазину та ТІС [5].

Приблизна структура основних форм організації вітчизняних інтернет-магазинів представлена на рис. 1.1 [5].



Рисунок 1.1 – Структура основних форм організації вітчизняних інтернет - магазинів [5]

## 1.2 Аналіз існуючих програмних рішень

### 1.2.1 Інтернет-магазин «Yabluka»

Інтернет-магазин «Yabluka» [6] представляє собою вебсервіс з продажу переважно техніки Apple та надає можливості покупки нової техніки та вживаних мобільних пристроїв, а також послуги прокату або обміну старих мобільних пристроїв на нові з доплатою. Зовнішній вигляд вебзастосунку (рис.1.2) має чорно-жовто-білу основну кольорову схему, два режими дений та нічний, у верхній частини розташовані основні вкладки, каталог товару та можливість пошуку по сайту, можливість реєстрації, порівняння моделей мобільних телефонів та кошик, далі йде карусель реклами та акцій, футер, або нижня частина містить повторення каталогу товарів, популярні товари, інформацію про кампанію та можливі вакансії, сервіси, які надаються компанією, контакти call-центру, поштову скриньку та іншу необхідну контактну інформацію.

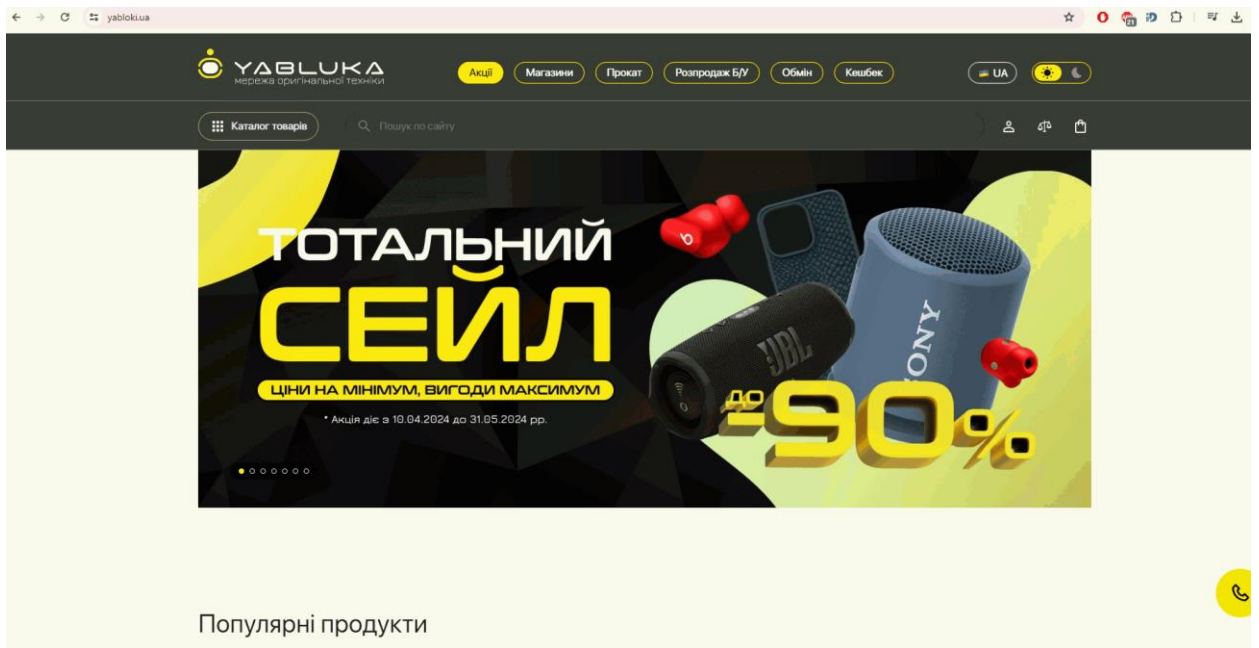


Рисунок 1.2 – Зовнішній вигляд інтернет-магазину «Yabluka»[6]

Сторінка товару представляє собою повноцінний landing page, односторінкову вітрину-представлення інформації про мобільний телефон вибраної моделі користувачем (рис.1.3).

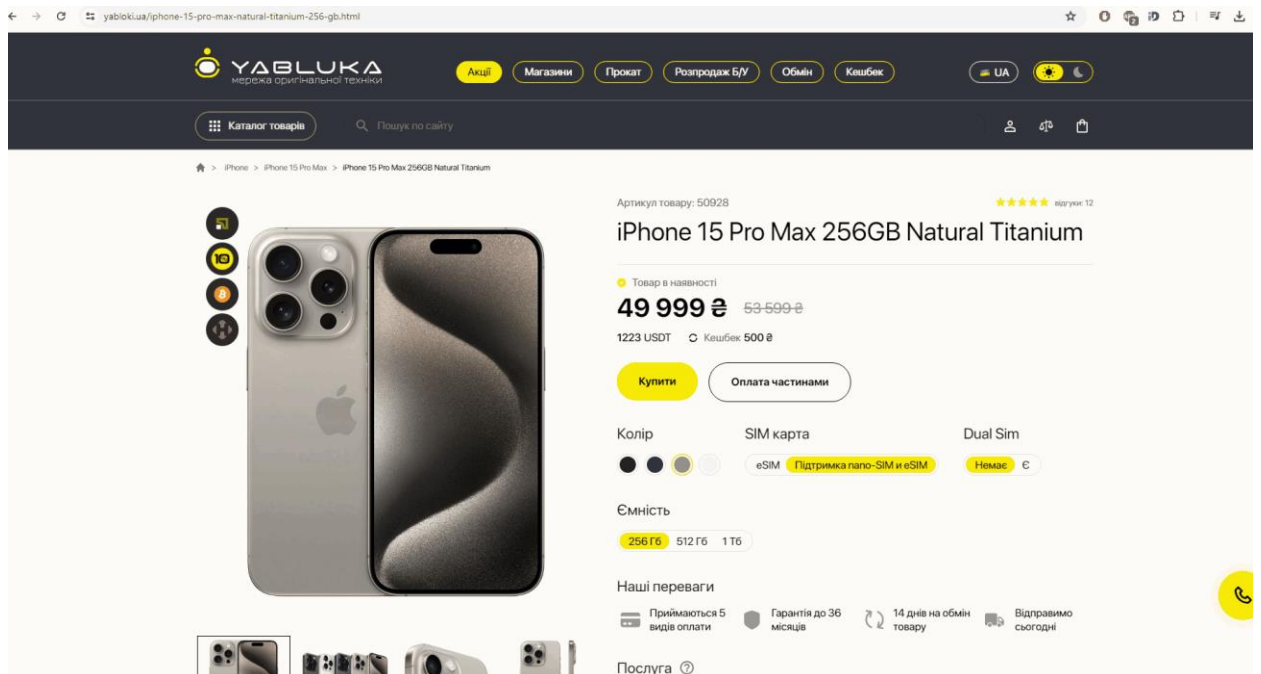


Рисунок 1.3 – Сторінка товару інтернет-магазину «Yabluka» [6]

Функціонал інтернет-магазину «Yabluka» розроблений за допомогою стеку технологій на основі фреймворку Symfony2.

### 1.2.2 Інтернет-магазин «Ябло»

Розглянемо ще одне аналогічне програмне рішення інтернет-магазину з продажу мобільних пристроїв – вебзастосунок «Ябло» [7], де представлена інформація про оригінальна техніку не тільки Apple, але і техніки Dyson, Sony, GoPro та інших брендів. Головна сторінка інтернет-магазину «Ябло» наведена на рисунку 1.4.

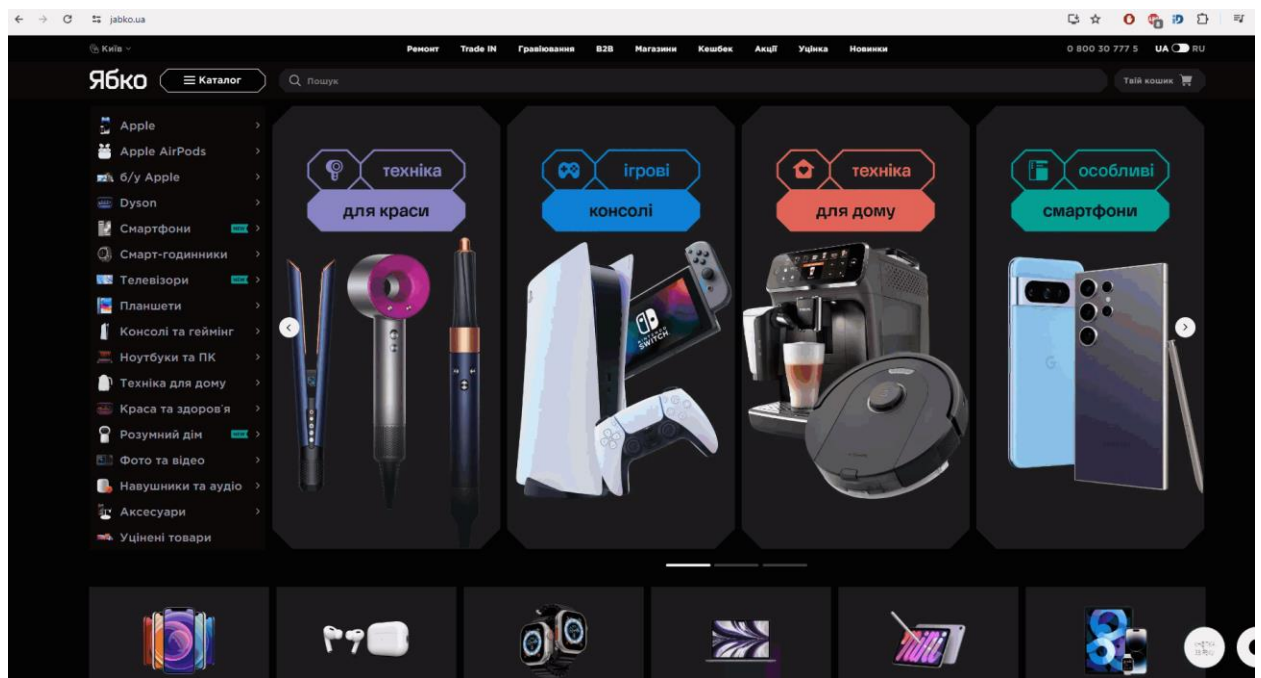


Рисунок 1.4 – Головна сторінка інтернет-магазину «Ябло» [7]

При виборі будь-якої категорії з правого боку відкривається фільтр з величезним переліком варіантів параметрів, частина з яких заточені під пошукову оптимізацію (рис.1.5.).

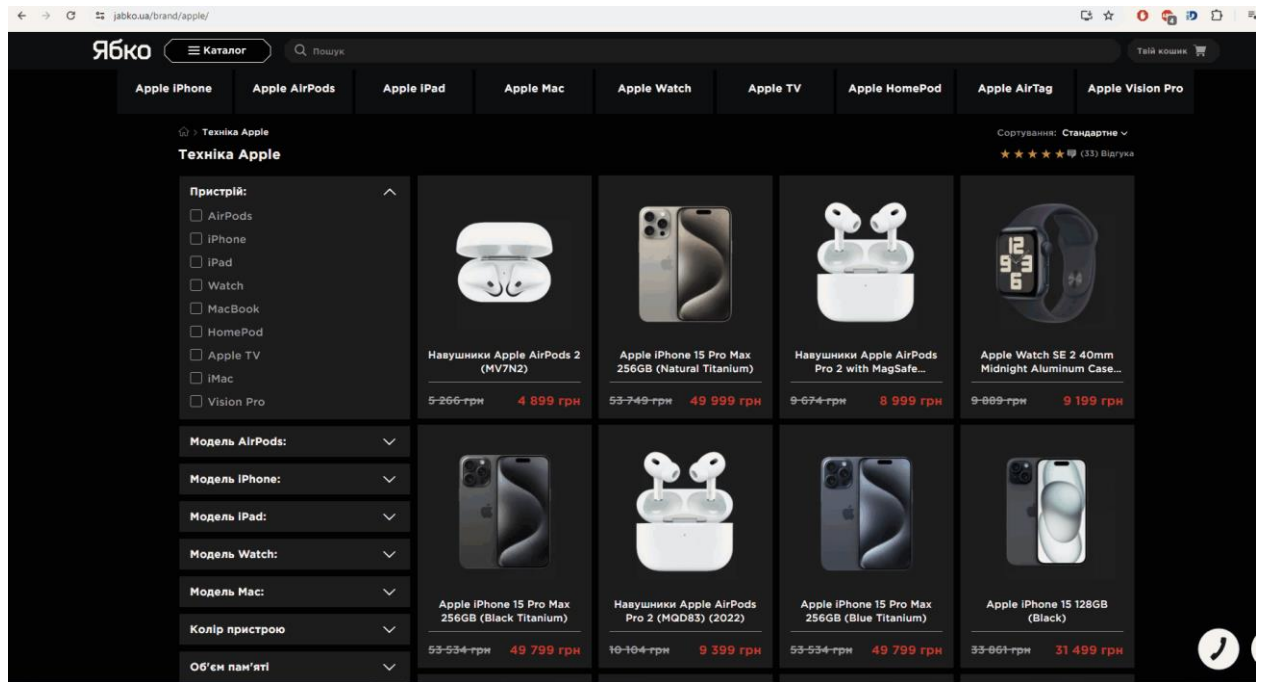


Рисунок 1.5 – Сторінка фільтрації товарів [7]

Далі можна обрати певний мобільний пристрій, відсортувавши його за ціною, кольором, обсягом пам'яті, версією sim-картки (рис.1.6).

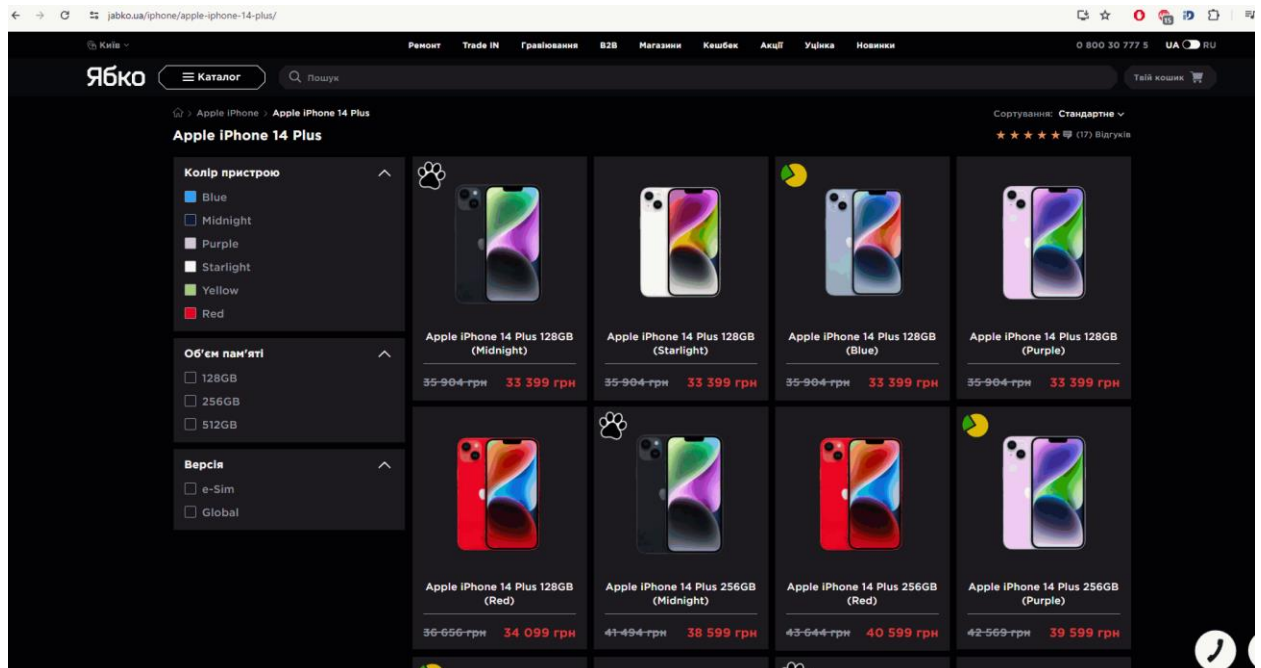


Рисунок 1.6 – Сторінка вибору телефону вибраної моделі [7]

Функціонал інтернет-магазину «Ябло» скоріше за все розроблений за допомогою стеку технологій LAMP – набору вільного програмного забезпечення з відкритим кодом, в який входять операційна система Linux, вебсервер Apache, систему керування базами даних MySQL, та інтерпретатор PHP або Python – основні компоненти для побудови серверної частини вебзастосунку.

### 1.2.3 Інтернет-магазин «Цитрус»

Вебсервіс «Цитрус» – український інтернет-магазин з продажу електронної та побутової техніки [8], містить в собі категорію ««Цитрус»» з продажу мобільних пристроїв, деякі популярні бренди винесені в окремі категорії «Samsung Galaxy» та «Apple» (рис.1.7).

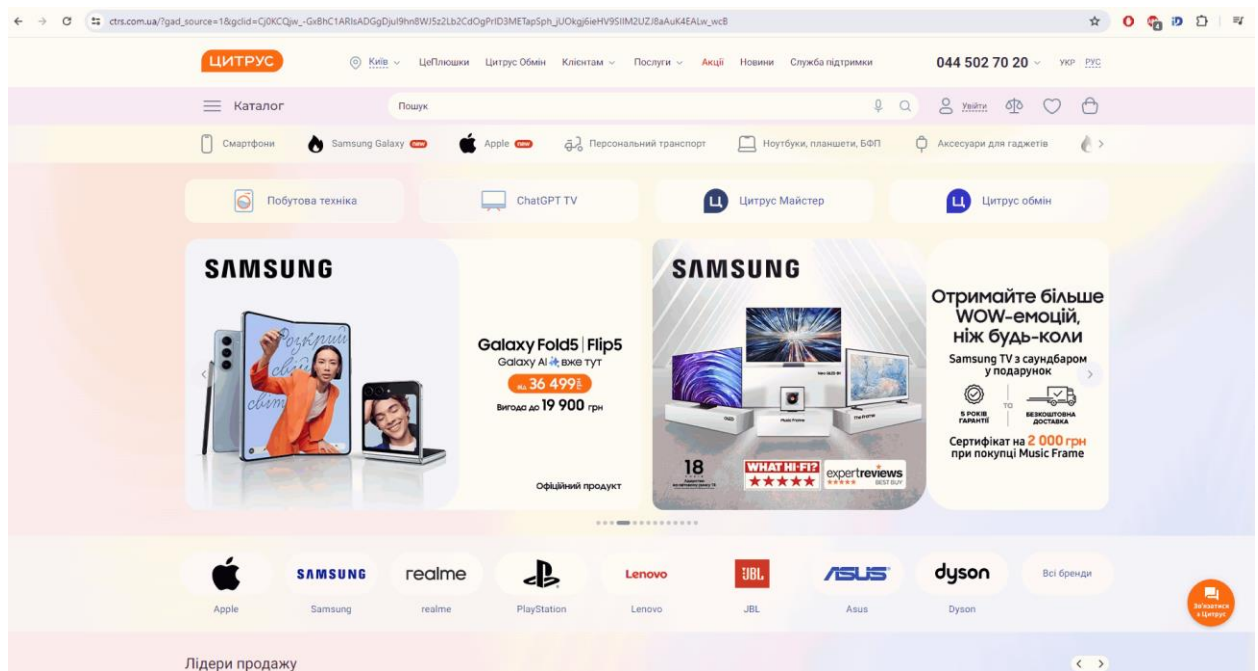


Рисунок 1.7 – Головна сторінка інтернет-магазину «Цитрус» [8]

При виборі категорії «Смартфони» можна побачити фільтр з величезним переліком варіантів параметрів, переважна частина з яких заточені під пошукову оптимізацію (рис.1.8). Далі можна обрати певний

мобільний пристрій, відсортувавши його за ціною, брендом, процесором, кольором, кількістю sim-карт та іншими параметрами.

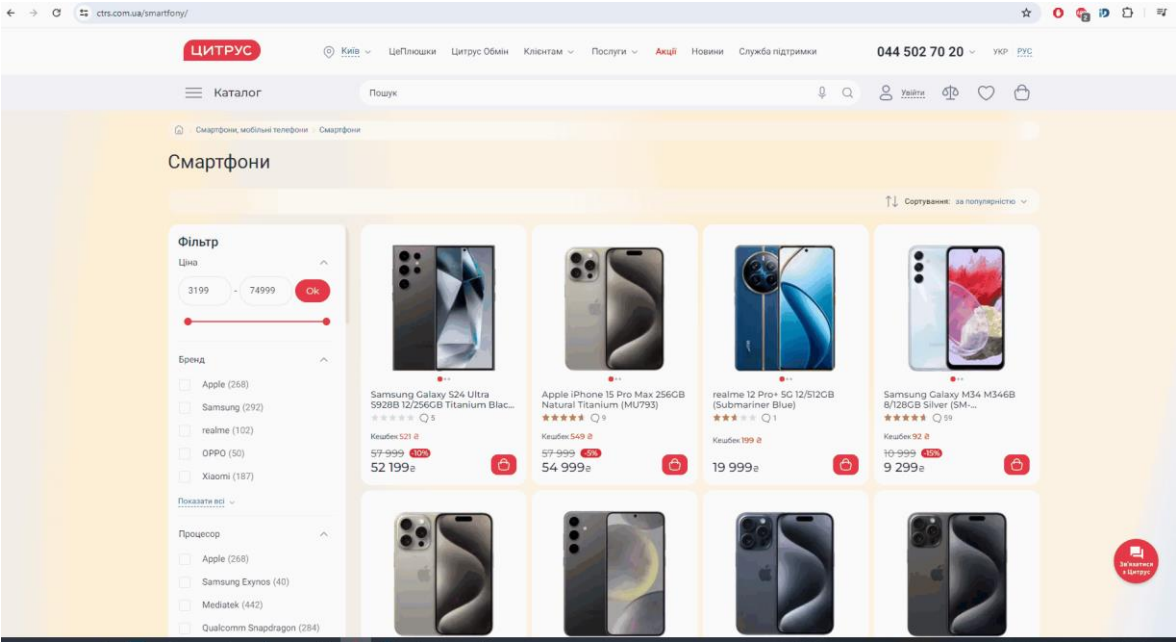


Рисунок 1.8 – Категорія «Смартфони» з фільтрацією [8]

При виборі певного телефону відображається сторінка товару, що представляє собою односторінкову вітрину-представлення інформації про мобільний телефон вибраної моделі користувачем (рис.1.9).

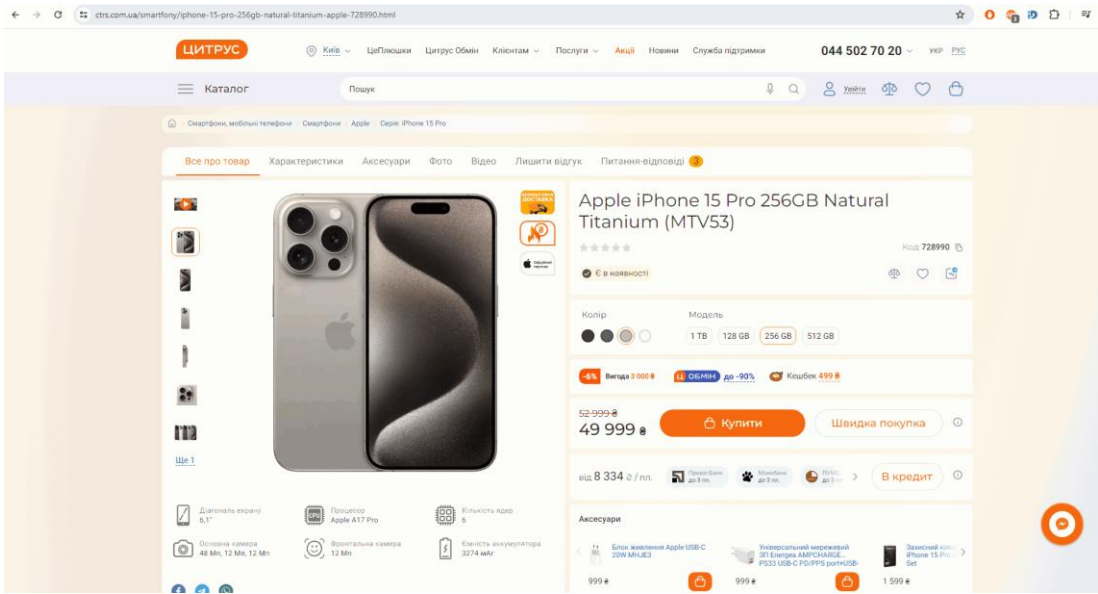


Рисунок 1.9 – Сторінка вибору телефону вибраної моделі [8]

Функціонал інтернет-магазину «Цитрус» розроблений за допомогою стеку технологій на основі стеку технологій LAMP (серверна частина) та фреймворку React.js (клієнтська частина).

Порівняльна характеристика існуючих аналогічних програмних рішень з продажу мобільних пристроїв наведена в табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика існуючих аналогічних програмних рішень [6] - [8]

| Параметр                                    | Інтернет-магазин<br>«Yabluka»                       | Інтернет-магазин<br>«Ябло»                          | Інтернет-магазин<br>«Цитрус»                        |
|---------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|
| Наявність каталогу                          | Так                                                 | Так                                                 | Так                                                 |
| Можливість реєстрації                       | Так                                                 | Так                                                 | Так                                                 |
| Переключення денний/нічний режим            | Так                                                 | Ні                                                  | Ні                                                  |
| Наявність кошику                            | Так                                                 | Так                                                 | Так                                                 |
| Наявність систем оплати                     | Visa або MasterCard                                 | Visa або MasterCard                                 | Visa або MasterCard                                 |
| Вид сторінки товару                         | односторінкова вітрина-представлення (landing page) | односторінкова вітрина-представлення (landing page) | односторінкова вітрина-представлення (landing page) |
| Функціонал вебзастосунку (серверна частина) | фреймворк Symfony2                                  | технологія LAMP                                     | технологія LAMP                                     |

### 1.3 Постановка задач дипломної роботи

В розділі було розглянуто поняття електронної комерції, виконано огляд існуючих аналогічних програмних рішень з продажу мобільних пристроїв, виділено перелік функціональних можливостей для неавторизованого та авторизованого користувачів, а також для менеджера (адміністратора) інтернет-магазину.

Мета даної дипломної кваліфікаційної роботи – розробити прототип інтернет-магазину для обліку продажу мобільних пристроїв.

Для досягнення поставленої мети в дипломній кваліфікаційній роботі потрібно вирішити наступні задачі:

- виконати вибір та обґрунтування засобів розробки інтернет-магазину;
- розглянути існуючі системи керування вмістом та фреймворки для розробки вебзастосунків;
- обрати відповідний архітектурний шаблон проєктування програмного забезпечення;
- обґрунтувати вибір системи керування базами даних та виконати проєктування схеми бази даних;
- розглянути засоби проєктування макету користувацького інтерфейсу вебзастосунків;
- спроектувати макет клієнтської частини за допомогою інструментів штучного інтелекту для проєктування користувацького інтерфейсу;
- розробити прототип інтернет-магазину з продажу мобільних пристроїв;
- виконати тестування розробленого програмного забезпечення.

## **2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Для розробки прототипу інтернет-магазину з продажу мобільних пристроїв, що буде містити серверну та клієнтську частини необхідно визначити функціональні можливості для користувачів, обрати інструментарій розробки, архітектурний шаблон програмного забезпечення, фреймворк для реалізації, інструментарій проєктування інтерфейсу користувача та систему керування базами даних.

### **2.1 Аналіз вимог до програмного забезпечення**

Розроблена система повинна мати наступні функціональні можливості для неавторизованого користувача:

- можливість реєстрації;
- перегляд галереї мобільних пристроїв;
- перегляд мобільних пристроїв за категоріями;
- пошук мобільного пристрою по ключовим словам.

Розроблене програмне забезпечення повинно надавати наступні функціональні можливості для авторизованого користувача:

- можливість авторизації;
- додавання мобільного пристрою до кошика;
- перегляд історії своїх замовлень;
- можливість вподобати та зберегти вибрані мобільні пристрої;
- редагування та оновлення особистої інформації;
- скидання паролю, використовуючи зареєстровану електронну пошту;
- оформлення та можливість оплати замовлення;
- перегляд галереї мобільних пристроїв;
- перегляд мобільних пристроїв за категоріями;
- пошук мобільного пристрою по ключовим словам.

Програмне забезпечення повинно мати наступні функціональні вимоги для менеджера(адміністратора):

- можливість введення інформації про нові категорії мобільних пристроїв;
- можливість додавання нових мобільних пристроїв та їх технічних характеристик та зображень;
- перегляд інформації про всі наявні мобільні пристрої;
- редагування та видалення інформації про мобільний пристрій;
- перегляд та редагування інформації про замовлення користувачів.

На основі наведених функціональних вимог було створені діаграми прецедентів за допомогою мови Unified Modelling Language (UML) – графічної мови для побудови діаграм, призначеної для визначення, візуалізації та документування моделей об'єктно-орієнтованих систем програмного забезпечення. UML розроблено для створення структури об'єктно-орієнтованого програмного забезпечення, включаючи інтернет-магазини [9].

Діаграми прецедентів представляють собою діаграми, що використовуються для візуалізації взаємодії між користувачами (акторами) та системою. Діаграма прецедентів відображає функціональні вимоги до системи та її поведінку з точки зору зовнішніх користувачів.

Діаграма прецедентів для неавторизованого користувача наведена на рис. 2.1.



Рисунок 2.1 – Діаграма прецедентів для неавторизованого користувача

Діаграма прецедентів для авторизованого користувача наведена на рис. 2.2.

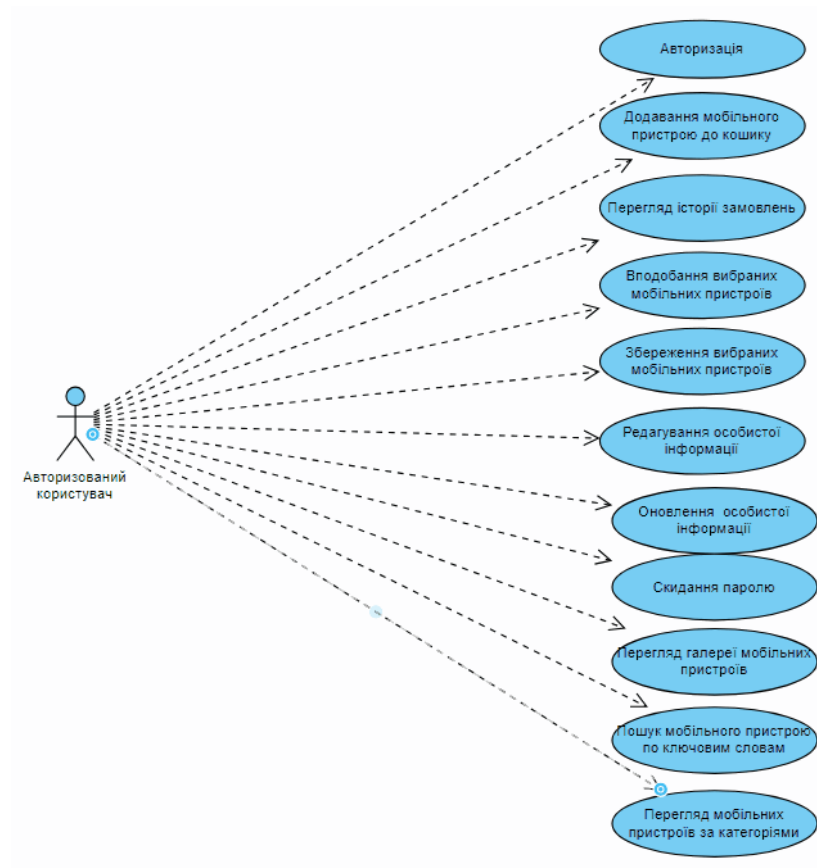


Рисунок 2.2 – Діаграма прецедентів для авторизованого користувача

Діаграма прецедентів для менеджера (адміністратора) наведена на рис. 2.3.

## 2.2 Вибір та обґрунтування засобів розробки інтернет-магазину

Для розробки інтернет-магазину з продажу мобільних пристроїв на сьогоднішній день використовують системи керування вмістом або спеціалізовані фреймворки(каркасні рішення) для розробки вебзастосунків.

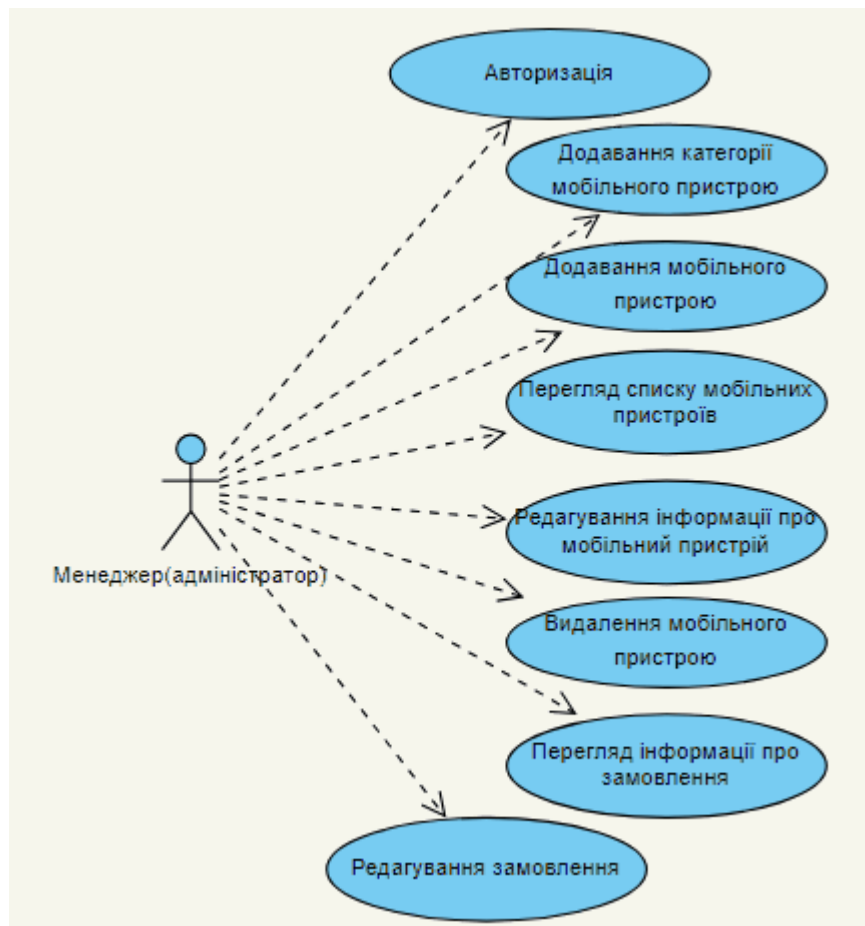


Рисунок 2.3 – Діаграма прецедентів для менеджера (адміністратора)

Розглянемо сучасні системи керування вмістом або Content Management System (CMS). На професійній мові CMS ще називають «рушій інтернет-магазину». CMS – це вебзастосунок, за допомогою якого можна створити та обслуговувати програмне забезпечення для інтернет-магазину. Основна перевага CMS – щоб створити інтернет-магазин не обов’язково знати програмування [10].

На сьогоднішній день на ринку CMS доступний широкий вибір систем для створення вебзастосунків, як платних, так і безкоштовних. Якщо вперше потрібно розробити вебзастосунок, вибір конкретної CMS серед всіх наявних платформ від різних розробників може бути складним.

Найбільш поширені CMS для розробки інтернет-магазину наступні: WordPress, Opencart та Magento.

Система керування контентом WordPress – є найбільш популярною платформою в сучасній веброзробці. Ця система має відкритий вихідний код і надає великі можливості розробникам для швидкого та простого керування вмістом вебсайтів [11]. WordPress застосовується у широкому спектрі сфер, від блогів до складних вебзастосунків. Вбудована система тем і плагінів, разом з гнучкою архітектурою, дозволяє створювати майже будь-які вебпроекти на основі WordPress. Ця платформа написана мовою програмування PHP і використовує базу даних MySQL. Початковий код WordPress поширюється за умовами ліцензії GNU General Public License [11].

Можна навести наступні переваги розробки вебзастосунку на WordPress [11]:

- можливість опублікувати статті за допомогою сторонніх програм та сервісів;
- швидка та проста процедура встановлення;
- забезпечення підтримки RSS, Atom, trackback, pingback;
- можливість підключення різноманітних додаткових плагінів;
- SEO-оптимізована система;
- широкий вибір тем для швидкої зміни зовнішнього вигляду вебсторінок та ін. [11].

Завдяки вищенаведеним функціональним можливостям WordPress –це гнучка система управління вмістом, яка підходить як для новачків, так і для досвідчених розробників, яку просто налаштувати під потрібні вимоги.

Система керування вмістом OpenCart не така популярна CMS як WordPress. Створити інтернет-магазин на OpenCart для розробника не складе особливих проблем завдяки наявності допоміжних інструментів та спеціалізованих модулів під розробку продуктів електронної комерції [12].

Основними перевагами OpenCart порівняно з WordPress є використання сучасної архітектури, що базується на шаблоні Model View Controller, підвищена швидкість роботи, функціональна адміністративна

панель управління контентом та економне використання серверних ресурсів [12].

OpenCart відома у комерційному сегменті як надійна та доступна система електронної торгівлі, яка підтримує роботу з усіма основними системами електронної оплати [12].

Система управління контентом OpenCart має ряд особливостей [13]:

- при низькому навантаженні на сервер забезпечується стабільна висока швидкість роботи;
- має власну систему формування звітів;
- легко навчити користувачів з різним рівнем програмування;
- на сторінках інтернет-магазину, побудованого на основі OpenCart, можна спостерігати зручну навігацію, перегляд історії покупок, статус замовлень, використання різних платіжних систем і багато іншого [13].

Можна виділити наступні недоліки OpenCart: проблеми із індексацією віддалених сторінок, відсутність підтримки контролю за дублікатами сайтів, можливі помилки у формуванні URL-адрес, а також нестабільна робота SEO-оптимізації без використання спеціальних розширень [13].

Magento – це відкрита eCommerce платформа з відкритим вихідним кодом, розроблена для створення інтернет-магазинів. Вона поширюється під ліцензією OSL 3.0 [14].

Magento є однією з найпопулярніших відкритих систем для організації електронної комерції в мережі: на основі цієї платформи було створено понад 100 тисяч інтернет-магазинів. Створено більше 2 тисяч розширень сторонніми розробниками. Спільнота проєкту налічує близько 375 тисяч учасників. Код платформи був завантажений більше 2,5 мільйонів разів. У 2010 році через системи на базі Magento було продано товарів на 25 мільярдів доларів [14].

Серед переваг Magento можна виділити:

- широкий функціонал, який дозволяє створювати різноманітні інтернет-магазини з різними потребами;

- велика кількість розширень та модулів, що розширюють можливості платформи;
- активна та велика спільнота користувачів та розробників, яка надає підтримку та розвиток платформи;
- висока масштабованість та можливість роботи з великою кількістю товарів і клієнтів.

До недоліків Magento можна віднести:

- складність у встановленні та налаштуванні, особливо для не досвідчених користувачів;
- високі вимоги до серверних ресурсів, що може призводити до повільної роботи великих магазинів;
- високі витрати на розробку та підтримку, зокрема через необхідність найму кваліфікованих розробників.

Одним з інструментом реалізації інтернет-магазинів є спеціалізовані фреймворки або каркасні рішення. Фреймворк – це готове рішення для розробки та створення програмного забезпечення інтернет-магазину, який дозволяє зробити весь процес розробки та програмування інтернет-магазину більш простим, налагодити його логіку, забезпечити безперебійну роботу бази даних, створити механізм реєстрації користувачів та інше [15].

Вибір фреймворку залежить від вибору архітектурного шаблону розробки програмного забезпечення та мови програмування. Тому далі розглянемо архітектурні шаблони для розробки вебзастосунків.

## **2.3 Обґрунтування вибору архітектурного шаблону**

Для розробки вебзастосунків використовують наступні архітектурні шаблони: Model View Controller (MVC), Model View Template (MVT) та Model View ViewModel (MVVM). Наведемо основні особливості наведених архітектурних шаблонів.

MVC – архітектурний шаблон (рис. 2.4), який використовується під час проєктування та розробки вебзастосунків.

За цим шаблоном програмну систему розділяють на три взаємопов'язані компоненти: модель даних, представлення або інтерфейс користувача та контролер. Використання цього шаблону дозволяє відокремлювати дані (модель) від інтерфейсу користувача (представлення), забезпечуючи таким чином мінімальний вплив змін інтерфейсу користувача на роботу з даними, а також можливість змін в моделі даних без впливу на інтерфейс користувача [16].

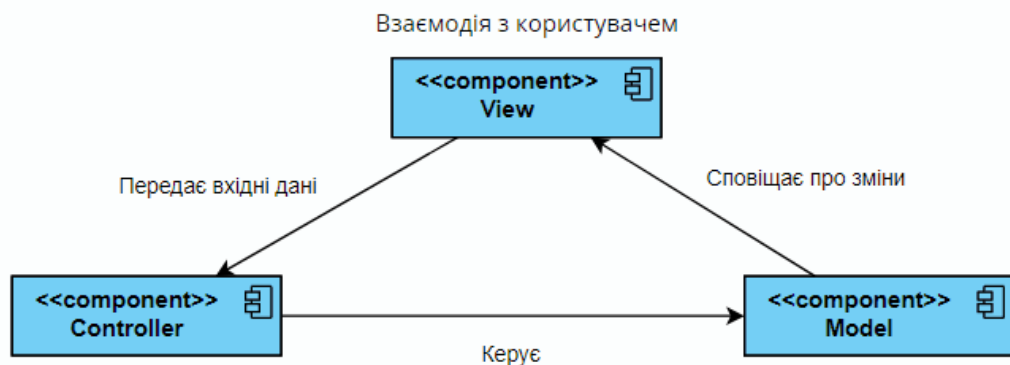


Рисунок 2.4 – Схема шаблону проєктування MVC [15]

За цим шаблоном програмну систему розділяють на три взаємопов'язані компоненти: модель даних, представлення або інтерфейс користувача та контролер. Використання цього шаблону дозволяє відокремлювати дані (модель) від інтерфейсу користувача (представлення), забезпечуючи таким чином мінімальний вплив змін інтерфейсу користувача на роботу з даними, а також можливість змін в моделі даних без впливу на інтерфейс користувача [16].

Даний шаблон дозволяє використовувати окремі компоненти програми повторно, спрощуючи процес розробки подальших змін чи

розширень. Застосування MVC у вебзастосунках впорядковує їх структуру та робить системи за рахунок зменшення складності більш зрозумілими [16].

Відповідно до MVC, вебзастосунок розбивається на три окремі, але взаємозалежні частини з розподілом функцій між компонентами [16].

Модель (Model) відповідає за зберігання даних та їх структуру, Представлення (View) відповідає за інтерфейс вебзастосунку, тобто представленням цих даних користувачеві, а Контролер (Controller) керує компонентами, отримуючи сигнали від користувача (наприклад, рух курсора миші, натискання кнопки або введення даних в текстове поле) та передаючи ці дані у модель [16].

Модель є головною компонентою MVC та відтворює поведінку програми, що є незалежною від інтерфейсу користувача. Крім того, цей компонент відповідає за пряме управління даними, логікою та правилами застосунку [16].

Можна виділити наступні найпопулярніші фреймворки, які підтримують архітектурний шаблон MVC [17]:

- Ruby on Rails – це набір інструментів для розробки вебзастосунків, який використовує мову програмування Ruby;
- Django – це фреймворк для розробки вебзастосунків на мові програмування Python;
- Laravel – це фреймворк для розробки вебзастосунків на мові програмування PHP;
- Spring MVC – це фреймворк для розробки вебзастосунків на мові програмування Java;
- Express.js – це фреймворк для розробки вебзастосунків на JavaScript з використанням платформи Node.js. Він допомагає створювати серверну частину застосунків з використанням шаблону MVC та використовувати JavaScript як на клієнтській, так і на серверній стороні;

– AngularJS – це фреймворк для створення клієнтської частини вебзастосунків на JavaScript. Він реалізує шаблон MVC на стороні клієнта та надає різноманітні інструменти для розробки вебінтерфейсів;

– React – це бібліотека для створення інтерфейсів на JavaScript. Використовуючи React разом із іншими бібліотеками та фреймворками (наприклад, Redux), можна реалізувати шаблон MVC на стороні клієнта [17].

Вищенаведені фреймворки та бібліотеки надають різні можливості та інструменти для розробки вебзастосунків з використанням архітектурного шаблону MVC. Вибір конкретного фреймворку залежить від мови програмування, потреб проєкту та особистих вподобань розробника.

MVT – архітектурний шаблон (рис. 2.5) [18]-[19], який складається з наступних компонентів: Модель (Model) – відповідає за підтримку даних або логічна структура даних, що лежить в основі застосунку і представлена реляційною базою даних; Представлення (View) – інтерфейс користувача(вебсторінки застосунку), Шаблон (Template) – статичні частини бажаних html-представлень, а також спеціального синтаксису, що описує, як буде вставлено динамічний вміст.

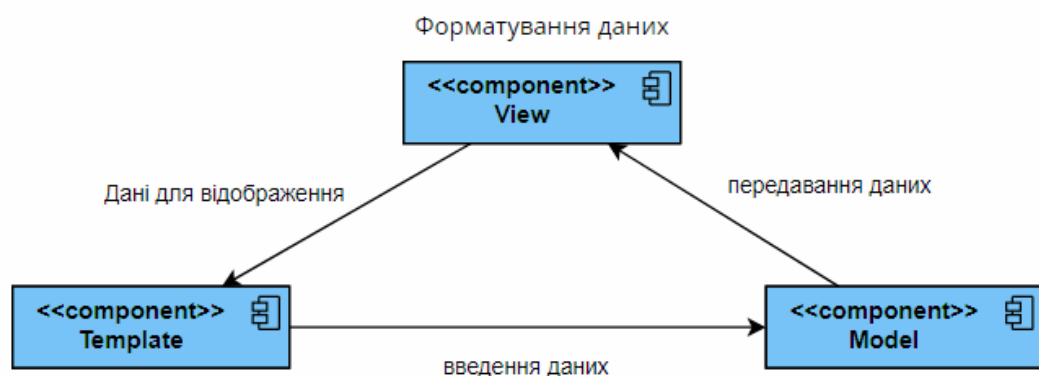


Рисунок 2.5 – Схема шаблону проєктування MVT [18]

Представлення в архітектурі MTV може виглядати як контролер, але призначення Представлення в цій архітектурі MTV це форматування даних

за допомогою Моделі. У свою чергу, Модель зв'язується з базою даних та даними, що передаються Шаблону для перегляду [18].

MVVM – шаблон проєктування, що представляє собою модифікацію шаблону Presentation Model [20].

MVVM спрощує розділення розробки графічного інтерфейсу від функціоналу серверної частини та відокремлення представлення від моделі. Модель представлення відповідає за обробку даних для їх подальшого використання і підтримки. Вона подібна до моделі, аніж до представлення, що дозволяє їй обробляти значну частину функціоналу відображення даних. Модель представлення іноді може бути реалізована як шаблон медіатора, який організовує доступ до функціоналу бекенду через набір правил використання, підтримуваних представленням [20].

MVVM складається з трьох компонентів: Моделі(Model), Відображення(View) та Модель Представлення (ViewModel), так використовується для відокремлення моделі та її відображення. Це необхідно для того, щоб можна було вносити зміни незалежно від інших. Наприклад, програміст працює над логікою обробки даних, тоді як дизайнер зосереджується на користувацькому інтерфейсі [20].

В MVC зміни у інтерфейсі користувача не впливають безпосередньо на модель, а йдуть через Контролер. У технологіях, таких як WPF та Silverlight, існує концепція "зв'язування даних", яка дозволяє зв'язувати дані з візуальними елементами в обидві сторони. Архітектура MVVM вирішує цю задачу шляхом чіткого розподілу відповідальностей [20]:

- розробка інтерфейсу користувача здійснюється дизайнером інтерфейсів за допомогою технології, наприклад використання файлів розмітки XML;

- функціонал інтерфейсу користувача реалізується розробником як компонент ViewModel;

- функціональні зв'язки між інтерфейсом користувача та ViewModel реалізуються через біндинги (bindings), які, по суті, є правилами типу «якщо

була натиснута певна кнопка, то повинен бути викликаний метод `onButtonAClick()` з `ViewModel`». Біндинги можуть бути написані в тексті програми або визначені декларативним шляхом [20].

Архітектура MVVM використовується в тому чи іншому вигляді усіма сучасними технологіями, наприклад Microsoft WPF і Silverlight, Oracle JavaFX, Adobe Flex, AJAX [20].

Шаблон MVVM ділиться на три частини [20]:

- Модель (Model), що представляє собою дані, що необхідні для роботи застосунку: зберігання даних, робота з базою даних та іншими зовнішніми ресурсами;

- Представлення (View) – це інтерфейс користувача, тобто вебсторінки, кнопки тощо;

- Модель Представлення (ViewModel, що означає «Model of View») служить проміжною ланкою між Моделлю та Представленням, містить Модель, яка перетворена до Представлення, а також містить у собі команди для маніпуляції даними, які відображаються у Представленні. ViewModel призначена для того, щоб здійснювати зв'язок між моделлю та вікном, відслідковувати зміни в даних, що зроблені користувачем, відпрацьовувати логіку роботи View (механізм команд).

Для реалізації програмного забезпечення прийнято рішення використати архітектурний шаблон MVT.

Структура вебзастосунку наведена на рис. 2.6 у вигляді діаграми пакетів.

Діаграма пакетів – це один з видів діаграм в UML, який використовується для представлення структури вебзастосунків. Вона показує розподіл елементів системи на пакети, їхню організацію та взаємозв'язки між ними. Кожен пакет може містити класи, модулі та шаблони вебсторінок, що допомагає впорядкувати та структурувати моделі програмного забезпечення. Діаграма пакетів дозволяє визначити залежності між різними частинами системи [9].

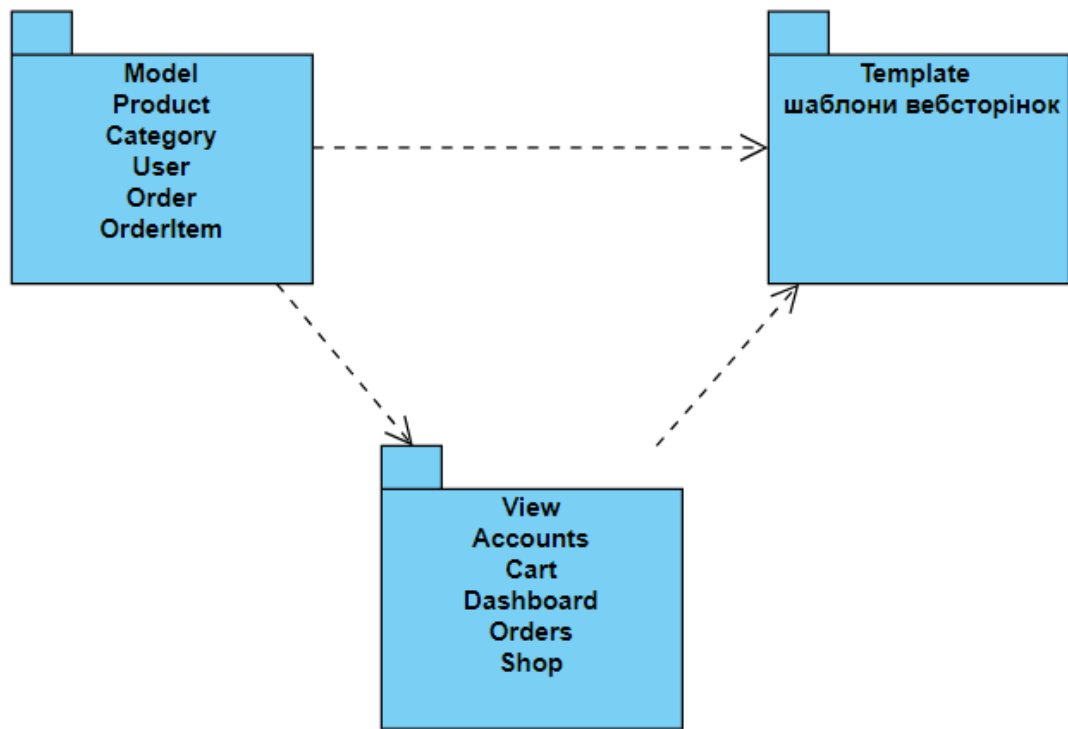
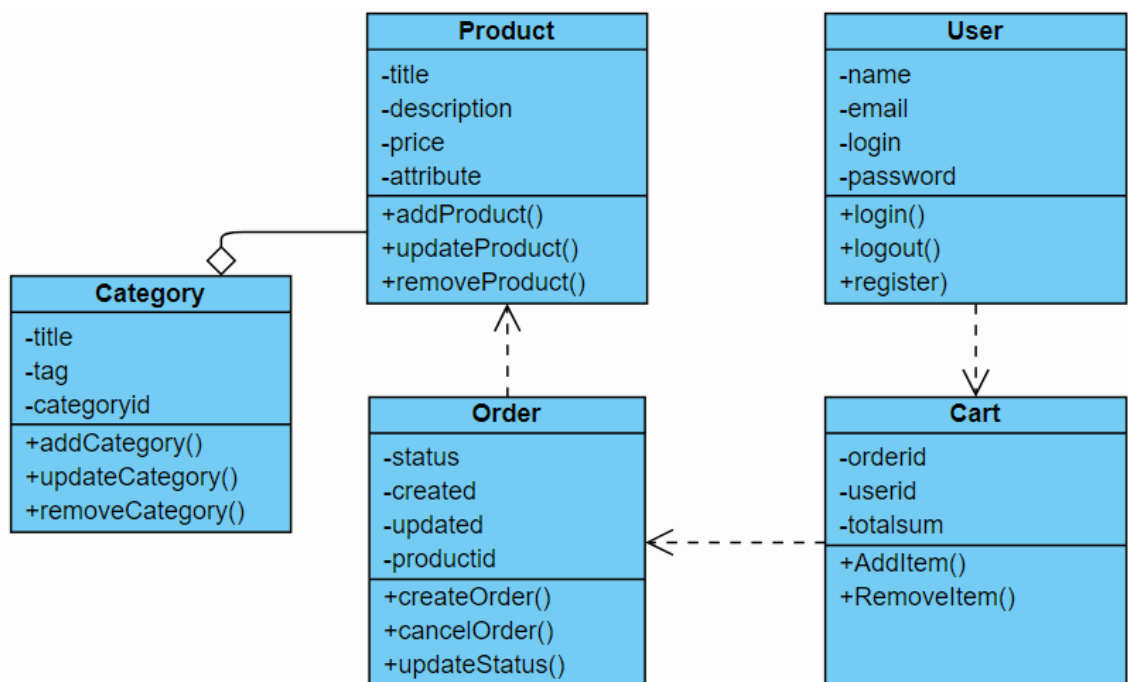


Рисунок 2.6 – Структура інтернет-магазину

Діаграми класів представляють собою діаграми на яких буде показано класи та зв'язки між ними (рис. 2.7).



АКТИВАІ

Рисунок 2.7 – Діаграма класів

На рис. 2.8 наведена діаграма компонент розробленого програмного забезпечення, що складається з компонентів Product(мобільний пристрій), Orders (замовлення), Cart (кошик), Accounts(облікові записи), Dashboard(адміністративний компонент).

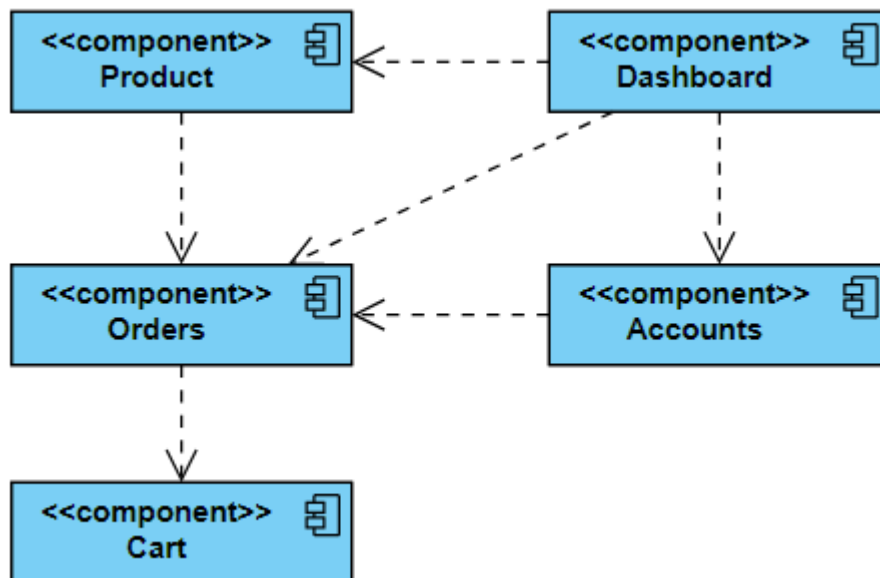


Рисунок 2.8 – Діаграма компонент

## 2.4 Обґрунтування вибору програмних засобів

### 2.4.1 Вибір мови програмування

Розглянемо наступні мови програмування, що використовуються для розробки вебзастосунків: PHP, Python та JavaScript.

PHP (англ. PHP: Hypertext Preprocessor – гіпертекстовий препроцесор), раніше відома як Personal Home Page Tools, є скриптовою мовою програмування, створеною для генерації HTML-сторінок на стороні вебсервера [21]. PHP є однією з найпоширеніших мов, що використовуються у веброботі поряд з Java, .NET, JavaScript, Python і Ruby. Переважна більшість хостинг-провайдерів підтримують PHP. PHP – проєкт з відкритим вихідним кодом. PHP-скрипти інтерпретуються вебсервером у HTML-код, який потім передається на клієнтську сторону. На відміну від JavaScript,

користувач не бачить PHP-коду, оскільки браузер отримує вже готовий HTML-код [21].

PHP – це мова програмування, яка дозволяє безпосередньо вбудовувати HTML-код сторінок у свій код, і такий код буде коректно оброблюватися PHP-інтерпретатором. Обробка PHP починається з виконання коду після відкриваючого тега (`<?php`) і триває до закриваючого тега. Завдяки великій різноманітності функцій PHP можна уникнути написання багаторядкових користувацьких функцій, як це часто відбувається в мовах програмування C або Pascal [21].

Python – інтерпретована об'єктно-орієнтована мова програмування високого рівня із суворою динамічною типізацією[22].

Вона була створена в 1990 році Гвідо ван Россумом. Завдяки своїм високорівневим структурам даних, динамічній семантиці та підтримці динамічного зв'язування, Python є ідеальним інструментом для швидкої розробки програм і інтеграції існуючих компонентів. Мова підтримує модулі та пакети, що сприяє модульності та повторному використанню коду. Інтерпретатор Python і стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі для всіх основних платформ. Python підтримує кілька парадигм програмування, включаючи об'єктно-орієнтовану, процедурну, аспектно-орієнтовану та функціональну [22].

Основні переваги мови Python наступні [22]:

- зрозумілий синтаксис (для виділення блоків використовуються відступи);
- переносимість програм (характерна для більшості інтерпретованих мов);
- стандартний дистрибутив містить безліч корисних модулів (включно з модулем для розробки графічного інтерфейсу);
- можливість використання Python у діалоговому режимі (дуже корисно для експериментів і розв'язання простих задач);

- стандартний дистрибутив включає просте, але водночас потужне середовище розробки під назвою IDLE, написане на Python;
- зручність у розв'язанні математичних задач (підтримка роботи з комплексними числами, оперування цілими числами довільної величини, можливість використання як потужного калькулятора в діалоговому режимі);
- відкритий код (можливість редагування іншими користувачами).

JavaScript – це динамічна, об'єктно-орієнтована прототипна мова програмування, яка реалізує стандарт ECMAScript. Вона найчастіше використовується для створення сценаріїв вебсторінок, що дозволяє взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, а також змінювати структуру та зовнішній вигляд вебсторінки на боці клієнта (пристрої кінцевого користувача). JavaScript класифікується як прототипна (підмножина об'єктно-орієнтованої) скриптова мова програмування з динамічною типізацією. Крім прототипної, JavaScript частково підтримує інші парадигми програмування (імперативну та частково функціональну) та має деякі відповідні архітектурні властивості, зокрема: динамічну та слабку типізацію, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу [23].

Порівняння мов програмування наведено у табл. 2.1.

Таблиця 2.1 – Порівняльна характеристика мов програмування

| Критерій | PHP                     | Python                          | JavaScript                                                                             |
|----------|-------------------------|---------------------------------|----------------------------------------------------------------------------------------|
| 1        | 2                       | 3                               | 4                                                                                      |
| Тип мови | Серверна скриптова мова | Універсальна мова програмування | Скриптова мова для веб (з використанням як на клієнтській, так і на серверній стороні) |

Продовження таблиці 2.1

| 1                          | 2                                    | 3                                                                     | 4                                                                                                    |
|----------------------------|--------------------------------------|-----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Використання               | Веброзробка,<br>серверні сценарії    | Веброзробка,<br>наука про дані,<br>автоматизація,<br>штучний інтелект | Веброзробка<br>(взаємодія з користувачем,<br>асинхронні операції),<br>серверні сценарії<br>(Node.js) |
| Типізація                  | Динамічна,<br>слабка                 | Динамічна,<br>сильна                                                  | Динамічна,<br>слабка                                                                                 |
| Парадигми<br>програмування | Імперативна,<br>об'єктно-орієнтована | Імперативна,<br>об'єктно-орієнтована,<br>функціональна,<br>процедурна | Імперативна,<br>об'єктно-орієнтована,<br>функціональна                                               |
| Стандартна<br>бібліотека   | Широка<br>підтримка<br>вебфункцій    | Дуже багата,<br>включає багато<br>модулів для<br>різних задач         | Багата, особливо<br>для роботи з веб                                                                 |
| Продуктивність             | Висока в<br>веброзробці              | Висока в різних<br>задачах                                            | Висока в<br>веброзробці,<br>особливо з<br>використанням<br>Node.js                                   |
| Керування<br>пам'яттю      | Автоматичне                          | Автоматичне                                                           | Автоматичне                                                                                          |

Кінець таблиці 2.1

| 1                     | 2                                 | 3                                | 4                                                |
|-----------------------|-----------------------------------|----------------------------------|--------------------------------------------------|
| Підтримка паралелізму | Обмежена                          | Модулі asyncio, threading        | Асинхронні операції, Promises                    |
| Середовище виконання  | Сервери з підтримкою PHP (Apache) | Інтерпретатор Python             | Браузери, Node.js                                |
| Модульність           | Підтримка через include і require | Підтримка через модулі та пакети | Підтримка через модулі (ES6), CommonJS (Node.js) |

Для реалізації програмного забезпечення прийнято рішення обрати мову програмування Python.

#### 2.4.2 Вибір середовища розробки

В якості середовища розробки проведемо огляд наступних інструментів Visual Studio Code, PyCharm та Atom.

Visual Studio Code (VS Code) – це потужний та популярний редактор вихідного коду, розроблений компанією Microsoft. Він побудований на базі Electron Framework і доступний для операційних систем Windows, Linux та macOS [24]. Visual Studio Code надає широкий спектр функціональності, включаючи підтримку налагодження, підсвічування синтаксису, інтелектуальне автозавершення коду, фрагменти коду, рефакторинг та вбудовану інтеграцію з системою контролю версій Git.

VS Code редактор дозволяє користувачам налаштовувати його під свої потреби, змінюючи теми, комбінації клавіш, параметри та встановлюючи розширення для розширення його можливостей. Visual Studio Code надійно

підтримує робочий процес розробника, забезпечуючи зручне середовище для написання, редагування та налагодження коду у різних мовах програмування.

PyCharm – це інтегроване середовище розробки, яке призначене для програмування на мові Python [25]. Воно має інструменти для аналізу коду, графічний зневаджувач, можливість виконання юніт-тестів і підтримку веброзробки на Django. Програма PyCharm розроблена компанією JetBrains з Чехії на базі IntelliJ IDEA. Це програмне забезпечення можна встановити на операційні системи Windows, Mac OS X і Linux.

Наведемо наступні можливості PyCharm [25]:

- проведення статичного аналізу коду, підсвічення синтаксичних помилок і виявлення потенційних проблем;
- зручна навігація у проєктах та початковому коді, включаючи відображення файлової структури, швидкий перехід між класами, методами та файлами;
- можливості рефакторингу, такі як перейменування, витяг методу, введення змінної, введення константи, підняття і опускання методів та ін.;
- інтеграція з фреймворком Django для зручної веброзробки;
- вбудований зневаджувач, що спрощує відлагодження коду на мові Python;
- вбудовані інструменти для проведення юніт-тестування;
- підтримка розробки з використанням Google App Engine;
- інтеграція з різними системами контролю версій, такими як Mercurial, Git, Subversion, Perforce і CVS, з можливістю використання загального користувацького інтерфейсу для керування змінами та злиттями [25].

PyCharm Professional Edition пропонує кілька різновидів ліцензій, які різняться за функціональністю, вартістю та умовами використання. Для освітніх закладів і проєктів з відкритим початковим кодом доступна безкоштовна версія PyCharm Professional Edition. Також існує безкоштовна

версія Community Edition з обмеженим функціоналом, що поширюється під ліцензією Apache 2 [25].

Atom – це вільний текстовий редактор і редактор коду, розроблений компанією "GitHub" [26]. Він може використовуватися як самодостатнє рішення для редагування текстів та коду, а також у якості технологічного стеку для створення різних спеціалізованих рішень. На платформі Atom базуються інші середовища розробки, зокрема, відомі Visual Studio Code від компанії Microsoft і Nuclide від Facebook. Проєкт був представлений компанією GitHub у лютому 2014 року, а його початковий код поширюється під ліцензією MIT [26]. Atom надає широкі можливості для редагування коду та текстових документів та має активну спільноту розробників, яка постійно вносить удосконалення та розширення до цього редактора.

Порівняльна характеристика середовищ розробки наведено у табл. 2.2.

Таблиця 2.2 – Порівняльна характеристика середовищ розробки

| Критерій          | Visual Studio Code                | PyCharm                                          | Atom                                             |
|-------------------|-----------------------------------|--------------------------------------------------|--------------------------------------------------|
| 1                 | 2                                 | 3                                                | 4                                                |
| Тип               | Редактор вихідного коду           | Інтегроване середовище розробки                  | Текстовий редактор і редактор коду               |
| Підтримувані мови | Багатомовний                      | Python, JavaScript, HTML, CSS та інші            | Різноманітні мови програмування та мови розмітки |
| Навігація         | Зручна навігація, швидкий перехід | Швидкий перехід між класами, методами та файлами | Швидкий перехід між рядками та файлами           |

Продовження таблиці 2.2

| 1                      | 2                                          | 3                                                  | 4                                                     |
|------------------------|--------------------------------------------|----------------------------------------------------|-------------------------------------------------------|
| Функціональність       | Зневаджувач, підсвічування синтаксису, Git | Аналіз коду, зневаджувач, рефакторинг, веброзробка | Можливості редагування та кодування, інтеграція з Git |
| Підтримка розширень    | Широкий вибір розширень                    | Розширений функціонал за допомогою плагінів        | Велика кількість плагінів для розширення можливостей  |
| Спільнота користувачів | Велика                                     | Велика                                             | Активна                                               |
| Вартість               | Безкоштовний                               | Платний                                            | Безкоштовний                                          |

В якості середовища розробки було прийнято рішення обрати Visual Studio Code.

### 2.4.3 Вибір системи керування базами даних

Бази даних є спеціалізованими сховищами, призначеними для зберігання різних типів даних [27]. Кожна база даних має певну модель (реляційну, документно-орієнтовану тощо), що забезпечує зручний доступ до інформації. Системи керування базами даних (СКБД) – це програмні застосунки, які керують базами даних різних розмірів і форматів.

Найбільш популярними СКБД на сьогоднішній день є:

- SQLite – дуже потужна вбудована система керування;
- MySQL – найпопулярніша і поширена СКБД;
- MariaDB – реляційна СКБД, створена як відгалуження MySQL;
- PostgreSQL – є однією з найбільш передових СКБД [28].

Переваги та недоліки найбільш відомих систем керування базами даних.

SQLite легко вбудовується в застосунки баз даних і, оскільки базується на файлах, вона надає широкий спектр інструментів для роботи, порівняно з мережевими СКБД. Використання SQLite передбачає безпосереднє звернення до файлів, а не до портів і гнізд мережеских СКБД, що робить її дуже швидкою та потужною завдяки технологіям обслуговуючих бібліотек [29].

Переваги використання SQLite:

- вся база даних складається з одного файлу, тому її легко переносити на різні машини;
- відмінно підходить для розробки та тестування.

Недоліки SQLite:

- відсутня система користувачів;
- відсутня можливість збільшення продуктивності.

SQLite краще використовувати, коли важлива можливість легкого перенесення програми, але не важлива масштабованість. Наприклад, одно користувальницькі застосунки, мобільні застосунки або ігри.

MySQL – це найпоширеніша повноцінна серверна СКБД. MySQL дуже функціональна, що вільно розповсюджується СКБД, яка успішно працює з різними сайтами і вебзастосунками [28]. Навчитися використанню цієї СУБД досить просто, так як на просторах інтернету про неї написано більшу кількість інформації.

Переваги MySQL:

- простота в роботі;
- GUI дозволяє досить легко працювати з базою даних (БД);
- багатий функціонал;
- безпека;
- масштабованість;
- швидкість.

Недоліки:

- проблеми з надійністю;
- повільна розробка (існують скарги на процес розробки);
- високий рівень небезпеки.

MySQL можна розглядати як альтернативу SQLite, особливо якщо потрібен розширений функціонал та можливість використання вихідного коду.

MariaDB є реляційною СКБД, яка з'явилася як відгалуження від MySQL на початку 2009 року. Розповсюджується MariaDB під вільною ліцензією GNU GPL [29]. Цей сервер баз даних використовується такими відомими користувачами, як Wikipedia, WordPress.com та Google. MariaDB зберігає високий рівень сумісності з MySQL, що означає, що більшість програм, створених для MySQL, працюватимуть із MariaDB без проблем. Сервер MariaDB дуже уважно ставиться до збереження зворотної сумісності для своїх користувачів. Оновлення старих версій MySQL до найновіших версій MariaDB підтримується, а сервер MariaDB навіть забезпечує режим сумісності синтаксису Oracle для запуску програм Oracle Database без змін.

PostgreSQL, у відміну від MariaDB, спочатку був дослідницьким проєктом, який акцентувався на функціональності, а не на продуктивності та стабільності. MariaDB, нащадок MySQL, приймала більш прагматичний підхід, з меншою кількістю функцій, але з вагою на продуктивність, стабільність та простоту використання. Згодом різниця між ними стала менш помітною: MariaDB зосередилася на більш повній реалізації стандарту ANSI SQL, тоді як PostgreSQL ставив перед собою завдання покращення продуктивності.

PostgreSQL є найпрофесійнішою із трьох розглянутих СКБД. Вона вільно поширюється та максимально відповідає стандартам SQL. PostgreSQL виділяється від інших СКБД підтримкою обох підходів до баз даних: об'єктно-орієнтованого та реляційного. Завдяки потужним технологіям PostgreSQL є дуже продуктивною [30].

### Переваги PostgreSQL:

- відкрите ПЗ, що відповідає стандарту SQL;
- велика кількість доповнень;
- можливість розширення функціоналу за рахунок збереження своїх процедур;

### Недоліки:

- низька продуктивність в порівнянні з MySQL;
- відносно низька популярність;
- складно знайти хостинг з підтримкою цієї СКБД.

PostgreSQL буде, кращим вибором, якщо вимоги «заточуються» на надійність і цілісність даних.

Порівняння основних функціональних можливостей СКБД наведено у табл. 2.3.

Таблиця 2.3 – Порівняльна характеристика СКБД

| Критерій            | SQLite                          | MySQL                           | MariaDB                         | PostgreSQL                      |
|---------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| 1                   | 2                               | 3                               | 4                               | 5                               |
| Тип                 | Вбудована база даних            | Серверна база даних             | Серверна база даних             | Серверна база даних             |
| Відкритість         | Відкрите програмне забезпечення | Відкрите програмне забезпечення | Відкрите програмне забезпечення | Відкрите програмне забезпечення |
| Сумісність з SQL    | Так                             | Так                             | Так                             | Так                             |
| Підтримка доповнень | Обмежена                        | Так                             | Так                             | Так                             |
| Продуктивність      | Низька                          | Висока                          | Висока                          | Висока                          |

Продовження таблиці 2.3

| 1                                        | 2           | 3        | 4        | 5   |
|------------------------------------------|-------------|----------|----------|-----|
| Підтримка великих обсягів даних          | Не найкраща | Так      | Так      | Так |
| Підтримка об'єктно-орієнтованого підходу | Не надає    | Не надає | Не надає | Так |
| Активний розвиток                        | Ні          | Так      | Так      | Так |

Для розробки бази даних прототипу інтернет-магазину з продажу мобільних пристроїв було прийнято вирішення обрати СКБД MySQL.

#### 2.4.4 Огляд використаних фреймворків

Для розробки серверної частини прототипу інтернет-магазину з продажу мобільних пристроїв необхідно обрати фреймворк.

На сьогоднішній день найбільш популярні фреймворки для розроблення вебзастосунків Flask та Django з використанням мови програмування Python.

Розглянемо функціональні можливості Flask та Django.

Flask вважається мікрофреймворком для вебзастосунків, що розробляються з використанням мови програмування Python та містить інструментарій Werkzeug та рушій шаблонів Jinja2 та поширюється відповідно до умов ліцензії BSD.

Програмний засіб Flask називають мікрофреймворком. Flask не містить спеціальних засобів чи бібліотек для роботи з базою даних, перевірки форм або інших компонентів сторонніх бібліотек. Проте Flask підтримує

розширення, які додають ці функції, наче вони були частиною Flask із самого початку. Існують розширення для об'єктно-реляційних зв'язків, перевірки форм, контролю завантаження, підтримки різних відкритих технологій аутентифікації та інших поширених інструментів для фреймворку [31]-[32].

Flask використовує механізм шаблонів Jinja для динамічного створення HTML-сторінок із використанням базового синтаксису мови програмування Python – змінні, цикли, списки тощо. Інструментарій Werkzeug – бібліотека на мові програмування Python, яка реалізує WSGI (Web Server Gateway Interface, універсальний інтерфейс для взаємодії між більшістю вебсерверів і вебзастосунками чи фреймворками), за допомогою можна створювати об'єкти `request`, `response` та використовувати інші допоможні функції.

В якості шаблону проєктування використовується подібний шаблон Model View Template, а саме обгортка навколо Werkzeug, що представляє собою обгорткою навколо чистого WSGI. Тобто обгортка з можливостями шаблону Model View Template.

Django – безкоштовний та відкритий фреймворк для створення вебзастосунків, написаний мовою програмування Python [33]. При реалізації вебзастосунку, завжди необхідний подібний набір компонентів: спосіб управління авторизацією користувача (створення акаунту, вхід, вихід), панель управління, форми, спосіб завантажити файли, тощо. Для використання подібних готових компонент був і розроблений фреймворк Django з метою зменшення часу для реалізації вебзастосунків.

Можна виділити наступні переваги Django [34]:

- легкість в налаштуванні та запуску;
- забезпечення готового до використання інтерфейсу користувача для адміністративної діяльності;
- можливість створення багатомовних вебсайтів за допомогою вбудованої системи інтернаціоналізації;
- можливість врахування великих потреби трафіку;

- використання для створення всіх типів систем керування вмістом, соціальних мереж, а також наукових обчислювальних платформ;
- можливість наскрізного тестування за стосунків;
- можливість документації розробленого API за допомогою виводу HTML;
- REST Framework має широку підтримку кількох протоколів автентифікації;
- наявність дозволів та політик обмеження;
- широке використання для обмеження швидкості запитів API від одного користувача.

Особливості фреймворку Django наступні:

- можливість визначення шаблони для URL-адрес у програмі;
- наявність простої, але потужної системи URL;
- наявність вбудованої системи аутентифікації;
- наявність бази даних на об'єктно-орієнтованій мові програмування, яка реалізує зберігання та пошук даних;
- можливість використання функції автоматичного інтерфейсу адміністратора, що дозволяє додавати, редагувати та видаляти елементи;
- можливість налаштування панелі адміністратора відповідно до потреб замовника;
- кеш-фреймворк має кілька механізмів кешу.

Архітектура фреймворку Django містить в собі шаблон проєктування програмного забезпечення Model View Template. Django сам керує частиною контролера (програмний код, який контролює взаємодію між моделлю та представленням). Шаблон – це файл HTML, який змішується з файлом мови шаблонів Django, який також називається DTL [34]. Усі компоненти шаблону Model View Template взаємодіють один з одним, щоб обслуговувати конкретний запит користувача, схема взаємодії наведена на рис. 2.9.

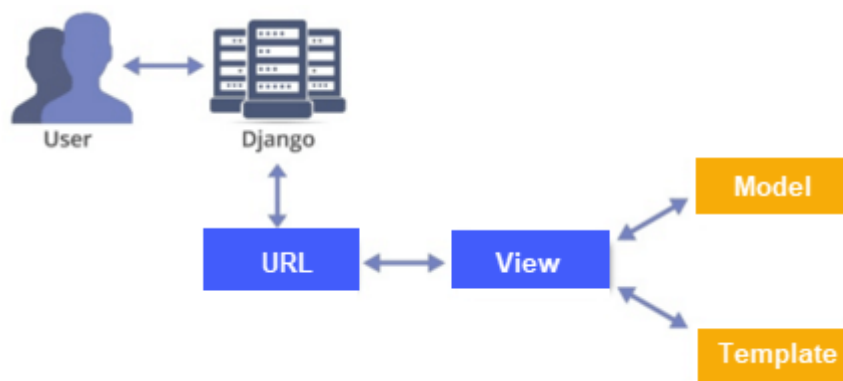


Рисунок 2.9 – Model View Template архітектура в Django [34]

Згідно з рисунком 2.9, користувач запитує ресурс у Django. Django діє як контролер, перевіряючи доступність ресурсу за URL-адресою. Якщо URL-адреса відповідає заданому маршруту, викликається представлення, яке взаємодіє з моделлю та шаблоном. Після цього Django формує відповідь і надсилає її користувачеві у вигляді шаблону.

## 2.5 Засоби проєктування макету користувацького інтерфейсу вебзастосунків

Для верстки макету користувацького інтерфейсу вебзастосунків на сьогоднішній день використовують спеціалізоване програмне забезпечення таке як Adobe Experience Design (Adobe XD) та Figma, а також засоби автоматичного створення макету інтернет-магазину на основі штучного інтелекту (ШІ) такі як Wix ADI (Artificial Design Intelligence), Uizard та ін.

Розглянемо спеціалізоване програмне забезпечення Adobe XD та Figma.

Adobe XD – високофункціональне програмне забезпечення, розроблене Adobe Systems, призначене для проєктування і прототипування інтерфейсів вебсайтів та мобільних застосунків, що надає широкий набір функцій і можливостей, які роблять процес дизайну та створення інтерактивних прототипів ефективнішим і зручнішим [35].

Можна створювати швидкі та прості дизайни на артбордах в Adobe XD. Потім з'єднати макети разом, щоб створити інтерактивний прототип, яким можна поділитися зі стейкхолдерами та повторювати його. Також можна використовувати плагіни для автоматизації повторюваних операцій або частин робочих процесів дизайнера, які є нудними, складними або повторюваними [36].

Adobe XD добре інтегрується з Photoshop, Illustrator та After Effects. Можна продовжувати проєктування в наведених програмах, перенести ресурси в XD, а потім використовувати XD для створення та обміну прототипами. XD підтримує SVG і растрові файли без втрати точності. Далі можна вдосконалити ресурси в Adobe XD або використовувати їх для розробки інтерактивних прототипів [36]. Adobe XD використовує ШІ для автоматизації рутинних завдань та пропонує інструменти, як Adobe Sensei, для аналізу та покращення дизайну.

Figma – векторний онлайн-сервіс розробки інтерфейсів та прототипування з можливістю організації спільної роботи, який працює у двох форматах: у браузері та як клієнтський застосунок на персональному комп'ютері користувача та дозволяє зберігати онлайн-версії файлів, з якими працював користувач [37], та надає широкий спектр функціональних можливостей і особливостей. Figma пропонує різні плагіни на основі ШІ, такі як Autoflow та Remove.bg, які автоматизують процес створення макетів і спрощують роботу дизайнерів.

Проведемо огляд засобів автоматичного створення макету інтернет-магазину на основі ШІ.

Wix ADI (Artificial Design Intelligence) – засіб на основі ШІ, що дозволяє автоматично створити унікальний дизайн вебзастосунку та вбудований в конструктор вебсайтів Wix, використовуючи інформацію від замовника та аналізуючи відповіді на питання щодо стилю та потреб користувача, автоматично створюючи дизайн інтернет-магазину, включаючи всі необхідні сторінки та елементи [38].

Доступ до Wix ADI можна отримати двома способами. Перший, найпростіший спосіб – перейти на сторінку конструктора сайтів зі ШІ і натиснути кнопку «Створити зі штучним інтелектом». Якщо відсутній акаунт у Wix конструкторі або не увійшли у свій обліковий запис, то буде перенаправлено на сторінку входу до Wix. Дотримуйтесь інструкцій, щоб створити безкоштовний акаунт Wix або увійти в існуючий. Після входу в обліковий запис буде запропоновано почати чат зі ШІ. Натисніть «Почати чат», щоб отримати доступ до конструктора сайтів [39].

Другий спосіб з Wix акаунту, для цього перейдіть на Wix.com і виберіть «Почати роботу». Далі буде запропоновано відповісти на кілька запитань про налаштування, після чого буде виконано перехід на інформаційну панель, де можна натиснути «Створити сайт». Далі можна дозволити Wix створити вебсайт автоматично або створити самостійно, використовуючи шаблон дизайну [39].

При наявності облікового запису на Wix, потрібно увійти до нього і натиснути «Створити новий сайт». На наступній сторінці необхідно натиснути «Почати чат», щоб почати створення за допомогою ШІ. Після завершення кроків налаштування буде здійснено перехід на панель управління сайтом, де зможете натиснути «Створити сайт» [39]. Безкоштовна версія конструктора Wix обмежена, платні версії мають більше функціональних можливостей.

Uizard – програмний інструмент, що дозволяє створювати дизайн інтернет-магазину на основі ескізів. Завдяки ШІ, він автоматично перетворює ескізи та макети у функціональні інтерфейси вебзастосунків. Даний програмний засіб дозволяє розробити унікальний дизайн вебзастосунку та надати можливість неспеціалістам у галузі дизайну створювати цифрові та інтерактивні продукти. [40]. Безкоштовна версія дозволяє створити 2 проекти дизайну та надає 10 безкоштовних шаблонів для використання.

Інструмент <https://builder.ai/> – програмний засіб, що дозволяє автоматизувати створення вебсайтів та мобільних застосунків, включаючи

інтернет-магазини. Використовуючи ШІ, він дозволяє генерувати дизайн та функціональність на основі вимог користувача, створювати застосунки з необмеженою кількістю налаштувань, надавати прототип програми, який можна доповнювати та розширювати. Демоверсія має обмежені функціональні можливості [41].

## **2.6 Висновок до другого розділу**

В якості архітектурного шаблону проєктування програмного забезпечення було обраний шаблон Model View Template, що використовується саме для розроблення вебзастосунків у фреймворках Flask та Django, було обрано мову програмування Python, середовища розробки – Visual Studio Code, фреймворк – Django, системи керування базами даних – MySQL.

## З КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для розробки серверної частини прототипу інтернет-магазину з продажу мобільних пристроїв було обрано фреймворк Django, шаблон Model View Template, мову програмування Python, в якості середовища розробки – Visual Studio Code та систему керування базами даних MySQL.

Розглянемо проєктування бази даних для прототипу інтернет-магазину з продажу мобільних пристроїв.

### 3.1 Розробка бази даних

Проєктування концептуальної моделі заснована на аналізі розв'язуваних завдань з обробки даних в інтернет-магазині з продажу мобільних пристроїв та вимог користувачів. Концептуальна модель допоможе візуалізувати структуру бази даних та встановити основні зв'язки між сутностями [27], [42]. Вона служить основою для подальшого розроблення фізичної моделі бази даних та реалізації інтернет-магазину.

Предметна галузь представляє собою каталог мобільних пристроїв. Користувач обирає потрібний мобільний пристрій із певної категорії і робить замовлення. Перелік сутностей та атрибутів предметної області наступний після початкового проєктування:

- мобільний пристрій: унікальний номер, назва, опис, ціна;
- категорія: унікальний номер, назва ;
- список замовлень: унікальний номер, дата, статус;
- замовлення: унікальний номер, ціна, кількість, посилання на замовлення, посилання на мобільний пристрій;
- користувач: унікальний номер, пароль, поштова скринька, повне ім'я.

Для відображення схеми бази даних використовувалась нотація «воронячі лапки». Схема бази даних зображена на рис. 3.1.

Для побудови схеми бази даних використовувався безкоштовний вебсервіс DrawSQL для створення діаграм [43].

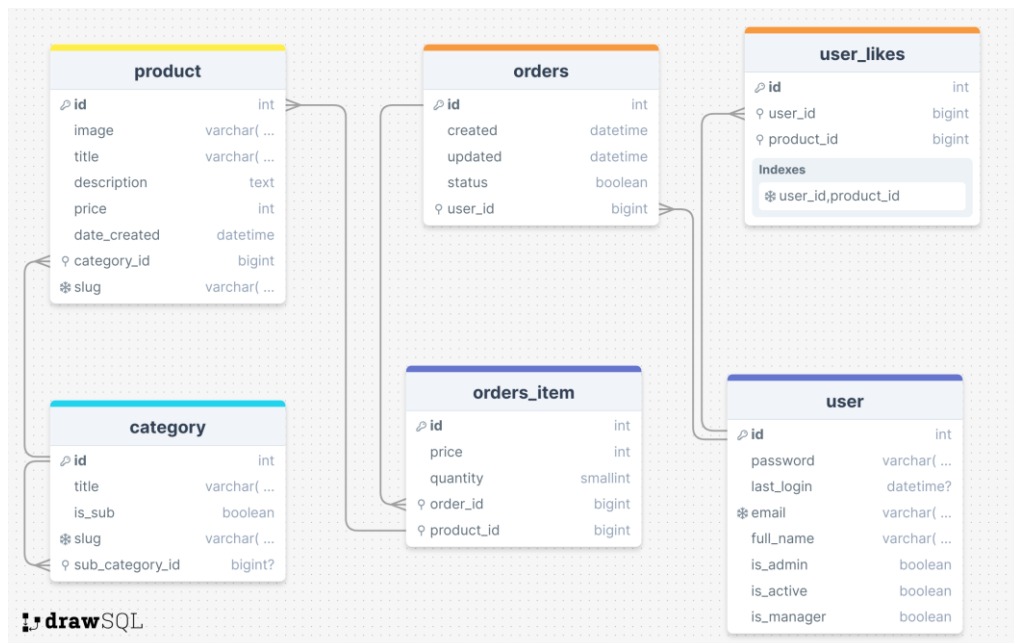


Рисунок 3.1 – Схема бази даних

База даних складається з наступних таблиць: **product**(мобільний пристрій), **category**(категорія), **orders**(замовлення), **orders\_item**(компонент замовлення), **user**(користувач), **user\_likes**(вподобання користувачів).

Таблиця 3.1 (**product**) зберігає в собі інформацію про мобільний пристрій, а саме: ідентифікатор мобільного пристрою(унікальний номер), шлях до зображення, назву, опис, ціна, дату створення, ідентифікатор категорії, тег категорії.

Таблиця 3.2 (**category**) зберігає в собі інформацію про категорію мобільного пристрою, а саме: ідентифікатор категорії(унікальний номер), назву, прапорець **is\_sub** чи це підкатегорія, тег категорії, ідентифікатор підкатегорії.

Таблиця 3.3 (**orders\_item**) зберігає в собі інформацію про замовлення, а саме: ідентифікатор замовлення(унікальний номер), ціна, кількість, ідентифікатор списку замовлення, ідентифікатор мобільного пристрою.

Таблиця 3.1 – Таблиця product

| Назва поля   | Тип даних   | Розмір     | Тип поля    |
|--------------|-------------|------------|-------------|
| id           | Лічильник   | Ціле       | Ключ        |
| title        | Текстовий   | 250        | Обов'язкове |
| description  | Текстовий   | 65535      | Обов'язкове |
| price        | Числовий    | Ціле       | Обов'язкове |
| image        | Текстовий   | 100        | Обов'язкове |
| date_created | Дата та час |            | Обов'язкове |
| category_id  | Числовий    | Довге ціле | Обов'язкове |
| slug         | Текстовий   | 50         | Обов'язкове |

Таблиця 3.2 – Таблиця category

| Назва поля      | Тип даних | Розмір     | Тип поля    |
|-----------------|-----------|------------|-------------|
| id              | Лічильник | Ціле       | Ключ        |
| title           | Текстовий | 200        | Обов'язкове |
| is_sub          | Логічний  |            | Обов'язкове |
| slug            | Текстовий | 200        | Обов'язкове |
| sub_category_id | Числовий  | Довге ціле | Обов'язкове |

Таблиця 3.3 – Таблиця orders\_item

| Назва поля | Тип даних | Розмір     | Тип поля    |
|------------|-----------|------------|-------------|
| id         | Лічильник | Ціле       | Ключ        |
| price      | Числовий  | Ціле       | Обов'язкове |
| quantity   | Числовий  | Ціле       | Обов'язкове |
| order_id   | Числовий  | Довге ціле | Обов'язкове |
| product_id | Числовий  | Довге ціле | Обов'язкове |

Таблиця 3.4 (orders) зберігає в собі інформацію про замовлення, а саме: ідентифікатор замовлення(унікальний номер), дату створення, дату оновлення, статус, ідентифікатор користувача.

Таблиця 3.4 – Таблиця orders

| Назва поля | Тип даних   | Розмір     | Тип поля    |
|------------|-------------|------------|-------------|
| id         | Лічильник   | Ціле       | Ключ        |
| created    | Дата та час |            | Обов'язкове |
| updated    | Дата та час |            | Обов'язкове |
| status     | Логічний    |            | Обов'язкове |
| user_id    | Числовий    | Довге ціле | Підстановка |

Таблиця 3.5 (user) зберігає в собі інформацію про користувача, а саме: ідентифікатор користувача(унікальний номер), пароль, дата останнього входу, поштова скринька, повне ім'я, прапорці перевірки: is\_admin, is\_active, is\_manager.

Таблиця 3.5 – Таблиця user

| Назва поля | Тип даних   | Розмір | Тип поля    |
|------------|-------------|--------|-------------|
| id         | Лічильник   | Ціле   | Ключ        |
| password   | Текстовий   | 128    | Обов'язкове |
| last_login | Дата та час |        | Обов'язкове |
| email      | Текстовий   | 100    | Обов'язкове |
| full_name  | Текстовий   | 100    | Обов'язкове |
| is_admin   | Логічний    |        | Обов'язкове |
| is_active  | Логічний    |        | Обов'язкове |
| is_manager | Логічний    |        | Обов'язкове |

Таблиця 3.6 (user\_likes) зберігає в собі інформацію про вподобання користувача, а саме: ідентифікатор вподобання користувача(унікальний номер), ідентифікатор користувача, ідентифікатор мобільного пристрою.

Таблиця 3.6 – Таблиця user\_likes

| Назва поля | Тип даних | Розмір     | Тип поля    |
|------------|-----------|------------|-------------|
| id         | Лічильник | Ціле       | Ключ        |
| user_id    | Числовий  | Довге ціле | Підстановка |
| product_id | Числовий  | Довге ціле | Підстановка |

### 3.2 Розробка класів моделей та представлень

В пакеті Model знаходяться класи моделей, що відповідають за сутності мобільний пристрій, категорія, користувач, замовлення, список замовлень.

Клас Product, що є об'єктним відображенням таблиці product.

Клас Category, що є об'єктним відображенням таблиці category.

Клас User, що є об'єктним відображенням таблиці user.

Клас Order, що є об'єктним відображенням таблиці orders.

Клас OrderItem, що є об'єктним відображенням таблиці orders\_item.

В пакеті Template знаходяться шаблони статичних частин бажаних html-представлень. Були розроблені наступні шаблони для представлення інформації, пов'язаних з відображенням дій користувача (табл. 3.7).

Для відображення кошика був розроблений наступний шаблон представлення інформації cart.html.

Для відображення інформації про замовлення були реалізовані наступні шаблони представлення (табл. 3.8).

Були розроблені наступні шаблони для представлення інформації, пов'язаних з відображенням каталогу мобільних пристроїв в інтернет-магазині (табл. 3.9).

Таблиця 3.7 – Шаблони представлень для користувача

| Назва html-представлення     | Призначення                                                        |
|------------------------------|--------------------------------------------------------------------|
| edit_profile.html            | вебсторінка для редагування профілю користувача                    |
| email_template.html          | вебсторінка для скидання паролю                                    |
| login.html                   | вебсторінка для авторизації користувача                            |
| manager_login.html           | вебсторінка для авторизації менеджера                              |
| password_reset_complete.html | вебсторінка для відображення, що пароль надіслано на пошту         |
| password_reset_confirm.html  | вебсторінка для підтвердження, що пароль скинуто успішно           |
| password_reset_done.html     | вебсторінка з інформацією, що пароль скинуто успішно               |
| password_reset.html          | вебсторінка для введення інформації, на яку пошту надсилати пароль |
| register.html                | вебсторінка для реєстрації користувача                             |

Таблиця 3.8 – Шаблони представлень для замовлення

| Назва html-представлення | Призначення                                        |
|--------------------------|----------------------------------------------------|
| checkout.html            | вебсторінка для оформлення та оплати замовлення    |
| user_orders.html         | вебсторінка для відображення замовлень користувача |

Таблиця 3.9 – Шаблони представлень для відображення каталогу мобільних пристроїв в інтернет-магазині

| Назва html-представлення | Призначення                                                                   |
|--------------------------|-------------------------------------------------------------------------------|
| 1                        | 2                                                                             |
| base.html                | вебсторінка для відображення каталогу мобільних пристроїв в інтернет-магазині |

Продовження таблиці 3.9

| 1                   | 2                                                                                      |
|---------------------|----------------------------------------------------------------------------------------|
| favorites.html      | вебсторінка для відображення вподобаних користувачем мобільних пристроїв               |
| home_page.html      | вебсторінка для відображення інформації про попередній та наступний мобільний пристрій |
| product_detail.html | вебсторінка для відображення інформації про мобільний пристрій                         |

Для відображення дій менеджера в адміністративній частині були розроблені наступні шаблони представлення інформації (табл. 3.10).

Таблиця 3.10 – Шаблони представлень дій менеджера в адміністративній частині

| Назва html-представлення | Призначення                                                      |
|--------------------------|------------------------------------------------------------------|
| add_category.html        | вебсторінка для відображення додавання категорії                 |
| add_product.html         | вебсторінка для відображення додавання мобільного пристрою       |
| dashboard.html           | вебсторінка для відображення меню адміністративної частини       |
| edit_product.html        | вебсторінка для редагування інформації про мобільний пристрій    |
| order_detail.html        | вебсторінка для відображення детальної інформації про замовлення |
| orders.html              | вебсторінка для відображення списку замовлень                    |
| products.html            | вебсторінка для відображення списку мобільних пристроїв          |

В пакеті View містяться скрипти для обробки інформації, пов'язаної що відображається на вебсторінках інтернет-магазину.

Для компоненту Accounts розроблений скрипт view.py, що містить перелік наступних функцій (табл. 3.11).

Таблиця 3.11 – Перелік функцій для View в компоненті Accounts

| Назва функції              | Призначення                           |
|----------------------------|---------------------------------------|
| def create_manager()       | створення облікового запису менеджера |
| def manager_login(request) | авторизація менеджера                 |
| def user_register(request) | реєстрація користувача                |
| def user_login(request)    | авторизація користувача               |
| def user_logout(request)   | вихід користувача                     |
| def edit_profile(request)  | редагування профілю користувача       |

Для компоненту Cart розроблений скрипт view.py, що містить перелік наступних функцій (табл. 3.12). Декоратор @login\_required потрібний для захисту перегляду в інтернет-магазині, змушуючи клієнта автентифікуватися з дійсним зареєстрованим користувачем.

Таблиця 3.12 – Перелік функцій для View в компоненті Cart

| Назва функції                                                   | Призначення                             |
|-----------------------------------------------------------------|-----------------------------------------|
| @login_required<br>def add_to_cart(request,<br>product_id)      | додавання мобільного пристрою до кошика |
| @login_required<br>def show_cart(request)                       | відображення сторінки кошика            |
| @login_required<br>def remove_from_cart(request,<br>product_id) | видалення мобільного пристрою з кошика  |

Для компоненту Dashboard розроблений скрипт view.py, що містить перелік наступних функцій (табл.3.13).

Таблиця 3.13 – Перелік функцій для View в компоненті Dashboard

| Назва функції                                                                       | Призначення                                                         |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------|
| def is_manager(user)                                                                | Перевірка, чи є користувач менеджером                               |
| @user_passes_test(is_manager)<br>@login_required                                    | відображення вебсторінки мобільних пристроїв в режимі менеджера     |
| @user_passes_test(is_manager)<br>@login_required<br>def add_product(request)        | додавання мобільного пристрою в режимі менеджера                    |
| @user_passes_test(is_manager)<br>@login_required<br>def delete_product(request, id) | видалення мобільного пристрою в режимі менеджера                    |
| @user_passes_test(is_manager)<br>@login_required<br>def edit_product(request, id)   | редагування інформації про мобільний пристрій в режимі менеджера    |
| @user_passes_test(is_manager)<br>@login_required<br>def add_category(request)       | додавання категорії мобільного пристрою в режимі менеджера          |
| user_passes_test(is_manager)<br>@login_required<br>def orders(request)              | відображення інформації про замовлення в режимі менеджера           |
| @user_passes_test(is_manager)<br>@login_required<br>def order_detail(request, id)   | відображення детальної інформації про замовлення в режимі менеджера |

Для компоненту Orders розроблений скрипт view.py, що містить перелік наступних функцій (табл.3.14).

Таблиця 3.14 – Перелік функцій для View в компоненті Orders

| Назва функції                                      | Призначення                                               |
|----------------------------------------------------|-----------------------------------------------------------|
| @login_required<br>def create_order(request)       | створення замовлення та відображення відповідної сторінки |
| @login_required<br>def checkout(request, order_id) | оформлення та оплата замовлення                           |
| @login_required<br>def user_orders(request)        | відображення переліку замовлень користувачів              |

Для компоненту Shop розроблений скрипт view.py, що містить перелік наступних функцій (табл.3.15).

Таблиця 3.15 – Перелік функцій для View в компоненті Shop

| Назва функції                                                     | Призначення                                    |
|-------------------------------------------------------------------|------------------------------------------------|
| 1                                                                 | 2                                              |
| def home_page(request)                                            | відображення відповідної сторінки              |
| def product_detail(request, slug)                                 | відображення інформації про мобільний пристрій |
| @login_required<br>def add_to_favorites(request, product_id)      | додавання мобільного пристрою до вибраного     |
| @login_required<br>def remove_from_favorites(request, product_id) | видалення мобільного пристрою з вибраного      |

Продовження таблиці 3.15

| 1                                     | 2                                      |
|---------------------------------------|----------------------------------------|
| @login_required                       | відображення відповідної сторінки з    |
| def favorites(request)                | вподобаними мобільними пристроями      |
| def search(request)                   | пошуку необхідного мобільного пристрою |
| def filter_by_category(request, slug) | фільтрація по категоріям               |

3.3 Схеми алгоритму функціонування системи

Схему алгоритму функціонування інтернет-магазину з продажу мобільних пристроїв наведено на рисунку 3.2.

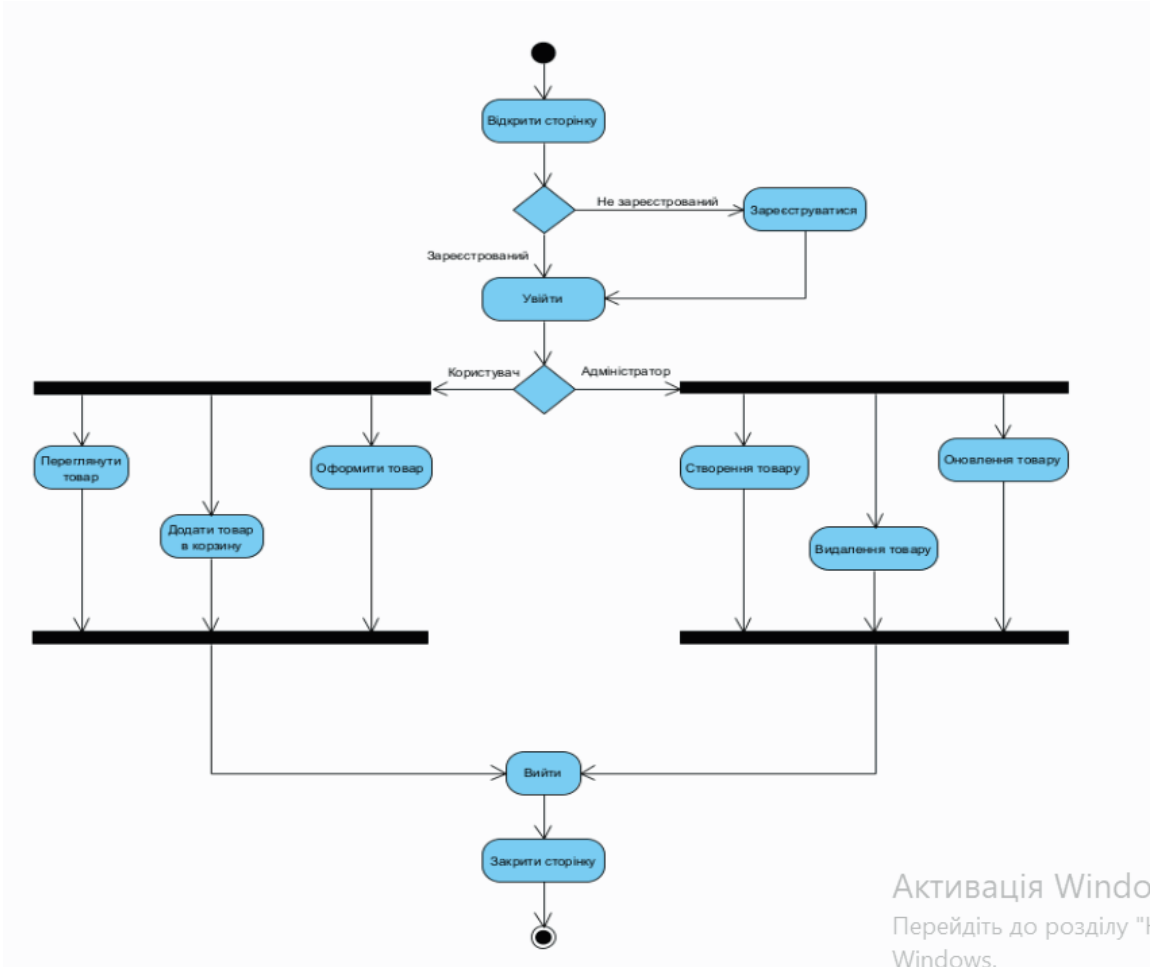


Рисунок 3.2 – Схему алгоритму функціонування інтернет-магазину з продажу мобільних пристроїв

Узагальнений алгоритм функціонування інтернет-магазину з продажу мобільних пристроїв з наступних етапів.

Етап 1. Вхід на сайт: користувач відкриває вебсторінку інтернет-магазину. Якщо користувач менеджер або адміністратор, то перехід на етап 8.

Етап 2. Реєстрація/Авторизація: якщо користувач новий, він реєструється, створюючи обліковий запис. Якщо користувач вже має обліковий запис, він авторизується.

Етап 3. Перегляд товарів: користувач переглядає каталог товарів, використовуючи категорії та фільтри для пошуку потрібних товарів.

Етап 4. Додавання товарів до кошика: користувач додає обрані товари до кошика. Кошик оновлюється в режимі реального часу, показуючи загальну суму покупки.

Етап 5. Оформлення замовлення Користувач переходить до кошика і натискає кнопку "Оформити замовлення". Заповнює форму з інформацією про доставку та вибирає спосіб оплати.

Етап 6. Оплата Користувач здійснює оплату, використовуючи вибраний спосіб оплати (кредитна картка, PayPal, інші платіжні системи). Система оплати обробляє транзакцію та підтверджує оплату.

Етап 7. Підтвердження замовлення Після успішної оплати користувач отримує підтвердження замовлення на екран і на електронну пошту. Замовлення зберігається в базі даних для подальшої обробки.

Етап 8. Обробка замовлення Адміністратор отримує сповіщення про нове замовлення. Замовлення готується до відправки (пакування товарів, створення накладної тощо). Також Адміністратор може створити категорію товару, додати новий товар, видалити товар, оновити інформацію про товар.

Етап 9. Кінець.

### 3.4 Розробка інтерфейсу

Розробка інтерфейсу інтернет-магазину складається з певних етапів, які повинні забезпечити зручність використання, просту навігацію та привабливий дизайн для користувачів. До основних етапів можна віднести – це аналіз цільової аудиторії, визначення основних функціональних можливостей, розробка схеми інтернет-магазину, проектування макету, розробка дизайну, тестування та оптимізація.

Для розробки макету було вирішено обрати програмний інструмент Uizard тому, що він надає можливість розробити 2 проекти дизайну за допомогою ШІ безкоштовно, для цього користувачем вводяться бажані вимоги у вигляді слів, та позначаються відповідні теги. В результаті отримали макети лендінгу основної вебсторінки (рис. 3.3).

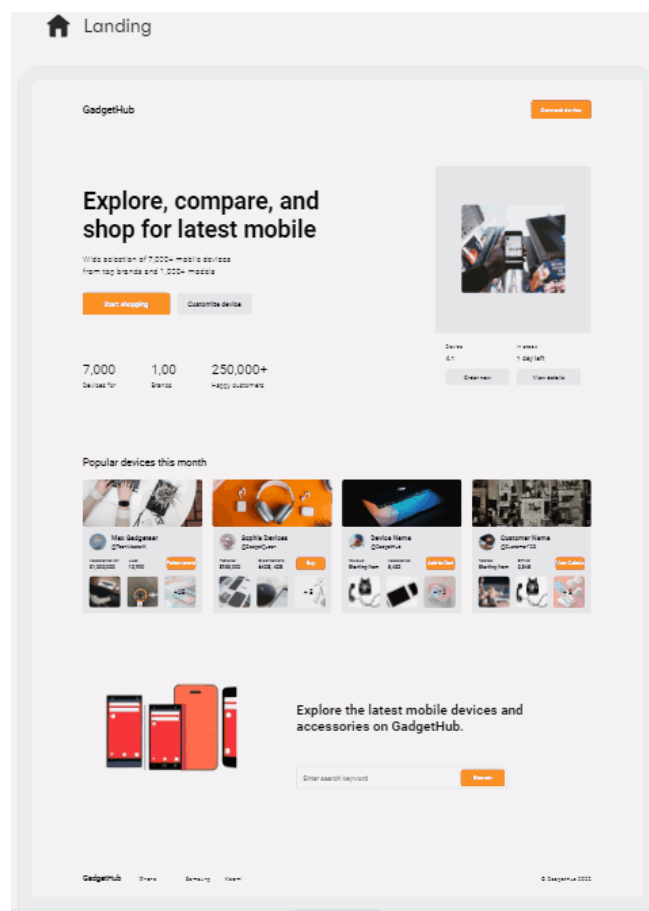


Рисунок 3.3 – Макет лендінг вебсторінки

Вебсторінку для пошуку наведено на рис. 3.4, вебсторінку результатів пошуку – на рис. 3.5.

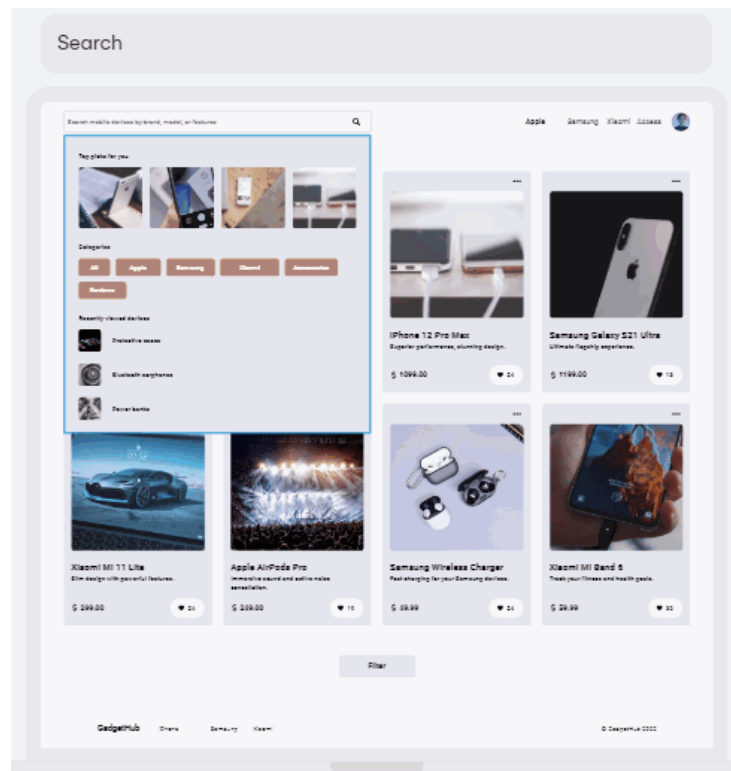


Рисунок 3.4 – Макет вебсторінки для пошуку

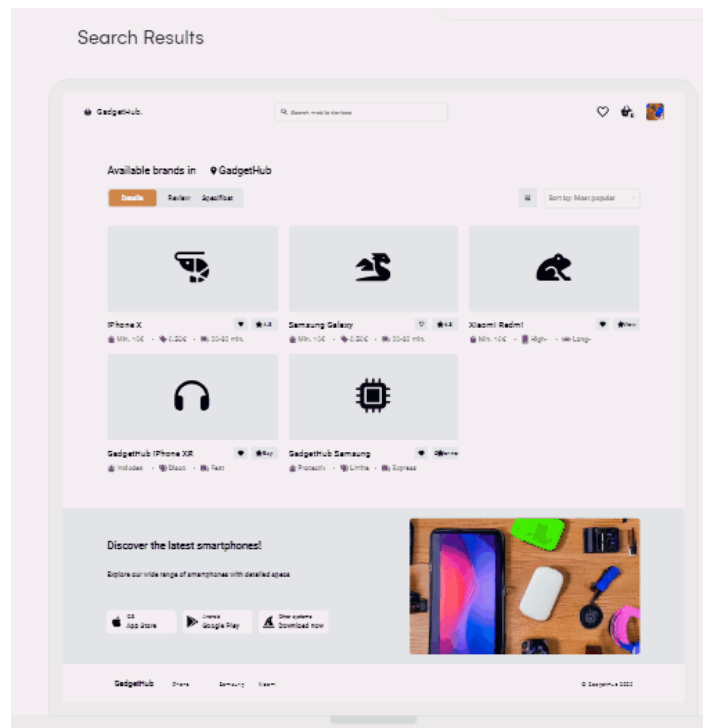


Рисунок 3.5 – Макет вебсторінки результатів пошуку

Вебсторінку з детальною інформацією про мобільний пристрій наведено на рис. 3.6, вебсторінку кошику – на рис. 3.7.

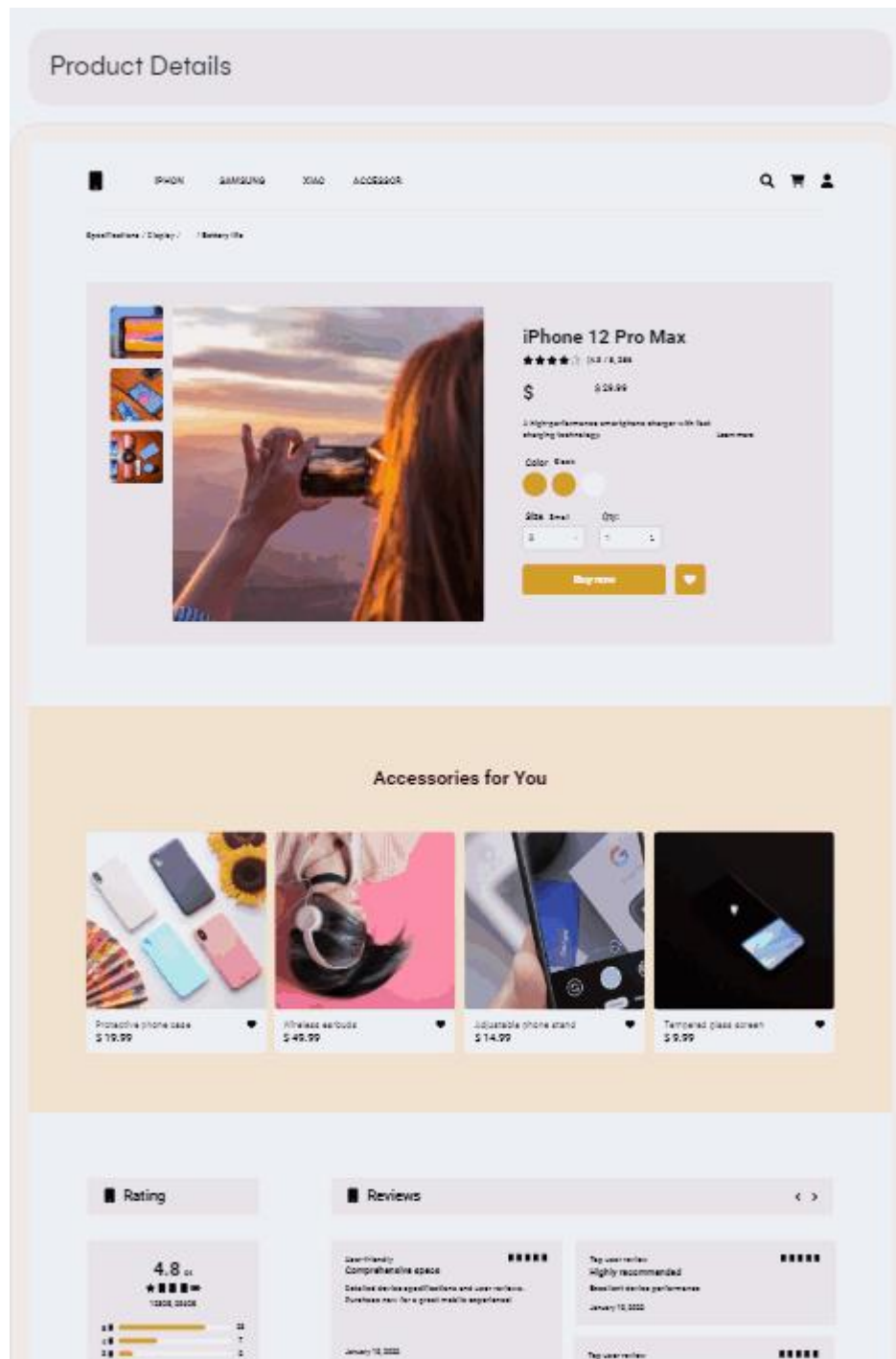


Рисунок 3.6 – Макет вебсторінки з детальної інформацією про мобільний пристрій

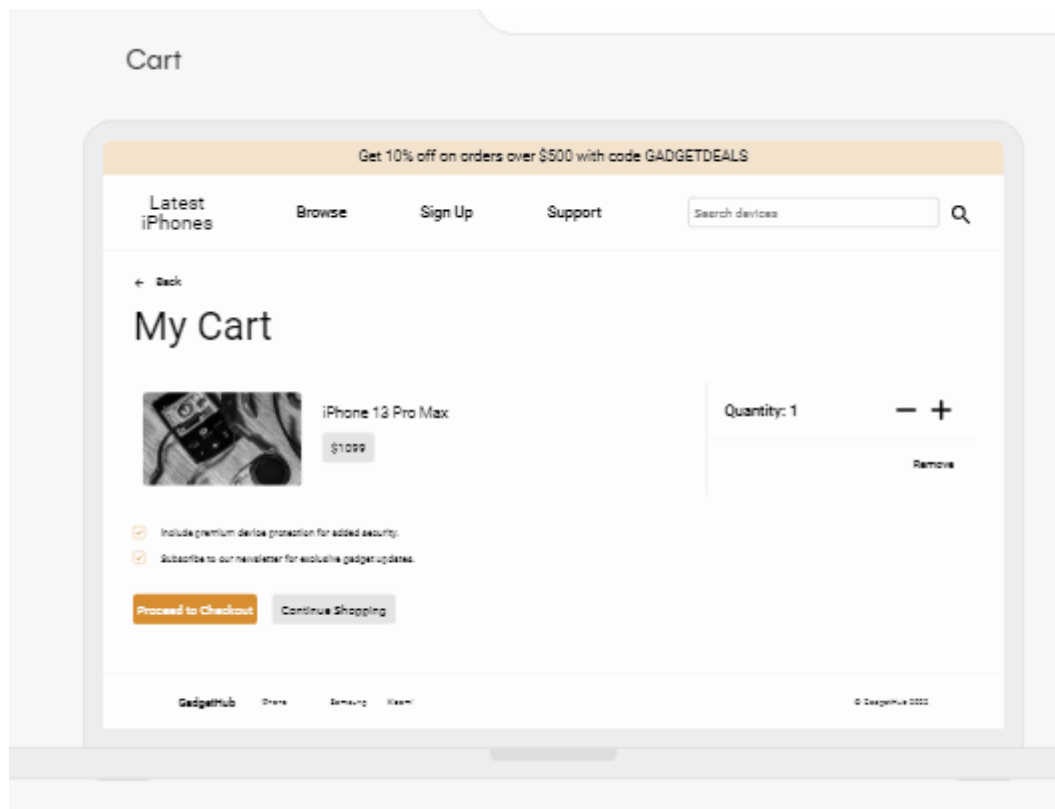


Рисунок 3.7 – Макет вебсторінки кошику

### 3.5 Висновок до третього розділу

В даному розділі було описано розроблені таблиці бази даних, спроектовано діаграму класів, компонентів та пакетів, наведено опис пакетів інтернет-магазину з продажу мобільних пристроїв згідно шаблону проєктування MVT, виконано проєктування макету користувацького інтерфейсу вебзастосунків за допомогою програмного інструменту Uizard.

## **4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ**

### **4.1 Призначення й умови застосування програми**

Розроблений прототип інтернет-магазину з продажу мобільних пристроїв дозволяє користувачам переглянути мобільні пристрої, їх технічні характеристик, вартість, зображення, додати мобільний пристрій до кошика, оформити замовлення, вподобати та зберегти свої улюблені мобільні пристрої.

Можна виділити наступні функціональні вимоги інтернет-магазину з продажу мобільних пристроїв для менеджера(адміністратора):

- можливість введення інформації про нові категорії мобільних пристроїв;
- можливість додавання нових мобільних пристроїв та їх технічних характеристик та зображень;
- перегляд інформації про всі наявні мобільні пристрої;
- редагування та видалення інформації про мобільний пристрій;
- перегляд та редагування інформації про замовлення користувачів.

Розроблене програмне забезпечення повинно надавати наступні функціональні можливості для авторизованого користувача:

- можливість авторизації;
- додавання мобільного пристрою до кошика;
- перегляд історії своїх замовлень;
- можливість вподобати та зберегти вибрані мобільні пристрої;
- редагування та оновлення особистої інформації;
- скидання паролю, використовуючи зареєстровану електронну пошту;
- оформлення та можливість оплати замовлення;
- перегляд галереї мобільних пристроїв;
- перегляд мобільних пристроїв за категоріями;

- пошук мобільного пристрою по ключовим словам.

Для неавторизованого користувача можна виділити наступні функціональні можливості:

- можливість реєстрації;
- перегляд галереї мобільних пристроїв;
- перегляд мобільних пристроїв за категоріями;
- пошук мобільного пристрою по ключовим словам.

#### **4.2 Вимоги до інтерфейсу**

Можна виділити наступні вимоги до інтерфейсу програмного забезпечення інтернет- магазину з продажу мобільних пристроїв:

- візуальне представлення повино бути простим, інтуїтивним та зрозумілим для користувачів;
- меню та основні навігаційні елементи повинні розташовані в верхній частині вебсторінки;
- інформаційні компоненти повинні підтримувати регулярну комунікацію з користувачами або видавати попередження про виконані дії або помилки [21];
- наявність елементів керування ведення інформації: текстові поля, перемикачі, кнопки зі списками або чекбокси [21];
- кожен мобільний пристрій повинен мати окрему сторінку з технічними характеристиками про нього та зображення;
- інформація про мобільні пристрої повинна бути структурована по категоріям та підкатегоріям;
- наявність простої форми замовлення;
- наявність адаптивного дизайну, тобто забезпечувати адаптацію під різні мобільні пристрої.

### 4.3 Умови виконання програми

Мінімальні вимоги до серверу для розгортання серверної частини розробленого програмного забезпечення:

- процесор Intel або AMD із тактовою частотою 2 GHz та більше;
- накопичувач 20 ГБ та більше;
- підтримка СКБД MySQL.

Мінімальні вимоги до програмно-технічних засобів для клієнтської частини наступні:

- персональний комп'ютер або смартфон, або планшетний комп'ютер із встановленою операційною системою;
- підключення до мережі Internet;
- оперативна пам'ять від 4GB;
- встановлений веббраузер.

### 4.4 Розгортання програмного забезпечення

Для розгортання розробленого програмного забезпечення з продажу мобільних пристроїв необхідно виконати наступні етапи: встановлення мови програмування Python та налаштування віртуального оточення, встановлення середовища Django та запуск самого інтернет-магазину.

Розгортання інтернет-магазину розглянемо на прикладі запуску програмного забезпечення під операційну систему Windows.

Для встановлення мови програмування Python необхідно зайти на офіційний сайт Python: [python.org](https://python.org), завантажити останню версію Python для операційної системи Windows. Під час встановлення необхідно обрати опцію "Add Python to PATH".

Далі необхідно створити віртуальне оточення, для цього потрібно відкрити командний рядок (натиснути комбінацію клавіш Win + R, ввести

cmd та натиснути Enter) або оболонку PowerShell(натиснути комбінацію клавіш Win + X та обрати PowerShell).

Наступним кроком необхідно перейти до директорії, де зберігається інтернет-магазин (рис. 4.1).

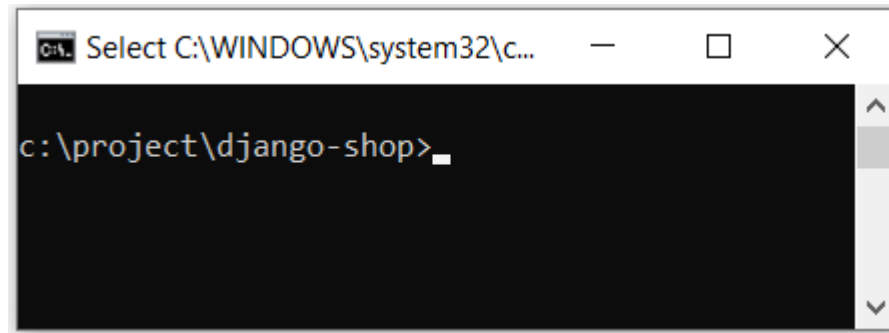


Рисунок 4.1 – Директорія розташування інтернет-магазину

Потім створити віртуальне оточення за допомогою команди: `python -m venv myenv` та активувати віртуальне оточення (рис. 4.2).

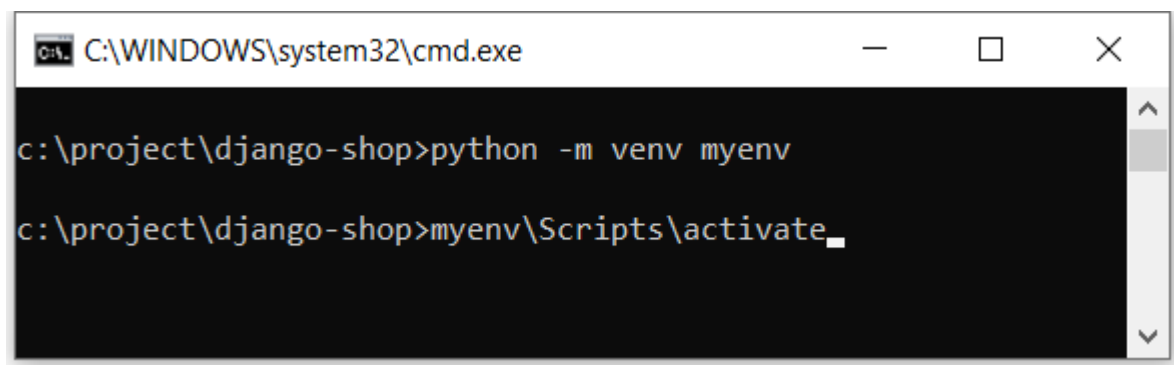


Рисунок 4.2 – Створення та активізація віртуального оточення для командного рядку

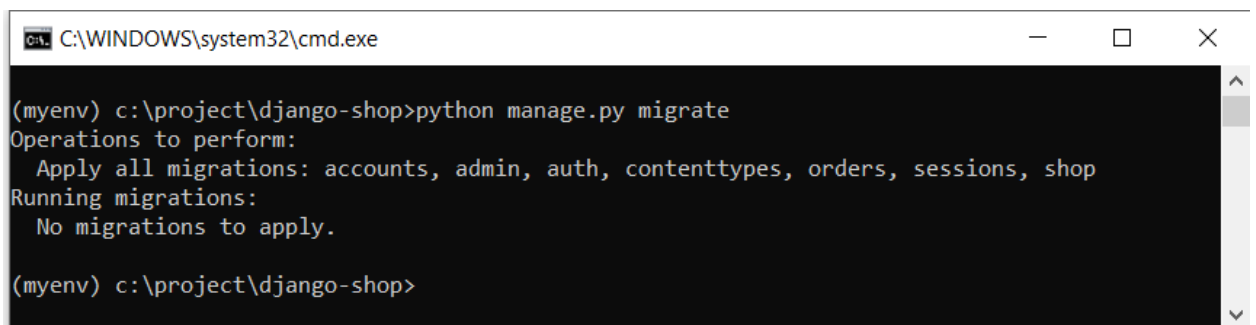
Наступним кроком необхідно підтягнути залежності з файлу `requirements.txt` за допомогою команди `pip install -r requirements.txt` (рис.4.3.)

```
(myenv) c:\project\django-shop>pip install -r requirements.txt
Ignoring backports.zoneinfo: markers 'python_version < "3.9"' don't match your environment
Collecting asgiref==3.4.1
  Using cached asgiref-3.4.1-py3-none-any.whl (25 kB)
Collecting Django==4.0
  Using cached Django-4.0-py3-none-any.whl (8.0 MB)
Collecting django-crispy-forms==1.13.0
  Using cached django_crispy_forms-1.13.0-py3-none-any.whl (122 kB)
Collecting Pillow==9.0.0
  Using cached Pillow-9.0.0-cp310-cp310-win_amd64.whl (3.2 MB)
Collecting sqlparse==0.4.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Collecting tzdata
  Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, Pillow, django-crispy-forms, asgiref, Django
Successfully installed Django-4.0 Pillow-9.0.0 asgiref-3.4.1 django-crispy-forms-1.13.0 sqlparse-0.4.2 tzdata-2024.1

[notice] A new release of pip available: 22.2.2 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Рисунок 4.3 – Встановлення залежностей

Для створення бази даних, таблиць, зміни стовпців, додавання індексів та виконання будь-яких інших операцій з базою даних, необхідних для відображення внесених змін, потрібно виконати команду `python manage.py migrate` (рис.4.4) для застосування міграцій для налаштування бази даних. Файл міграції містить текст програми на мові програмування Python, який представляє зміни в схемі бази даних, такі як створення нових таблиць, зміна існуючих таблиць або додавання нових полів.



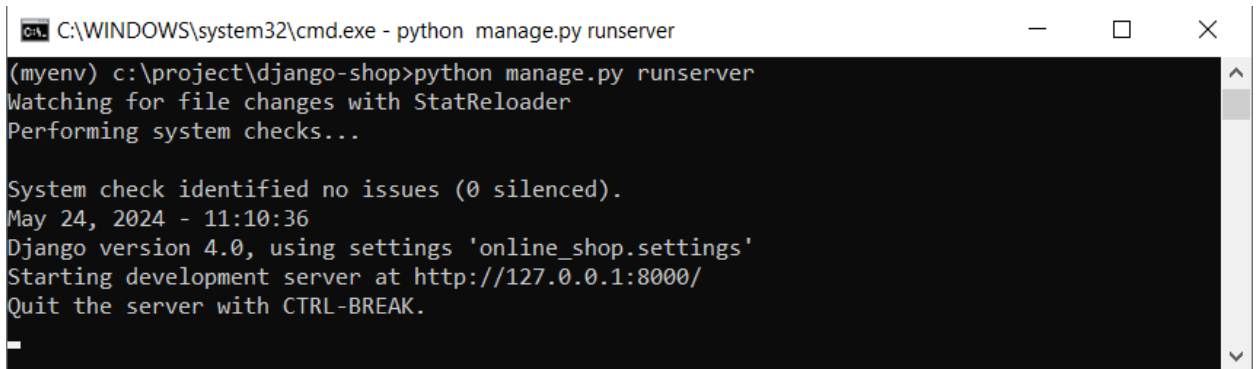
```
C:\WINDOWS\system32\cmd.exe

(myenv) c:\project\django-shop>python manage.py migrate
Operations to perform:
  Apply all migrations: accounts, admin, auth, contenttypes, orders, sessions, shop
Running migrations:
  No migrations to apply.

(myenv) c:\project\django-shop>
```

Рисунок 4.4 – Результат виконання команди `python manage.py migrate`

Для запуску локального серверу розробки необхідно виконати команду `python manage.py runserver` (рис.4.5).



```

C:\WINDOWS\system32\cmd.exe - python manage.py runserver
(myenv) c:\project\django-shop>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 24, 2024 - 11:10:36
Django version 4.0, using settings 'online_shop.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```

Рисунок 4.5 – Запуск локального серверу розробки

Наступним кроком необхідно відкрити браузер та перейти до інтернет-магазину за посиланням <http://127.0.0.1:8000/> (рис.4.6).

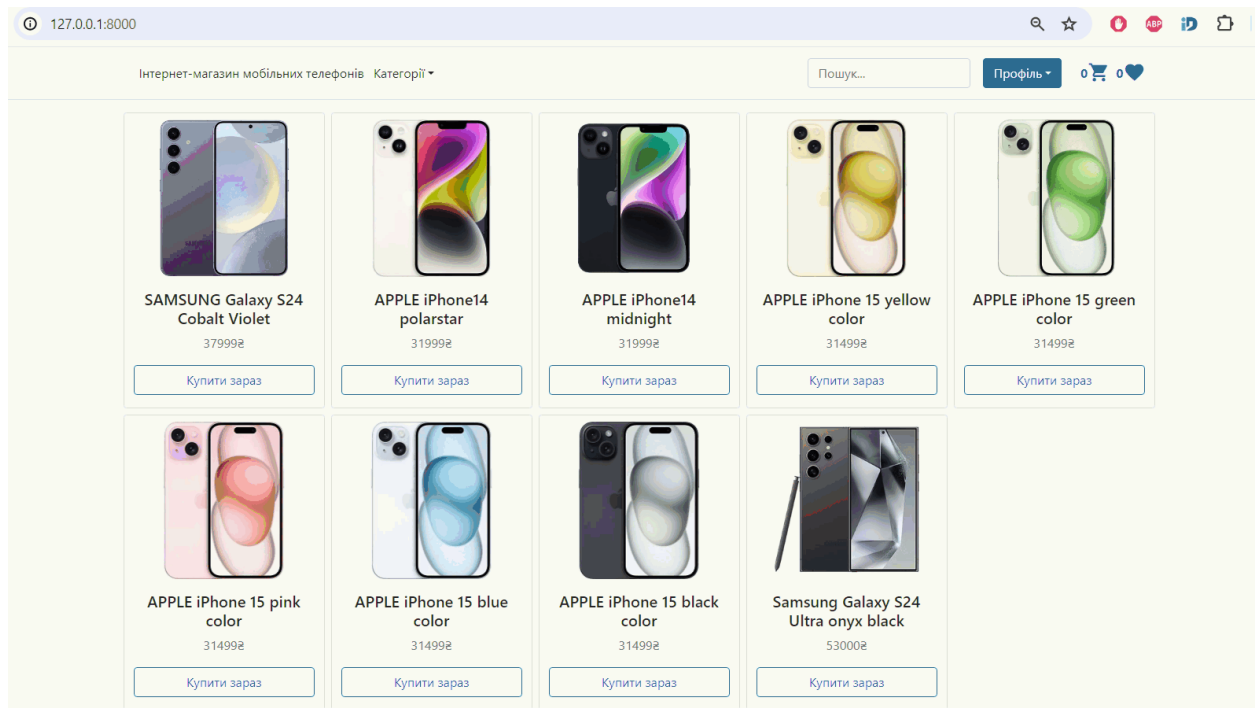


Рисунок 4.6 – Головна сторінка інтернет-магазину

В розробленому інтернет-магазині з продажу мобільних пристроїв реалізовано дві частини графічного інтерфесу користувача: адміністративна панель та авторизованого користувача.

В адміністративній частини менеджер може вносити інформацію про нові категорії мобільних пристроїв, додавати нові мобільні пристрої та їх технічні характеристики, зображення, побачити інформацію про всі наявні

мобільні пристрої, редагувати та видалити інформацію про мобільний пристрій, побачити та редагувати інформацію про замовлення користувачів.

Авторизований користувач може вносити мобільний пристрій до кошика, переглядати історію своїх замовлень, може вподобати та зберегти вибрані мобільні пристрої, редагувати та оновлювати особисту інформацію, скинути свій пароль, використовуючи зареєстровану електронну пошту.

#### 4.5 Методика та результати тестування

Під час тестування було проведено наступні методики роботи користувача з системою: відстеження коректності роботи інтернет-магазину та базою даних, відстеження обробки помилкових дій користувача, обробки стандартних дій користувача: використання навігаційних елементів, кнопок та ін.

Головна сторінка інтернет-магазину з продажу мобільних пристроїв наведена на рис. 4.6.

Сторінка авторизації користувача показана на рисунку 4.7.

Інтернет-магазин мобільних телефонів Категорії ▾

Пошук...

Авторизація Sign-up

### Авторизація

Email\*

email

Password\*

password

Відправити

Забули пароль?

Ви новий користувач? Створити акаунт

Рисунок 4.7 – Сторінка авторизації користувача

Якщо у користувача відсутній акаунт в інтернет-магазині, то можна створити акаунт, натиснувши на «Створити акаунт». Далі відкриється сторінка створення нового акаунту користувача, яка показана на рисунку 4.8.

← → ↻ 127.0.0.1:8000/accounts/register/ ☆ 🔍 🛒 ❤️

Інтернет-магазин мобільних телефонів Категорії ▾ Пошук... Авторизація Sign-up

### Створити акаунт

Email\*  
email

Full name\*  
full name

Password\*  
password

**Відправити**

Вже маєте обліковий запис? [Авторизація](#)

Рисунок 4.8 – Сторінка створення нового акаунту користувача

Введемо електрону пошту, повне ім'я, пароль і натиснемо кнопку «Відправити» (рис. 4.9.). Як видно з рисунку, відбувається валідація даних, якщо введена некоректно електрона пошта.

← → ↻ 127.0.0.1:8000/accounts/register/ 🌐 ☆ 🔍 🛒 ❤️

Інтернет-магазин мобільних телефонів Категорії ▾ Пошук... Авторизація Sign-up

### Створити акаунт

Email\*  
test2

❗ Електронна адреса має містити знак "@". В електронній адресі "test2" знака "@" немає.

Full name\*  
.....

Password\*  
....

**Відправити**

Вже маєте обліковий запис? [Авторизація](#)

Рисунок 4.9 – Сторінка валідації даних

Введемо коректну електрону пошту і авторизуємося в інтернет-магазин, при цьому введемо неправильний пароль (рис.4.10). На сторінці авторизації можна побачити опцію відновлення пароля, у випадку коли користувач забув або загубив пароль.

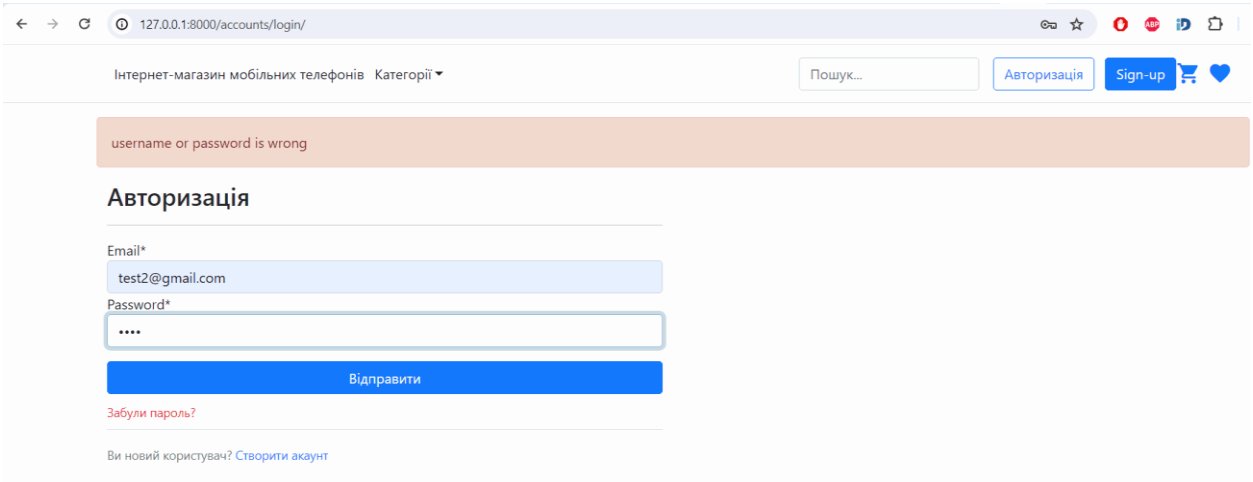


Рисунок 4.10 – Сторінка відображення помилки у випадку введення некоректного паролю

Після авторизації користувач може переглянути кошик, історію своїх замовлень, може вподобати та зберегти вибрані мобільні пристрої, редагувати та оновлювати особисту інформацію, скинути свій пароль, використовуючи зареєстровану електронну пошту (рис.4.11).

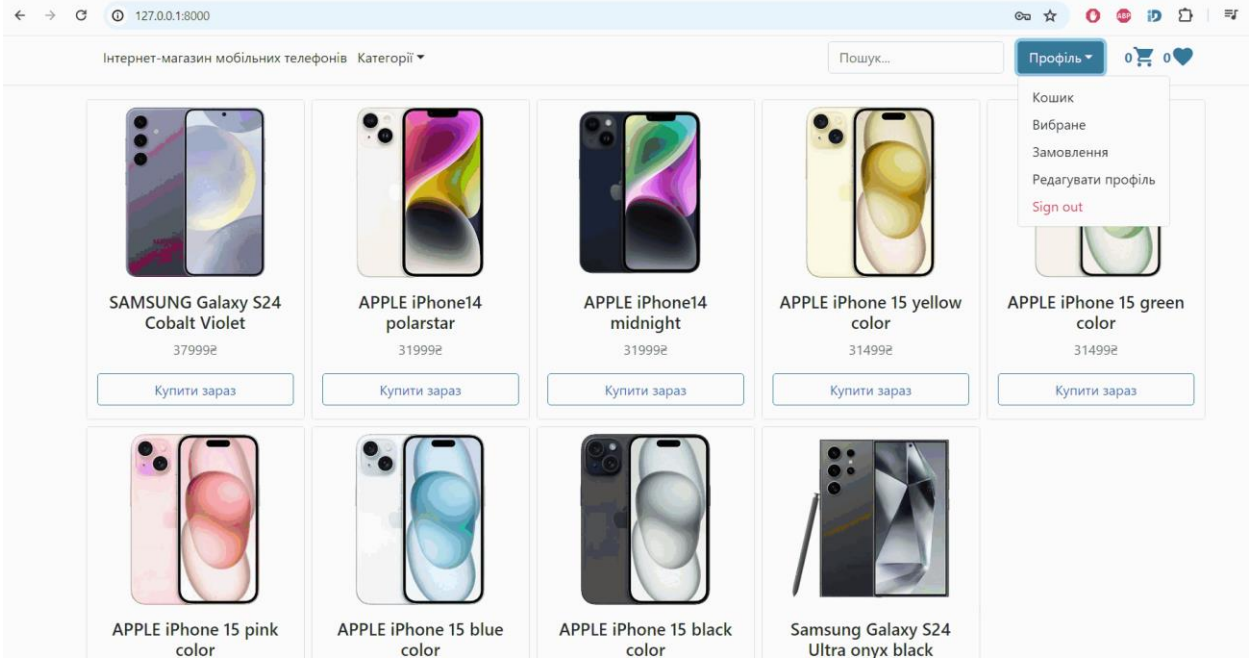


Рисунок 4.11 – Сторінка профілю авторизованого користувача

Виберемо будь-який мобільний пристрій та додамо його до кошика, обрати кількість та натиснувши на кнопку «Додати до кошика» (рис. 4.12).

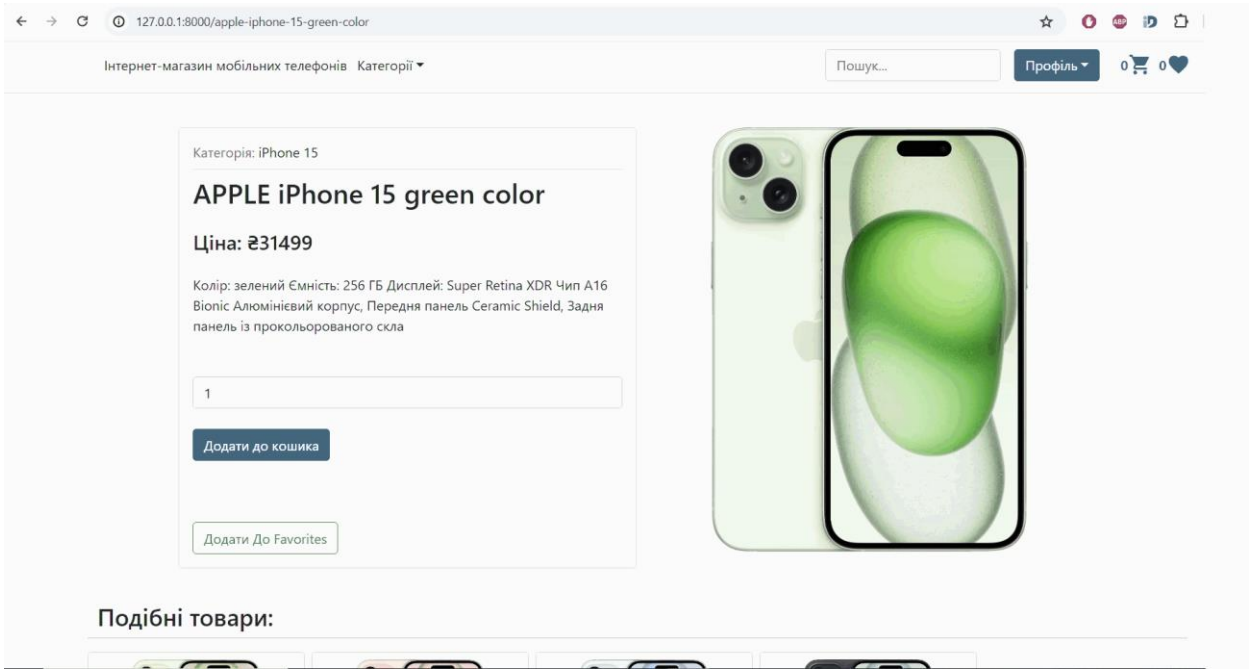


Рисунок 4.12 – Сторінка додавання мобільного пристрою до кошика

Користувач побачить повідомлення, що мобільний пристрій успішно додано до кошику. Перейдемо до кошика (рис.4.13).

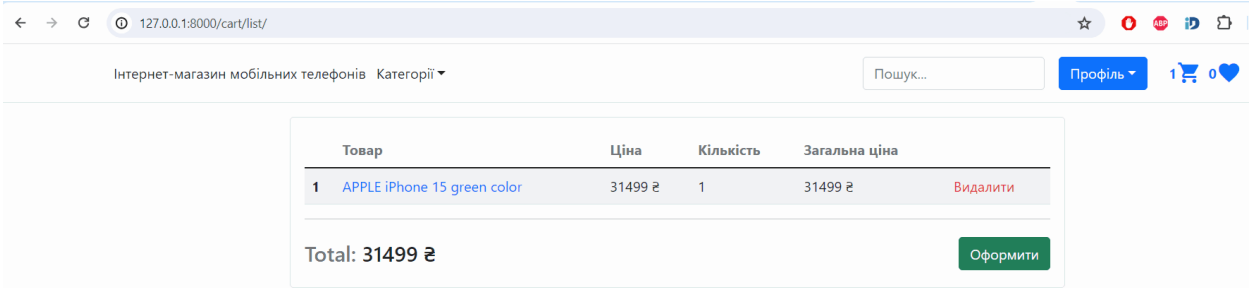


Рисунок 4.13 – Сторінка оформлення замовлення

Натиснемо кнопку «Оформити» і перейдемо до сторінки замовлення користувача (рис.4.14).

Також авторизований користувач може вподобати та зберегти вибрані мобільні пристрої в «Вподобання» (рис.4.15).

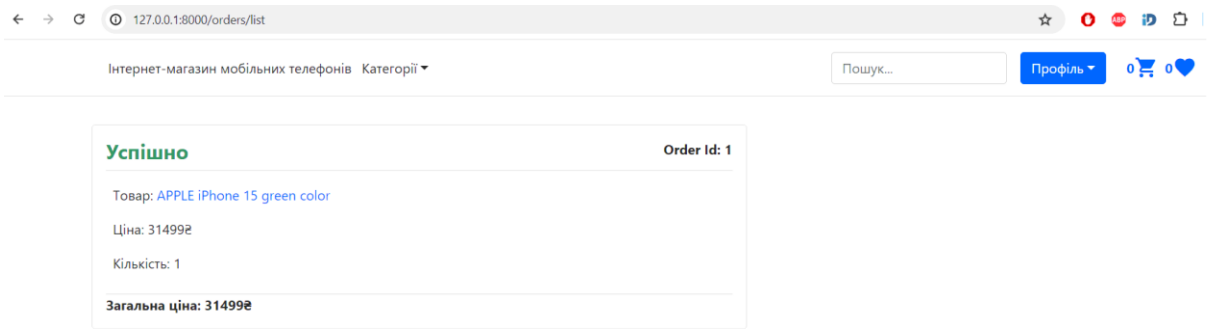


Рисунок 4.14 – Сторінка замовлень користувача

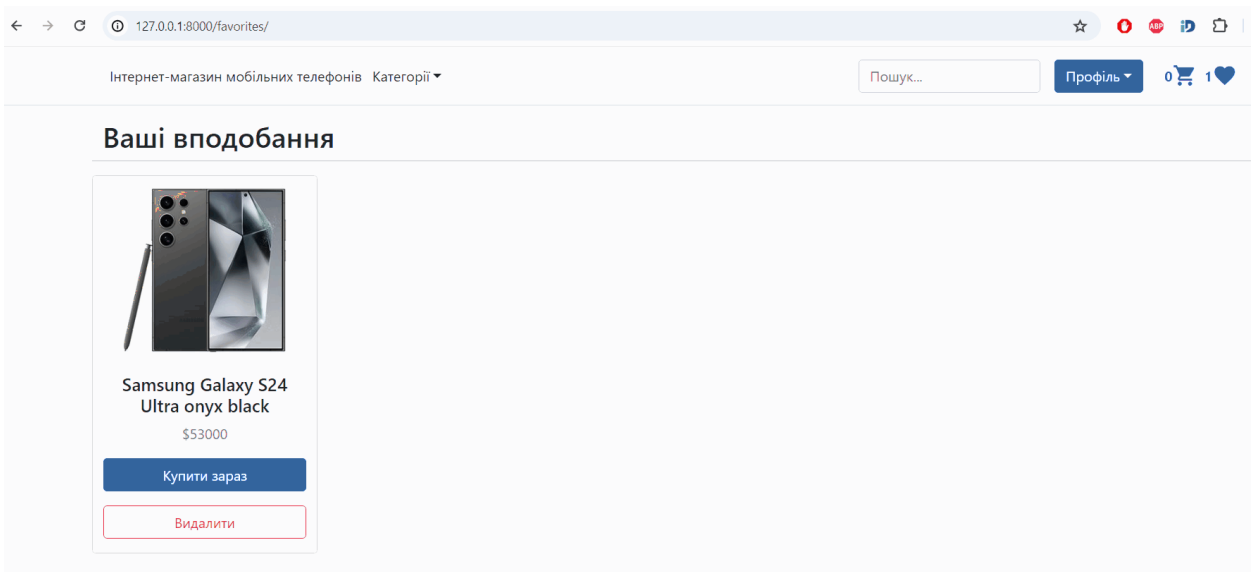


Рисунок 4.15 – Додавання вибраного мобільного пристрою в «Ваші вподобання»

В адміністративній частині можна побачити перелік мобільних пристроїв в інтернет-магазині (рис. 4.16).

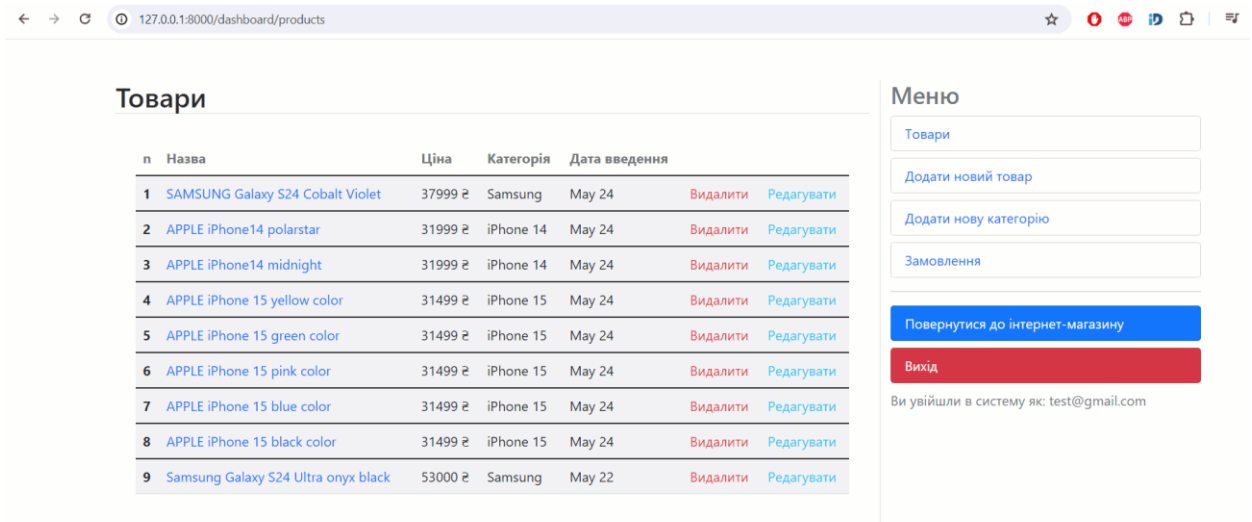


Рисунок 4.16 – Сторінка адміністративної частини

В адміністративній частини менеджер може додавати нову категорію мобільних пристроїв, нові мобільні пристрої та їх технічні характеристики, зображення, переглянути всі наявні мобільні пристрої, редагувати та видалити інформацію про мобільний пристрій, переглянути та редагувати інформацію про замовлення користувачів.

До найпоширеніших методів тестування вебзастосунків відносяться функціональне тестування, яке включає перевірку роботи функціональних можливостей програми, ad-hoc тестування, тестування візуальної частини, таке як верстка, доступність, працездатність чи відсутність скриптів на JavaScript, а також тестування контенту та інші [45].

Для перевірки працездатності інтернет-магазину були використані функціональне тестування та тестування візуальної частини.

Під час тестування можна побачити, що візуально всі елементи інтерфейсу, розділи меню, категорії інтернет-магазину відображені коректно, помилки налаштування відсутні. Також було проведено тестування відображення інтернет-магазину на екранах різних розмірів та з різною роздільною здатністю. Це дозволило оцінити, як дизайн інтернет-магазину виглядатиме на комп'ютерах потенційних відвідувачів, включаючи такі

питання, як наявність горизонтальної смуги прокрутки на малих екранах та стабільність розміщення елементів при різних роздільних здатностях тощо.

Також перевірка працездатності інтернет-магазину відбувалась на мобільному пристрої. Для цього використовувалися функціональні можливості веббраузера Google Chrome, натиснувши кнопку F12, переходимо до режиму розробника (рис. 4.17) та обираємо мобільний пристрій для тестування відображення сторінок інтернет-магазину (рис. 4.18).

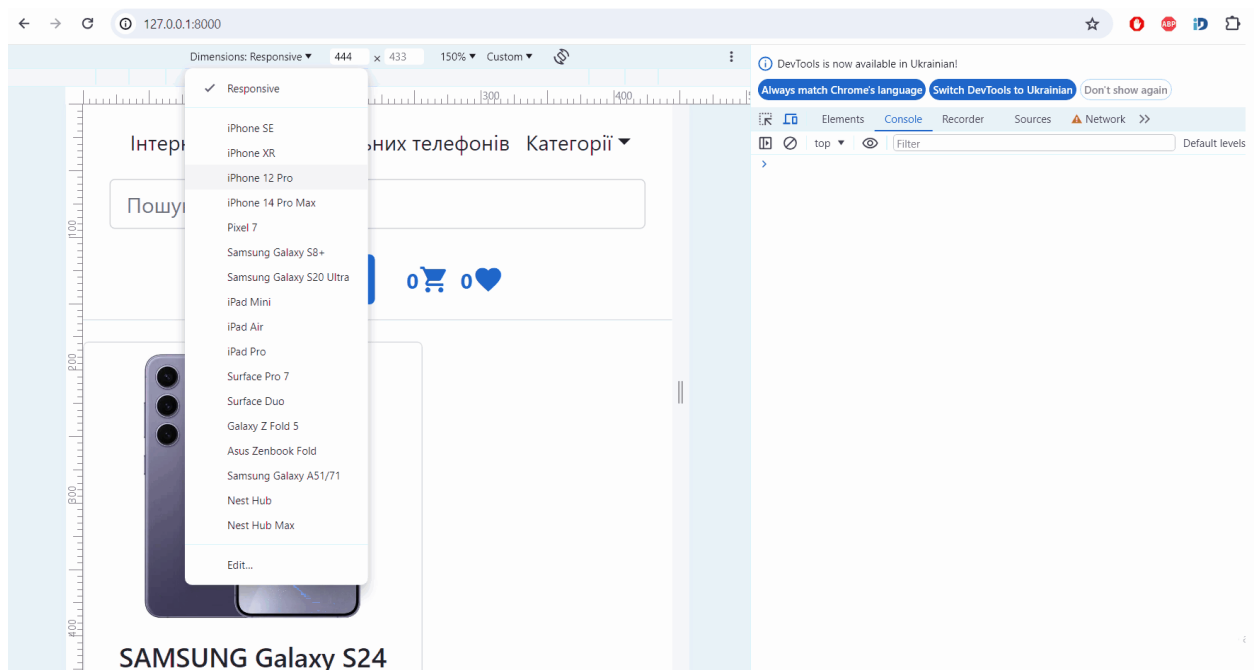


Рисунок 4.17 – Режим розробника в Google Chrome

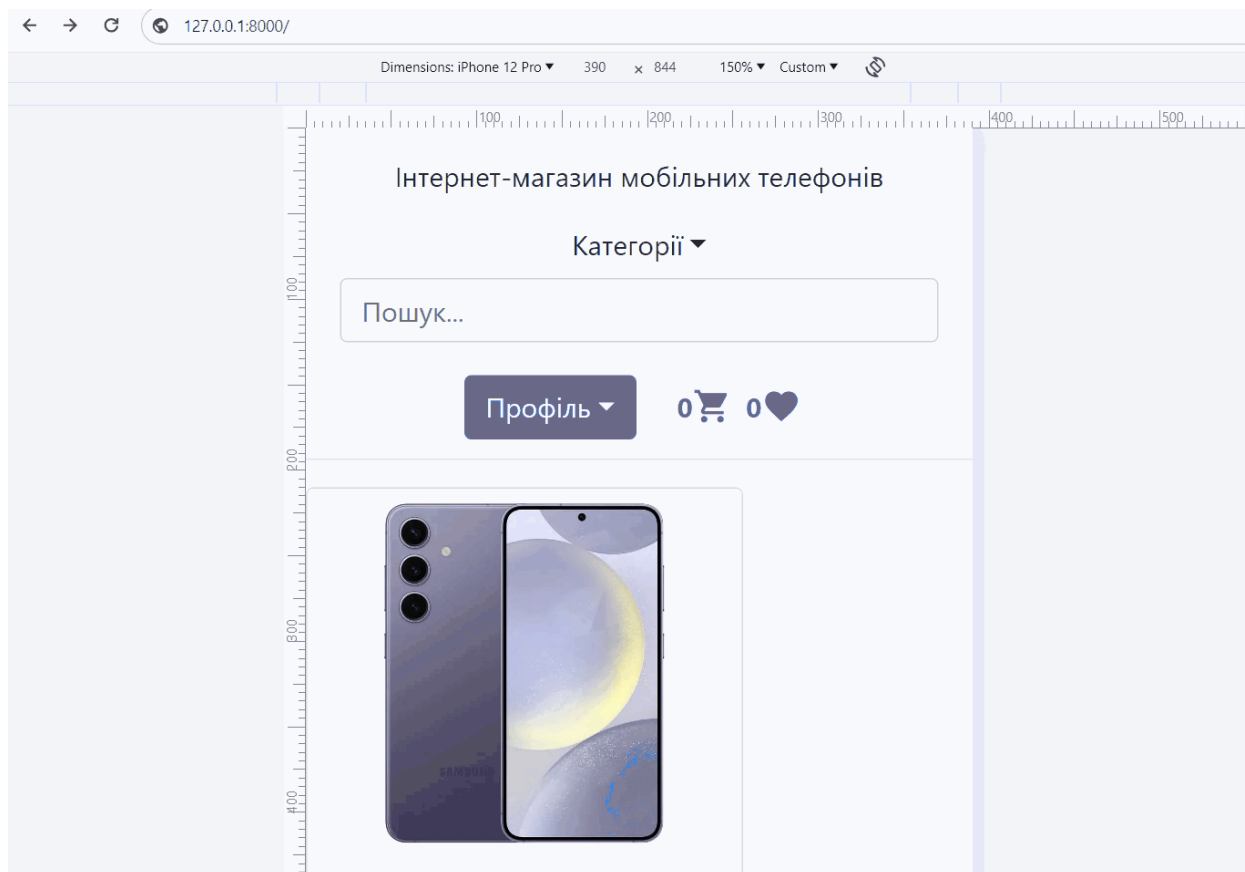


Рисунок 4.18 – Зовнішній вигляд головної сторінки інтернет-магазину під мобільний пристрій iPhone12

Під час тестування сторінок інтернет-магазину під мобільний пристрій iPhone12 можна побачити, що візуально всі елементи інтерфейсу, розділи меню, категорії інтернет-магазину відображені коректно, помилки налаштування відсутні. Перегляд програмного забезпечення адаптовано під мобільний пристрій.

#### 4.6 Висновок до четвертого розділу

В даному розділі наведені функціональні вимоги інтернет-магазину з продажу мобільних пристроїв для менеджера(адміністратора), для авторизованого та неавторизованого користувачів, вимоги до інтерфейсу, опис розгортання програмного забезпечення, методика роботи з інтернет-магазином та виконано тестування програмного забезпечення.

Подальші напрямки розвитку розробленого програмного забезпечення полягають в розширенні функціональності, а саме: додавання брендбуку інтернет-магазину згідно зпроєктованого макету в підрозділі 3.3, можливості залишити відгук, форми зворотного зв'язку, платіжних систем, варіантів доставки, кредитних пропозицій, акцій та знижок, інформацію про програми лояльності, інструментів пошукової оптимізації, розгортання та викладення розробленого інтернет-магазину на хостингу та ін.

## ВИСНОВКИ

В результаті виконання дипломної кваліфікаційної роботи було розроблено прототип інтернет-магазину з продажу мобільних пристроїв, що об'єднує серверну частину, реалізовану з використанням мови програмування Python та фреймворку Django, систему керування базами даних MySQL, а також клієнтську частину, розроблену за допомогою інструментів штучного інтелекту для проектування користувацького інтерфейсу.

Практична цінність роботи полягає в тому, що впровадження розробленого кросплатформного програмного забезпечення дозволить підвищити ефективність введення обліку продажів мобільних пристроїв в інтернет-магазині.

В подальшій роботі можливо розширити функціонал програмного забезпечення шляхом додавання брэндбуку інтернет-магазину, можливості залишити відгук, форми зворотного зв'язку, платіжних систем, варіантів доставки, кредитних пропозицій, акцій та знижок, інформації про програми лояльності, інструментів пошукової оптимізації, а також розгортання та викладення розробленого інтернет-магазину на хостингу.

Результати роботи було представлено на щорічній науково-практичній конференції викладачів, науковців, молодих учених, аспірантів та студентів НУЗП «Тиждень науки-2024» [46].

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. В Україні оживають інтернет-магазини, активність йде на рекорд: названо топ товарів та ключові тенденції [Електрон. ресурс]. – Режим доступу: <https://www.obozrevatel.com/ukr/ekonomika-glavnaya/economy/v-ukraini-ozhivayut-internet-magazini-aktivnist-jde-na-rekord-nazvano-top-tovariv-ta-klyuchovi-tendentsii.htm>.
2. Особливості електронної комерції та стан її розвитку в сучасних економічних умовах України [Електрон. ресурс]. – Режим доступу: [http://www.economy.nayka.com.ua/pdf/12\\_2018/76.pdf](http://www.economy.nayka.com.ua/pdf/12_2018/76.pdf).
3. Інтернет-магазин [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Інтернет-магазин>.
4. Електронна комерція [Електрон. ресурс]. – Режим доступу: [https://uk.wikipedia.org/Електронна\\_комерція](https://uk.wikipedia.org/Електронна_комерція).
5. Шалева, О.І. Електронна комерція - 3.1. Поняття Internet-магазину та особливості його функціонування [Електрон. ресурс] / О.І. Шалева. – Режим доступу: <http://westudents.com.ua/glavy/27092-31-ponyattya-Internet-magazinu-ta-osoblivost-yogo-funktsonuvannya.html>.
6. Yabluka, мережа оригінальної техніки [Електрон. ресурс]. – Режим доступу: <https://yabloki.ua/>.
7. Інтернет-магазин «Ябко» [Електрон. ресурс]. – Режим доступу: <https://jabko.ua/>.
8. Інтернет-магазин «Цитрус» [Електрон. ресурс]. – Режим доступу: <https://www.ctrс.com.ua/>.
9. Розділ 2. Основи UML [Електрон. ресурс]. – Режим доступу: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-basics.html>.
10. Що таке CMS сайту [Електрон. ресурс]. – Режим доступу: <https://hostiq.ua/wiki/ukr/cms/>.
11. WordPress [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/WordPress>.

12. OpenCart [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/OpenCart>.
13. На якій платформі зробити сайт, яку CMS вибрати для сайту? [Електрон. ресурс]. – Режим доступу: <https://webbuilding.pro/ukr/blog/site-platforms>.
14. Magento [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Magento>.
15. Для чого потрібен фреймворк і як його вибрати [Електрон. ресурс]. – Режим доступу: <https://owlweb.com.ua/news/cikavevidsov/dlya-chogo-potriben-freymvork-i-yak-yogo-vibrati>.
16. Модель-вид-контролер [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Модель-вид-контролер>.
17. Побудова динамічних вебзастосунків за допомогою MVC [Електрон. ресурс]. – Режим доступу: <https://blog.ithillel.ua/articles/building-dynamic-web-applications-using-mvc>.
18. Working Structure of Django MTV Architecture [Electronic resource]. – Access mode: <https://towardsdatascience.com/working-structure-of-django-mtv-architecture-a741c8c64082>.
19. Web Development with Django: A definitive guide to building modern Python web applications using Django 4 / [Ben Shaw, Saurabh Badhwar, Chris Guest, Bharath Chandra K. S.]. – Birmingham: Packt Publishing, 2023. – 764 p.
20. Model-View-ViewModel [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Model-View-ViewModel>.
21. PHP [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/PHP>.
22. Python [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Python>.
23. JavaScript [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/JavaScript>.

24. Visual Studio Code [Електрон. ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://uk.wikipedia.org/wiki/Visual_Studio_Code).
25. PyCharm [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/PyCharm>.
26. Atom (текстовий редактор) [Електрон. ресурс]. – Режим доступу: [https://uk.wikipedia.org/wiki/Atom\\_\(текстовий\\_редактор\)](https://uk.wikipedia.org/wiki/Atom_(текстовий_редактор)).
27. Корнієнко С. К. Системи баз даних: організація та проектування : навч. посібник / С. К. Корнієнко. – Запоріжжя : ЗНТУ, 2006. – 252 с.
28. Мамалига Н. Є. Система керування базами даних в сучасних умовах ІТ-індустрії / Н. Є. Мамалига, А. І. Катаєва // Вісник студентського наукового товариства ДонНУ імені Василя Стуса. – 2021. – №13(1). – С.275-279.
29. MariaDB [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/MariaDB>.
30. PostgreSQL [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/PostgreSQL>.
31. Flask [Electronic resource]. – Access mode: <https://flask.palletsprojects.com/en/3.0.x/>.
32. Flask [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Flask>.
33. Що таке Django? [Електрон. ресурс]. – Режим доступу: <https://tutorial.djangogirls.org/uk/django/>.
34. Підручник Django для початківців: особливості, Archітектура та історія [Електрон. ресурс]. – Режим доступу: <https://www.guru99.com/uk/django-tutorial.html>.
35. Інструменти веб-дизайну [Електрон. ресурс]. – Режим доступу: <https://lemon.school/blog/instrumenty-veb-dyzajnu>.
36. Design, prototype, and share with Adobe XD [Electronic resource]. – Access mode: <https://helpx.adobe.com/xd/help/adobe-xd-overview.html>.

37. Figma [Електрон. ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Figma>.
38. WIX [Електрон. ресурс]. – Режим доступу: <https://uk.wix.com/>.
39. How Wix ADI elevates website creation for everyone [Electronic resource]. – Access mode: <https://www.wix.com/blog/wix-artificial-design-intelligence>.
40. Visualize product ideas fast and easy with AI [Electronic resource]. – Access mode: <https://uizard.io/>.
41. We make building an app so easy, anyone can do it [Electronic resource]. – Access mode: <https://www.builder.ai/>.
42. Connolly, T. M. Database Systems: A Practical Approach to Design, Implementation, and Management/ Connolly, T. M., Begg, C. E. – London: Pearson, 2004. – 211 p.
43. DrawSQL [Electronic resource]. – Access mode: <https://drawsql.app/>.
44. Інтерфейс Користувача (UI): Значення, складові та вплив на продажі інтернет-магазину [Електрон. ресурс]. – Режим доступу: <https://wezom.com.ua/ua/blog/chto-takoe-ui-i-kak-polzovatelskij-interfejs-vliyaet-na-prodazhi-internet-magazina>.
45. Тестування веб-проектів: основні етапи та поради [Електрон. ресурс]. – Режим доступу: <https://qalight.ua/baza-znaniy/testuvannya-veb-proektiv-osnovni-etapi-ta-poradi/>.
46. Черніговцев М.О. Розробка програмного забезпечення інтернет-магазину з продажу мобільних пристроїв / М.О. Черніговцев, А.В. Пархоменко // Тиждень науки-2024. Факультет комп'ютерних наук і технологій: щоріч. наук.-практ. конф., 15-19 квітня 2024 р.: тези доп. / Редкол.: В. Шаломеев (відпов.ред.) Електрон.дані. – Запоріжжя: НУ «Запорізька політехніка», 2024 (в друці).

**ДОДАТОК А**  
**Текст програми**

### A.1 Текст файла view.py компоненты Dashboard

```
from django.shortcuts import render, redirect, get_object_or_404
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from django.contrib.auth.decorators import user_passes_test
from django.http import Http404

from shop.models import Product
from accounts.models import User
from orders.models import Order, OrderItem
from .forms import AddProductForm, AddCategoryForm, EditProductForm

def is_manager(user):
    try:
        if not user.is_manager:
            raise Http404
        return True
    except:
        raise Http404

@user_passes_test(is_manager)
@login_required
def products(request):
    products = Product.objects.all()
    context = {'title': 'Товары', 'products': products}
    return render(request, 'products.html', context)

@user_passes_test(is_manager)
@login_required
```

```

def add_product(request):
    if request.method == 'POST':
        form = AddProductForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            messages.success(request, 'Товар успішно додано!')
            return redirect('dashboard:add_product')
    else:
        form = AddProductForm()
    context = {'title': 'Додати товар', 'form': form}
    return render(request, 'add_product.html', context)

```

```

@user_passes_test(is_manager)
@login_required
def delete_product(request, id):
    product = Product.objects.filter(id=id).delete()
    messages.success(request, 'Товар видалено!', 'success')
    return redirect('dashboard:products')

```

```

@user_passes_test(is_manager)
@login_required
def edit_product(request, id):
    product = get_object_or_404(Product, id=id)
    if request.method == 'POST':
        form = EditProductForm(request.POST, request.FILES, instance=product)
        if form.is_valid():
            form.save()
            messages.success(request, 'Товар оновлено', 'success')
            return redirect('dashboard:products')
    else:

```

```

    form = EditProductForm(instance=product)
    context = {'title': 'Редагувати товар', 'form':form}
    return render(request, 'edit_product.html', context)

```

```

@user_passes_test(is_manager)
@login_required
def add_category(request):
    if request.method == 'POST':
        form = AddCategoryForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Категорія успішно додана!')
            return redirect('dashboard:add_category')
        else:
            form = AddCategoryForm()
    context = {'title':'Додати категорію', 'form':form}
    return render(request, 'add_category.html', context)

```

```

@user_passes_test(is_manager)
@login_required
def orders(request):
    orders = Order.objects.all()
    context = {'title':'Замовлення', 'orders':orders}
    return render(request, 'orders.html', context)

```

```

@user_passes_test(is_manager)
@login_required
def order_detail(request, id):
    order = Order.objects.filter(id=id).first()
    items = OrderItem.objects.filter(order=order).all()

```

```
context = {'title': 'Деталі замовлення', 'items': items, 'order': order}
return render(request, 'order_detail.html', context)
```

## A.2 Текст файлу form.py компоненту Dashboard

```
from django import forms
from django.forms import ModelForm

from shop.models import Product, Category

class AddProductForm(ModelForm):
    class Meta:
        model = Product
        fields = ['category', 'image', 'title', 'description', 'price']

    def __init__(self, *args, **kwargs):
        super(AddProductForm, self).__init__(*args, **kwargs)
        for visible in self.visible_fields():
            visible.field.widget.attrs['class'] = 'form-control'

class AddCategoryForm(ModelForm):
    class Meta:
        model = Category
        fields = ['title', 'sub_category', 'is_sub']

    def __init__(self, *args, **kwargs):
        super(AddCategoryForm, self).__init__(*args, **kwargs)
        self.fields['is_sub'].widget.attrs['class'] = 'form-check-input'
        self.fields['sub_category'].widget.attrs['class'] = 'form-control'
```

```
self.fields['title'].widget.attrs['class'] = 'form-control'
```

```
class EditProductForm(ModelForm):
```

```
    class Meta:
```

```
        model = Product
```

```
        fields = ['category', 'image', 'title', 'description', 'price']
```

```
    def __init__(self, *args, **kwargs):
```

```
        super(EditProductForm, self).__init__(*args, **kwargs)
```

```
        for visible in self.visible_fields():
```

```
            visible.field.widget.attrs['class'] = 'form-control'
```

### A.3 Текст файлы dashboard.html компоненты Dashboard

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
    <meta charset="UTF-8">
```

```
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
        <meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
                                <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"                                integrity="sha384-
1BmE4kWBq78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jI
W3" crossorigin="anonymous">
```

```
    <title>{{title}}</title>
```

```
</head>
```

```
<body>
```

```
    <div class="row mt-5">
```

```

<div class="col-md-1"></div>
<!-- content -->
<div class="col-md-7">
  <h2 class="border-bottom text-capitalize">{{ title }}</h2>
  <div class="p-4">
    <!-- messages -->
    {% if messages %}
    {% for message in messages %}
      <div class="alert alert-{{ message.tags }}">{{ message
}}</div>

    {% endfor %}
    {% endif %}
    {% block content %}{% endblock %}
  </div>
</div>
<!-- sidebar -->
<div class="col-md-3 border-start">
  <h3 class="text-muted">Меню</h3>
  <a class="text-decoration-none" href="{% url 'dashboard:products'
%}">

    <li class="rounded list-group-item text-primary">Товари</li>
  </a>

    <a class="text-decoration-none" href="{% url
'dashboard:add_product' %}">

    <li class="rounded list-group-item mt-2 text-primary">Додати
новий товар</li>
  </a>

    <a class="text-decoration-none" href="{% url
'dashboard:add_category' %}">

```

```

        <li class="rounded list-group-item mt-2 text-primary">Додати
нову категорію</li>
    </a>
    <a class="text-decoration-none" href="{% url 'dashboard:orders'
%}">
        <li class="rounded list-group-item mt-2 text-
primary">Замовлення</li>
    </a>
    <hr>
    <a class="text-decoration-none" href="{% url 'shop:home_page'
%}">
        <li class="rounded list-group-item bg-primary mt-2 text-
white">Повернутися до інтернет-магазину</li>
    </a>
    <a class="text-decoration-none" href="{% url 'accounts:user_logout'
%}">
        <li class="rounded list-group-item bg-danger mt-2 text-
white">Вихід</li>
    </a>
    <div class="text-muted mt-2">Ви увійшли в систему як: {{
request.user.email }}</div>
</div>
<div class="col-md-1 "></div>
</div>
</body>
</html>

```

**ДОДАТОК Б**  
**Слайди презентації**

Національний університет  
«Запорізька Політехніка»  
Кафедра програмних засобів

Дипломна кваліфікаційна робота  
на тему:

**Розробка програмного забезпечення  
інтернет-магазину з продажу мобільних  
пристроїв**



Виконав  
студент групи КНТ-220

Михайло ЧЕРНІГОВЦЕВ

Керівник  
к.т.н., доцент

Анжеліка ПАРХОМЕНКО

Рисунок Б.1 – Слайд 1



**Актуальність роботи**

| Параметр                                    | Інтернет-магазин<br>«Yabluka»                       | Інтернет-магазин<br>«Яблоко»                        | Інтернет-магазин<br>«Цитрус»                        |
|---------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|-----------------------------------------------------|
| Наявність каталогу                          | Так                                                 | Так                                                 | Так                                                 |
| Можливість реєстрації                       | Так                                                 | Так                                                 | Так                                                 |
| Переключення денний/нічний режим            | Так                                                 | Ні                                                  | Ні                                                  |
| Наявність кошику                            | Так                                                 | Так                                                 | Так                                                 |
| Наявність систем оплати                     | Visa або MasterCard                                 | Visa або MasterCard                                 | Visa або MasterCard                                 |
| Вид сторінки товару                         | односторінкова вітрина-представлення (landing page) | односторінкова вітрина-представлення (landing page) | односторінкова вітрина-представлення (landing page) |
| Функціонал вебзастосунку (серверна частина) | фреймворк Symfony2                                  | технологія LAMP                                     | технологія LAMP                                     |

2

Рисунок Б.2 – Слайд 2



## Мета та задачі роботи

**Мета роботи** – розробка прототипу інтернет-магазину для обліку продажу мобільних пристроїв.

**Задачі роботи:**

- виконати вибір та обґрунтування засобів розробки інтернет-магазину;
- розглянути існуючі системи керування вмістом та фреймворки для розробки вебзастосунків;
- обрати відповідний архітектурний шаблон проєктування програмного забезпечення;
- обґрунтувати вибір системи керування базами даних та виконати проєктування схеми бази даних;
- розглянути засоби проєктування макету користувацького інтерфейсу вебзастосунків;
- спроєктувати макет клієнтської частини за допомогою інструментів штучного інтелекту для проєктування користувацького інтерфейсу;
- розробити прототип інтернет-магазину з продажу мобільних пристроїв;
- виконати тестування розробленого програмного забезпечення.

3

Рисунок Б.3 – Слайд 3



## Аналіз вимог

Необхідно розробити прототип інтернет-магазину з продажу мобільних пристроїв, що буде містити серверну та клієнтську частини.

| Функціональні вимоги                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Неавторизований користувач                                                                                                                                                                                                              | Авторизований користувач                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Адміністратор(менеджер)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <ul style="list-style-type: none"><li>– можливість реєстрації;</li><li>– перегляд галереї мобільних пристроїв;</li><li>– перегляд мобільних пристроїв за категоріями;</li><li>– пошук мобільного пристрою по ключовим словам.</li></ul> | <ul style="list-style-type: none"><li>– можливість авторизації;</li><li>– додавання мобільного пристрою до кошика;</li><li>– перегляд історії своїх замовлень;</li><li>– можливість вподобати та зберегти вибрані мобільні пристрої;</li><li>– редагування та оновлення особистої інформації;</li><li>– скидання паролю, використовуючи зареєстровану електронну пошту;</li><li>– оформлення та можливість оплати замовлення;</li><li>– перегляд галереї мобільних пристроїв;</li><li>– перегляд мобільних пристроїв за категоріями;</li><li>– пошук мобільного пристрою по ключовим словам.</li></ul> | <ul style="list-style-type: none"><li>– можливість авторизації;</li><li>– додавання мобільного пристрою до кошика;</li><li>– перегляд історії своїх замовлень;</li><li>– можливість вподобати та зберегти вибрані мобільні пристрої;</li><li>– редагування та оновлення особистої інформації;</li><li>– скидання паролю, використовуючи зареєстровану електронну пошту;</li><li>– оформлення та можливість оплати замовлення;</li><li>– перегляд галереї мобільних пристроїв;</li><li>– перегляд мобільних пристроїв за категоріями;</li><li>– пошук мобільного пристрою по ключовим словам.</li></ul> |

4

Рисунок Б.4 – Слайд 4



5

Рисунок Б.5 – Слайд 5

| Параметр                | PHP                               | Python                                                        | JavaScript                                                                                |
|-------------------------|-----------------------------------|---------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Тип мови                | Серверна скриптова мова           | Універсальна мова програмування                               | Скриптова мова для веб (з використанням як на клієнтській, так і на серверній стороні)    |
| Використання            | Веб-розробка, серверні сценарії   | Веб-розробка, наука про дані, автоматизація, штучний інтелект | Веб-розробка (взаємодія з користувачем, асинхронні операції), серверні сценарії (Node.js) |
| Типізація               | Динамічна, слабка                 | Динамічна, сильна                                             | Динамічна, слабка                                                                         |
| Парадигми програмування | Імперативна, об'єктно-орієнтована | Імперативна, об'єктно-орієнтована, функціональна, процедурна  | Імперативна, об'єктно-орієнтована, функціональна                                          |
| Стандартна бібліотека   | Широка підтримка веб-функцій      | Дуже багата, включає багато модулів для різних задач          | Багата, особливо для роботи з веб                                                         |

6

Рисунок Б.6 – Слайд 6



## Вибір мови програмування

| Параметр              | PHP                               | Python                           | JavaScript                                       |
|-----------------------|-----------------------------------|----------------------------------|--------------------------------------------------|
| Керування пам'яттю    | Автоматичне                       | Автоматичне                      | Автоматичне                                      |
| Підтримка паралелізму | Обмежена                          | Модулі asyncio, threading        | Асинхронні операції, Promises                    |
| Середовище виконання  | Сервери з підтримкою PHP (Apache) | Інтерпретатор Python             | Браузери, Node.js                                |
| Модульність           | Підтримка через include і require | Підтримка через модулі та пакети | Підтримка через модулі (ES6), CommonJS (Node.js) |

7

Рисунок Б.7 – Слайд 7



## Вибір середовища розробки

| Параметр               | Visual Studio Code                         | PyCharm                                             | Atom                                                  |
|------------------------|--------------------------------------------|-----------------------------------------------------|-------------------------------------------------------|
| Тип                    | Редактор вихідного коду                    | Інтегроване середовище розробки                     | Текстовий редактор і редактор коду                    |
| Підтримувані мови      | Багатомовний                               | Python, JavaScript, HTML, CSS та інші               | Різноманітні мови програмування та мови розмітки      |
| Навігація              | Зручна навігація, швидкий перехід          | Швидкий перехід між класами, методами та файлами    | Швидкий перехід між рядками та файлами                |
| Функціональність       | Зневаджувач, підсвічування синтаксису, Git | Аналіз коду, зневаджувач, рефакторинг, веб-розробка | Можливості редагування та кодування, інтеграція з Git |
| Підтримка розширень    | Широкий вибір розширень                    | Розширений функціонал за допомогою плагінів         | Велика кількість плагінів для розширення можливостей  |
| Спільнота користувачів | Велика                                     | Велика                                              | Активна                                               |
| Вартість               | Безкоштовний                               | Платний                                             | Безкоштовний                                          |

8

Рисунок Б.8 – Слайд 8



## Вибір системи керування базами даних

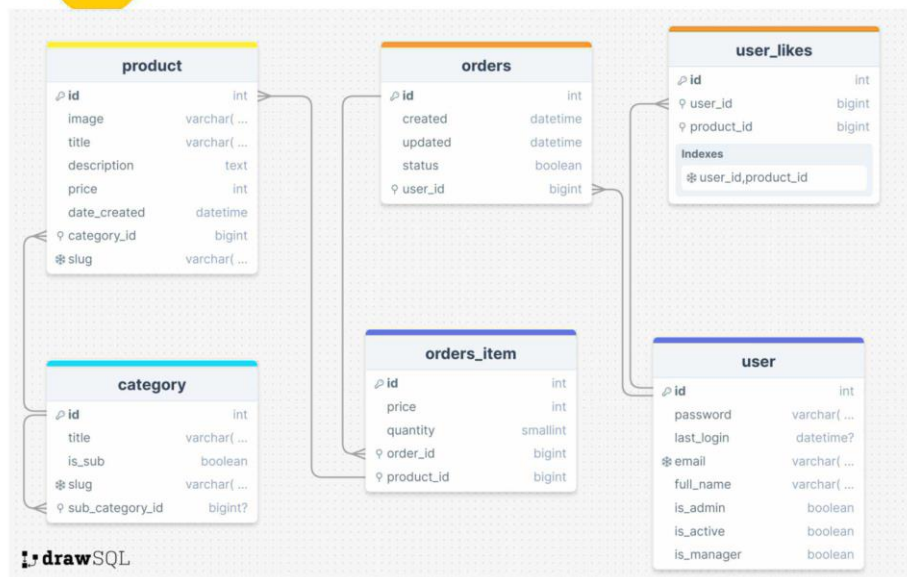
| Критерій                                 | SQLite                          | MySQL                           | MariaDB                         | PostgreSQL                      |
|------------------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Тип                                      | Вбудована база даних            | Серверна база даних             | Серверна база даних             | Серверна база даних             |
| Відкритість                              | Відкрите програмне забезпечення | Відкрите програмне забезпечення | Відкрите програмне забезпечення | Відкрите програмне забезпечення |
| Сумісність з SQL                         | Так                             | Так                             | Так                             | Так                             |
| Підтримка доповнень                      | Обмежена                        | Так                             | Так                             | Так                             |
| Продуктивність                           | Низька                          | Висока                          | Висока                          | Висока                          |
| Підтримка великих обсягів даних          | Не найкраща                     | Так                             | Так                             | Так                             |
| Підтримка об'єктно-орієнтованого підходу | Не надає                        | Не надає                        | Не надає                        | Так                             |
| Активний розвиток                        | Ні                              | Так                             | Так                             | Так                             |

9

Рисунок Б.9 – Слайд 9



## Схема бази даних

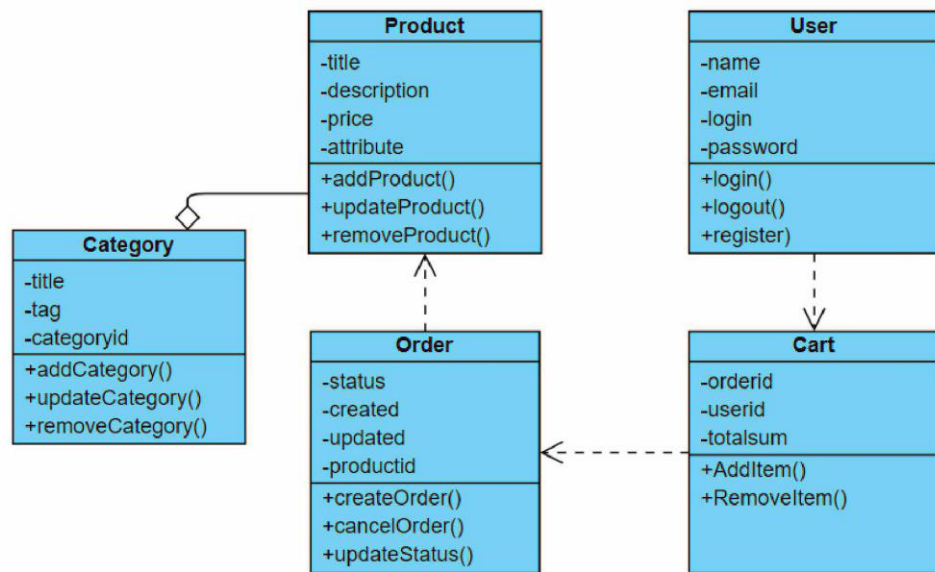


10

Рисунок Б.10 – Слайд 10



## Діаграма класів



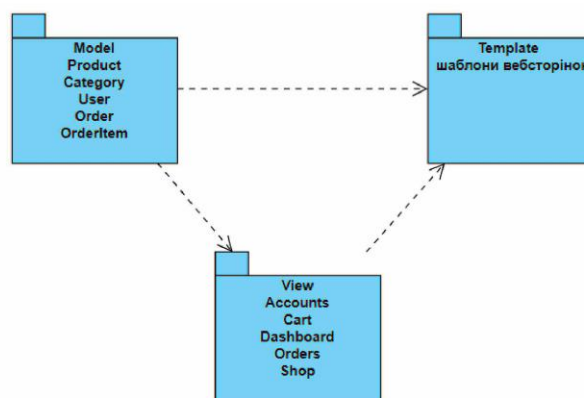
АКТИРАІ

11

Рисунок Б.11 – Слайд 11



## Структура інтернет-магазину

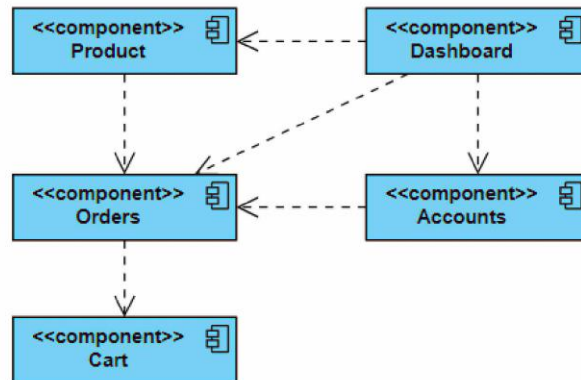


12

Рисунок Б.12 – Слайд 12



## Діаграма компонент

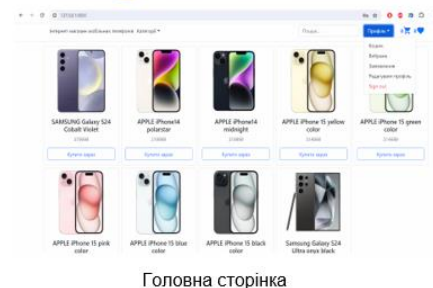


13

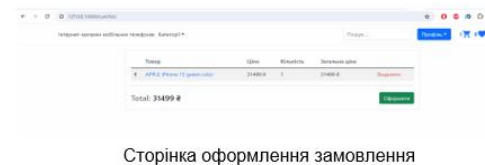
Рисунок Б.13 – Слайд 13



## Інтерфейс застосунку



Головна сторінка



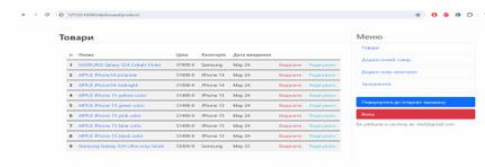
Сторінка оформлення замовлення



Сторінка замовлень користувача



Сторінка додавання мобільного пристрою до кошика



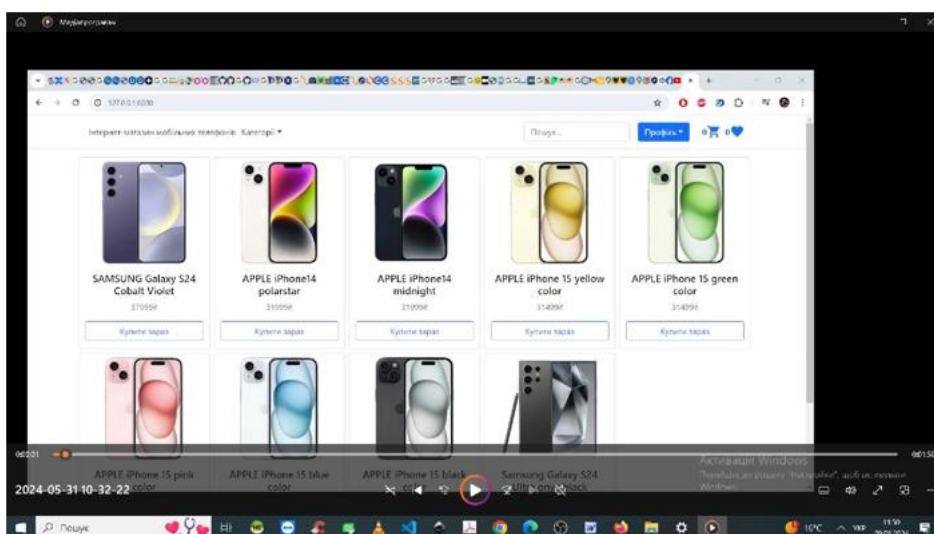
Сторінка адміністративної частини

14

Рисунок Б.14 – Слайд 14



## Демонстрація роботи програми



15

Рисунок Б.15 – Слайд 15



## Висновки

В результаті виконання дипломної кваліфікаційної роботи було розроблено прототип інтернет-магазину з продажу мобільних пристроїв, що об'єднує серверну частину, реалізовану з використанням мови програмування Python та фреймворку Django, систему керування базами даних MySQL, а також клієнтську частину, розроблену за допомогою інструментів штучного інтелекту для проекткування користувацького інтерфейсу.

Практична цінність роботи полягає в тому, що впровадження розробленого кроссплатформного програмного забезпечення дозволить підвищити ефективність введення обліку продажів мобільних пристроїв в інтернет-магазині.

В подальшій роботі можливо розширити функціонал програмного забезпечення шляхом додавання брендбуку інтернет-магазину, можливості залишити відгук, форми зворотного зв'язку, платіжних систем, варіантів доставки, кредитних пропозицій, акцій та знижок, інформації про програми лояльності, інструментів пошукової оптимізації, а також розгортання та викладення розробленого інтернет-магазину на хостингу.

Результати роботи було представлено на щорічній науково-практичній конференції викладачів, науковців, молодих учених, аспірантів та студентів НУЗП «Тиждень науки-2024».

16

Рисунок Б.16 – Слайд 16