

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Інтернет магазин для продажу мобільних телефонів

Назва теми

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного  
забезпечення»

Шифр КвРПЗ.190142.01.19.ПЗ

Виконав студент IV курсу, група ПЗ-19-1

  
Підпис

В. Б.Трембач  
Ініціали, прізвище

Керівник д-р фіз.-мат. наук, проф.  
Науковий ступінь, звання

  
Підпис

Л.П. Бедратюк  
Ініціали, прізвище

Нормоконтролер канд. тех. наук, доцент  
Науковий ступінь, звання

  
Підпис

І. В. Гурман  
Ініціали, прізвище

До захисту допускаю:  
Завідувач кафедри інженерії  
програмного забезпечення

  
Підпис

Л. П. Бедратюк  
Ініціали, прізвище

5 червня 2023 р.

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри Л. П. Бедратюк

Л. П. Бедратюк

02 01 2023 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ (ПРОЕКТ)

Трембачу Владіславу Богдановичу

Прізвище, ім'я, по батькові студента

1. Тема роботи (проекту) Інтернет магазин для продажу мобільних телефонів

Керівник роботи (проекту) Бедратюк Леонід Петрович, д-р фіз.-мат. наук, проф

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Матеріали переддипломної практики

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

Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Три креслення у форматі А3: UML-Діаграма варіантів використання, ER-діаграма бази даних, макети сторінок додатку.

6. Консультанти розділів кваліфікаційної роботи

| Розділ        | Прізвище, ініціали та посада консультанта | Підпис, дата      |                   |
|---------------|---|-------------------|-------------------|
|               |   | завдання видав    | завдання прийняв  |
| Нормоконтроль | Гурман І. В., канд.тех.наук. доцент       | 3.06.23 <i>ГВ</i> | 3.06.23 <i>ГВ</i> |
| Антиплагіат   | Гурман І. В., канд.тех.наук. доцент       | 2.06.23 <i>ГВ</i> | 2.06.23 <i>ГВ</i> |

7. Дата видачі завдання « 02 » січня 2023р.

КАЛЕНДАРНИЙ ПЛАН

| Назва етапів (розділів) кваліфікаційної роботи (проекту)   | Строк виконання етапів проєкту (роботи) | Примітка |
|--|---|----------|
| 1 Ознайомлення з тематикою кваліфікаційної роботи (КвР), визначення та узгодження індивідуальних тем КвР   | 01.12– 31.12.2022                       |          |
| 2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання | 02.01 – 31.01.2023                      |          |
| 3 Проєктування програмного забезпечення  | 01.02 – 28.02.2023                      |          |
| 4 Програмна реалізація з використанням відповідних засобів розробки  | 01.03 – 10.04.2023                      |          |
| 5 Тестування програмного забезпечення  | 11.04 – 30.04.2023                      |          |
| 6 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог стандартів                                       | 01.05 – 25.05.2023                      |          |
| 7 Попередній захист КвР  | Травень 2023 (згідно графіка)           |          |
| 8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки                                   | 26.05 – 30.05.2023                      |          |
| 9 Здача КвР на кафедру; підготовка КвР для розміщення у репозитарії ХНУ; підготовка до захисту та захист КвР   | з 01.06.2023                            |          |

Студент

  
Підпис

В.Б. Трембач  
Ініціали, прізвище

Керівник проєкту (роботи)

  
Підпис

Л.П.Бедратюк  
Ініціали, прізвище

## Анотація

Тема кваліфікаційної роботи: «Інтернет магазин для продажу мобільних телефонів».

Автор роботи: Трембач Владіслав Богданович.

Керівник роботи: Бедратюк Леонід Петрович.

Пояснювальна записка: 73 с., 33 рис., 3 табл., 3 дод., 40 джерел.

Графічна частина: 14 презентаційних слайдів.

Інтернет магазин для продажу мобільних телефонів

Мета кваліфікаційної роботи: розробка Інтернет магазин для продажу мобільних телефонів.

У кваліфікаційній роботі проведено аналіз предметної області та її інформаційного забезпечення, визначено вимоги до програмного забезпечення, розроблено його архітектуру та спроектована структура бази даних інтернет-магазину.

При розробці програмної системи використано мови програмування C# фреймворк ASP.NET MVC, сервер бази даних SQL Server.

Практична значимість отриманих результатів полягає у розробці готового до використання програмного продукту у вигляді веб-сайту.

Впровадження системи дозволяє оптимізувати роботу інтернет-магазину, надає можливість роботи адміністратора з сайтом.

1.06.2023  
Дата

Трембач  
Підпис

## ВІДОМІСТЬ ДОКУМЕНТІВ

| № рядка | Формат | Позначення документа    | Найменування документа                     | К-сть аркушів | № екз. | Примітка |
|---------|--------|-------------------------|--|---------------|--------|----------|
|         |        |                         | <u>Текстові документи</u>                  |               |        |          |
| 1       | A4     | КвРІПЗ. 190142.01.19.ПЗ | Пояснювальна записка                       | 73            |        |          |
| 2       | A4     |                         | Завдання на кваліфікаційну роботу          | 1             |        |          |
| 3       | A4     |                         | Анотація                                   | 1             |        |          |
|         |        |                         | <u>Графічні документи</u>                  |               |        |          |
| 4       | A4     |                         | Презентаційні матеріали                    | 12            |        |          |
| 5       | A3     | КвРІПЗ.190142.01.19.Е8  | UML – діаграма -<br>Варіантів використання | 1             |        |          |
| 6       | A3     | КвРІПЗ.190142.01.19.Е8  | ER – діаграма бази даних                   | 1             |        |          |
| 7       | A3     | КвРІПЗ.190142.01.19.Е8  | Макети сторінок<br>застосунку              | 1             |        |          |

|  |      |               |         |      |
|--|------|---------------|---------|------|
| <b>КвРІПЗ.190142.01.19.ВД</b>                    |      |               |         |      |
| Змн  | Арк. | № докум.      | Підпис  | Дата |
| Розроб   |      | Трембач В.Б.  |         | 5.06 |
| Перевір.   |      | Бедратюк Л.П. |         | 5.06 |
| Н. Контр.  |      | Гурман І.В.   |         | 5.06 |
| Затверд.   |      | Бедратюк Л.П. |         | 5.06 |
| Інтернет магазин для продажу мобільних телефонів |      |               | Лім.    | Арк. |
| Відомість документів                             |      |               |         | 1    |
|  |      |               | Аркушів | 1    |
| <b>ХНУ, ІПЗ-19-1</b>                             |      |               |         |      |

## ЗМІСТ

|   |    |
|---|----|
| <b>ВСТУП</b> .....  | 6  |
| <b>1. Дослідження предметної області</b> .....  | 8  |
| 1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей ..... | 8  |
| 1.2 Аналіз наявного програмно-технічного забезпечення предметної області .....                | 11 |
| 1.3 Визначення функціональних вимог до програмного забезпечення .....                         | 18 |
| 1.4 Висновки до розділу «Дослідження предметної області».....                                 | 22 |
| <b>1 Проектування веб-застосунку</b> .....  | 23 |
| 2.1 Аналіз та вибір архітектури веб-додатка.....  | 23 |
| 2.2 Опис структури даних та моделі бази даних .....   | 27 |
| 2.3 Проектування серверної частини веб-додатка.....   | 31 |
| 2.4 Проектування клієнтської частини веб-додатка.....   | 33 |
| 2.5 Створення макета веб-застосунка та дизайн.....  | 35 |
| 2.6 Аналіз та вибір технологій і методів реалізації веб-додатка.....                          | 40 |
| 2.7. Висновки до розділу “Проектування веб-застосунку” .....                                  | 44 |
| <b>3. Програмна реалізація</b> .....  | 45 |
| 3.1 Розробка бази даних .....   | 45 |
| 3.2 Розробка програмних модулів .....   | 46 |
| 3.3 Керівництво користувача .....   | 51 |
| 3.4 Технічні характеристики веб-застосунка. ....  | 60 |
| 3.5. Аналіз методів тестування веб- застосунку та розробка тестів .....                       | 61 |
| 3.6 Висновки до розділу Програмна реалізація .....  | 64 |
| <b>ВИСНОВКИ</b> .....   | 65 |
| <b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</b> .....   | 66 |
| <b>ДОДАТОК А</b> .....  | 70 |
| <b>ДОДАТОК Б</b> .....  | 74 |

|           |      |               |        |      |  |      |      |         |
|-----------|------|---------------|--------|------|--|------|------|---------|
|           |      |               |        |      | <b>КвРІПЗ.190142.01.19.ПЗ</b>                    |      |      |         |
| Змн.      | Арк. | № докум.      | Підпис | Дата |  |      |      |         |
| Розроб.   |      | Трембач В.Б.  |        | 5.06 | Інтернет магазин для продажу мобільних телефонів | Літ. | Арк. | Акрушів |
| Перевір.  |      | Бедратюк Л.П. |        | 5.06 |  |      | 5    | 73      |
| Н. Контр. |      | Гурман І.В.   |        | 5.06 | ХНУ, ІПЗ-19-1                                    |      |      |         |
| Затверд.  |      | Бедратюк Л.П. |        | 5.06 |  |      |      |         |

|             |             |                 |               |             |                               |             |
|-------------|-------------|-----------------|---------------|-------------|-------------------------------|-------------|
|             |             |                 |               |             | <i>КвРІПЗ.190142.01.19.ПЗ</i> | <i>Арк.</i> |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                               | 6           |

## ВСТУП

Сучасні технології швидко змінюють світ навколо нас, а електронна комерція не є винятком. Інтернет-магазини стають все більш популярними та необхідними для людей, які хочуть придбати товари без виходу з дому.

Однією з головних тенденцій сучасного світу є збільшення кількості користувачів Інтернету. За даними статистики, понад 4 мільярди людей у світі використовують Інтернет, і це число постійно зростає. За останні кілька років також спостерігається зростання кількості покупців, які обирають онлайн-шопінг. Мобільні телефони, в свою чергу, є невід'ємною частиною життя сучасної людини. Їх використовують не тільки для зв'язку, але й для роботи, розваг та покупок.

Проблема, на розв'язання якої спрямована кваліфікаційна робота, полягає у створенні ефективного та зручного інтернет-магазину з продажу мобільних телефонів, який зможе задовольнити потреби сучасного споживача. Окреслюючи завдання, необхідні для вирішення цієї проблеми, можна зазначити розробку зручного інтерфейсу, визначення оптимальної системи оплати та доставки товарів, підвищення безпеки та зручності користування сайтом.

Розробка інтернет-магазину з продажу мобільних телефонів є дуже актуальною в наш час, оскільки сучасний споживач шукає швидкий та зручний спосіб придбання товарів. Завдяки Інтернету він має можливість обирати товари з великого асортименту та купувати їх з будь-якої точки світу, не виходячи з дому. Розробка інтернет-магазину з продажу мобільних телефонів буде корисною не тільки для клієнтів, але й для компанії-власника, оскільки це збільшить їхню потенційну аудиторію, а разом із цим і кількість продаж та можливість здійснювати продажі в режимі 24/7 в будь-яку точку країни де є поштове відділення.

Отже, розробка інтернет-магазину з продажу мобільних телефонів може

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
|      |      |          |        |      |                        | 7    |
| Змн. | Арк. | № докум. | Підпис | Дата |                        |      |

бути дуже актуальною та перспективною задачею в сучасному світі. У кваліфікаційній роботі ми зосередимося на розробці ефективного та зручного інтерфейсу, системі оплати та доставки товарів, підвищенні безпеки та зручності користування сайтом. Наш проект зможе забезпечити якісний сервіс для клієнтів та сприяти розвитку електронної комерції в Україні.

Метою кваліфікаційної роботи є розробка функціонального і ефективного веб-застосунка для торгівлі мобільними телефонами, який власникам малого та середнього бізнесу розширити свою справу та збільшити обсяг продажів. Вони зможуть отримати більше замовлень та підвищити свій прибуток завдяки ефективній онлайн-продажі. Крім того, вони можуть збільшити свою базу клієнтів завдяки доступності онлайн-магазину та зручності його використання. Кінцева версія веб-застосунка повинна забезпечувати функції онлайн-магазину, що дозволить власникам бізнесу продавати товари та встановлювати зв'язок зі своїми клієнтами.

Завдання, які необхідно вирішити для досягнення мети:

- Провести комплексний аналіз предметної області для визначення її головних особливостей;
- провести аналіз наявного програмного забезпечення для визначення вимог до застосунку;
- здійснити огляд технологій, методологій та практик розробки веб-застосунків та вибрати підходящі для майбутнього використання при розробці проекту;
- сформулювати технічне завдання;
- розробити структуру та функціональну модель проекту;
- виконати програмну реалізацію проекту;
- провести тестові випробування застосунка;
- зробити документацію, що описує процес виконання проекту.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|      |      |          |        |      |                               | 8    |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

## 1. Дослідження предметної області

### 1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Мобільні телефони з'явилися в 80-х роках минулого століття і пройшли значний еволюційний шлях, перетворившись зі звичайних засобів зв'язку на сучасні розумні пристрої, що вміють виконувати різноманітні завдання. Первісно, перші версії мобільних телефонів пропонували лише основні функції, такі як здійснення дзвінків і надсилання коротких текстових повідомлень (SMS). Проте, з розвитком технологій, мобільні телефони стали набагато складнішими пристроями і отримали широкий спектр функціональних можливостей. Вони здатні підключатися до Інтернету, мають фото- та відеокамери, дозволяють запускати різноманітні ігри, прослуховувати музику, переглядати відео, здійснювати навігацію за допомогою GPS, а також мають доступ до великої кількості різноманітних додатків, що забезпечують виконання різних функцій - від простих записників та секундомірів до потужних програм, що нічим не поступають функціоналом ПЗ на персональних комп'ютерах.

За останні десятиліття спостерігається значне зростання обсягів продажу мобільних телефонів. Згідно з даними Statista, у 2020 році світовий ринок мобільних телефонів досяг практично 1,57 мільярда одиниць, що порівняно з 1,41 мільярда одиниць у 2016 році. Це свідчить про високу популярність мобільних телефонів серед споживачів.

Однією з ключових причин зростання популярності мобільних телефонів є їхній широкий функціонал. Розумні телефони здатні виконувати більшість завдань, які раніше виконувались на комп'ютерах. Крім того, вони надають можливість бути постійно на зв'язку, що дозволяє людям швидко та зручно отримувати інформацію і спілкуватися з іншими.

Взагалі, розвиток мобільних телефонів і зростання їх продажів є

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
|      |      |          |        |      |                        | 9    |
| Змн. | Арк. | № докум. | Підпис | Дата |                        |      |

складним явищем, яке має як позитивні, так і негативні наслідки. Щоб забезпечити максимальну користь від цього розвитку, важливо продовжувати прогресувати в галузі нових технологій та створювати ефективні й безпечні для споживачів мобільні телефони. Також потрібно приймати заходи для зменшення впливу масового використання мобільних телефонів на довкілля та здоров'я людей.

Крім того, розвиток мобільних телефонів і зростання їх використання сприяють розвитку цифрової економіки та інтернет-технологій. Вони дозволяють людям легко отримувати доступ до інформації, спілкуватися з іншими людьми, здійснювати онлайн-операції та здійснювати покупки, що робить їхнє життя більш зручним і ефективним.

Зростання обсягу продажів мобільних телефонів також суттєво впливає на глобальну економіку, оскільки цей сектор став одним з найбільш прибуткових. Він забезпечує зайнятість мільйонам людей по всьому світу, від розробників до продавців, і створює значні прибутки для компаній, що виробляють мобільні телефони.

Крім того, важливо забезпечити доступність мобільних телефонів для всіх верств населення і країн, особливо для тих, хто проживає в віддалених або менш розвинених регіонах. Це дозволить забезпечити рівні можливості та доступ до інформації, сприяючи розвитку і прогресу країн.

Узагальнюючи, розвиток мобільних телефонів і зростання кількості їх продажів мають значний вплив на наше життя та економіку. Вони забезпечують зручність та доступність до інформації та технологій, але також можуть мати негативний вплив на здоров'я та довкілля. Крім того, зростання продажів мобільних телефонів впливає на економіку країн. Виробництво та продаж цих пристроїв стає джерелом доходу для багатьох компаній та країн, що забезпечує підвищення рівня життя населення та розвиток технологічного сектору. Також збільшення кількості продажів мобільних телефонів сприяє росту ринку мобільних застосунків та інших послуг, що розширює можливості користувачів та сприяє інноваційному розвитку.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КєРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 10   |

Основним завданням проекту є розробка детальної архітектури та компонентів платформи для мобільної інтернет-торгівлі з мобільними телефонами. Для досягнення цієї мети було проведено дослідження предметної області з метою з'ясування вимог до програмного продукту та його користувачів.

У процесі дослідження були визначені основні категорії користувачів програмного продукту, причому першою і найбільшою категорією є клієнти магазину. Вони будуть мати доступ до каталогу товарів та зможуть здійснювати покупки. Другою категорією користувачів є адміністратори, які будуть відповідати за додавання та редагування товарів.

Оскільки кожна категорія користувачів матиме свої функціональні можливості, планується реалізація механізму реєстрації та авторизації в системі, щоб забезпечити відокремлення користувачів та надати їм відповідний рівень доступу.

На зображенні 1.1 наведено контекстну діаграму онлайн-магазину, де потоки показують, звідки надходять дані та основну інформацію, яка виходить. Вхідні дані включають інформацію, що надходить від користувачів, їх реєстраційні дані, запити до системи щодо даних про конкретні товари або загальну інформацію.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 11   |

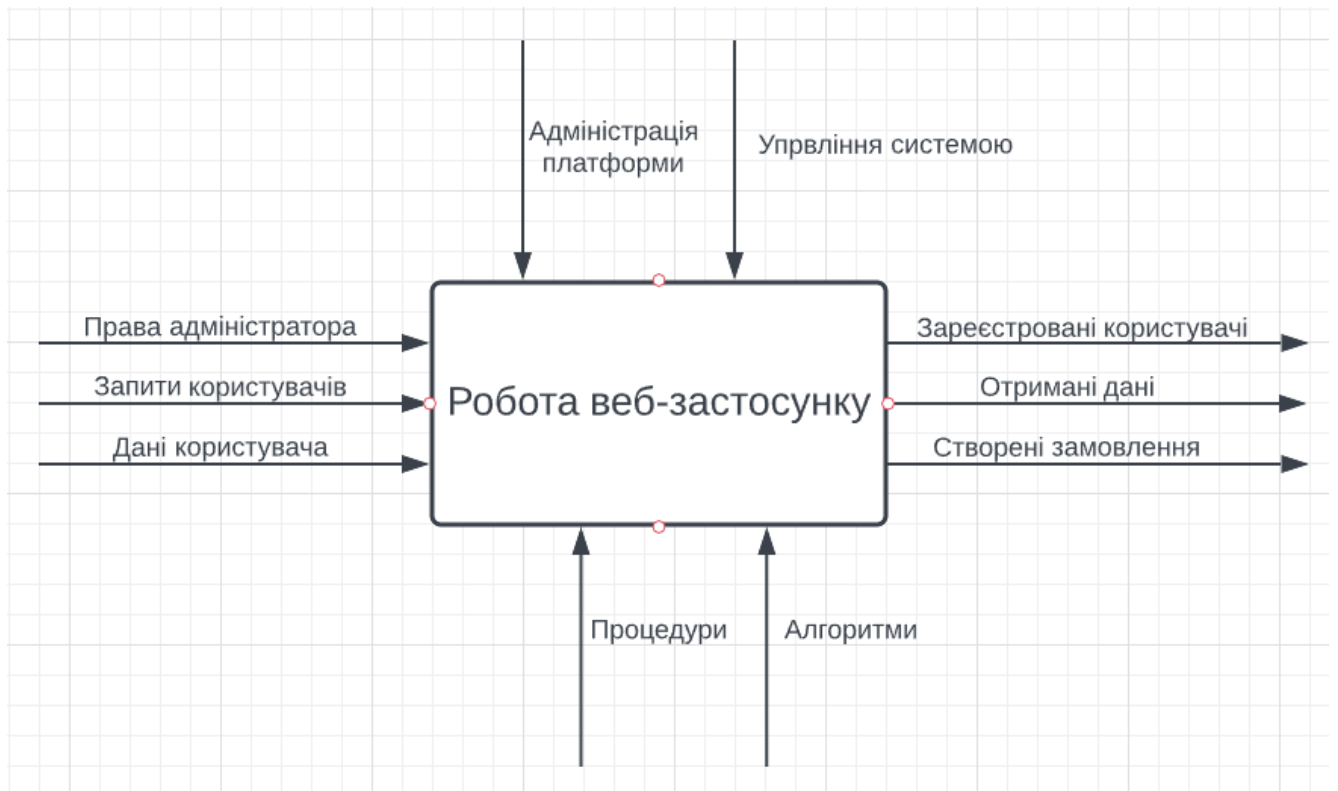


Рисунок 1.1 – Діаграма потоків даних.

Система надає основні функціональні можливості, такі як перегляд детальної інформації про товари, замовлення товарів, додавання їх до кошика та оплата. Крім цього, система має додатковий функціонал, спрямований на обслуговуючий персонал, який забезпечує їм можливість додавання та редагування товарів.

Задля детального зображення процесів дії користувача у системі, була використана діаграма декомпозиції першого рівня (рис.1.2).

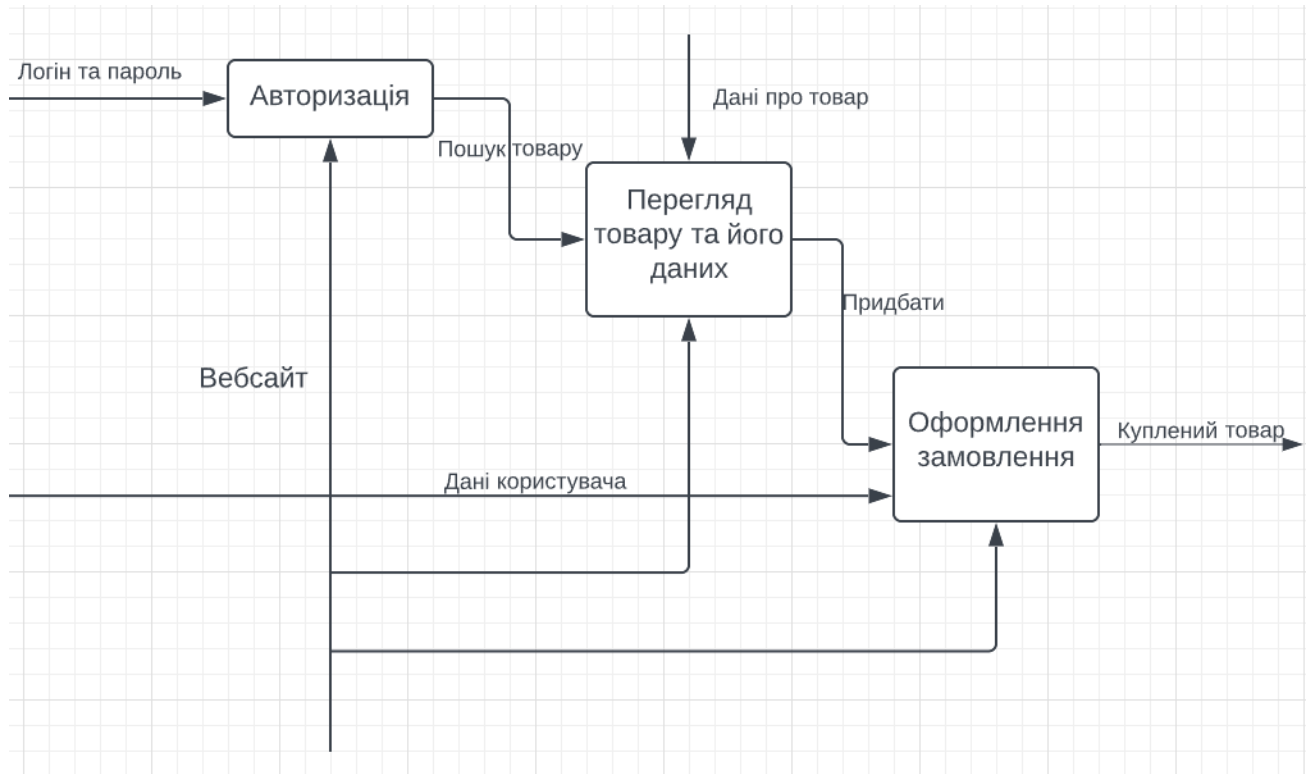


Рисунок 1.2 – Діаграма декомпозиції першого рівня.

Це рішення, створити діаграму декомпозиції першого рівня, допомогло отримати один із основних сценаріїв використання сайту.

## 1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Аналіз ринку інтернет-магазинів мобільних телефонів в Україні свідчить про наявність жорсткої конкуренції серед учасників ринку. Україна є однією з країн, де користувачі мобільних телефонів швидко переходять на нові моделі та прагнуть оновлювати свої пристрої. Тому ринок мобільних телефонів в Україні є динамічним і активним.

За даними порталу E-Commerce.ua, в Україні функціонує понад 100 інтернет-магазинів мобільних телефонів. Проте точно визначити, скільки з них успішно працюють і надають якісний сервіс, є складним завданням. Найбільші учасники ринку, такі як Rozetka, Citrus, Foxtrot, Allo і Comfy, займають значну

частку ринку, проте нові інтернет-магазини з'являються з часом і намагаються зайняти свою нішу.

У останні роки спостерігається зростання популярності китайських брендів, таких як Xiaomi, Oppo і Vivo, на українському ринку. Ці бренди набувають популярності завдяки своїм високоякісним, але доступним за ціною продуктам. Багато інтернет-магазинів, спеціалізуючись на мобільних пристроях, пропонують широкий вибір смартфонів цих брендів.

Українські інтернет-магазини мобільних телефонів часто надають різноманітні послуги, що включають безкоштовну доставку, оплату при отриманні товару та можливість повернення пристрою протягом певного періоду. Крім того, деякі магазини пропонують додаткові сервіси, такі як налаштування пристрою та гарантійний ремонт.

Одним з відомих українських інтернет-магазинів є Rozetka, який пропонує широкий асортимент товарів, починаючи від електроніки та побутової техніки і досягаючи косметики та спортивних товарів. Сайт Rozetka розроблений з урахуванням зручного та швидкого пошуку товарів, оформлення замовлень та сплати за них.

Rozetka є веб-застосунком, створеним для здійснення торгівельної діяльності в Інтернеті, зокрема, для продажу товарів (Рисунок 1.3). Використовуване програмне забезпечення на сайті дозволяє користувачам зручно шукати товари, оформляти замовлення, здійснювати оплату та відстежувати статус доставки своїх замовлень.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 14   |

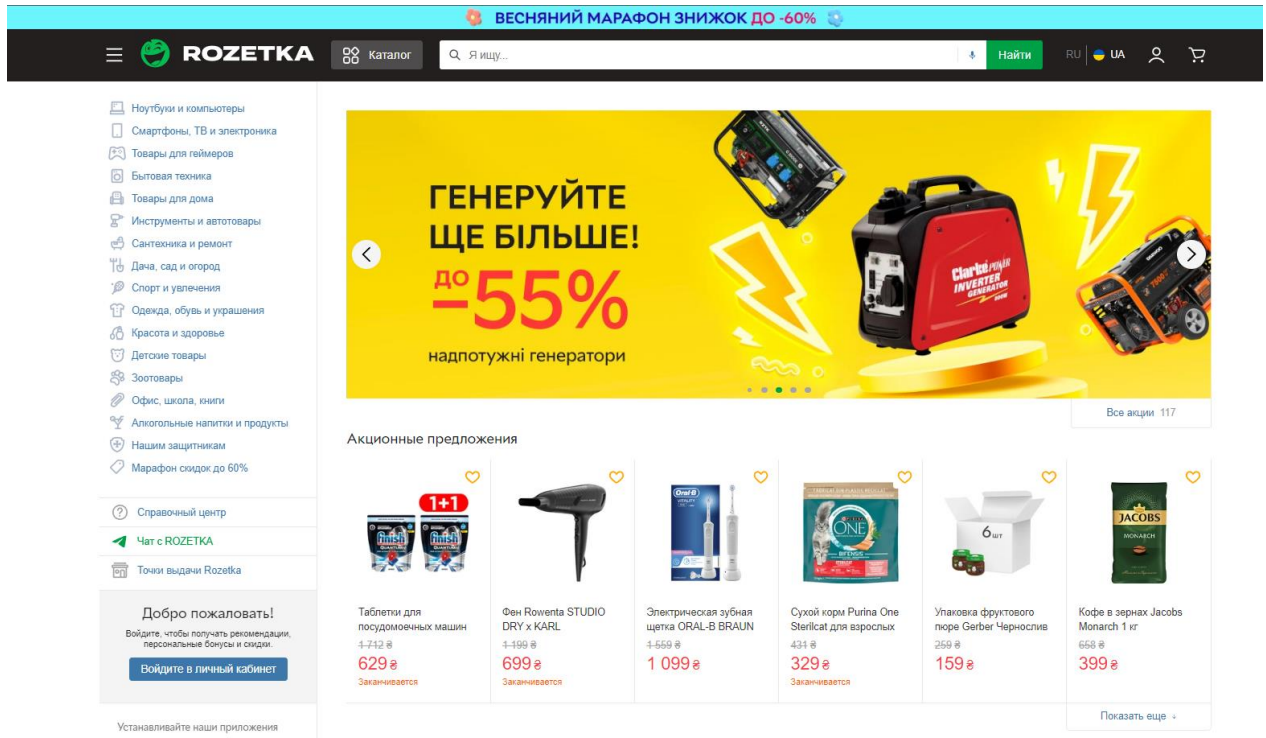


Рисунок 1.3 –головна сторінки Rozetka <https://rozetka.com.ua>.

Розробку сайту було здійснено компанією «Rozetka Group», яка була заснована у 2005 році. Зараз ця організація стала однією з провідних компаній в Україні, спеціалізуючись на онлайн-торгівлі та наданні послуг у сфері логістики та кур'єрської доставки.

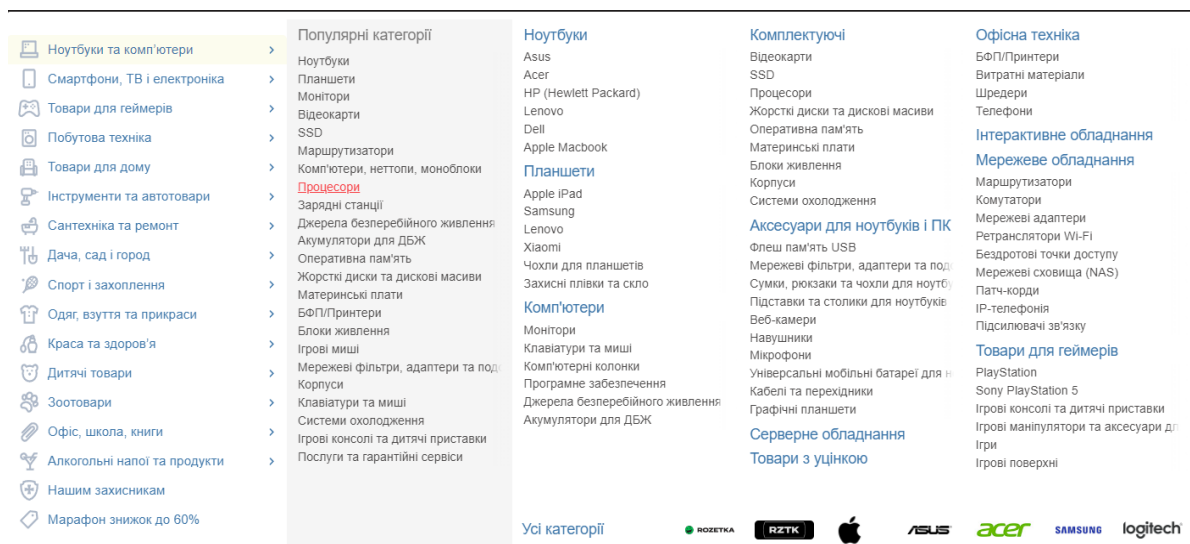


Рисунок 1.4 –каталог сайту Rozetka <https://rozetka.com.ua/ua/>

Сайт пропонує простий і зручний інтерфейс, що дозволяє користувачам легко знаходити потрібні товари та здійснювати замовлення. На головній сторінці сайту розміщені найпопулярніші категорії товарів, що дозволяє клієнтам швидко перейти до розділу, що їх цікавить (Рисунок 1.4). Також на сайті присутні фільтри, які допомагають зручно вибирати товари за різними критеріями. Крім того, користувачі мають можливість переглядати відгуки про товари, що допомагає їм зробити обґрунтований вибір. Користувачам також доступний показник доступності товарів у магазинах різних міст України.

Серед переваг сайту можна виділити величезний вибір товарів з різних категорій, наявність системи відгуків та коментарів, які допомагають користувачам при виборі товару, а також наявність онлайн-чату з підтримкою, що надає допомогу в разі виникнення проблем з товаром або оплатою. Крім того, сайт пропонує доставку товарів у будь-яке місце України.

Однак, серед недоліків сайту можна зазначити, що ціни на товари часто є вищими, ніж у конкурентів. Також час доставки товарів може залежати від регіону проживання клієнта, що може бути недоцільним для деяких користувачів.

Сайт Comfy представляє собою електронний магазин, спеціалізований на продажу електроніки та побутової техніки (Рисунок 1.5).

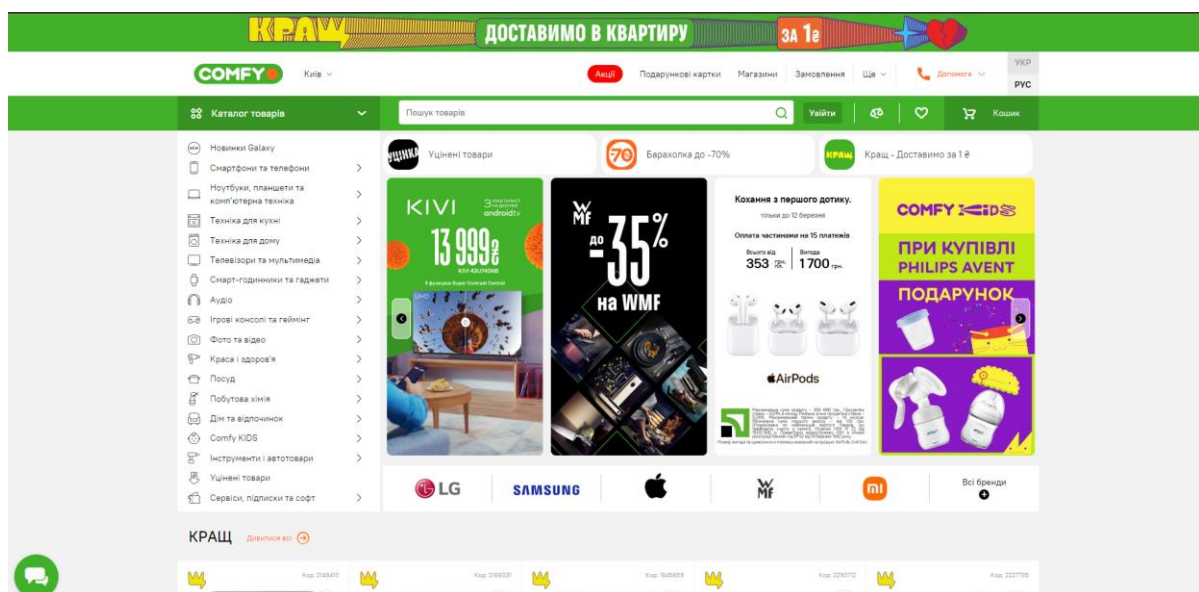


Рисунок 1.5 – Головна сторінка Comfy <https://comfy.ua/ua/>

|      |      |          |        |      |  |  |  |  |  |      |
|------|------|----------|--------|------|--|--|--|--|--|------|
|      |      |          |        |      |  |  |  |  |  | Арк. |
|      |      |          |        |      |  |  |  |  |  | 16   |
| Змн. | Арк. | № докум. | Підпис | Дата |  |  |  |  |  |      |

КвРІПЗ.190142.01.19.ПЗ

Для забезпечення роботи сайту використовується комплексне програмне забезпечення, яке складається з веб-сервера, бази даних та інтерфейсу користувача. Розробку сайту та його функціоналу виконує українська компанія Comfy, яка має багаторічний досвід в електронній комерції, працюючи на ринку понад 20 років

Інтерфейс Comfy вражає своєю сучасністю та функціональністю. Головна сторінка сайту пропонує найпопулярніші товари та акційні пропозиції, а також забезпечує швидкий доступ до категорій товарів. На сайті також присутній зручний пошуковий рядок, який дозволяє швидко знайти потрібний товар.

Переваги сайту Comfy включають:

- Великий вибір товарів у різних категоріях.
- Конкуентоспроможні ціни на багато товарів.
- Безкоштовна доставка при замовленні на суму від 1500 грн.
- Можливість отримати кешбек до 5% від вартості замовлення.

Недоліки сайту Comfy:

- Не завжди надається достатня детальна інформація про товар на сайті.
- Деякі клієнти скаржаться на низьку якість обслуговування.
- Сайт часто видає помилки при виконанні певних дій.
- Відсутність додаткових сервісів та послуг, таких як встановлення чи настройка техніки.

Загалом, сайт Comfy є досить функціональним і зручним для користувачів, але може потребувати покращень у деяких аспектах.

Сайт Citrus є онлайн-магазином електроніки та гаджетів, який надає широкий вибір продуктів від провідних виробників (Рисунок 1.6). Програмне забезпечення, яке використовується на сайті, складається з веб-сервера, бази даних та інтерфейсу користувача. Останній включає сторінки товарів, кошик покупок, пошук, реєстрацію та інші необхідні функції. Компанія Citrus.ua є розробником сайту.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 17   |

Інтерфейсний дизайн сайту Citrus є простим і зрозумілим. Головне меню розташоване у верхній частині сторінки і містить основні категорії товарів.

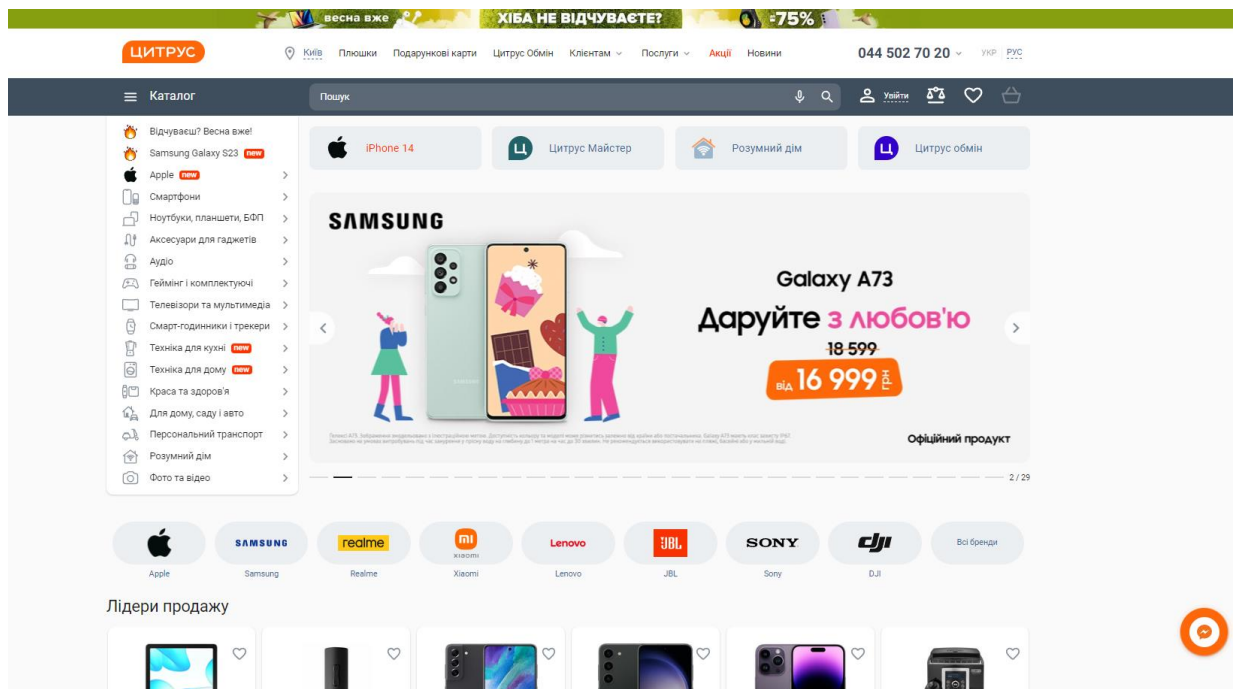


Рисунок 1.6 – Вигляд сайту Citrus <https://www.ctr.com.ua/>.

На сайті також доступні швидкий пошук товарів за назвою або кодом товару, а також система фільтрів (Рисунок 1.7). Кожна сторінка товару містить багато інформації про продукт, включаючи його характеристики та фотографії. Додатково, сайт пропонує зручні фільтри, які дозволяють користувачам шукати товари за різними параметрами, такими як ціна, виробник, модель і т. д.

Кожна сторінка товару містить багато інформації про продукт, включаючи його характеристики та фотографії. Додатково, сайт пропонує зручні фільтри, які дозволяють користувачам шукати товари за різними параметрами, такими як ціна, виробник, модель і т. д.

Переваги сайту Citrus включають наступні аспекти:

- Широкий вибір товарів від провідних виробників, що дозволяє клієнтам знайти бажаний продукт;
- швидкий та зручний пошук товарів, що допомагає користувачам ефективно здійснювати пошук і знаходити потрібні товари;

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
|      |      |          |        |      |                        | 18   |
| Змн. | Арк. | № докум. | Підпис | Дата |                        |      |

– інформативні сторінки товарів з детальними описами та фотографіями, що надають користувачам достатньо інформації для прийняття рішення про покупку;

– різноманітні способи оплати та доставки товарів, що забезпечують зручність і гнучкість для клієнтів;

– Програма лояльності для постійних клієнтів, що надає додаткові переваги і заохочує повторні покупки.

Недоліки сайту Citrus:

– Зайве завантаження інформацією на деяких сторінках товарів, що може створювати враження перенасиченості для деяких користувачів;

– повільна робота сайту у періоди великого попиту на товари, що може викликати затримки та незручності для користувачів.

Фільтр

Ціна

2999 - 79999 OK

Бренд

- Apple (201)
- Samsung (163)
- Xiaomi (70)
- OPPO (30)
- realme (36)

Показати всі

Кількість SIM-карт

- 1 SIM (139)
- 2 SIM (494)

Тип SIM-карти

- Micro-SIM (3)
- Nano-SIM (630)
- e-SIM (74)

Тип слоту

- SIM + SIM (87)

Рисунок 1.7 -Реалізація фільтрації на сайті Цитрус . <https://www.ctr.com.ua/>

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                        | 19   |

Після проведеного аналізу подібного програмного забезпечення встановлено, що розроблюваний застосунок повинен мати наступні характеристики:

- Простий, зручний та інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко навігуватися та використовувати застосунок ок без зайвих зусиль;
- повний опис доступних товарів, що дозволяє користувачам отримати всю необхідну інформацію про продукти перед придбанням;
- система пошуку та фільтрів, що допомагає користувачам здійснювати зручний пошук товару.

### 1.3 Визначення функціональних вимог до програмного забезпечення

Проаналізуємо вимоги до розроблюваного додатку та покажемо, які дії користувачі зможуть виконати за допомогою діаграми варіантів використання (ВВ). В таблиці 1.1 наведено опис двох основних акторів (користувачів додатку) і створено таблицю варіантів використання.

Таблиця 3.1 –коротка характеристика акторів.

| Актор         | Короткий опис   |
|---------------|---|
| Покупець      | Основними акторами є клієнти, які виконують декілька основних дій. Вони можуть шукати товари, додавати їх до кошика та оформляти замовлення. Клієнти поділяються на дві категорії: зареєстровані користувачі та гості.  |
| Адміністратор | Адміністратори відповідають за управління інтернет-магазином. Їхні обов'язки включають додавання нових продуктів до каталогу, оновлення існуючих продуктів, обробку замовлень від клієнтів, контроль запасу товару та забезпечення роботи магазину без зупинок. |

Для визначення вимог до додатку було розглянуто дії, які користувачі зможуть виконувати за допомогою додатку, а також обов'язки та функціональні можливості. За допомогою таблиці варіантів використання описано, які операції доступні користувачам.

В таблиці 1.1 надано опис двох основних акторів: користувачів додатку. Для кожного актора ми вказуємо їхні основні функції та обов'язки.

Крім того було створено таблицю варіантів використання(таблиця 1.2), де описано конкретні дії, які актори можуть виконувати в системі.

Таблиця 3.2 – Детальний опис варіантів використання.

| Актор  | Назва ВВ                                  | Опис ВВ  |
|--------|---|--|
| Клієнт | Пошук та вибір товару                     | Клієнти можуть зручно знаходити товари за ключовими словами, характеристиками, ціною, виробниками та іншими параметрами.   |
|        | Додавання товарів у кошик та їх видалення | Клієнти мають можливість додавати товари до свого кошика, що дозволяє зберігати їх там до моменту покупки, та видаляти їх з кошик у тому випадку, якщо передумали їх купляти.  |
|        | Створення замовлень                       | Клієнти мають можливість здійснювати замовлення, вказуючи свої контактні дані, адресу доставки та іншу необхідну інформацію. Це дозволяє їм зручно та точно вказати деталі, необхідні для обробки та доставки замовлення. Крім того, ця інформація допомагає забезпечити взаємозв'язок між клієнтом та магазином для будь-яких подальших комунікацій або питань, пов'язаних з замовленням. |
|        | Оплата                                    | Клієнти мають можливість обрати зручний для них спосіб оплати. Вони можуть скористатися різноманітними методами оплати, такими як кредитні картки, електронні гроші, банківський переказ та інші. Це дає клієнтам гнучкість і зручність при здійсненні оплати за покупки в інтернет-магазині.  |

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Змн. | Арк. | № докум. | Підпис | Дата |

КєРІПЗ.190142.01.19.ПЗ

Арк.

21

Продовження таблиці 3.2.

|               |                                |   |
|---------------|--------------------------------|---|
|               | Відстежування стану замовлення | Клієнти мають можливість відстежувати стан свого замовлення, починаючи від моменту відправлення його з магазину і до моменту доставки.  |
| Адміністратор | Додавання товарів              | Адміністратори мають можливість додавати нові товари до системи інтернет-магазину. Вони можуть включати в цей процес важливу інформацію, таку як опис товару, фотографії, ціну, наявність на складі та інші характеристики, що допомагають клієнтам зробити інформований вибір під час покупки. |
|               | Керування замовленнями         | Адміністрація має можливість переглядати та відстежувати статуси замовлень, призначати доставку та інші дії.  |
|               | Керування оплатою              | Адміністратори мають змогу здійснювати контроль над процесом оплати і відстежувати здійснені платежі.   |
|               | Звітність та аналітика         | Адміністрація має можливість генерувати звіти та проводити аналіз продажів з метою підвищення ефективності бізнесу і розуміння потреб клієнтів.   |

До переліку акторів також входять:

**Постачальники:** Ці актори відповідають за постачання товарів до інтернет-магазину. Вони забезпечують наявність товарів, надають інформацію про ціни та наявність продуктів.

**Система оплати:** Цей актор відповідає за безпечну та швидку оплату замовлень. Він може бути інтегрований з різними платіжними системами та банками, забезпечуючи зручність і безпеку платежів.

**Система доставки:** Цей актор відповідає за доставку продуктів до клієнтів вчасно та без пошкоджень. Вона може включати в себе як власних кур'єрів, так

і користуватися послугами зовнішніх служб доставки. Головна мета системи доставки - забезпечити швидку та надійну доставку товарів.

На основі описаних акторів та варіантів використання була побудована діаграма, яка наведена на рисунку 1.8

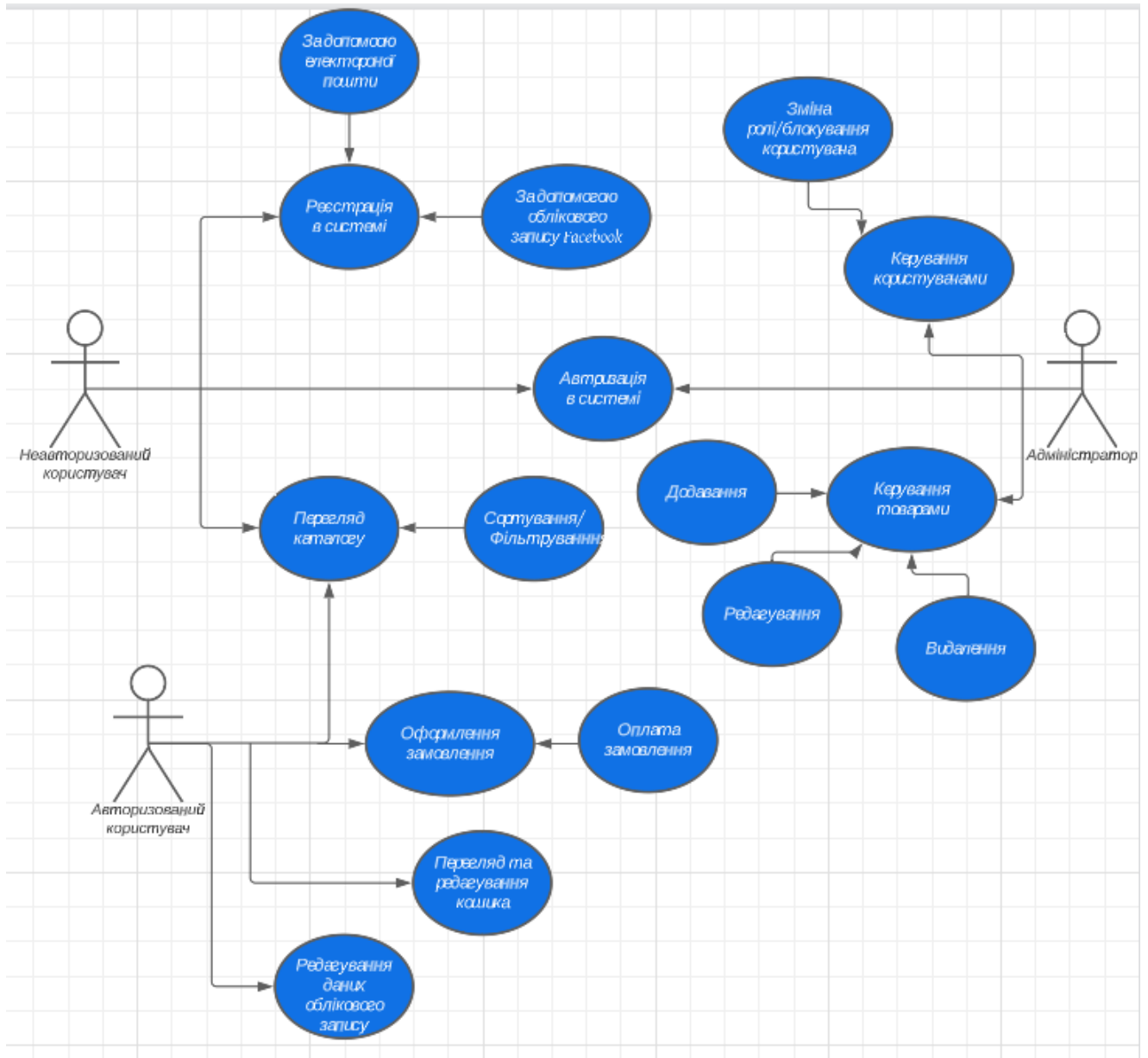


Рисунок 1.8 – Діаграма варіантів використання.

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Змн. | Арк. | № докум. | Підпис | Дата |

#### 1.4 Висновки до розділу «Дослідження предметної області».

У цьому розділі було проведено дослідження предметної області і виконано аналіз існуючих рішень, пов'язаних з темою кваліфікаційної роботи. Було проаналізовано кілька найбільш популярних веб-застосунків, що стосуються теми продажу мобільних телефонів.

Крім того, були визначені основні вимоги до програмного забезпечення. Ці вимоги враховують потреби і вимоги користувачів і сприяють розробці ефективного та функціонального веб-застосунку. Важливим етапом цього процесу було визначення основних акторів програмного забезпечення, тобто осіб або систем, які взаємодіють з веб-застосунком. На основі цих визначень була побудована діаграма варіантів використання, яка ілюструє основні сценарії взаємодії між акторами та програмним забезпеченням. Написання технічного завдання (додаток А) стало заключним етапом цього розділу.

Поставлені задачі:

- Створити систему каталогу товарів, щоб користувачі могли переглядати доступні товари.
  - Реалізувати можливість реєстрації та авторизації користувачів;
  - Забезпечити функціонал налаштування особистого профілю користувача, де вони зможуть змінювати свої особисті дані та налаштування;
  - Розробити систему кошика товарів;
  - Забезпечити можливість оплати, щоб користувачі могли безпечно та зручно оплачувати свої замовлення;
  - Створити функціонал керування сайтом для адміністратора, який дозволить додавати, редагувати та видаляти записи на сайті.
  - Реалізувати функціонал для керування та відстежування замовлень
- Необхідно забезпечити простий та зрозумілий інтерфейс користувача, який буде легким у використанні як для звичайних користувачів, так і для адміністраторів.

|      |      |          |        |      |  |                        |      |
|------|------|----------|--------|------|--|------------------------|------|
|      |      |          |        |      |  | КвРІПЗ.190142.01.19.ПЗ | Арк. |
|      |      |          |        |      |  |                        | 24   |
| Змн. | Арк. | № докум. | Підпис | Дата |  |                        |      |

## 2. Проектування веб-застосунку

### 2.1 Аналіз та вибір архітектури веб-додатка

Багаторівнева архітектура (або «n-рівнева»)- це підхід до розробки програмного забезпечення, який полягає у розділенні функціональності програми на логічні рівні або шари(Рисунок 2.1). Кожен рівень виконує певну задачу та взаємодіє з іншими рівнями, утворюючи ієрархічну структуру.

Типова n-tier архітектура включає такі рівні:

**Presentation Layer (рівень представлення):** Це верхній рівень, який відповідає за взаємодію з користувачем. Він включає веб-інтерфейс або інші інструменти, які дозволяють користувачам спілкуватися з додатком.

**Application Layer (рівень додатку):** Цей рівень виконує бізнес-логіку додатку. Він обробляє запити, отримані від Presentation Layer, і взаємодіє з іншими компонентами, такими як база даних або зовнішні сервіси. Тут зазвичай розміщуються контролери або служби додатку.

**Data Layer (рівень даних):** Цей рівень відповідає за доступ до даних. Він може включати базу даних або інші джерела даних, такі як файлові системи або зовнішні сервіси. Рівень даних забезпечує зберігання та маніпуляцію даними, які використовуються додатком.

Використання n-tier архітектури в розробці веб-додатків має кілька переваг:

**Модульність:** Кожен рівень може бути розроблений та підтримуватися окремо. Це спрощує розподіл роботи між командами розробників та полегшує впровадження змін у окремих компонентах без впливу на інші.

**Масштабованість:** Кожен рівень може бути масштабований окремо залежно від потреб додатку. Наприклад, можна збільшити кількість серверів у рівні додатку для обробки більшого обсягу запитів без необхідності змінювати інші компоненти.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 25   |

Безпека: Розподіл функціональності на рівні дозволяє встановлювати різні рівні доступу до ресурсів. Наприклад, можна встановити обмеження доступу до бази даних з рівня додатку та забезпечити захист даних.

Переносимість: Завдяки розділенню рівнів, можна замінити або модифікувати окремі компоненти без впливу на інші. Наприклад, можна замінити базу даних без необхідності змінювати код додатку.

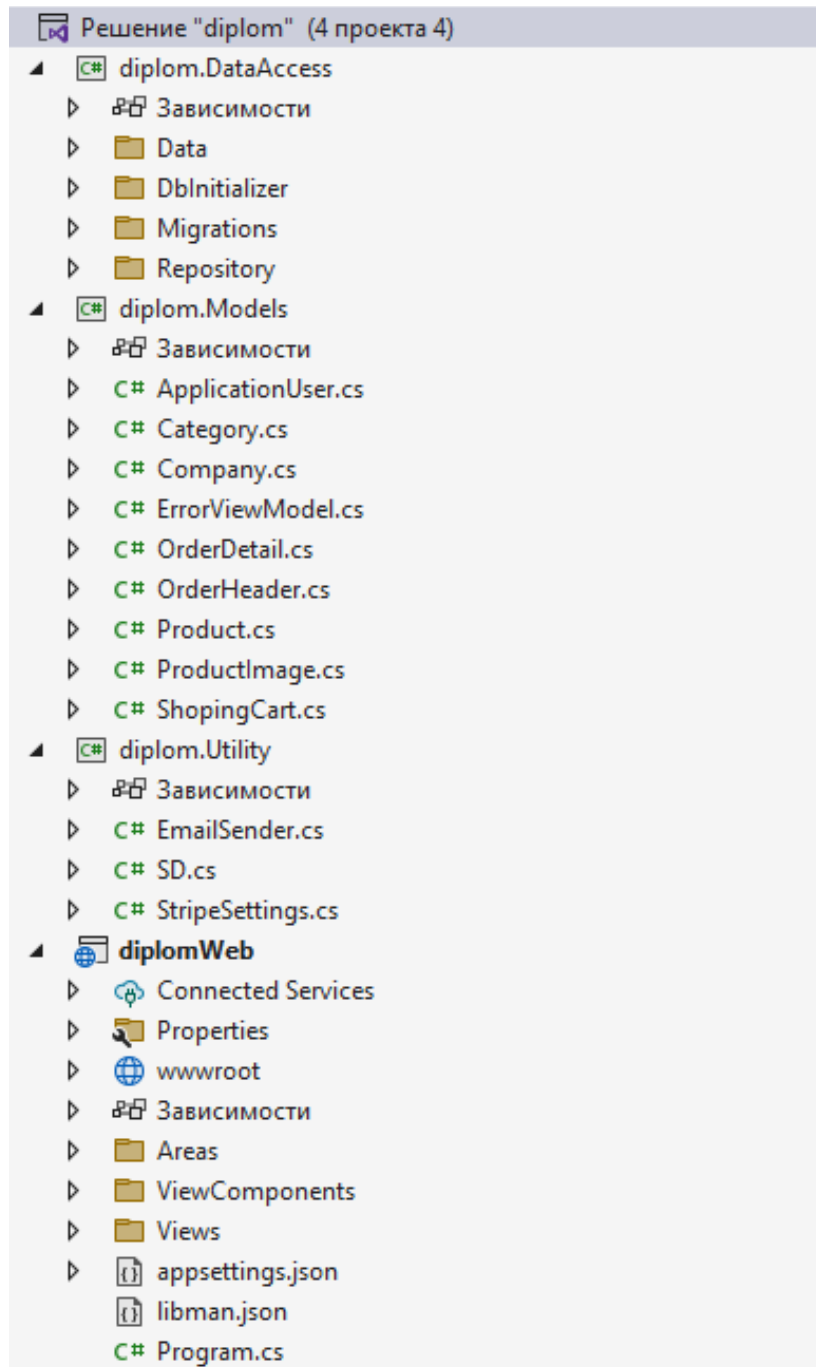


Рисунок 2.1 Приклад використання n-рівневої архітектури.

Загалом, n-рівнева архітектура дозволяє розділити функціональність додатку на окремі компоненти, полегшуючи розробку, підтримку, масштабування та забезпечуючи більшу гнучкість та контроль над додатком.

Усі існуючі Інтернет-платформи та будь-які веб сервіси побудовані на клієнт-серверній архітектурі.

В основі клієнт-серверної архітектури лежать два компоненти: клієнт і сервер.

Клієнт – комп'ютер на стороні користувача, який відправляє запит до сервера для надання інформації або виконання певних дій.

Сервер – більш потужний комп'ютер або обладнання, призначене для вирішення певних завдань з виконання програмних кодів, виконання сервісних функцій за запитом клієнтів, надання користувачам доступу до певних ресурсів, зберігання інформації і баз даних.

Модель такої системи полягає в тому, що клієнт відправляє запит на сервер, де він обробляється, і готовий результат відправляється клієнтові. Сервер може обслуговувати кілька клієнтів одночасно. Якщо одночасно приходять більше одного запиту, то вони встановлюються в чергу і виконуються сервером послідовно. Іноді запити можуть мати пріоритети. Запити з більш високими пріоритетами повинні виконуватися раніше.

Функції, які реалізуються на сервері:

- зберігання, доступ, захист і резервне копіювання даних;
- обробка клієнтського запиту;
- відправлення результату (відповіді) клієнту.
- Функції, які реалізуються на стороні клієнта:
- надання користувальницького інтерфейсу;
- формулювання запиту до сервера і його відправка;
- отримання результатів запиту і відправка додаткових команд

(запитів на додавання, оновлення або видалення даних).

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|      |      |          |        |      |                               | 27   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

Архітектура клієнт-сервер визначає принципи спілкування між комп'ютерами, а правила і взаємодії визначені в протоколі.

Мережевий протокол – це набір правил, за якими відбувається взаємодія між комп'ютерами в мережі.

Існують різні концепції побудови системи клієнт-сервер:

Слабкий клієнт – потужний сервер. У такій моделі вся обробка інформації перенесена на сервер, а у клієнта права доступу суворо обмежені. Сервер відправляє відповідь, яка не вимагає додаткової обробки. Клієнт взаємодіє з користувачем: складає та відправляє запит, приймає результат і виводить інформацію на екран.

Сильний клієнт – концепція, в якій частина обробки інформації надається клієнтові. У такому випадку сервер виступає сховищем даних, а вся робота по обробці та подання інформації переноситься на комп'ютер клієнта.

Обґрунтування вибору клієнт-серверної архітектури для розроблюваного інтернет-магазину базується на наступних перевагах:

Масштабованість: Клієнт-серверна архітектура дозволяє гнучко масштабувати додаток, окремо збільшуючи потужність серверної інфраструктури або розподіляючи навантаження між декількома серверами.

Розділення відповідальності: Розподіл функцій між клієнтом та сервером дозволяє краще розподіляти відповідальність між компонентами системи. Клієнтська сторона відповідає за користувацький інтерфейс та взаємодію з користувачем, тоді як серверна сторона забезпечує обробку даних.

Ефективність: Клієнт-серверна архітектура дозволяє оптимізувати використання ресурсів шляхом розподілу обробки між клієнтом і сервером. Це дозволяє знизити навантаження на клієнтські пристрої і забезпечити кращу продуктивність додатка.

Безпека: Клієнт-серверна архітектура дозволяє використовувати різні заходи безпеки, такі як аутентифікація, авторизація та шифрування, для захисту веб-додатка та даних користувачів.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 28   |

Підтримка багатокористувацькості: Завдяки клієнт-серверній архітектурі можна підтримувати багатокористувацькі сесії та обробляти запити від багатьох користувачів одночасно, що є важливим аспектом для інтернет-магазину з великою кількістю відвідувачів.

Розширюваність: Клієнт-серверна архітектура дозволяє додавати нові функціональні можливості до серверної сторони, не впливаючи на клієнтську частину. Це спрощує процес розширення та підтримки додатка в майбутньому.

Враховуючи всі переваги, клієнт-серверна архітектура є найбільш відповідною для розробки інтернет-магазину. Вона забезпечує гнучкість, масштабованість, ефективність, безпеку та можливість розширення, що дозволяє створити стабільний, функціональний та надійний веб-додаток для продажу товарів онлайн.

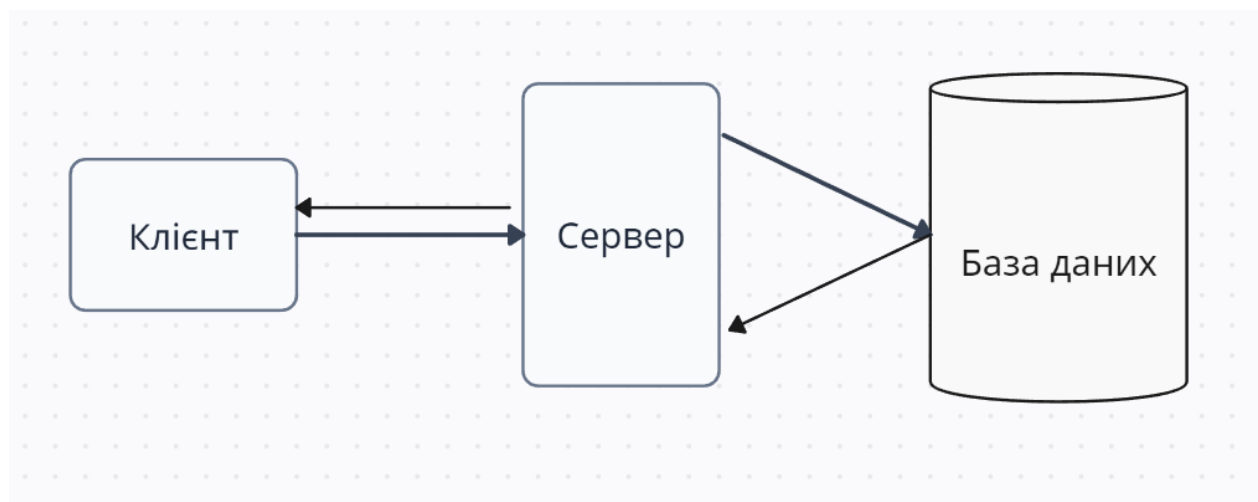


Рисунок 2.2 – Схема клієнт-серверної архітектури

## 2.2 Опис структури даних та моделі бази даних

Було створені таблицьки для коректної роботи програми відповідно до необхідних вхідних даних.

Відповідно до предметної області було створено такі головні таблиці як:

- Users;
- Roles;

- UserRoles;
- Product;
- ProductImage;
- Order;
- ShopingCart.

Users – таблиця потрібна для того, щоб зберігати всю інформацію про користувачів. Ця таблиця містить такі поля як:

- id – унікальний номер або ідентифікатор, що ідентифікує кожного користувача в базі даних;
- name рядок, що відображає ім'я користувача;
- streetadress - рядок, що містить інформацію про адресу користувача адресу користувача;
- city–рядок, що містить інформацію про місто користувача;
- postalcode – поштовий код користувача;
- username – логін користувача;
- email - рядок, що містить електронну адресу користувача;
- passwordhash – зашифрований рядок, що використовується для аутентифікації користувача;
- phonenumber – телефонний номер користувача.

Roles – таблиця що зберігає інформацію про ролі які можуть отримувати користувачі веб-застосунку. Ця таблиця містить такі поля як:

- Id – унікальний ідентифікатор ролі;
- Name – рядок, що відображає назву ролі.

UserRoles – допоміжна таблиця що зберігає інформацію про роль користувача. Ця таблиця містить такі поля як:

- UserId – ідентифікатор користувача в базі даних;
- RoleId – унікальний ідентифікатор ролі.

Product – таблиця у якій зберігається вся інформацію про товари. Ця таблиця містить такі поля як:

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|      |      |          |        |      |                               | 30   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

- Id – унікальний ідентифікатор товару;
- Name – назва товару;
- Description – детальний опис товару;
- Screen – розмір екрану, та його роздільна здатність;
- Camera – характеристика камер телефону;
- RAM – об'єм оперативної пам'яті телефону;
- Memory – об'єм пам'яті телефону;
- CPU – характеристики процесору телефону;
- OS – інформація про операційну систему пристрою;
- Price – ціна товару.

ProductImage – таблиця у якій зберігаються зображення товарів. Ця таблиця містить такі поля як:

- Id – унікальний ідентифікатор зображення;
- ImageUrl – посилання на зображення;
- ProductId – унікальний ідентифікатор товару.

Order – таблиця у якій зберігаються інформація про замовлення. Ця таблиця містить такі поля як:

- Id – унікальний ідентифікатор замовлення;
- UserId – ідентифікатор користувача в базі даних;
- ProductId – унікальний ідентифікатор товару;
- OrderDate – дата та час замовлення;
- ShippingDate – дата та час доставки;
- OrderStatus – статус замовлення;
- PaymentDate – статус оплати замовлення.

ShoppingCart – таблиця що зберігає у собі інформацію про вміст кошика користувачів. Ця таблиця містить такі поля як:

- Id – унікальний ідентифікатор кошика;
- ProductId – унікальний ідентифікатор товару

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|      |      |          |        |      |                               | 31   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |



Для опису сутностей призначені атрибути. На ER-діаграмі сутність зображується у вигляді прямокутника, всередині якого міститься ім'я сутності у формі іменника в однині. ER-діаграми поділяються на концептуальні та фізичні. На відміну від фізичних, у концептуальних ER-діаграмах не враховуються особливості конкретної бази даних. Згодом сутності концептуальних ER-діаграм стають таблицями, атрибути - стовпцями, а зв'язки реалізуються шляхом міграції ключових атрибутів батьківських сутностей та створення зовнішніх ключів. Розроблена ER-діаграма подана на рисунку 2.3.

### 2.3 Проектування серверної частини веб-додатка

Проектування серверної частини веб-додатку є важливим етапом у розробці сучасних веб-систем. Цей процес включає в себе визначення архітектури, вибір технологій, налаштування інфраструктури та розробку серверного програмного забезпечення.

Вибір серверного середовища для веб-додатку інтернет-магазину для продажу телефонів є важливим етапом проектування, оскільки воно визначає, як додаток буде функціонувати, які можливості будуть доступні розробникам і які вимоги до сервера.

Існує кілька популярних варіантів серверних середовищ, які можна використовувати для розробки веб-додатків. Нижче наведено два з них, які можуть бути варто розглянути для інтернет-магазину з продажу телефонів:

LAMP (Linux, Apache, MySQL, PHP): LAMP є одним з найпоширеніших серверних середовищ для веб-розробки. Він включає операційну систему Linux, веб-сервер Apache, базу даних MySQL та мову програмування PHP. Apache є стабільним і потужним веб-сервером, який забезпечує обробку запитів клієнтів та доставку веб-сторінок. PHP є популярною мовою програмування для серверного кодування, а MySQL - надійною системою керування базами даних.

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                        | 33   |

Використання LAMP надає розробникам широкий набір інструментів та документації для розробки веб-додатків.

MEAN/MERN (MongoDB, Express, Angular/React, Node.js): MEAN або MERN стеки є популярними серверними середовищами для розробки сучасних веб-додатків. Вони використовують MongoDB як базу даних, Express.js або React.js для побудови серверної частини, а Angular або React для розробки користувацького інтерфейсу. Node.js дозволяє виконувати серверний код JavaScript. Цей стек відкриває широкі можливості для створення швидких та масштабованих веб-додатків, а також дозволяє використовувати одну мову програмування на всіх рівнях розробки.

Вибір серверного середовища ASP.NET для веб-додатку інтернет-магазину для продажу телефонів є відмінним варіантом, особливо якщо ви вже маєте досвід розробки на платформі Microsoft або вам більш комфортно працювати з технологіями .NET. Ось кілька переваг використання серверного середовища ASP.NET для вашого веб-додатку:

- Масштабованість: ASP.NET забезпечує масштабованість, що дозволяє вашому інтернет-магазину ефективно працювати з ростом трафіку та об'єму даних. Він підтримує розподілені системи, навантажувальне балансування та інші механізми, що сприяють швидкодії та надійності додатка;

- Безпека: ASP.NET має вбудовану систему безпеки, що дозволяє захистити ваш інтернет-магазин від різноманітних атак, включаючи хакерські атаки, перекриття, внедрення зловмисного коду та інші загрози. Ви можете використовувати механізми аутентифікації, авторизації, шифрування даних та інші заходи для забезпечення безпеки вашого магазину;

- Інтеграція з іншими сервісами Microsoft: Якщо ви вже використовуєте інші продукти або сервіси Microsoft, такі як SQL Server для бази даних або Azure для хостингу, то ASP.NET надає пряму інтеграцію з цими рішеннями. Це спрощує розробку, управління та масштабування вашого інтернет-магазину;

– Багатофункціональність: ASP.NET пропонує багатий набір функціональностей та інструментів, що дозволяють вам швидко розробити різноманітні функції вашого інтернет-магазину, такі як обробка платежів, управління замовленнями, взаємодія з поштовими службами, аналітика та інше;

– Розширюваність: ASP.NET має велику спільноту розробників і підтримку від Microsoft, що дозволяє вам знайти багато готових компонентів, бібліотек і розширень для швидкого розширення функціональності вашого інтернет-магазину.

Загалом, використання серверного середовища ASP.NET для вашого веб-додатку дозволить вам розробити потужний, безпечний та гнучкий інтернет-магазин для продажу телефонів, використовуючи різноманітні можливості технології .NET.

## 2.4 Проектування клієнтської частини веб-додатка

Усі Інтернет-платформи мають загальноприйняті шаблони структури, які складаються з декількох частин і є актуальними для всіх сторінок Інтернет-платформи. Перед появою HTML5, всю розмітку здійснювалося за допомогою елементів <div>, до яких привласнювалися класи та ідентифікатори. Це допомагало структурувати та наочно розміщувати всі елементи, такі як нижні та верхні колонтитули, бокові панелі, різні навігаційні панелі та багато іншого.

З появою нового стандарту HTML5 з'явилися й нові елементи для структурування, угруповання різного контенту та розмітки текстового вмісту. Ці нові елементи допомогли покращити структуру Інтернет-платформи. Елементи можна стилізувати за власним бажанням, оскільки не задано жодних правил для відображення зовнішнього вигляду.

Проте, для деяких сучасних браузерів елементи HTML5 не завжди є зрозумілими. Якщо задати просто ширину та висоту таких елементів, вони можуть неправильно відобразитися у браузері. Щоб вирішити цю проблему,

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 35   |

необхідно для всіх використовуваних елементів HTML5 вказати {display: block;}.

Веб-додаток загалом складається з трьох основних частин:

Хедер (шапка) - це верхня частина сторінки, яка зазвичай розташовується в тегах <header>. Хедер є першою і однією з ключових частин сайту, оскільки користувач звертає на нього увагу при першому відвідуванні. Грамотне оформлення цієї частини має велике значення, оскільки в ній розміщуються назва сайту, коротка інформація, кнопки авторизації та реєстрації. Також у хедері можуть бути розміщені кнопки навігації або вони можуть бути виділені в окрему частину.

Навігація по сторінці - зазвичай ця частина розташовується під хедером або в межах самого хедеру, а оформлення головної навігації на сторінці здійснюється за допомогою тегу <nav>. Доброю практикою є використання елементів списку для розміщення навігаційних посилань.

Основна частина - це головна частина сайту, яка займає найбільший простір на сторінці. Вся необхідна та цікава для користувача інформація розташовується в цій частині. Текстовий контент зазвичай поміщається в тег <article>, який використовується для структурування блоку з контентом. Цей тег може бути використаний незалежно від контексту сторінки і розміщуватися в інших частинах сторінки.

Сайдбари або колонки з віджетами: Сайдбари є бічними панелями, які відображають графічні елементи для управління програмним забезпеченням або відображення додаткової інформації. Вони знаходяться збоку (праворуч або ліворуч) на сайті. Сайдбар не є обов'язковим, але він часто зустрічається на різних сайтах і має інформативну цінність. Для кожного окремого елемента сайдбара використовується тег <aside>.

Футер або підвал сайту: Футер є нижнім колонтитулом сторінки і зазвичай розміщується у нижній частині. Він містить інформацію про автора, авторські права, дату публікації, контактну інформацію та іншу правову інформацію.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 36   |

HTML5 має багато інших тегів, які допомагають розмістити різну інформацію на сторінці. Згадані вище теги представляють основну структуру сайту. Різноманітні теги допомагають поліпшити структуру сайту і зробити його більш привабливим для пошукових систем. Це, в свою чергу, сприяє покращенню рейтингу сайту в пошукових результатах.

Щоб представити загальний вигляд структури сайту з усіма частинами було створено рисунок 2.4.

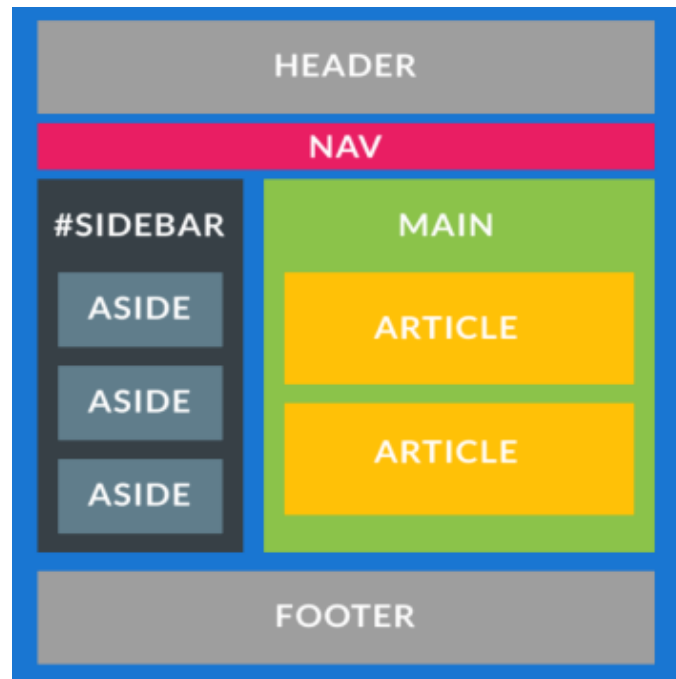


Рисунок 2.4 – Схематична структура сторінки HTML 5

## 2.5 Створення макета веб-застосунка та дизайн

Для проектування інтерфейсу користувача доцільно користуватись прототипуванням (розробкою макету ресурсу). Існує динамічне та статичне прототипування. Це два різних підходи до розробки прототипів інтерфейсу користувача. Кожен з цих підходів має свої переваги та використовується на різних етапах проектування.

Статичне прототипування використовується для створення статичних зображень або наборів зображень, що відображають зовнішній вигляд та

структуру інтерфейсу. Цей підхід може використовувати графічні редактори або спеціалізовані інструменти для проектування інтерфейсів. Статичні прототипи можуть бути представлені у вигляді макетів, зображень екранів або навіть простих наборів веб-сторінок. Вони зазвичай не мають функціональності та не можуть взаємодіяти з користувачем. Статичні прототипи часто використовуються на початкових етапах проекту для визначення загального вигляду та розташування елементів інтерфейсу.

Динамічне прототипування включає розробку інтерактивних прототипів, які можуть взаємодіяти з користувачем. Цей підхід застосовується для відображення поведінки та функціональності інтерфейсу. Для створення динамічних прототипів можуть використовуватися спеціалізовані інструменти, які дозволяють додавати імітовані дії, переходи між екранами, анімацію та взаємодію з реальними елементами інтерфейсу. Динамічні прототипи дають змогу перевірити функціональність, потоки користувача та інші аспекти взаємодії з інтерфейсом ще до його реалізації. Це дозволяє зробити корекції та вдосконалення до остаточного вигляду інтерфейсу.

Обидва підходи мають свої переваги і використовуються на різних етапах проектування. Статичне прототипування дозволяє швидко візуалізувати загальний вигляд інтерфейсу, визначити розташування елементів та структуру сторінок. Динамічне прототипування надає можливість перевірити функціональність та взаємодію з інтерфейсом, що допомагає виявити можливі проблеми та зробити необхідні зміни ще до реалізації проекту. Комбінація обох підходів може бути ефективною для досягнення оптимального результату при розробці інтерфейсу користувача.

Для взаємодії користувача з системою розроблено прототип інтерфейсу користувача, який дозволяє працювати з різними пристроями. Він повинен бути зручним у використанні та адаптивним. Робота з системою починається з головної сторінки(рисунок 2.5), де покупець може переглянути наявний список товару і ознайомитись з ним.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 38   |

Зверху всіх сторінок сайту знаходиться header на якому містяться кнопки навігації по сайту, такі як: Home, Manage Order, Shopping Cart, Login, Register. Якщо користувач має роль адміністратора в нього буде додаткова кнопка Content Managment, яка перекидує його до сторінки керування наповненням сайту.

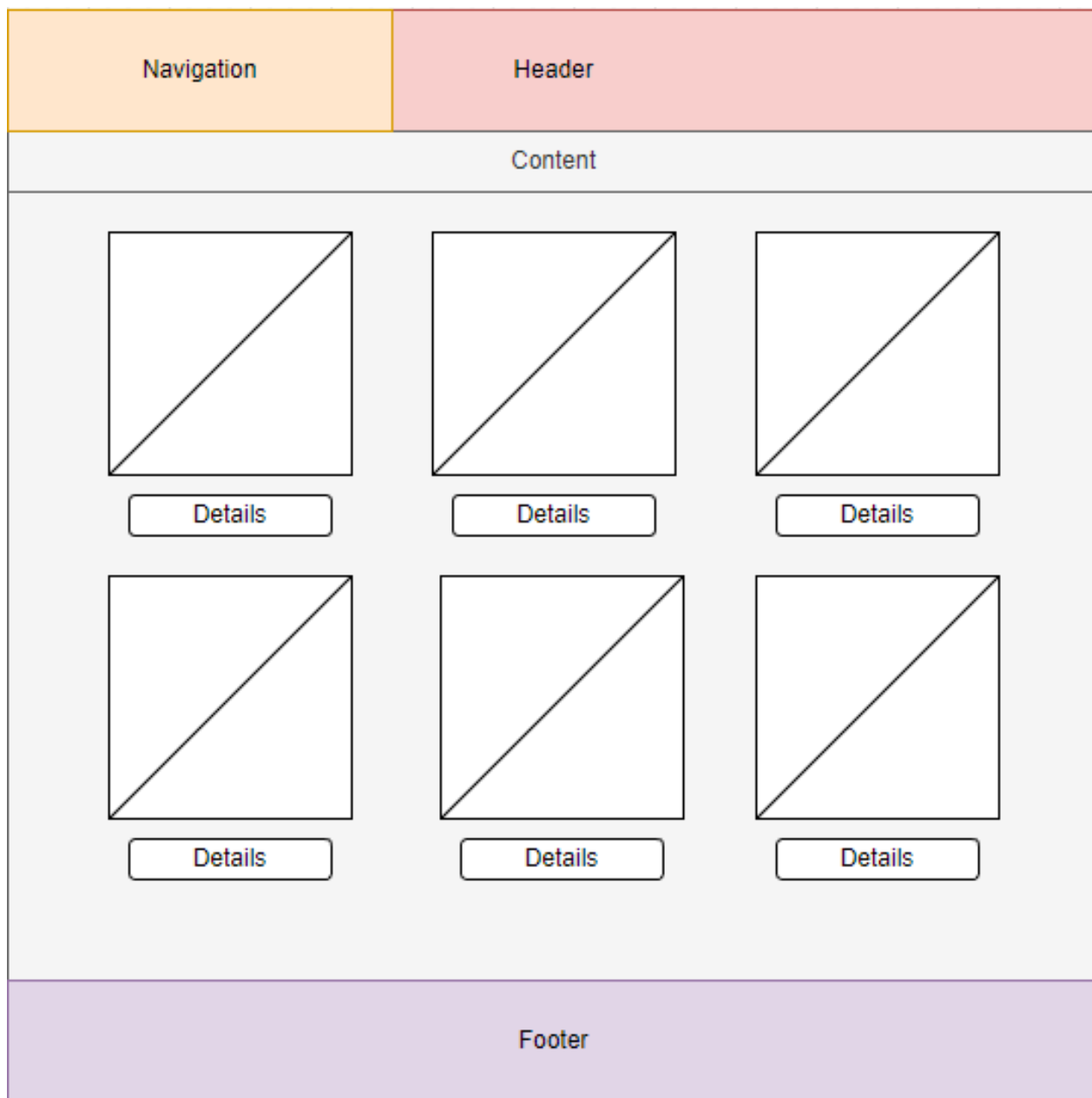


Рисунок 2.5 – Макет головної сторінки.

В середині сторінки знаходиться список товарів доступних для покупки. Для детального ознайомлення з характеристиками товару покупку потрібно

натиснути кнопку Details, після цього користувач переходить на сторінку з детальним описом товару(Рисунок 2.6).

Знизу знаходиться footer на якому розміщення інформація про контакти, а також посилання на соціальні мережі та довідкові сторінки інтернет-магазину.

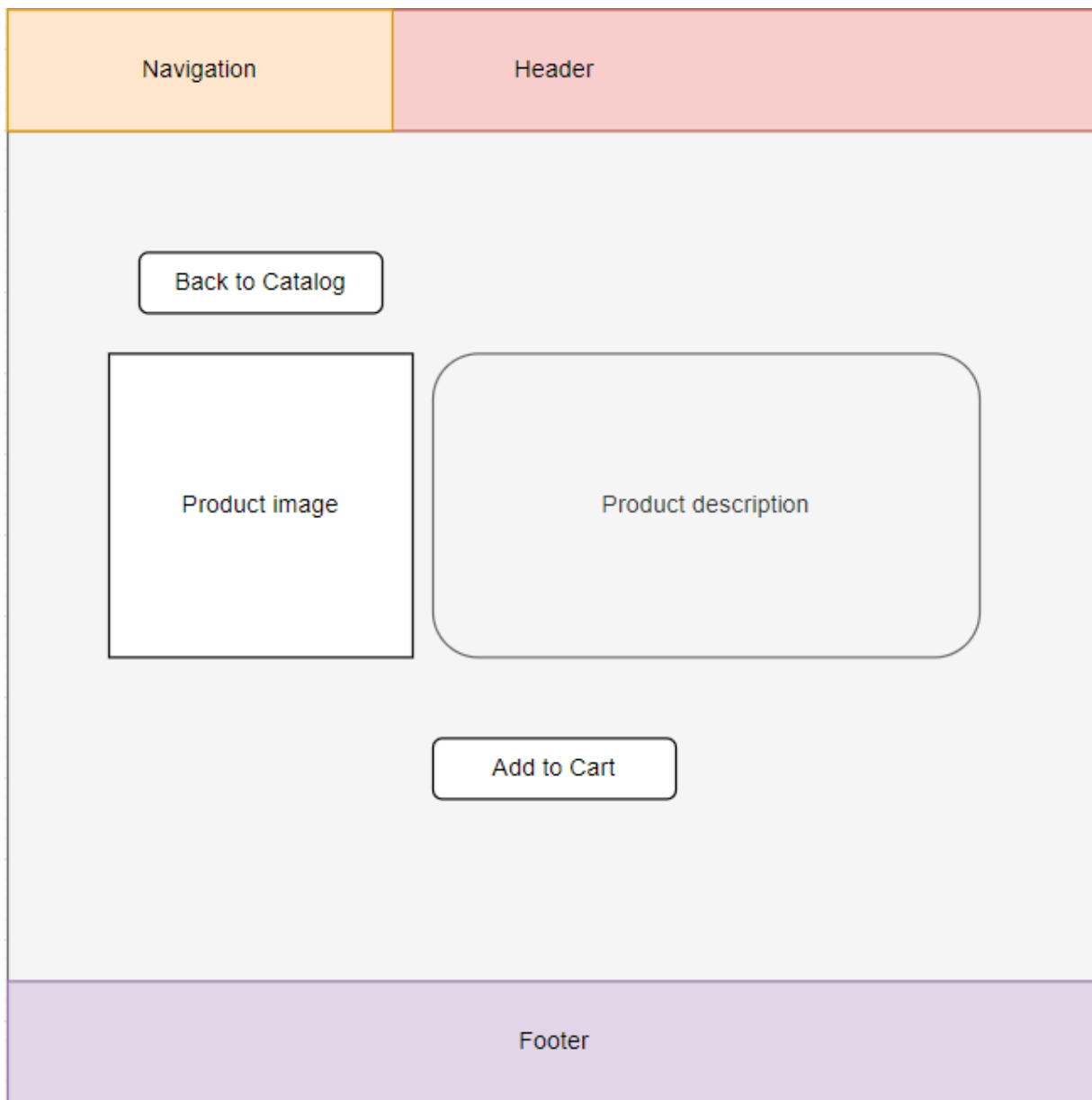


Рисунок 2.6 – Макет сторінки опису товару.

На цій сторінці знаходиться повний опис товару та його характеристик, знаходиться одне або кілька зображень товару, є кілька кнопок. Кнопка “Back to Catalog” повертає користувача назад на сторінку каталогу. Кнопка Add to Cart додає товар до кошика.

На сторінці кошику (Рисунок 2.7) користувач бачить список товарі, які він додав до нього. У списку вказано назву товару, його короткий опис та зображення товару. Також є кнопка Delete, що видаляє товар з кошику. Є навігаційна кнопка Continue Shopping, що повертає користувача до каталогу. Та кнопка Confirm Order, що підтверджує замовлення та переадресовує користувача на сторінку оплати.

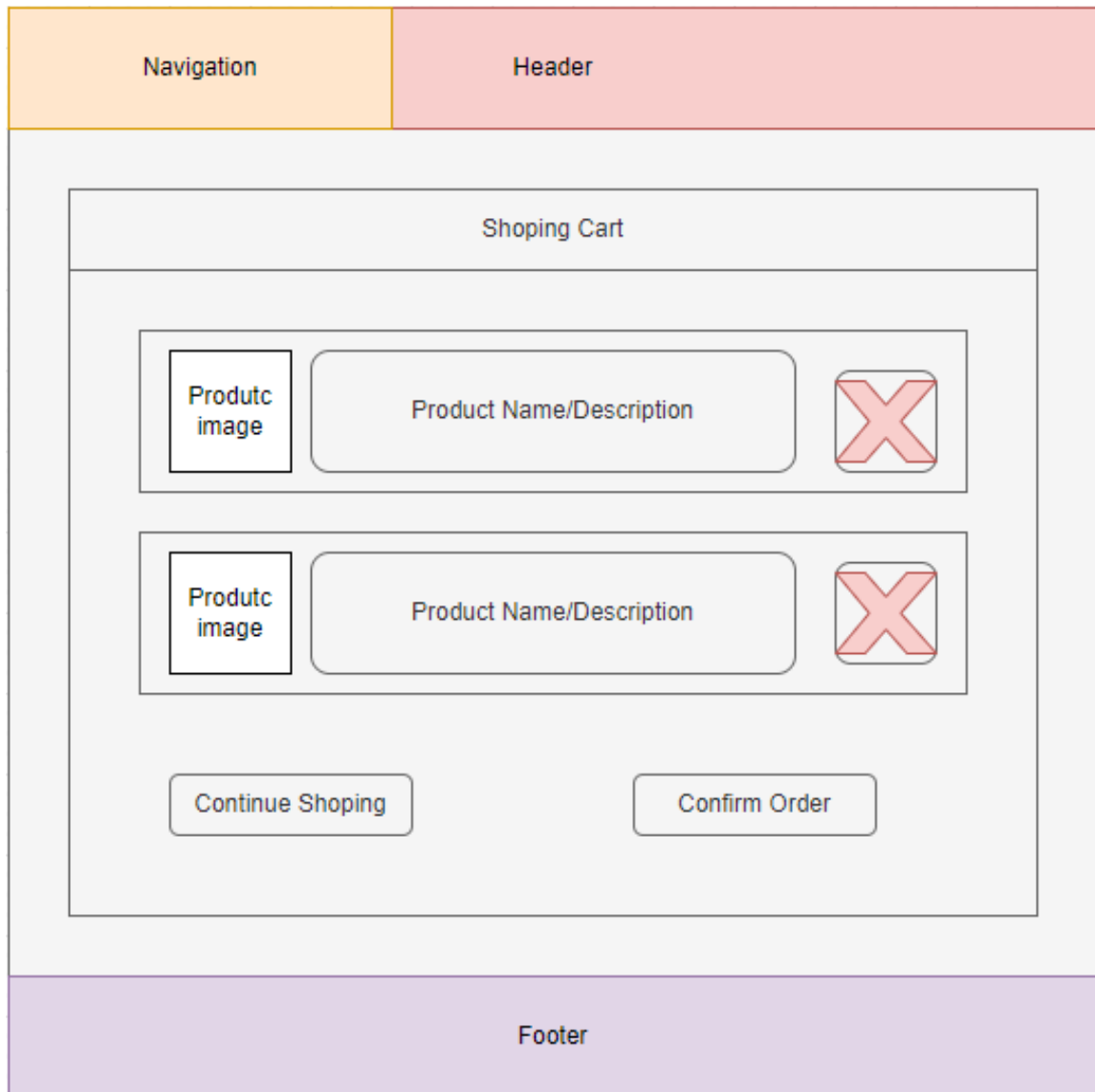


Рисунок 2.7 – макет сторінки кошика.

Адміністратор сайту має можливість перейти на сторінку керування товарами(Рисунок 2.8). В центрі цієї сторінки знаходиться список товарів, що розміщені у каталозі. Навпроти назви товару знаходяться кнопки Edit та Delete, що дозволяють редагувати чи видаляти товар.

Над списком товарів знаходиться поле пошуку, що спрощує пошук потрібної моделі, та кнопка Create New Product, що дозволяє створити нове оголошення.

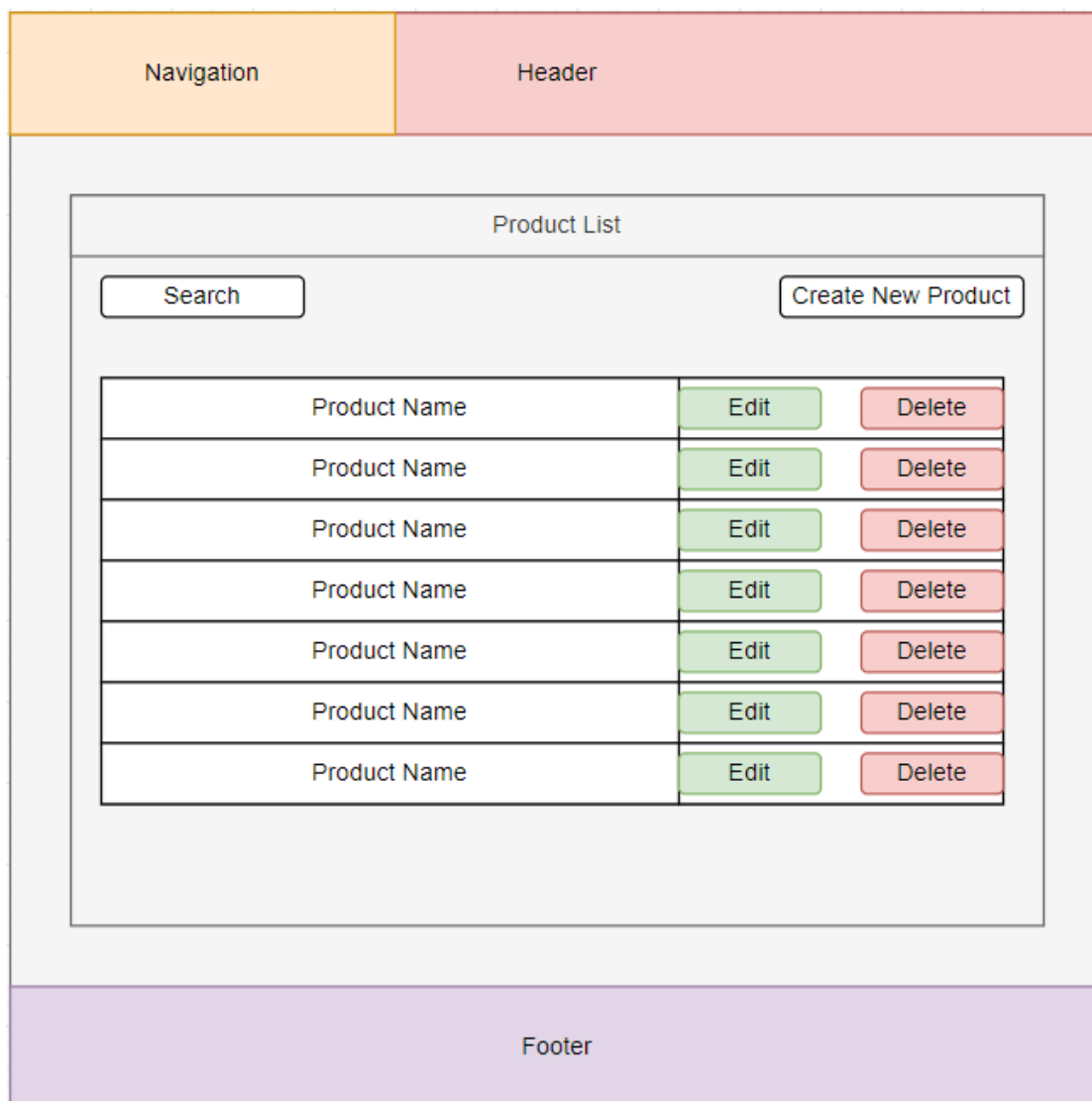


Рисунок 2.8 – Макет сторінки керування товарами.

## 2.6 Аналіз та вибір технологій і методів реалізації веб-додатка.

У даному розділі проведений огляд, аналіз та обґрунтування вибору технологій та засобів для розробки веб-додатка для продажу телефонів на платформі ASP.NET MVC. Розглянуті наступні складові і методи реалізації веб-додатка: мова розмітки HTML, складові DHTML, каскадні таблиці стилів (CSS),

об'єктна модель документа (DOM), методи створення JavaScript-сценаріїв у складі веб-сторінок, керування елементами сторінок на основі DOM, організація взаємодії з користувачем на основі подій, засоби бібліотеки jQuery для організації взаємодії JavaScript та HTML, мова програмування PHP та відповідні фреймворки, методи взаємодії веб-сценаріїв та СКБД, мова запитів до баз даних MySQL, підхід до побудови інтерактивних інтерфейсів користувача з використанням технології Аяx, протоколи та формати обміну даними у мережі Інтернет та методи їх обробки.

Мова розмітки HTML (Hypertext Markup Language) є основною мовою для створення веб-сторінок. Вона використовується для структуризації та відображення контенту на веб-сторінках. HTML є стандартною мовою для розробки веб-додатків і відповідає за відображення тексту, зображень, посилань та інших елементів на сторінці.

Складові DHTML (Dynamic HTML) є поєднанням HTML, CSS і JavaScript для створення динамічних інтерфейсів користувача. Використання DHTML дозволяє змінювати вигляд та поведінку елементів сторінки без перезавантаження всієї сторінки. DHTML може бути використаний для створення анімації, зміни вмісту сторінки на льоту та інших інтерактивних ефектів.

Каскадні таблиці стилів CSS (Cascading Style Sheets) використовується для опису вигляду і форматування веб-сторінок. Використання CSS дозволяє розділити структуру та зовнішній вигляд сторінки. Це дозволяє легко змінювати стиль і макет всіх сторінок веб-додатка, не змінюючи їх HTML-код.

Об'єктна модель документа DOM (Document Object Model) є стандартом, що визначає структуру та доступ до елементів веб-сторінки. Використання DOM дозволяє змінювати вміст та структуру сторінки за допомогою JavaScript. DOM надає доступ до елементів сторінки, їх властивостей та методів, що дозволяє здійснювати маніпуляції з вмістом сторінки з використанням JavaScript.

Методи створення JavaScript-сценаріїв у складі веб-сторінок JavaScript є мовою програмування, що використовується для створення динамічних функцій

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 43   |

та інтерактивності на веб-сторінках. JavaScript можна вставляти безпосередньо у HTML-код сторінки або зовнішніми файлами скриптів. Це дозволяє виконувати різні операції на стороні клієнта, такі як валідація даних, анімація, маніпуляція елементами сторінки та взаємодія з користувачем.

Керування елементами сторінок на основі DOM надає методи та властивості для керування елементами сторінки з використанням JavaScript. За допомогою DOM можна змінювати вміст, стилі, атрибути та положення елементів сторінки. Це дозволяє динамічно змінювати сторінку під час взаємодії з користувачем або при отриманні даних з сервера.

Організація взаємодії з користувачем на основі подій Веб-додаток може взаємодіяти з користувачем на основі подій, таких як клік мишею, введення тексту, натискання клавіші тощо. JavaScript дозволяє встановлювати обробники подій, які виконують певні дії при виникненні певних подій. Це дозволяє реалізувати інтерактивність та відповідати на дії користувача.

Засоби бібліотеки jQuery для організації взаємодії JavaScript та HTML jQuery є популярною бібліотекою JavaScript, яка спрощує взаємодію з HTML-документом, обробку подій, анімацію та звернення до сервера. Використання jQuery дозволяє скоротити кількість коду, необхідного для взаємодії з HTML та DOM, і забезпечити кросбраузерну сумісність.

Мова програмування PHP є мовою програмування, спеціально розробленою для розробки веб-додатків. Використання PHP дозволяє здійснювати обробку форм, доступ до бази даних, генерацію вмісту та інші серверні операції. Крім того, існують різні фреймворки, такі як Laravel, Symfony або CodeIgniter, які надають розширені можливості для швидкої та ефективної розробки веб-додатків на платформі PHP.

Методи взаємодії веб-сценаріїв та СКБД, мова запитів до баз даних MySQL Для забезпечення зберігання та доступу до даних веб-додатка використовуються системи керування базами даних (СКБД). MySQL є однією з популярних СКБД, яка підтримує мову запитів SQL. Веб-сценарії можуть взаємодіяти з базою

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 44   |

даних, виконуючи запити, отримуючи та зберігаючи дані, оновлюючи і видаляючи записи.

Підхід до побудови інтерактивних інтерфейсів користувача з використанням технології Ajax Ajax (Asynchronous JavaScript and XML) є технологією, яка дозволяє оновлювати частину сторінки без перезавантаження всієї сторінки. Використання Ajax дозволяє створювати інтерактивні веб-додатки, які реагують на дії користувача в реальному часі та взаємодіють з сервером для отримання або збереження даних.

У мережі Інтернет використовуються різні протоколи та формати для обміну даними. Наприклад, протокол HTTP використовується для передачі веб-сторінок та взаємодії з сервером. Формати даних, такі як JSON або XML, використовуються для представлення та обміну даними між клієнтом та сервером. Для обробки цих протоколів та форматів можуть використовуватися відповідні бібліотеки та методи програмування.

Вибір вищезазначених технологій і методів реалізації веб-додатка для продажу телефонів на платформі ASP.NET MVC обґрунтовується їх популярністю, широким спектром можливостей та підтримкою у веб-розробці. Комбінація HTML, CSS і JavaScript дозволяє створити зовнішній вигляд та інтерактивність веб-сторінок. Використання PHP та фреймворків на цій мові програмування дозволяє реалізувати серверну логіку, включаючи обробку форм, доступ до бази даних та інші операції на серверному рівні. MySQL виступає як СКБД для зберігання та управління даними. Технологія Ajax дозволяє створити багатофункціональний та інтерактивний веб-додаток, який реагує на дії користувача без перезавантаження сторінки.

Всі ці технології та методи добре поєднуються між собою та забезпечують необхідні можливості для створення веб-застосунку для продажу телефонів на платформі ASP.NET MVC.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 45   |

## 2.7. Висновки до розділу “Проектування веб-застосунку”

Під час роботи над розділом 2 було виконано детальний аналіз та вибір архітектури веб-застосунку з метою забезпечення його ефективної роботи. Була розглянута схема бази даних (СБД) і проведена нормалізація СБД для оптимального збереження даних. Для візуалізації структури бази даних була побудована ER-діаграма, яка надає графічне представлення взаємозв'язків між таблицями.

Наступним кроком було спроектовано серверну та клієнтську частини веб-застосунку.

Завершальним етапом розділу було проведення аналізу та вибір методів та технологій для реалізації веб-застосунку. Було розглянуто різні підходи та інструменти, які дозволять ефективно реалізувати функціонал веб-застосунку. Враховуючи особливості проекту, були обрані оптимальні методи та технології, які забезпечать успішну реалізацію веб-застосунку з урахуванням вимог проекту та потреб користувачів.

|             |             |                 |               |             |                               |      |
|-------------|-------------|-----------------|---------------|-------------|-------------------------------|------|
|             |             |                 |               |             | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|             |             |                 |               |             |                               | 46   |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                               |      |

### 3. Програмна реалізація

#### 3.1 Розробка бази даних

Розділ «Розробка бази даних». є ключовим етапом у процесі створення інтернет-магазину з продажу мобільних телефонів. Його основна мета полягає у ретельному спроектовані структури та взаємозв'язків таблиць у базі даних, щоб забезпечити ефективну роботу та необхідний функціонал для магазину.

Для створення бази даних для веб-застосунку, ми будемо використовувати мову програмування C# та фреймворк ASP.NET MVC, який містить необхідні інструменти для взаємодії з базами даних. Одним з основних завдань цього етапу є вибір оптимальної системи управління базами даних (СУБД), яка зможе задовільнити всі потреби інтернет-магазину.

Однією з найпопулярніших систем управління базами даних (СУБД), яку ми обрали для нашого проекту, є Microsoft SQL Server. Ця СУБД відома своїм широким спектром функціональних можливостей, які дозволяють ефективно створювати та керувати базами даних. Вона також повністю підтримує мову запитів SQL і має надзвичайно зручну інтеграцію з фреймворком ASP.NET MVC.

У базі даних цього веб-застосунку будуть зберігатися основні сутності, які включають в себе різноманітну інформацію про товари, клієнтів, замовлення та додаткові дані, пов'язані з ними. Наприклад, для таблиці «Users» ми можемо використовувати різні поля, що містять важливі дані про користувачів. Серед цих полів буде присутній унікальний ідентифікатор користувача, який дозволить однозначно відокремити кожного користувача в системі. Крім того, у таблиці будуть зберігатися дані, такі як ім'я користувача, електронна пошта та інша відповідна інформація, необхідна для повноцінного представлення користувачів у системі. (рис. 3.1).

|             |             |                 |               |             |                               |      |
|-------------|-------------|-----------------|---------------|-------------|-------------------------------|------|
|             |             |                 |               |             | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|             |             |                 |               |             |                               | 47   |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                               |      |

|   | Column Name          | Data Type     | Allow Nulls                         |
|---|----------------------|---------------|-------------------------------------|
| 🔑 | Id                   | nvarchar(450) | <input type="checkbox"/>            |
|   | Discriminator        | nvarchar(MAX) | <input type="checkbox"/>            |
|   | Name                 | nvarchar(MAX) | <input checked="" type="checkbox"/> |
|   | StreetAddress        | nvarchar(MAX) | <input checked="" type="checkbox"/> |
|   | City                 | nvarchar(MAX) | <input checked="" type="checkbox"/> |
|   | State                | nvarchar(MAX) | <input checked="" type="checkbox"/> |
|   | PostalCode           | nvarchar(MAX) | <input checked="" type="checkbox"/> |
|   | UserName             | nvarchar(256) | <input checked="" type="checkbox"/> |
|   | Email                | nvarchar(256) | <input checked="" type="checkbox"/> |
|   | EmailConfirmed       | bit           | <input type="checkbox"/>            |
|   | PasswordHash         | nvarchar(MAX) | <input checked="" type="checkbox"/> |
|   | PhoneNumber          | nvarchar(MAX) | <input checked="" type="checkbox"/> |
|   | PhoneNumberConfirmed | bit           | <input type="checkbox"/>            |
|   | TwoFactorEnabled     | bit           | <input type="checkbox"/>            |

Рисунок 3.1 – Таблиця Users в Базі Даних.

### 3.2 Розробка програмних модулів

Цей розділ присвячений розробці програмних модулів для веб-застосунку. Використовуючи мову програмування C# та фреймворк ASP.NET MVC, ми маємо можливість розробити потужні та функціональні модулі, які забезпечують правильну роботу сервісу та зручний користувацький інтерфейс.

Один з головних необхідних модулів, які потрібно створити, це модуль авторизації та реєстрації(рисунок). Цей модуль має за мету забезпечити безпеку та контроль доступу до певних функціональних можливостей веб-застосунку. Він надає користувачам можливість створювати свої особисті облікові записи, входити до них, змінювати персональні дані та керувати особистими налаштуваннями. Також було реалізовано можливість виходу з облікового запису.

```

public class AccountController : Controller
{
    private readonly UserManager<ApplicationUser> _userManager;
    private readonly SignInManager<ApplicationUser> _signInManager;

    Ссылка 0
    public AccountController(UserManager<ApplicationUser> userManager, SignInManager<ApplicationUser> signInManager)
    {
        _userManager = userManager;
        _signInManager = signInManager;
    }

    [HttpGet]
    Ссылка 0
    public IActionResult Register()
    {
        return View();
    }

    [HttpPost]
    Ссылка 0
    public async Task<IActionResult> Register(UserRegistrationModel model)
    {
        if (ModelState.IsValid)
        {
            var user = new ApplicationUser { Username = model.Username, Email = model.Email };
            var result = await _userManager.CreateAsync(user, model.Password);

            if (result.Succeeded)
            {
                await _signInManager.SignInAsync(user, isPersistent: false);
                return RedirectToAction("Index", "Home");
            }

            foreach (var error in result.Errors)
            {
                ModelState.AddModelError("", error.Description);
            }
        }
    }

    return View(model);
}

```

Рисунок 3.2 – реалізація реєстрації користувача.

Ще один необхідний модуль – модуль керування товаром (рисунок). Модуль керування товарами дозволяє адміністраторам і власникам магазину додавати нові товари до каталогу. Вони можуть ввести всі необхідні характеристики товару, такі як назва, опис, ціна, зображення тощо. Це дозволяє поповнювати асортимент магазину і представляти нові товари користувачам. Модуль дозволяє змінювати характеристики товарів, такі як ціна, опис, наявність тощо. Адміністратори можуть вносити зміни в інформацію про товари, якщо, наприклад, ціна змінилась або товар більше не доступний. Це дозволяє підтримувати актуальну інформацію про товари в магазині.

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
|      |      |          |        |      |                        | 49   |
| Змн. | Арк. | № докум. | Підпис | Дата |                        |      |

```

public class AccountController : Controller
{
    private readonly SignInManager<ApplicationUser> _signInManager;

    Ссылка: 0
    public AccountController(SignInManager<ApplicationUser> signInManager)
    {
        _signInManager = signInManager;
    }

    [HttpGet]
    Ссылка: 0
    public IActionResult Login()
    {
        return View();
    }

    [HttpPost]
    Ссылка: 0
    public async Task<IActionResult> Login(LoginViewModel model, string returnUrl)
    {
        if (ModelState.IsValid)
        {
            var result = await _signInManager.PasswordSignInAsync(model.Username, model.Password, model.RememberMe, LockoutOnFailure: false);

            if (result.Succeeded)
            {
                if (!string.IsNullOrEmpty(returnUrl) && Url.IsLocalUrl(returnUrl))
                {
                    return Redirect(returnUrl);
                }
                else
                {
                    return RedirectToAction("Index", "Home");
                }
            }

            ModelState.AddModelError(string.Empty, "Invalid login attempt.");
        }

        return View(model);
    }

    [HttpPost]
    Ссылка: 0
    public async Task<IActionResult> Logout()
    {
        await _signInManager.SignOutAsync();
        return RedirectToAction("Index", "Home");
    }
}

```

Рисунок 3.3- реалізація авторизації та виходу з аккаунту користувача.

Модуль керування товарами дозволяє вести облік запасів товарів. Адміністратори можуть встановлювати кількість доступних товарів, вказувати, коли товар закінчується або повністю розпродано. Це допомагає уникнути продажу товарів, які вже недоступні, і підтримувати актуальність інформації про наявність товарів магазину.

Отже, модуль керування товарами є важливою складовою інтернет-магазину, оскільки він дозволяє зручно та ефективно управляти асортиментом, оновлювати інформацію про товари та забезпечувати високу якість обслуговування користувачів.

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                        | 50   |

```

public IActionResult Upsert(ProductVM productVM, List<IFormFile> files)
{
    if (ModelState.IsValid)
    {
        if (productVM.Product.Id == 0) {
            _unitOfWork.Product.Add(productVM.Product);
        }
        else {
            _unitOfWork.Product.Update(productVM.Product);
        }
    }

    _unitOfWork.Save();

    string wwwRootPath = _webHostEnvironment.WebRootPath;
    if (files != null)
    {
        foreach(IFormFile file in files)
        {
            string fileName = Guid.NewGuid().ToString() + Path.GetExtension(file.FileName);
            string productPath = @"images\products\product-" + productVM.Product.Id;
            string finalPath = Path.Combine(wwwRootPath, productPath);

            if (!Directory.Exists(finalPath))
                Directory.CreateDirectory(finalPath);
            using (var fileStream = new FileStream(Path.Combine(finalPath, fileName), FileMode.Create)) {
                file.CopyTo(fileStream);
            }

            ProductImage productImage = new() {
                ImageUrl = @"\" + productPath + @"\" + fileName,
                ProductId=productVM.Product.Id,
            };
            if (productVM.Product.ProductImages == null)
                productVM.Product.ProductImages = new List<ProductImage>();

            productVM.Product.ProductImages.Add(productImage);

            _unitOfWork.Product.Update(productVM.Product);
            _unitOfWork.Save();
        }
        TempData["success"] = "Product created/updated successfully";
        return RedirectToAction("Index");
    }
    else
    {
        productVM.CategoryList = _unitOfWork.Category.GetAll().Select(u => new SelectListItem
        {
            Text = u.Name,
            Value = u.Id.ToString()
        });
        return View(productVM);
    }
}

```

Рисунок 3.4 – Реалізація додавання/редагування товару.

Ще один модуль додавання товару у кошик є важливою складовою частиною інтернет-магазину. Він забезпечує користувачам зручний спосіб вибору та збереження товарів, які вони бажають придбати. Модуль додавання товару у кошик надає користувачам простий та зрозумілий спосіб вибрати

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                        | 51   |

бажані товари. Вони можуть легко переглядати асортимент, додавати товари у кошик і продовжувати шукати інші. Це дозволяє користувачам зручно скласти своє замовлення. Користувачі можуть додавати товари у кошик, але не негайно здійснювати покупку. Це дає їм можливість зберегти свій вибір і продовжити покупки в майбутньому.

Модуль дозволяє зберігати обрані товари користувача, навіть якщо він закриває веб-сайт або виходить з облікового запису. Після додавання товарів у кошик, користувачі мають можливість переглянути своє замовлення та здійснити покупку. Модуль додавання товару у кошик надає можливість керувати обраними товарами, змінювати їх кількість або видаляти певні товари перед оформленням замовлення.

```
[HttpPost]
[Authorize]
Ссылка: 0
public IActionResult Details(ShoppingCart shoppingCart)
{
    var claimsIdentity = ((ClaimsIdentity)User.Identity);
    var userId = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier).Value;
    shoppingCart.ApplicationUserId= userId;

    ShoppingCart cartFromDb = _unitOfWork.ShoppingCart.Get(u=>u.ApplicationUserId == userId &&
    u.ProductId==shoppingCart.ProductId);

    if (cartFromDb != null) {
        //shopping cart exists
        cartFromDb.Count += shoppingCart.Count;
        _unitOfWork.ShoppingCart.Update(cartFromDb);
        _unitOfWork.Save();
    }
    else {
        //add cart record
        _unitOfWork.ShoppingCart.Add(shoppingCart);
        _unitOfWork.Save();
        HttpContext.Session.SetInt32(SD.SessionCart,
        _unitOfWork.ShoppingCart.GetAll(u => u.ApplicationUserId == userId).Count());
    }
    TempData["success"] = "Cart updated successfully";

    return RedirectToAction(nameof(Index));
}
```

Рисунок 3.5 – Реалізація додавання товару до кошика.

### 3.3 Керівництво користувача

При запуску інтернет-магазину користувач потрапляє на головну сторінку, яка є основною точкою входу до нашого сайту. Головна сторінка представляє собою центральний пункт навігації та інформації для користувача. Вона включає різні елементи і можливості, які допомагають зручно взаємодіяти з нашим інтернет-магазином.

Основна частина головної сторінки - це каталог товарів. Він представлений у вигляді сітки з мініатюрами фотографій телефонів, де кожен товар має свою назву, короткий опис та ціну. Користувач може переглядати товари, прокручуючи сторінку вниз.

Кожен товар у каталозі має посилання або кнопку «Details», яка дозволяє користувачеві перейти на сторінку товару з більш докладною інформацією, такою як технічні характеристики, фотографії з різних кутів тощо.

У нижній частині головної сторінки розташовані додаткові блоки інформації. Ці посилання допомагають користувачам знайти додаткову інформацію про компанію, її політику, зв'язатися з нами або знайти довідку. Зовнішній вигляд головної сторінки, зображений на рисунку 3.6.

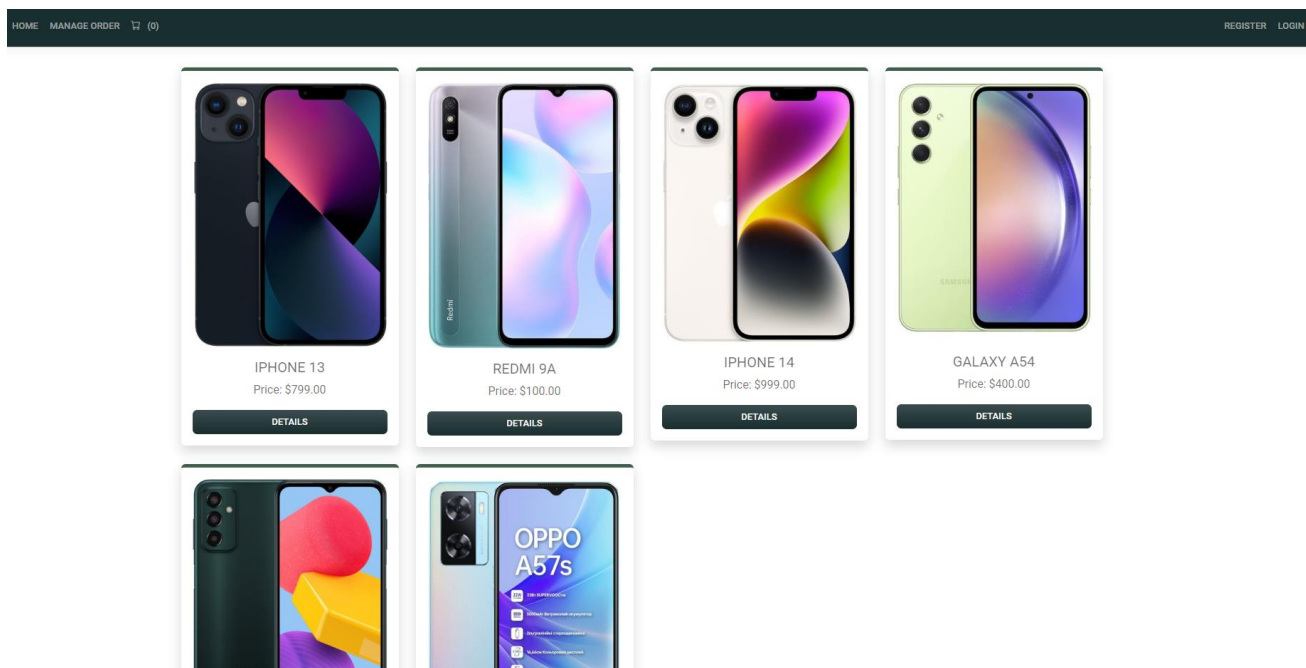


Рисунок 3.6 – Загальний вигляд головної сторінки.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <b>КвРІПЗ.190142.01.19.ПЗ</b> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 53   |

При натисканні користувачем кнопки «Login» відбувається дія, яка відкриває сторінку авторизації. На цій сторінці користувачеві потрібно ввести свої аутентифікаційні дані, які складаються з логіну та пароля. Ці дані необхідні для встановлення ідентифікації користувача. Зовнішній вигляд сторінки авторизації зображено на рисунку 3.7

Рисунок 3.7 – Вигляд сторінки авторизації.

Коли користувач вперше відвідує ваш магазин і не має створеного раніше акаунту, він має можливість натиснути кнопку «Register». Це відкриє сторінку реєстрації, де користувачу потрібно буде ввести всі необхідні дані для створення нового акаунту.

На сторінці створення акаунту присутні наступні поля для заповнення:

Ім'я користувача: Тут користувач повинен ввести своє ім'я, яке буде використовуватися для ідентифікації в системі.

Електронна пошта: Користувач повинен ввести свою дійсну електронну адресу, на яку будуть надсилатися повідомлення та сповіщення щодо його акаунту.

Пароль: Користувач повинен вибрати надійний пароль, який буде використовуватися для захисту його акаунту. Пароль повинен містити мінімум

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <b>КвРІПЗ.190142.01.19.ПЗ</b> | Арк. |
|      |      |          |        |      |                               | 54   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

вісім символів і комбінацію букв, цифр і спеціальних символів для забезпечення безпеки.

Підтвердження пароля: Користувач повторно вводить пароль для підтвердження, що він введений правильно. Це допомагає уникнути помилок при введенні паролю.

Інші дані, Адресу доставки, контактний номер телефону. Ці дані корисні для обробки замовлень та забезпечення зручного обслуговування користувача.

Після того, як користувач введе всі необхідні дані для реєстрації, сторінка перевіряє правильність введених даних.

Якщо всі дані введені правильно і пройшли успішну перевірку, користувачу буде створено новий акаунт, і він буде перенаправлений на головну сторінку магазину. Рисунок 3.8, відображає вигляд сторінки реєстрації.

The image shows a web registration form. At the top, there is a navigation bar with 'HOME', 'MANAGE ORDER', and a cart icon with '(0)'. On the right side of the navigation bar are 'REGISTER' and 'LOGIN' links. The main content area has a dark header with the word 'Register' in white. Below the header, it says 'Create a new account.' The form consists of several input fields: 'Email', 'Full Name', 'Phone Number', 'Password', 'Confirm Password', 'Street Address', 'City', 'State', and 'Postal Code'. Below these fields is a dark 'REGISTER' button. Underneath the button, there is an 'or' separator and two social login buttons: 'FACEBOOK' and 'MICROSOFT'. At the bottom of the form, there is a footer with contact information: 'Contact phone +123123123; Email mobilephone@shop.com'.

Рисунок 3.8 – Вигляд сторінки реєстрації.

У інтернет-магазині є можливість редагування облікового запису користувача та підключення подвійної автентифікації.

Після того, як користувач зареєструвався на веб-сайті, йому надається обліковий запис, з яким він може увійти та використовувати сервіс. Однак, часом користувач може потребувати змінити свої особисті дані або налаштування свого профілю.

Для того, щоб змінити дані свого облікового запису, користувач повинен перейти на головну сторінку сервісу та знайти кнопку, яка вказує на його профіль. Після кліку на цю кнопку користувач буде перенаправлений на сторінку керування профілем.

На сторінці керування профілем користувач може побачити свої поточні дані, такі як ім'я, електронну пошту, пароль і інші особисті відомості. Він може редагувати ці дані, вибираючи відповідні поля для редагування. Наприклад, користувач може змінити своє ім'я, введенням нового імені у відповідному полі. Він також може оновити свою електронну пошту або змінити пароль.

Крім зміни особистих даних, на сторінці керування профілем користувач також може знайти опцію підключення подвійної автентифікації. Подвійна автентифікація - це додатковий рівень захисту облікового запису, який вимагає введення додаткового коду після введення звичайного пароля. Цей додатковий код може бути надісланий на мобільний телефон користувача або генеруватися за допомогою спеціального програмного забезпечення.

Якщо користувач бажає підключити подвійну автентифікацію, він зазвичай має натиснути на відповідну кнопку на сторінці керування профілем. Потім йому буде запропоновано вибрати метод підключення подвійної автентифікації, наприклад, через SMS-повідомлення або використання спеціального додатку. Користувач повинен буде надати додаткові дані або налаштувати відповідний додаток, щоб активувати подвійну автентифікацію для свого облікового запису.

Після зміни даних або підключення подвійної автентифікації користувач може зберегти зміни та повернутися до свого профілю або основного інтерфейсу сервісу. Відтепер його обліковий запис буде оновлений з відповідними змінами, а подвійна автентифікація буде встановлена для додаткового захисту. Зовнішній вигляд сторінки редагування профілю зображено на рисунку 3.9

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 56   |

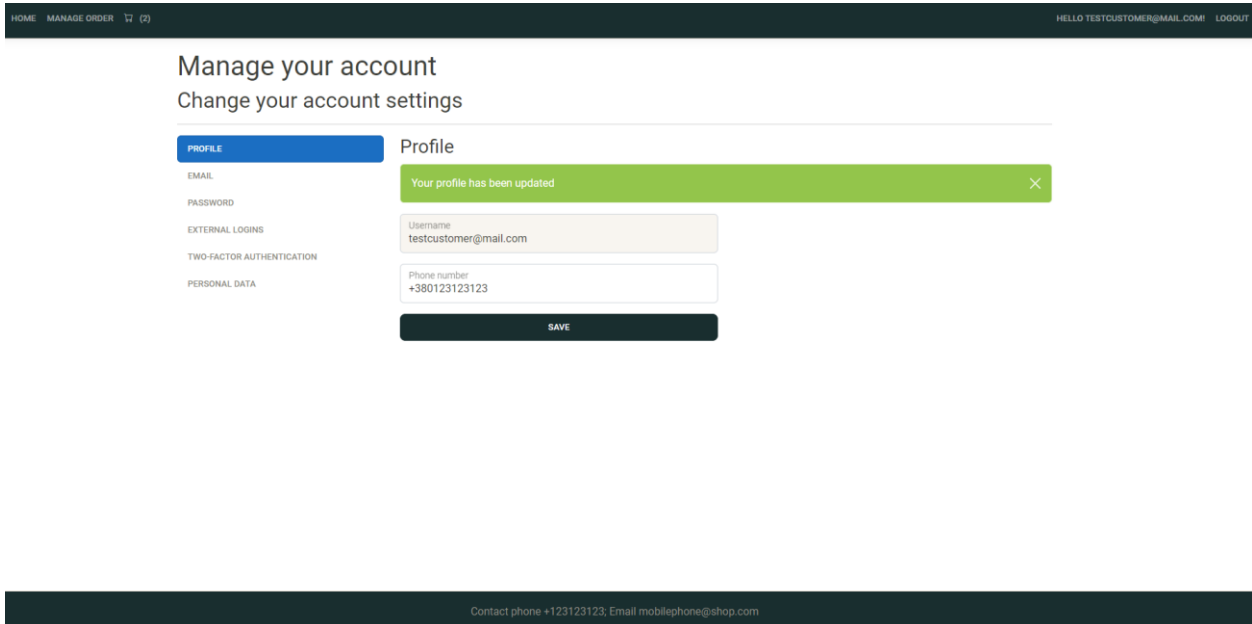


Рисунок 3.9 – Вигляд сторінки редагування профілю.

При натисненні кнопки Detail на сторінці каталогу, користувач потрапляє на сторінку детального опису товару, на якій вказано технічні характеристики мобільного пристрою та його ціну. За допомогою кнопки Add to Cart користувач може додати товар до свого кошику. Вигляд сторінки товару зображено на рисунку 3.10.

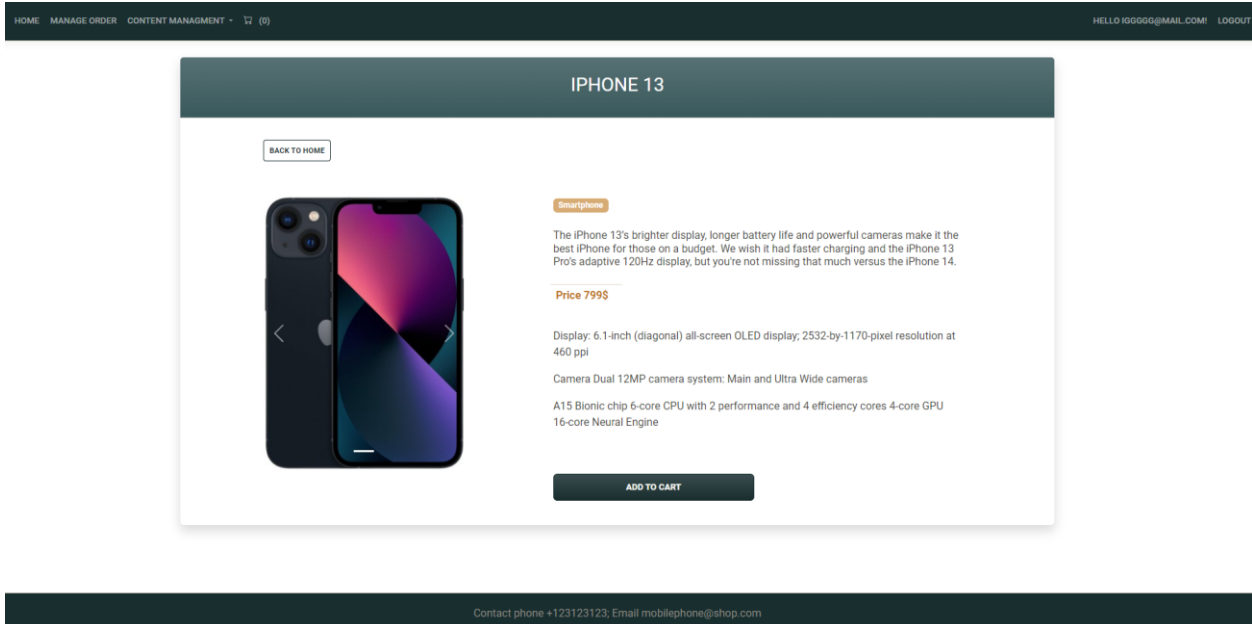


Рисунок 3.10 – Вигляд сторінки опису товару.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <b>КвРІПЗ.190142.01.19.ПЗ</b> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 57   |

Веб-застосунок запам'ятовує товари, які користувач додає у кошик. Користувач може переглянути цей список натиснувши на кнопку кошика. Він побачить список товарів які додав, їх ціну, кількість та загальну суму замовлення, якщо користувача все влаштовує він може підтвердити замовлення натиснувши кнопку Summary. Вигляд сторінки кошик зображено на рисунку 3.11.

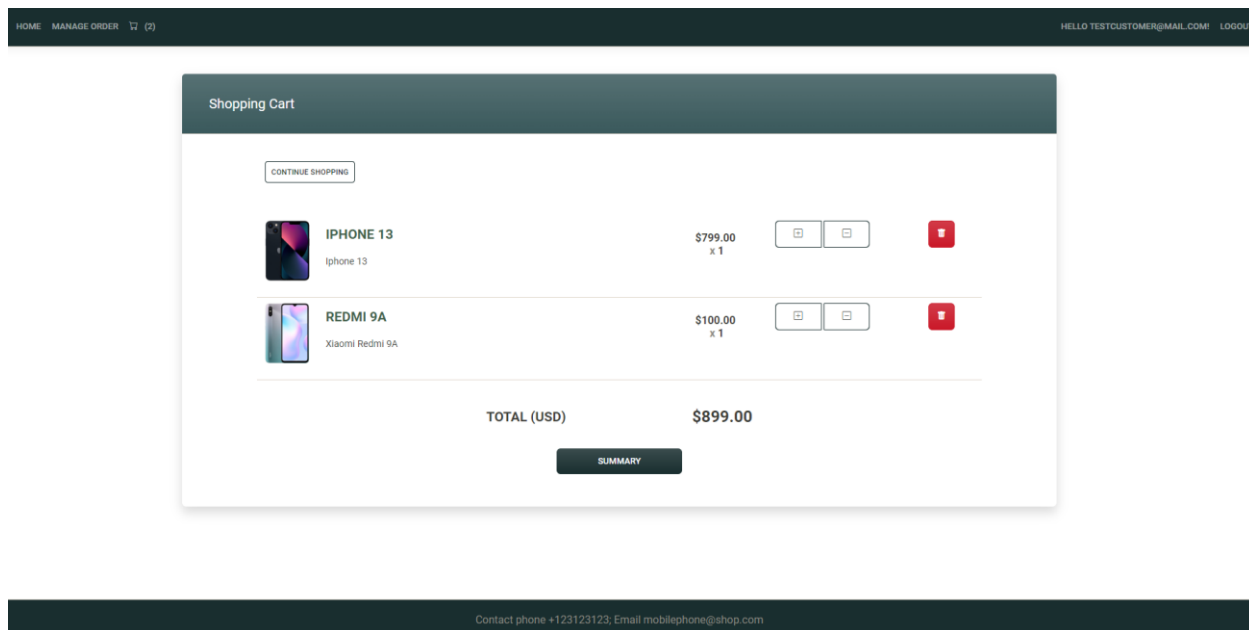


Рисунок 3.11 – Вигляд сторінки кошику.

Головна сторінка адміністратора сайту має особливі функції та можливості, які відрізняють її від користувацької. На головній сторінці адміністратора присутнє додаткове навігаційне меню, яке називається «Content Managment» або «Управління контентом». Це меню надає доступ до інструментів та функцій, що дозволяють адміністратору керувати вмістом сайту. Рисунок 3 ілюструє зовнішній вигляд головної сторінки адміністратора. Відкривши це меню користувач може перейти на сторінку керування товарами натиснувши кнопку Product. На цій сторінці знаходиться список всіх товарів у магазині, є можливість зручного пошуку цих товарів, та кнопки Edit та Delete, які відповідають за редагування та видалення товарів відповідно. Зовнішній вигляд сторінки редагування товарів зображено на рисунку 3.12.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <b>КвРІПЗ.190142.01.19.ПЗ</b> | Арк. |
|      |      |          |        |      |                               | 58   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

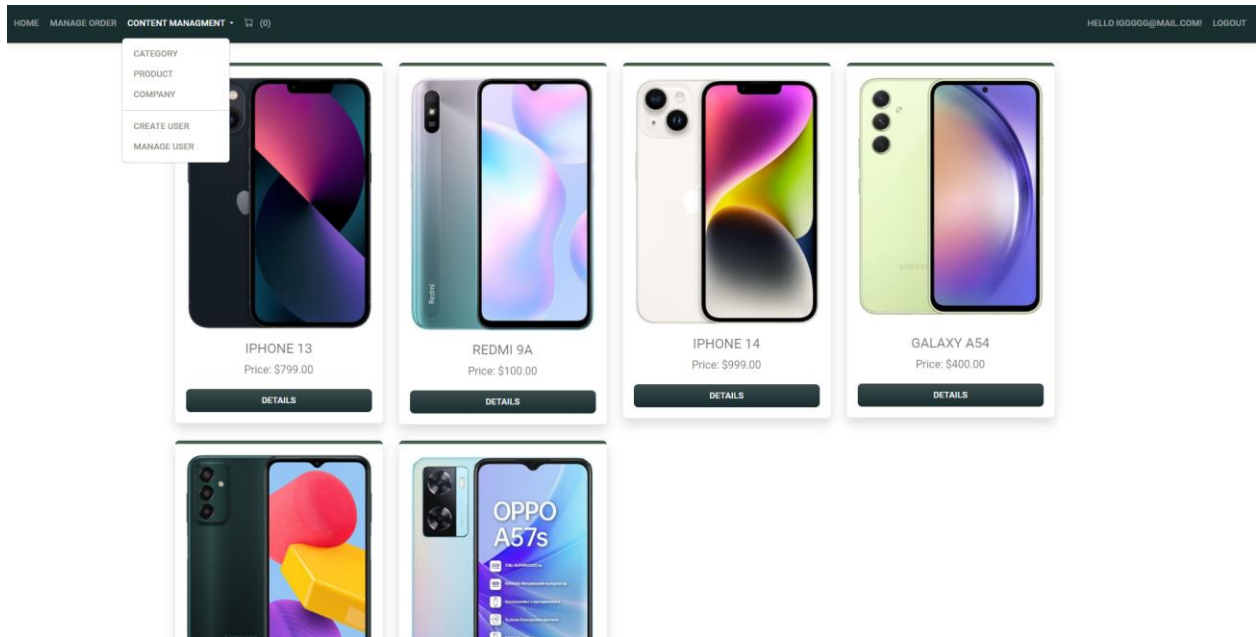


Рисунок 3.12 – Вигляд головної сторінки для адміністратора.

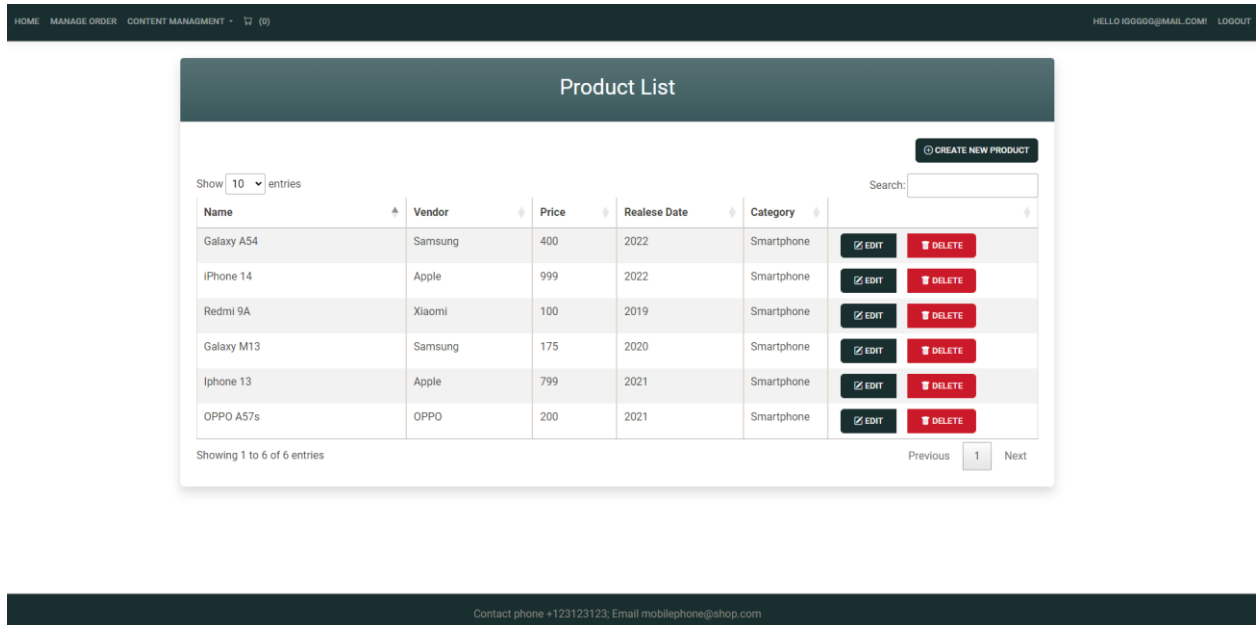


Рисунок 3.13 – Загальний вигляд сторінки керування товарами.

Натиснувши кнопку Edit адміністратор потрапляє на сторінку редагування товару. На цій сторінці знаходяться поля з характеристиками та ціною товару, які можна редагувати. Також на цій сторінці адміністратор може змінити або додати нові зображення товару. На рисунку 3.13 зображено загальний вигляд сторінки редагування товару.

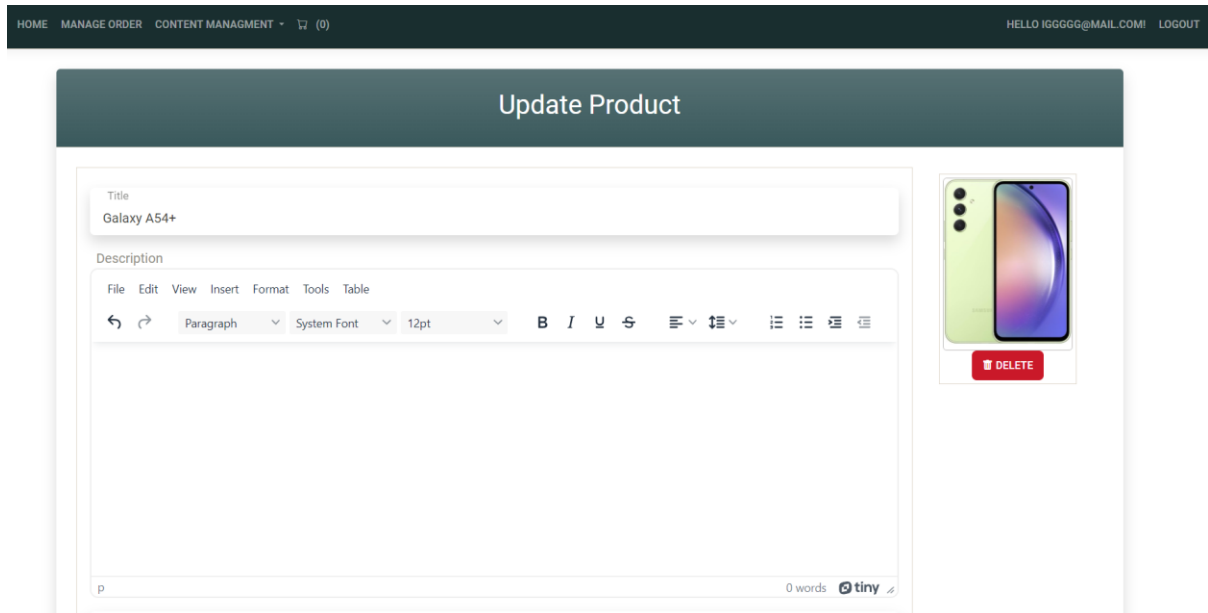


Рисунок 3.14 – Загальний вигляд сторінки редагування товару

Якщо внесення змін в опис товару відбулось успішно, адміністратор отримає відповідне повідомлення у правому верхньому куті екрану. Це повідомлення зображено на рисунку 3.15

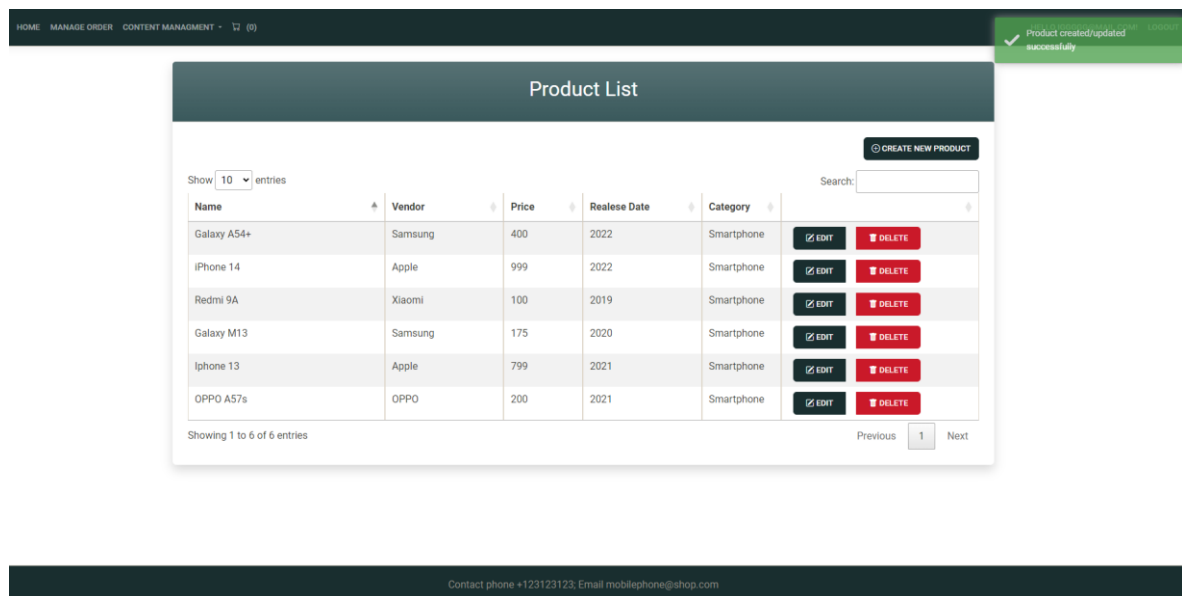


Рисунок 3.15 – Повідомлення про успішне додавання товару.

Також адміністратор може створити новий обліковий запис який матиме роль адміністратора. Для цього йому потрібно натиснути кнопку Create User, після чого він потрапить на сторінку створення аккаунта, де може ввести дані

для створення нового облікового запису. На рисунку 3.1 зображено загальний вигляд сторінки створення облікового запису адміністратора.

Рисунок 3.16 – вигляд сторінки створення облікового запису адміністратора.

Також адміністратор може редагувати вже створені облікові записи користувачів. Для цього йому потрібно натиснути кнопку Manage User, після чого він потрапить на сторінку керування користувачами.

| Name  | Email                 | Phone          | Company | Role     |                   |
|-------|-----------------------|----------------|---------|----------|-------------------|
| adada | iggggg@mail.com       | +3801231231231 |         | Admin    | UNLOCK PERMISSION |
| Test  | testcustomer@mail.com | +380123123123  |         | Customer | UNLOCK PERMISSION |

Рисунок 3.17 – Сторінка керування користувачами.

Тут він може редагувати вже створені облікові записи та видавати чи забирати у них права адміністратора. Також є можливість заблокувати обліковий запис, після чого з його допомогою не можна буде авторизуватись на сайті

### 3.4 Технічні характеристики веб-застосунка.

Для забезпечення правильного функціонування веб-застосунку з клієнтської сторони, необхідно враховувати певні технічні вимоги. Розглянемо ці вимоги більш детально та переформулюємо їх:

- Операційна система: Веб-застосунок підтримує операційну систему Windows 7 або більш пізню версію. Для оптимальної роботи рекомендується використовувати найновішу доступну версію операційної системи;
- Інтернет-браузер: Веб-застосунок може коректно працювати в будь-якому сучасному інтернет-браузері, який підтримує веб-стандарти, такі як Chrome, Firefox, Safari або Edge;
- Стабільне підключення до мережі Інтернет: Для безперебійної роботи веб-застосунку необхідно мати стабільне та надійне підключення до Інтернету.
- 2-ядерний процесор: Мінімальним вимогам до процесора для клієнтської сторони веб-застосунку відповідає наявність двох ядер. Це дозволить забезпечити ефективну обробку запитів та відгук на клієнтському пристрої;
- 1 ГБ оперативної пам'яті: Для нормальної роботи веб-застосунку клієнтському пристрою потрібно мати щонайменше 1 ГБ оперативної пам'яті. Це забезпечить достатній ресурс для виконання операцій та завантаження ресурсів.

Тепер розглянемо технічні вимоги для правильного функціонування серверної сторони веб-застосунку:

- Операційна система: Серверна частина веб-застосунку підтримує операційну систему Windows 7 або більш пізню версію;
- Встановлений ASP.NET та необхідні фреймворки: Для коректної роботи серверної сторони веб-застосунку потрібно мати встановлений фреймворк ASP.NET, а також всі необхідні додаткові фреймворки, такі як ASP.NET MVC та Entity Framework;

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 62   |

- SQL Server Management Studio: Для управління базою даних веб-застосунку необхідно мати встановлений SQL Server Management Studio, що дозволяє зручно працювати з базою даних;
- SQL Server 8.0 та вище: Для зберігання та управління даними веб-застосунку використовується SQL Server версії 8.0 або вище;
- Підключення до мережі Інтернет зі швидкістю не менше 15Мбіт/с: Для надійної роботи серверної сторони веб-застосунку потрібне стабільне підключення до мережі Інтернет зі швидкістю не менше 15 Мбіт/с;
- 4-ядерний процесор: Мінімальним вимогам до процесора для серверної сторони веб-застосунку відповідає наявність чотирьох ядер, що забезпечує потужність для обробки запитів та завантаження;
- 4ГБ оперативної пам'яті: Для нормальної роботи серверної сторони веб-застосунку необхідно мати щонайменше 4 ГБ оперативної пам'яті. Це забезпечить ефективну обробку запитів та завантаження ресурсів;
- 10ГБ внутрішньої пам'яті: Для зберігання даних та забезпечення їх доступу серверній стороні веб-застосунку потрібно мати наявність щонайменше 10 ГБ внутрішньої пам'яті.

Всі вказані вимоги є мінімальними для нормального функціонування веб-застосунку і враховуються для забезпечення оптимальної продуктивності та стабільності системи.

### 3.5. Аналіз методів тестування веб- застосунку та розробка тестів

Тестування веб-застосунків включає в себе різні методи та підходи, які спрямовані на перевірку функціональності, надійності, продуктивності та інших аспектів веб-додатків. Ось декілька основних методів тестування веб-застосунків:

Функціональне тестування: Цей метод перевіряє, чи працює веб-застосунок згідно з очікуваннями і відповідає функціональним вимогам. Це

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|      |      |          |        |      |                               | 63   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

включає тестування всіх функцій, таких як навігація, взаємодія з користувачем, обробка форм, перехід між сторінками та інші функції.

Тестування користувацького інтерфейсу (UI): Цей метод перевіряє, чи працює інтерфейс веб-застосунку правильно, а також чи є він зручним для користувачів. Тут оцінюється вигляд, розташування елементів, зручність навігації та інші аспекти, що стосуються користувацького досвіду.

Тестування продуктивності: Цей метод перевіряє, як швидко веб-застосунок відповідає на запити користувачів та обробляє навантаження. Включає тестування часу завантаження сторінок, швидкості відгуку, пропускну здатності та інших показників продуктивності.

Тестування сумісності: Цей метод перевіряє, як веб-застосунок працює на різних платформах, операційних системах, браузерах та пристроях. Включає тестування на різних версіях браузерів, розмірах екранів, мобільних пристроях тощо.

Тестування безпеки: Цей метод перевіряє, чи є веб-застосунок захищеним від потенційних загроз безпеки, таких як злам, витік інформації, вразливості веб-додатку та інші атаки. Включає перевірку наявності захисту від хакерських атак, перевірку правильності автентифікації та авторизації, обробку введення користувача та інші аспекти безпеки.

Автоматизоване тестування: Використання спеціальних інструментів і фреймворків для автоматизації процесу тестування веб-застосунків. Це дозволяє автоматично виконувати тестові сценарії, забезпечує більшу швидкість і ефективність тестування.

Тестування внутрішньої структури: Цей метод перевіряє внутрішню структуру веб-застосунку, таку як бази даних, взаємодія з сервером, логіку програми та інші аспекти. Включає тестування API, валідацію даних, тестування баз даних та інші аспекти.

Було проведено функціональне тестування сайту з метою перевірки його коректної роботи. У таблиці 3.1 наведені розглянуті тестові сценарії, які використовувалися для цього.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 64   |

Таблиця 3.1 – тестовий сценарій №1

| Дія  | Очікуваний результат  | Отриманий результат   |
|--|---|---|
| Запуск веб-застосунку                                | Відкриття початкової сторінки інтернет-магазину   | Сторінка відкрилася без затримок, усі дані коректно відобразилися   |
| Натискання кнопки «Login»                            | Відкриття сторінки веб-застосунку з формою авторизації  | Сторінка відкрилася, усі елементи форми авторизації відобразилися   |
| Вхід користувача до облікового запису                | Після заповнення усіх полів форми авторизації відкриється головна сторінка сайту з функціоналом авторизованого користувача              | Авторизація пройшла успішно, відкрилася головна сторінка, з'явилися кнопки «Shopping Cart» та «Profile»   |
| Отримання доступу до детальної характеристики товару | Після натискання кнопки «Details», відкриється сторінка з повною характеристикою мобільного пристрою                                    | Сторінка товару відкрилася, уся інформація про товар відобразилася на екрані. З'явилися кнопки «Add to cart» та «Back to Catalog»                           |
| Отримання доступу до детальної характеристики товару | Після натискання кнопки «Details», відкриється сторінка з повною характеристикою мобільного пристрою                                    | Сторінка товару відкрилася, уся інформація про товар відобразилася на екрані. Кнопки «Add to cart» та «Back to Catalog» відобразилися у відповідних місцях. |
| Натискання кнопки «Back to Catalog»                  | Відбудеться повернення користувача до головної сторінки інтернет-магазину   | Перехід до сторінки каталогу Користувач, який мав авторизацію, продовжив перебувати у статусі авторизованого  |
| Вихід користувача з облікового запису                | Після натискання кнопки «Logout», користувач залишиться на головній сторінці, але без доступу до функціоналу авторизованого користувача | Користувач залишився на головній сторінці, на місці кнопки «Profile» з'явилася кнопка «Login»   |

Після завершення тестування можна зробити висновок, що весь функціонал веб-застосунку відповідає всім вимогам до програмного

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                        | 65   |

забезпечення. Усі модулі застосунку працюють безперешкодно, і жодна проблема з їхнім функціонуванням не залишилася невирішеною.

### 3.6 Висновки до розділу Програмна реалізація

Під час розробки розділу 3 було пройдено кілька етапів, щоб забезпечити належну функціональність веб-застосунку. По-перше, була виконана конструкція фізичної бази даних, а також її успішне підключення до модулів застосунку. Цей процес включав створення таблиць з відповідними структурами і відношеннями, а також встановлення правильних залежностей між ними. Додатково, було забезпечено належну інтеграцію бази даних з різними модулями веб-застосунку, щоб забезпечити зручну та ефективну роботу системи. Наступним кроком було реалізовано різні модулі веб-застосунку, а також детально описано їх функціональність та роботу. Кожен модуль був розроблений з урахуванням вимог і функціональних можливостей, необхідних для веб-застосунку. Опис був створений з урахуванням потреб користувачів та документував усі можливості, які надаються кожним модулем. Крім того, додатково були включені пояснювальні рисунки для полегшення розуміння функцій та взаємозв'язків між модулями.

Останнім кроком було визначення та виконання тестів для перевірки роботи веб-застосунку. Це включало тестування різних аспектів функціональності застосунку

|             |             |                 |               |             |                               |      |
|-------------|-------------|-----------------|---------------|-------------|-------------------------------|------|
|             |             |                 |               |             | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                               | 66   |

## ВИСНОВКИ

Під час виконання кваліфікаційної нам вдалося досягти поставленої мети, яка полягала в розробці інтернет-магазину для продажу мобільних телефонів.

У першому розділі роботи проведено дослідження предметної області, визначено функціональні та нефункціональні вимоги до веб-застосунку. Також було сформовано технічне завдання та постановку задачі, проаналізовано наявні рішення у сфері інтернет магазинів, оцінено їх позитивні та негативні аспекти.

У другому розділі було проведено проектування інтернет-магазину, включаючи розробку архітектури, структури, бази даних та інтерфейсу користувача. Також здійснено аналіз і вибрано технології та методи реалізації, що забезпечать коректну роботу системи.

У третьому розділі ми здійснили програмну реалізацію інтернет магазину. Було представлено структуру та функціональне призначення модулів програмного засобу, розроблено та впроваджено базу даних. Також було проведено тестування створеного програмного засобу в широкому діапазоні. Були виконані різноманітні тест-кейси, що дозволило перевірити коректність роботи системи та виявити та усунути можливі дефекти.

В результаті виконання цієї кваліфікаційної роботи було досягнуто мети, інтернет магазин для продажу мобільних телефонів було успішно розроблено та протестовано.

|             |             |                 |               |             |                               |      |
|-------------|-------------|-----------------|---------------|-------------|-------------------------------|------|
|             |             |                 |               |             | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|             |             |                 |               |             |                               | 67   |
| <i>Змн.</i> | <i>Арк.</i> | <i>№ докум.</i> | <i>Підпис</i> | <i>Дата</i> |                               |      |

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Л. П. Бедратюк. Дипломний проект: методичні вказівки щодо його виконання для студентів спеціальності 121 «Інженерія програмного забезпечення» / Л. П. Бедратюк, Г. І. Радельчук, Ю. В. Форкун, О. М. Яшина. Хмельницький: ХНУ, 2020. – 77 с.
2. Андрющенко, О.О., Коваленко, В.В. "ASP.NET MVC. Практичні аспекти розробки веб-додатків" / О.О. Андрющенко, В.В. Коваленко. - Київ: Видавництво "Професіонал", 2014. - 352 с.
3. Барановський, В.І. "ASP.NET MVC 5: від початківця до професіонала" / В.І. Барановський. - Київ: Видавничий дім "Ін Юре", 2015. - 512 с.
4. Войтович, І.А. "ASP.NET MVC: від початківця до професіонала" / І.А. Войтович. - Київ: Комп'ютерний прес, 2012. - 360 с.
5. Гавриленко, В.В. "ASP.NET MVC 4: Розробка веб-додатків з використанням Entity Framework" / В.В. Гавриленко. - Київ: Видавничий дім "Ін Юре", 2013. - 384 с.
6. Данилюк, В.А. "ASP.NET MVC 5: Повний курс" / В.А. Данилюк. - Київ: Дім Книги, 2014. - 416 с.
7. Жовтобрюх, М.О. "ASP.NET MVC: Концепції та прийоми розробки веб-додатків" / М.О. Жовтобрюх. - Київ: Дім Книги, 2017. - 432 с.
8. Коломoeць, О.М. "ASP.NET MVC 5: Повний посібник з програмування" / О.М. Коломoeць. - Київ: ТОВ "Видавничий дім "Гельветика", 2016. - 464 с.
9. Люлько, В.І. "ASP.NET MVC 5: Розробка веб-додатків на мові С#" / В.І. Люлько. - Київ: Центр навчальної літератури, 2015. - 352 с.
10. Мельник, О.О., Черевко, І.В. "ASP.NET MVC: Потужний фреймворк для веб-розробки" / О.О. Мельник, І.В. Черевко. - Київ: Видавництво "Бук Трейд", 2011. - 240 с.

|      |      |          |        |      |                        |      |
|------|------|----------|--------|------|------------------------|------|
|      |      |          |        |      | КвРІПЗ.190142.01.19.ПЗ | Арк. |
|      |      |          |        |      |                        | 68   |
| Змн. | Арк. | № докум. | Підпис | Дата |                        |      |

11. Новіков, В.П. "ASP.NET MVC: Сучасні технології веб-програмування" / В.П. Новіков. - Київ: Дім Книги, 2016. - 304 с.
12. Олійник, А.Ю. "ASP.NET MVC 5: Захоплюючі проекти для професіоналів" / А.Ю. Олійник. - Київ: Видавничий дім "Ін Юре", 2017. - 400 с.
13. Петров, В.С. "ASP.NET MVC 4: Навчальний курс" / В.С. Петров. - Київ: Видавництво "Бук Трейд", 2014. - 320 с.
14. Романовський, М.О., Борисенко, О.В. "ASP.NET MVC 5: Професійний підхід" / М.О. Романовський, О.В. Борисенко. - Київ: ТОВ "Видавничий дім "Гельветика", 2015. - 560 с.
15. Семенов, І.А. "ASP.NET MVC: Кращі практики програмування веб-додатків" / І.А. Семенов. - Київ: Комп'ютерний прес, 2013. - 352 с.
16. Тарасюк, В.А. "ASP.NET MVC 5: Навчальний посібник" / В.А. Тарасюк. - Київ: Видавництво "Бук Трейд", 2016. - 288 с.
17. Усачов, А.В. "ASP.NET MVC 4: Програмування веб-додатків" / А.В. Усачов. - Київ: Дім Книги, 2013. - 336 с.
18. Федоренко, О.В. "ASP.NET MVC 5: Короткий курс" / О.В. Федоренко. - Київ: Комп'ютерний прес, 2014. - 224 с.
19. Хом'як, І.В. "ASP.NET MVC 5: Повна модель розробки" / І.В. Хом'як. - Київ: Видавництво "Професіонал", 2015. - 336 с.
20. Циліорик, А.А. "ASP.NET MVC 4: Навчальний курс" / А.А. Циліорик. - Київ: Видавничий дім "Ін Юре", 2013. - 304 с.
21. Червяков, Д.М., Іванов, В.П. "ASP.NET MVC: Розробка веб-додатків" / Д.М. Червяков, В.П. Іванов. - Київ: Дім Книги, 2012. - 400 с.
22. Шевчук, С.С. "ASP.NET MVC 4: Розробка веб-додатків на мові С#" / С.С. Шевчук. - Київ: ТОВ "Видавничий дім "Гельветика", 2013. - 416 с.
23. Шевчук, С.С., Кірієнко, В.О. "ASP.NET MVC 5: Розробка веб-додатків на мові С#" / С.С. Шевчук, В.О. Кірієнко. - Київ: ТОВ "Видавничий дім "Гельветика", 2016. - 560 с.
24. Adam Freeman. "Pro ASP.NET MVC 5" / Adam Freeman. - Київ: Apress, 2013. - 792 с.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                               | 69   |

25. Andrew Troelsen, Philip Japikse. "Pro ASP.NET MVC 5 Platform" / Andrew Troelsen, Philip Japikse. - Київ: Apress, 2014. - 936 с.

26. Jon Galloway, Brad Wilson, K. Scott Allen, David Matson. "Professional ASP.NET MVC 5" / Jon Galloway, Brad Wilson, K. Scott Allen, David Matson. - Київ: Wiley, 2014. - 624 с.

27. Scott Hanselman, Scott Hunter, David Matson. "Professional ASP.NET MVC 5" / Scott Hanselman, Scott Hunter, David Matson. - Київ: Wiley, 2014. - 624 с.

28. Adam Freeman, Steven Sanderson. "Pro ASP.NET MVC 4" / Adam Freeman, Steven Sanderson. - Київ: Apress, 2012. - 756 с.

29. Jon Galloway, Phil Haack, Brad Wilson, K. Scott Allen. "Professional ASP.NET MVC 4" / Jon Galloway, Phil Haack, Brad Wilson, K. Scott Allen. - Київ: Wiley, 2012. - 624 с.

30. Matthew MacDonald. "Pro ASP.NET MVC 4" / Matthew MacDonald. - Київ: Apress, 2012. - 756 с.

31. Steven Sanderson. "Pro ASP.NET MVC 3 Framework" / Steven Sanderson. - Київ: Apress, 2011. - 756 с.

32. Scott Guthrie, Scott Hanselman, Phil Haack, Steven Sanderson. "Professional ASP.NET MVC 3" / Scott Guthrie, Scott Hanselman, Phil Haack, Steven Sanderson. - Київ: Wiley, 2011. - 624 с.

33. Adam Freeman. "Pro ASP.NET MVC 3 Framework" / Adam Freeman. - Київ: Apress, 2011. - 756 с.

34. Andrew Troelsen, Steven Sanderson. "Pro ASP.NET MVC 2 Framework" / Andrew Troelsen, Steven Sanderson. - Київ: Apress, 2010. - 768 с.

35. Scott Guthrie, Scott Hanselman, Phil Haack, Steven Sanderson. "Professional ASP.NET MVC 2" / Scott Guthrie, Scott Hanselman, Phil Haack, Steven Sanderson. - Київ: Wiley, 2010. - 624 с.

36. Rob Conery, Scott Hanselman, Phil Haack, Scott Guthrie. "Professional ASP.NET MVC" / Rob Conery, Scott Hanselman, Phil Haack, Scott Guthrie. - Київ: Wiley, 2009. - 576 с.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КєРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|      |      |          |        |      |                               | 70   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

37. Steve Sanderson. "Pro ASP.NET MVC Framework" / Steve Sanderson. - Київ: Apress, 2009. - 550 с.

38. Rob Conery, Scott Hanselman, Phil Haack, Scott Guthrie. "Professional ASP.NET MVC 1.0" / Rob Conery, Scott Hanselman, Phil Haack, Scott Guthrie. - Київ: Wiley, 2009. - 576 с.

39. Dino Esposito. "Programming Microsoft ASP.NET MVC" / Dino Esposito. - Київ: Microsoft Press, 2014. - 528 с.

40. Imar Spaanjaars. "Beginning ASP.NET MVC 1.0" / Imar Spaanjaars. - Київ: Wrox, 2009. - 480 с.

|      |      |          |        |      |                               |      |
|------|------|----------|--------|------|-------------------------------|------|
|      |      |          |        |      | <i>КвРІПЗ.190142.01.19.ПЗ</i> | Арк. |
|      |      |          |        |      |                               | 71   |
| Змн. | Арк. | № докум. | Підпис | Дата |                               |      |

# ДОДАТОК А

## Технічне завдання

### Введення

Робота виконується в рамках проекту розробки веб-застосунку для продажу мобільних телефонів, у вигляді сайту, що дозволяє здійснювати онлайн покупки.

### 1 Підстава для розробки

Підставою для розробки є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри інженерії програмного забезпечення. Найменування розробки: Інтернет-магазин для продажу мобільних телефонів.

### 2 Призначення розробки

Призначення веб-застосунку для продажу мобільних телефонів полягає у представленні каталогу товарів, інформації про товар(опис та характеристики), здійснення покупок товарів та оформлення доставки придбаних товарів. Користувачами веб-застосунку є користувач мережі Інтернет

Функціональне призначення полягає у: додаванні товарів до каталогу, редагування інформації про товар, реєстрації та авторизації користувачів, оформленні замовлення та оплати замовлення.

Редагування наповнення та будь-якої інформації на сайті може відбуватися адміністратором відповідно до вимог. Для використання ПЗ не потрібно ніякого додаткового налаштування і воно працює на усіх комп'ютерах, які відповідають технічним вимогам.

### 3 Вимоги до програми

#### 3.1 Вимоги до функціональних характеристик

Веб-застосунок для продажу мобільних телефонів має забезпечувати наступні функції:

- додавання товарів до каталогу;
- реєстрація та авторизація у системі;
- налаштування особистого профілю(редагування даних, зміна паролю);
- відображення товарів та інформації про них;
- фільтрація для пошуку товарів;
- додавання товарів до кошику;
- оформлення замовлення;

### **3.2 Вимоги до надійності**

Система має виконувати наступні вимоги до надійності:

- розмежування прав користувачів у системі;
- валідація на стороні сервера для контролю інформації, що вводиться та виключення введення помилкових даних користувачем;
- підтримання захисту від несанкціонованого доступу;
- розмежування прав користувачів у системі;

### **3.3 Умови експлуатації**

Умови експлуатації мають відповідати санітарним і технічним нормам експлуатації персонального комп'ютера, при температурі та відносній вологості навколишнього середовища, визначених для персональної обчислювальної техніки згідно з ГОСТ 15150-69 [7].

Для обслуговування веб-застосунку допускаються тільки спеціально навчені адміністратори або розробники. До користування веб-застосунком допускаються користувачі мережі Інтернет.

### **3.4 Вимоги до складу та параметрів технічних засобів**

- ОС Windows 7 та вище;
- Дискретний або інтегрований відеоадаптер;
- Оперативна пам'ять 1ГБ;
- Жорсткий диск 1ГБ;
- Девайси(миша, клавіатура);
- Монітор;
- Під'єднання до мережі Інтернет;

### **3.5 Вимоги до інформаційної та програмної сумісності**

Для створення клієнтської частини веб-застосунку використовувалася мова розмітки HTML, каскадні таблиці стилів CSS та мова програмування JS

Для створення серверної частини веб-застосунку використовувалися технології платформи ASP.NET, мова програмування C# та фреймворки ASP.NET MVC і Entity Framework

### **3.6 Спеціальні вимоги**

Інтуїтивно-зрозумілий інтерфейс, простий дизайн та легкий у користуванні функціонал застосунку.

## **4 Вимоги до програмної документації**

Набір документації, що надається у момент здачі проекту:

- Опис модулів програми з демонстрацією їх роботи
- Текст програми з коментарями
- Керівництво користувача для звичайних користувачів та адміністратора
- Довідкова інформація
- Керівництво для розробника

## **5 Стадії та етапи розробки**

Стадії та етапи розробки веб-застосунку для продажу мобільних телефонів продемонстровано у таблиці 1

Таблиця 1 - стадії та етапи розробки

| Стадія розробки   | Етапи робіт   | Зміст робіт   |
|---|---|---|
| Технічне завдання<br>02.01 – 31.01.2023                   | Обґрунтування<br>необхідності розробки<br>програми      | Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання |
| Ескізний проект<br>01.02 – 14.02 2023                     | Розробка ескізного<br>проекту                           | Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися                  |
| Технічний проект<br>15.02 – 28.02 2023                    | Розробка технічного<br>проекту                          | Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів                                 |
| Робочий проект<br>01.03 – 10.04.2023                      | Розробка програмного<br>забезпечення                    | Реалізація програмного забезпечення; відладка; проведення попереднього тестування   |
| Розробка програмної<br>документації<br>11.04 – 20.04.2023 | Розробка документації<br>до програмного<br>забезпечення | Розробка необхідної документації, передбаченої технічним завданням  |
| Тестування системи<br>21.04 – 30.04.2023                  | Проведення тестування<br>програмного<br>забезпечення    | Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення   |
| Впровадження  | Підготовка і передача<br>програми                       | Підготовка і передача програмного забезпечення; навчання персоналу використуванню програмного забезпечення; внесення коректувань в програмне забезпечення і документацію                      |

## ДОДАТОК Б

### КОД (ЛІСТИНГ) ПРОГРАМИ

Код UserController.cs

```
Using Diplom.DataAccess.Repository.IRepository;
using Diplom.DataAccess.Data;
using Diplom.Models;
using Diplom.Models.ViewModels;
using Diplom.Utility;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Data;

namespace DiplomWeb.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]
    public class UserController : Controller
    {
        private readonly UserManager<IdentityUser> _userManager;
        private readonly RoleManager<IdentityRole> _roleManager;
        private readonly IUnitOfWork _unitOfWork;
        public UserController(UserManager<IdentityUser> userManager, IUnitOfWork
unitOfWork, RoleManager<IdentityRole> roleManager) {
            _unitOfWork = unitOfWork;
            _roleManager = roleManager;
            _userManager = userManager;
        }
        public IActionResult Index()
        {
            return View();
        }

        public IActionResult RoleManagment(string userId) {
            RoleManagmentVM RoleVM = new RoleManagmentVM() {
                ApplicationUser = _unitOfWork.ApplicationUser.Get(u => u.Id == userId,
includeProperties: "Company"),
                RoleList = _roleManager.Roles.Select(i => new SelectListItem {
                    Text = i.Name,
                    Value = i.Name
                }),
                CompanyList = _unitOfWork.Company.GetAll().Select(i => new SelectListItem
{
                    Text = i.Name,
                    Value = i.Id.ToString()
                }),
            };

            RoleVM.ApplicationUser.Role =
_userManager.GetRolesAsync(_unitOfWork.ApplicationUser.Get(u=>u.Id==userId))
                .GetAwaiter().GetResult().FirstOrDefault();
            return View(RoleVM);
        }
    }
}
```

```

[HttpPost]
public IActionResult RoleManagment(RoleManagmentVM roleManagmentVM) {

    string oldRole =
_userManager.GetRolesAsync(_unitOfWork.ApplicationUser.Get(u => u.Id ==
roleManagmentVM.ApplicationUser.Id))
        .GetAwaiter().GetResult().FirstOrDefault();

    ApplicationUser applicationUser = _unitOfWork.ApplicationUser.Get(u => u.Id
== roleManagmentVM.ApplicationUser.Id);

    if (!(roleManagmentVM.ApplicationUser.Role == oldRole)) {
        //a role was updated
        if (roleManagmentVM.ApplicationUser.Role == SD.Role_Company) {
            applicationUser.CompanyId =
roleManagmentVM.ApplicationUser.CompanyId;
        }
        if (oldRole == SD.Role_Company) {
            applicationUser.CompanyId = null;
        }
        _unitOfWork.ApplicationUser.Update(applicationUser);
        _unitOfWork.Save();

        _userManager.RemoveFromRoleAsync(applicationUser,
oldRole).GetAwaiter().GetResult();
        _userManager.AddToRoleAsync(applicationUser,
roleManagmentVM.ApplicationUser.Role).GetAwaiter().GetResult();
    }
    else {
        if(oldRole==SD.Role_Company && applicationUser.CompanyId !=
roleManagmentVM.ApplicationUser.CompanyId) {
            applicationUser.CompanyId =
roleManagmentVM.ApplicationUser.CompanyId;
            _unitOfWork.ApplicationUser.Update(applicationUser);
            _unitOfWork.Save();
        }
    }

    return RedirectToAction("Index");
}

#region API CALLS

[HttpGet]
public IActionResult GetAll()
{
    List<ApplicationUser> objUserList =
_unitOfWork.ApplicationUser.GetAll(includeProperties: "Company").ToList();

    foreach(var user in objUserList) {

        user.Role=
_userManager.GetRolesAsync(user).GetAwaiter().GetResult().FirstOrDefault();

        if (user.Company == null) {
            user.Company = new Company() {
                Name = ""
            };
        }
    }
}

```

```

        return Json(new { data = objUserList });
    }

    [HttpPost]
    public IActionResult LockUnlock([FromBody]string id)
    {
        var objFromDb = _unitOfWork.ApplicationUser.Get(u => u.Id == id);
        if (objFromDb == null)
        {
            return Json(new { success = false, message = "Error while
Locking/Unlocking" });
        }

        if(objFromDb.LockoutEnd!=null && objFromDb.LockoutEnd > DateTime.Now) {
            //user is currently locked and we need to unlock them
            objFromDb.LockoutEnd = DateTime.Now;
        }
        else {
            objFromDb.LockoutEnd = DateTime.Now.AddYears(1000);
        }
        _unitOfWork.ApplicationUser.Update(objFromDb);
        _unitOfWork.Save();
        return Json(new { success = true, message = "Operation Successful" });
    }

    #endregion
}
}

```

## Код Product Controller.cs

```

using Diplom.DataAccess.Repository.IRepository;
using Diplom.DataAccess.Data;
using Diplom.Models;
using Diplom.Models.ViewModels;
using Diplom.Utility;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using System.Collections.Generic;
using System.Data;

namespace DiplomWeb.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize(Roles = SD.Role_Admin)]
    public class ProductController : Controller
    {
        private readonly IUnitOfWork _unitOfWork;
        private readonly IWebHostEnvironment _webHostEnvironment;
        public ProductController(IUnitOfWork unitOfWork, IWebHostEnvironment
webHostEnvironment)
        {
            _unitOfWork = unitOfWork;
            _webHostEnvironment = webHostEnvironment;
        }
        public IActionResult Index()
        {

```

```

        List<Product> objProductList =
        _unitOfWork.Product.GetAll(includeProperties: "Category").ToList();

        return View(objProductList);
    }

    public IActionResult Upsert(int? id)
    {
        ProductVM productVM = new()
        {
            CategoryList = _unitOfWork.Category.GetAll().Select(u => new
SelectListItem
        {
            Text = u.Name,
            Value = u.Id.ToString()
        }
        ),
            Product = new Product()
        };
        if (id == null || id == 0)
        {
            //create
            return View(productVM);
        }
        else
        {
            //update
            productVM.Product =
        _unitOfWork.Product.Get(u=>u.Id==id,includeProperties: "ProductImages");
            return View(productVM);
        }
    }

    [HttpPost]
    public IActionResult Upsert(ProductVM productVM, List<IFormFile> files)
    {
        if (ModelState.IsValid)
        {
            if (productVM.Product.Id == 0) {
                _unitOfWork.Product.Add(productVM.Product);
            }
            else {
                _unitOfWork.Product.Update(productVM.Product);
            }

            _unitOfWork.Save();

            string wwwRootPath = _webHostEnvironment.WebRootPath;
            if (files != null)
            {
                foreach(IFormFile file in files)
                {
                    string fileName = Guid.NewGuid().ToString() +
Path.GetExtension(file.FileName);
                    string productPath = @"images\products\product-" +
productVM.Product.Id;
                    string finalPath = Path.Combine(wwwRootPath, productPath);

                    if (!Directory.Exists(finalPath))
                        Directory.CreateDirectory(finalPath);

                    using (var fileStream = new FileStream(Path.Combine(finalPath,
fileName), FileMode.Create)) {
                        file.CopyTo(fileStream);
                    }
                }
            }
        }
    }

```

```

        }

        ProductImage productImage = new() {
            ImageUrl = @"\" + productPath + @"\" + fileName,
            ProductId=productVM.Product.Id,
        };

        if (productVM.Product.ProductImages == null)
            productVM.Product.ProductImages = new List<ProductImage>();

        productVM.Product.ProductImages.Add(productImage);
    }

    _unitOfWork.Product.Update(productVM.Product);
    _unitOfWork.Save();

}

TempData["success"] = "Product created/updated successfully";
return RedirectToAction("Index");
}
else
{
    productVM.CategoryList = _unitOfWork.Category.GetAll().Select(u => new
SelectListItem
    {
        Text = u.Name,
        Value = u.Id.ToString()
    });
    return View(productVM);
}
}

public IActionResult DeleteImage(int imageId) {
    var imageToBeDeleted = _unitOfWork.ProductImage.Get(u => u.Id == imageId);
    int productId = imageToBeDeleted.ProductId;
    if (imageToBeDeleted != null) {
        if (!string.IsNullOrEmpty(imageToBeDeleted.ImageUrl)) {
            var oldImagePath =
                Path.Combine(_webHostEnvironment.WebRootPath,
                    imageToBeDeleted.ImageUrl.TrimStart('\\'));

            if (System.IO.File.Exists(oldImagePath)) {
                System.IO.File.Delete(oldImagePath);
            }
        }

        _unitOfWork.ProductImage.Remove(imageToBeDeleted);
        _unitOfWork.Save();

        TempData["success"] = "Deleted successfully";
    }

    return RedirectToAction(nameof(Upsert), new { id = productId });
}

#region API CALLS

[HttpGet]

```

```

public IActionResult GetAll()
{
    List<Product> objProductList = _unitOfWork.Product.GetAll(includeProperties:
"Category").ToList();
    return Json(new { data = objProductList });
}

[HttpDelete]
public IActionResult Delete(int? id)
{
    var productToBeDeleted = _unitOfWork.Product.Get(u => u.Id == id);
    if (productToBeDeleted == null)
    {
        return Json(new { success = false, message = "Error while deleting" });
    }

    string productPath = @"images\products\product-" + id;
    string finalPath = Path.Combine(_webHostEnvironment.WebRootPath,
productPath);

    if (Directory.Exists(finalPath)) {
        string[] filePaths = Directory.GetFiles(finalPath);
        foreach (string filePath in filePaths) {
            System.IO.File.Delete(filePath);
        }

        Directory.Delete(finalPath);
    }

    _unitOfWork.Product.Remove(productToBeDeleted);
    _unitOfWork.Save();

    return Json(new { success = true, message = "Delete Successful" });
}

#endregion
}
}

```

## Код OrderController.cs

```

using Diplom.DataAccess.Repository.IRepository;
using Diplom.Models;
using Diplom.Models.ViewModels;
using Diplom.Utility;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Mvc;
using Stripe;
using Stripe.Checkout;
using System.Diagnostics;
using System.Security.Claims;

namespace BulkyBookWeb.Areas.Admin.Controllers {
    [Area("admin")]
    [Authorize]
    public class OrderController : Controller {

        private readonly IUnitOfWork _unitOfWork;
        [BindProperty]
        public OrderVM OrderVM { get; set; }
    }
}

```

```

public OrderController(IUnitOfWork unitOfWork)
{
    _unitOfWork = unitOfWork;
}

public IActionResult Index() {
    return View();
}

public IActionResult Details(int orderId) {
    OrderVM = new() {
        OrderHeader = _unitOfWork.OrderHeader.Get(u => u.Id == orderId,
includeProperties: "ApplicationUser"),
        OrderDetail = _unitOfWork.OrderDetail.GetAll(u => u.OrderHeaderId ==
orderId, includeProperties: "Product")
    };

    return View(OrderVM);
}
[HttpPost]
[Authorize(Roles =SD.Role_Admin+", "+SD.Role_Employee)]
public IActionResult UpdateOrderDetail() {
    var orderHeaderFromDb = _unitOfWork.OrderHeader.Get(u => u.Id ==
OrderVM.OrderHeader.Id);
    orderHeaderFromDb.Name = OrderVM.OrderHeader.Name;
    orderHeaderFromDb.PhoneNumber = OrderVM.OrderHeader.PhoneNumber;
    orderHeaderFromDb.StreetAddress = OrderVM.OrderHeader.StreetAddress;
    orderHeaderFromDb.City = OrderVM.OrderHeader.City;
    orderHeaderFromDb.State = OrderVM.OrderHeader.State;
    orderHeaderFromDb.PostalCode = OrderVM.OrderHeader.PostalCode;
    if (!string.IsNullOrEmpty(OrderVM.OrderHeader.Carrier)) {
        orderHeaderFromDb.Carrier = OrderVM.OrderHeader.Carrier;
    }
    if (!string.IsNullOrEmpty(OrderVM.OrderHeader.TrackingNumber)) {
        orderHeaderFromDb.Carrier = OrderVM.OrderHeader.TrackingNumber;
    }
    _unitOfWork.OrderHeader.Update(orderHeaderFromDb);
    _unitOfWork.Save();

    TempData["Success"] = "Order Details Updated Successfully.";

    return RedirectToAction(nameof(Details), new {orderId=
orderHeaderFromDb.Id});
}

[HttpPost]
[Authorize(Roles = SD.Role_Admin + ", " + SD.Role_Employee)]
public IActionResult StartProcessing() {
    _unitOfWork.OrderHeader.UpdateStatus(OrderVM.OrderHeader.Id,
SD.StatusInProgress);
    _unitOfWork.Save();
    TempData["Success"] = "Order Details Updated Successfully.";
    return RedirectToAction(nameof(Details), new { orderId =
OrderVM.OrderHeader.Id });
}

[HttpPost]
[Authorize(Roles = SD.Role_Admin + ", " + SD.Role_Employee)]
public IActionResult ShipOrder() {

    var orderHeader = _unitOfWork.OrderHeader.Get(u => u.Id ==
OrderVM.OrderHeader.Id);
    orderHeader.TrackingNumber = OrderVM.OrderHeader.TrackingNumber;

```

```

orderHeader.Carrier = OrderVM.OrderHeader.Carrier;
orderHeader.OrderStatus = SD.StatusShipped;
orderHeader.ShippingDate = DateTime.Now;
if (orderHeader.PaymentStatus == SD.PaymentStatusDelayedPayment) {
    orderHeader.PaymentDueDate = DateTime.Now.AddDays(30);
}

_unitOfWork.OrderHeader.Update(orderHeader);
_unitOfWork.Save();
TempData["Success"] = "Order Shipped Successfully.";
return RedirectToAction(nameof(Details), new { orderId =
OrderVM.OrderHeader.Id });
}
[HttpPost]
[Authorize(Roles = SD.Role_Admin + ", " + SD.Role_Employee)]
public IActionResult CancelOrder() {

    var orderHeader = _unitOfWork.OrderHeader.Get(u => u.Id ==
OrderVM.OrderHeader.Id);

    if (orderHeader.PaymentStatus == SD.PaymentStatusApproved) {
        var options = new RefundCreateOptions {
            Reason = RefundReasons.RequestedByCustomer,
            PaymentIntent = orderHeader.PaymentIntentId
        };

        var service = new RefundService();
        Refund refund = service.Create(options);

        _unitOfWork.OrderHeader.UpdateStatus(orderHeader.Id, SD.StatusCancelled,
SD.StatusRefunded);
    }
    else {
        _unitOfWork.OrderHeader.UpdateStatus(orderHeader.Id, SD.StatusCancelled,
SD.StatusCancelled);
    }
    _unitOfWork.Save();
    TempData["Success"] = "Order Cancelled Successfully.";
    return RedirectToAction(nameof(Details), new { orderId =
OrderVM.OrderHeader.Id });
}

[ActionName("Details")]
[HttpPost]
public IActionResult Details_PAY_NOW()
{
    OrderVM.OrderHeader = _unitOfWork.OrderHeader
        .Get(u => u.Id == OrderVM.OrderHeader.Id, includeProperties:
"ApplicationUser");
    OrderVM.OrderDetail = _unitOfWork.OrderDetail
        .GetAll(u => u.OrderHeaderId == OrderVM.OrderHeader.Id,
includeProperties: "Product");

    //stripe logic
    var domain = Request.Scheme + "://" + Request.Host.Value + "/";
    var options = new SessionCreateOptions {
        SuccessUrl = domain +
$"admin/order/PaymentConfirmation?orderId={OrderVM.OrderHeader.Id}",
        CancelUrl = domain +
$"admin/order/details?orderId={OrderVM.OrderHeader.Id}",
        LineItems = new List<SessionLineItemOptions>(),
        Mode = "payment",

```

```

};

foreach (var item in OrderVM.OrderDetail) {
    var sessionLineItem = new SessionLineItemOptions {
        PriceData = new SessionLineItemPriceDataOptions {
            UnitAmount = (long)(item.Price * 100), // $20.50 => 2050
            Currency = "usd",
            ProductData = new SessionLineItemPriceDataProductDataOptions {
                Name = item.Product.Title
            }
        },
        Quantity = item.Count
    };
    options.LineItems.Add(sessionLineItem);
}

var service = new SessionService();
Session session = service.Create(options);
_unitOfWork.OrderHeader.UpdateStripePaymentID(OrderVM.OrderHeader.Id,
session.Id, session.PaymentIntentId);
_unitOfWork.Save();
Response.Headers.Add("Location", session.Url);
return new StatusCodeResult(303);
}

public IActionResult PaymentConfirmation(int orderHeaderId) {

    OrderHeader orderHeader = _unitOfWork.OrderHeader.Get(u => u.Id ==
orderHeaderId);
    if (orderHeader.PaymentStatus == SD.PaymentStatusDelayedPayment) {
        //this is an order by company

        var service = new SessionService();
        Session session = service.Get(orderHeader.SessionId);

        if (session.PaymentStatus.ToLower() == "paid") {
            _unitOfWork.OrderHeader.UpdateStripePaymentID(orderHeaderId,
session.Id, session.PaymentIntentId);
            _unitOfWork.OrderHeader.UpdateStatus(orderHeaderId,
orderHeader.OrderStatus, SD.PaymentStatusApproved);
            _unitOfWork.Save();
        }

    }

    return View(orderHeaderId);
}

#region API CALLS

[HttpGet]
public IActionResult GetAll(string status) {
    IEnumerable<OrderHeader> objOrderHeaders;

    if (User.IsInRole(SD.Role_Admin) || User.IsInRole(SD.Role_Employee)) {
        objOrderHeaders = _unitOfWork.OrderHeader.GetAll(includeProperties:
"ApplicationUser").ToList();
    }
    else {

```

```

var claimsIdentity = (ClaimsIdentity)User.Identity;
var userId = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier).Value;

objOrderHeaders = _unitOfWork.OrderHeader
    .GetAll(u => u.ApplicationUserId == userId, includeProperties:
"ApplicationUser");
    }

    switch (status) {
        case "pending":
            objOrderHeaders = objOrderHeaders.Where(u => u.PaymentStatus ==
SD.PaymentStatusDelayedPayment);
            break;
        case "inprocess":
            objOrderHeaders = objOrderHeaders.Where(u => u.OrderStatus ==
SD.StatusInProgress);
            break;
        case "completed":
            objOrderHeaders = objOrderHeaders.Where(u => u.OrderStatus ==
SD.StatusShipped);
            break;
        case "approved":
            objOrderHeaders = objOrderHeaders.Where(u => u.OrderStatus ==
SD.StatusApproved);
            break;
        default:
            break;
    }

    return Json(new { data = objOrderHeaders });
}

#endregion
}
}

```

# ДОДАТОК В

## ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Хмельницький Національний Університет  
Факультет Інформаційних технологій  
Кафедра Інженерії програмного забезпечення

Кваліфікаційна робота, на тему:  
«Інтернет-магазин для продажу мобільних телефонів»

Студент: Трембач Владіслав Богданович

Керівник: Бедраюк Л.П. д-р фіз-мат. наук, професор

## ВСТУП

Сучасні технології швидко змінюють світ навколо нас, і електронна комерція не є винятком. Інтернет-магазини стають все більш популярними та необхідними для людей, які хочуть обрати та придбати товари без виходу з дому.

Однією з головних тенденцій сучасного світу є збільшення кількості користувачів Інтернету. За даними статистики, понад 4,5 мільярди людей у світі використовують Інтернет, і це число постійно зростає. За останні кілька років також спостерігається зростання кількості покупців, які обирають онлайн-шопінг. Мобільні телефони, в свою чергу, є невід'ємною частиною життя сучасної людини. Їх використовують не тільки для зв'язку, але й для роботи, розваг та покупок.

# АКТУАЛЬНІСТЬ КВАЛІФІКАЦІЙНОЇ РОБОТИ

- Розробка інтернет-магазину з продажу мобільних телефонів є актуальною в наш час, оскільки сучасний споживач шукає швидкий та зручний спосіб придбання товарів. Завдяки Інтернету він має можливість обирати товари з великого асортименту та купувати їх з будь-якої точки світу, не виходячи з дому.
- Також розробка інтернет-магазину з продажу мобільних телефонів буде корисною не тільки для клієнтів, але й для компанії-власника, оскільки це збільшить їхню потенційну аудиторію, а разом із цим і кількість продаж та можливість здійснювати продажі в режимі 24/7 в будь-яку точку світу де є поштове відділення.

## МЕТА ТА ЗАВДАННЯ

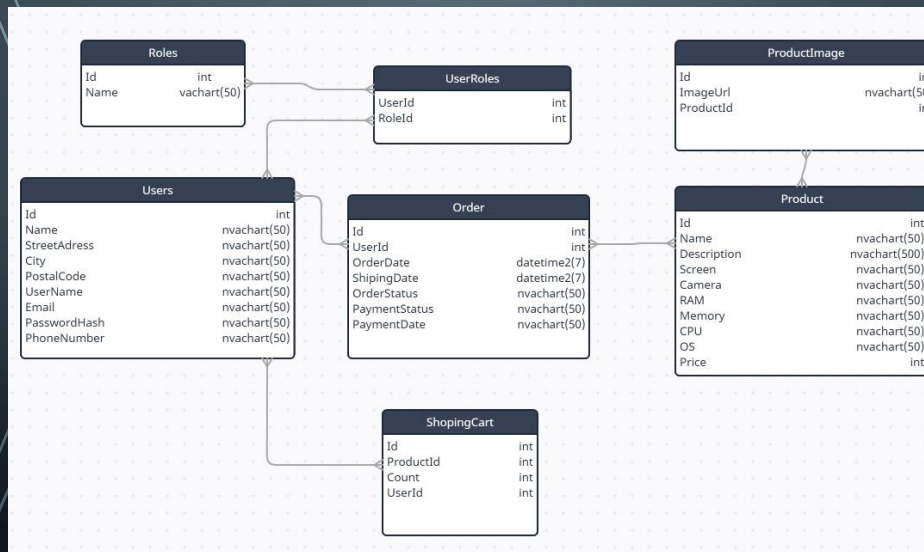
Метою кваліфікаційної роботи є розробка функціонального і ефективного веб-застосунку для торгівлі мобільними телефонами, який дозволить власникам малого та середнього бізнесу розширити свою справу та збільшити обсяг продажів. Вони можуть збільшити свою базу клієнтів завдяки доступності онлайн-магазину та зручності його використання. Кінцева версія веб-додатку повинна забезпечувати функції інтернет-магазину, що дозволить власникам бізнесу продавати товари та встановлювати зв'язок зі своїми клієнтами.

### Завдання кваліфікаційної роботи

- Провести аналіз предметної області для визначення її головних особливостей;
- провести аналіз наявного програмного забезпечення для визначення вимог до додатку;
- здійснити огляд технологій, методологій та практик розробки веб-застосунків та обрати найкращі для майбутнього використання при розробці проекту;
- сформулювати технічне завдання;
- розробити структуру та функціональну модель проекту;
- виконати програмну реалізацію проекту;
- провести тестові випробування застосунку;
- створити документацію, що описує процес виконання проекту.



# ЛОГІЧНА МОДЕЛЬ БАЗИ ДАНИХ



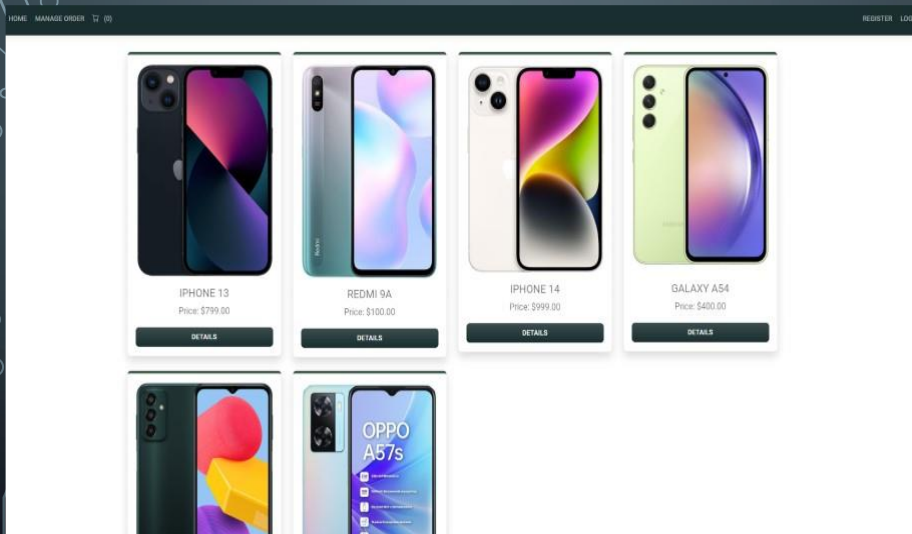
## ВИБІР ЗАСОБІВ РОЗРОБКИ

Для розробки проекту було обрано фреймворк ASP.NET MVC. Це популярний фреймворк для розробки веб-додатків, який дотримується шаблону проектування Model-View-Controller, що забезпечує чітку організацію коду та його повторне використання. Це дозволяє легко розбивати проект на компоненти та підтримувати чистоту коду.

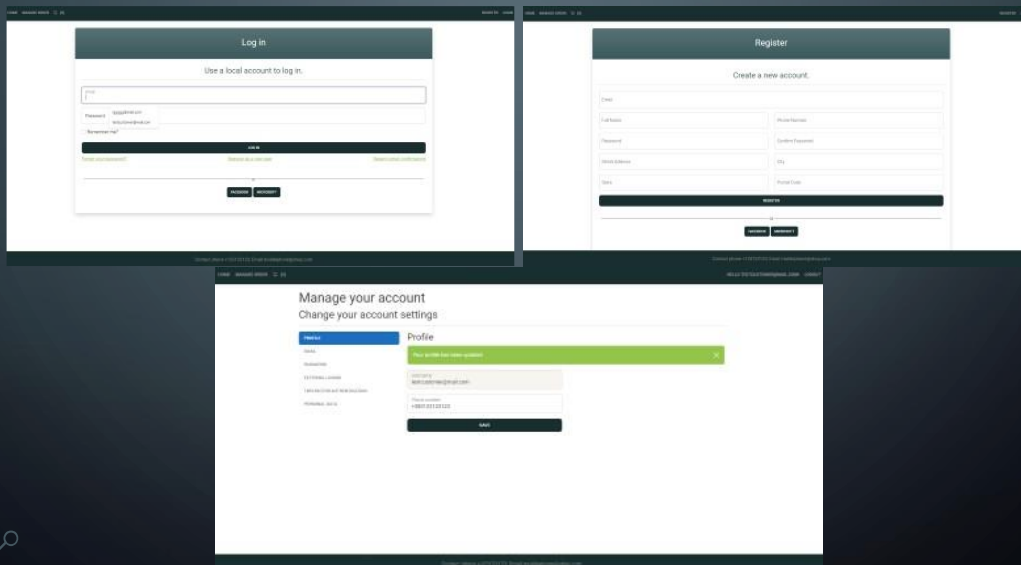
Також ASP.NET MVC забезпечує гнучкість у виборі інструментів та технологій. При розробці можна використовувати різні рішення для реалізації моделі, представлення та контролера в залежності від вимог проекту. Крім того, фреймворк легко розширюється за допомогою власних або сторонніх компонентів.

ASP.NET MVC легко інтегрується з іншими компонентами платформи .NET, такими як Entity Framework для доступу до бази даних, ASP.NET Identity для автентифікації та авторизації користувачів та багатьма іншими. Окрім цього має вбудовану підтримку заходів безпеки, таких як перехоплення міжсайтових скриптів, захист від впровадження коду і не тільки.

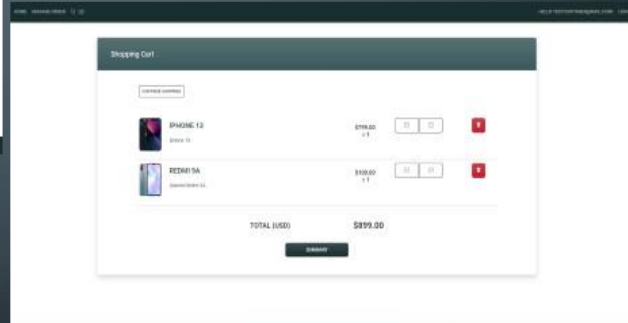
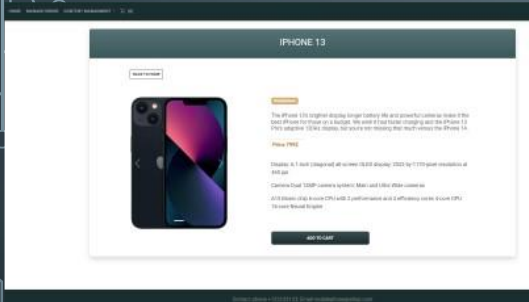
# ІНТЕРФЕЙС ДОДАТКУ- ГОЛОВНА СТОРІНКА



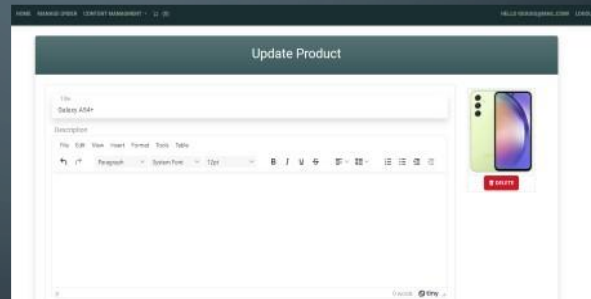
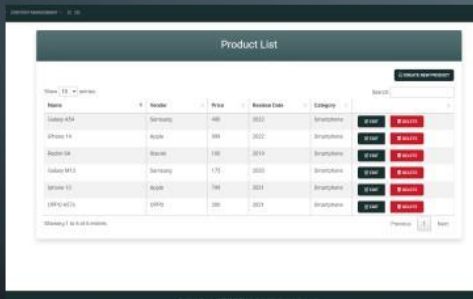
# ІНТЕРФЕЙС ДОДАТКУ – СТОРІНКИ АВТОРИЗАЦІЇ/РЕЄСТРАЦІЇ ТА КЕРУВАННЯ ПРОФІЛЕМ



# ІНТЕРФЕЙС ДОДАТКУ- СТОРІНКА ТОВАРУ ТА КОШИКУ



# ІНТЕРФЕЙС ДОДАТКУ- СТВОРЕННЯ/ РЕДАГУВАННЯ ЗАПИСІВ

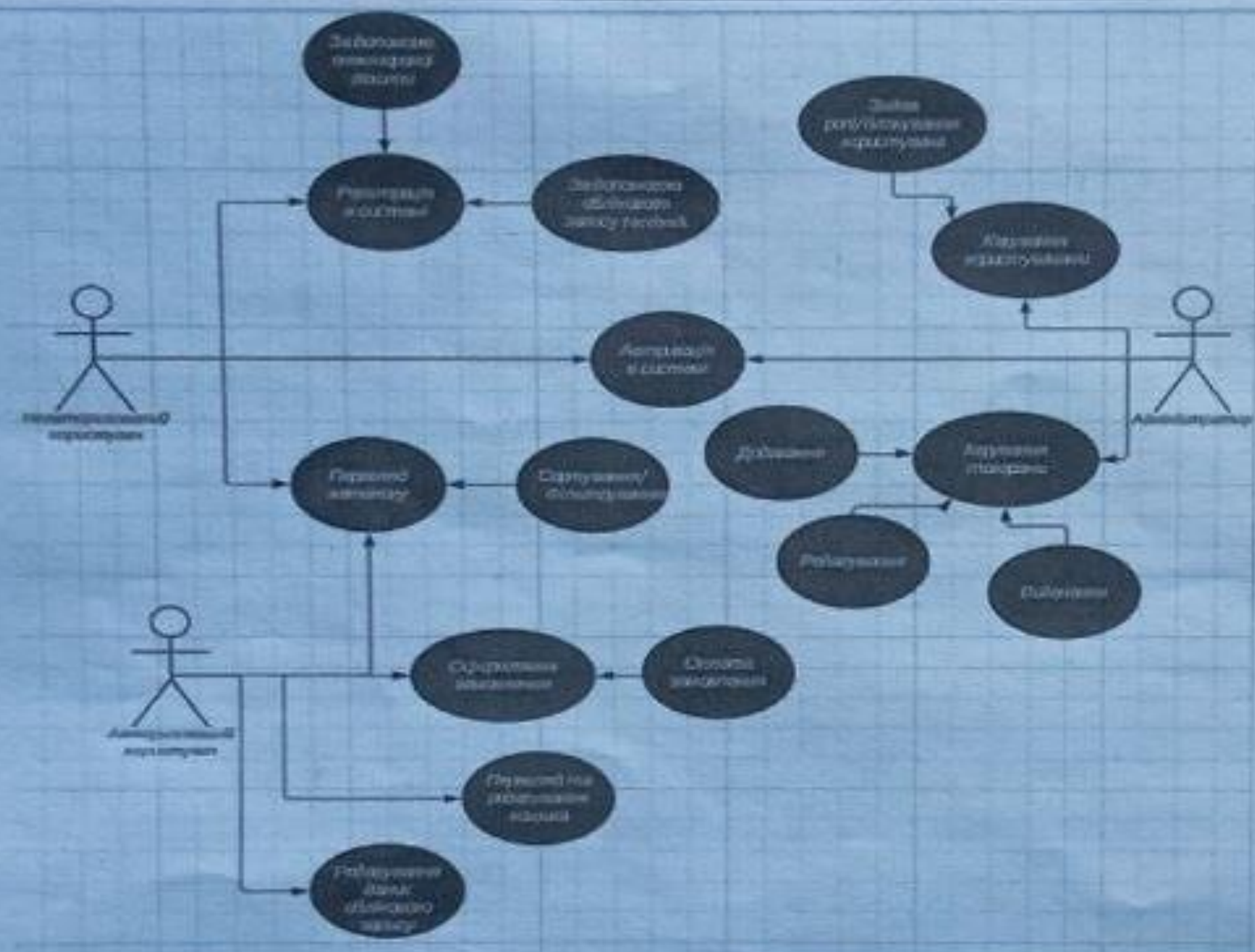




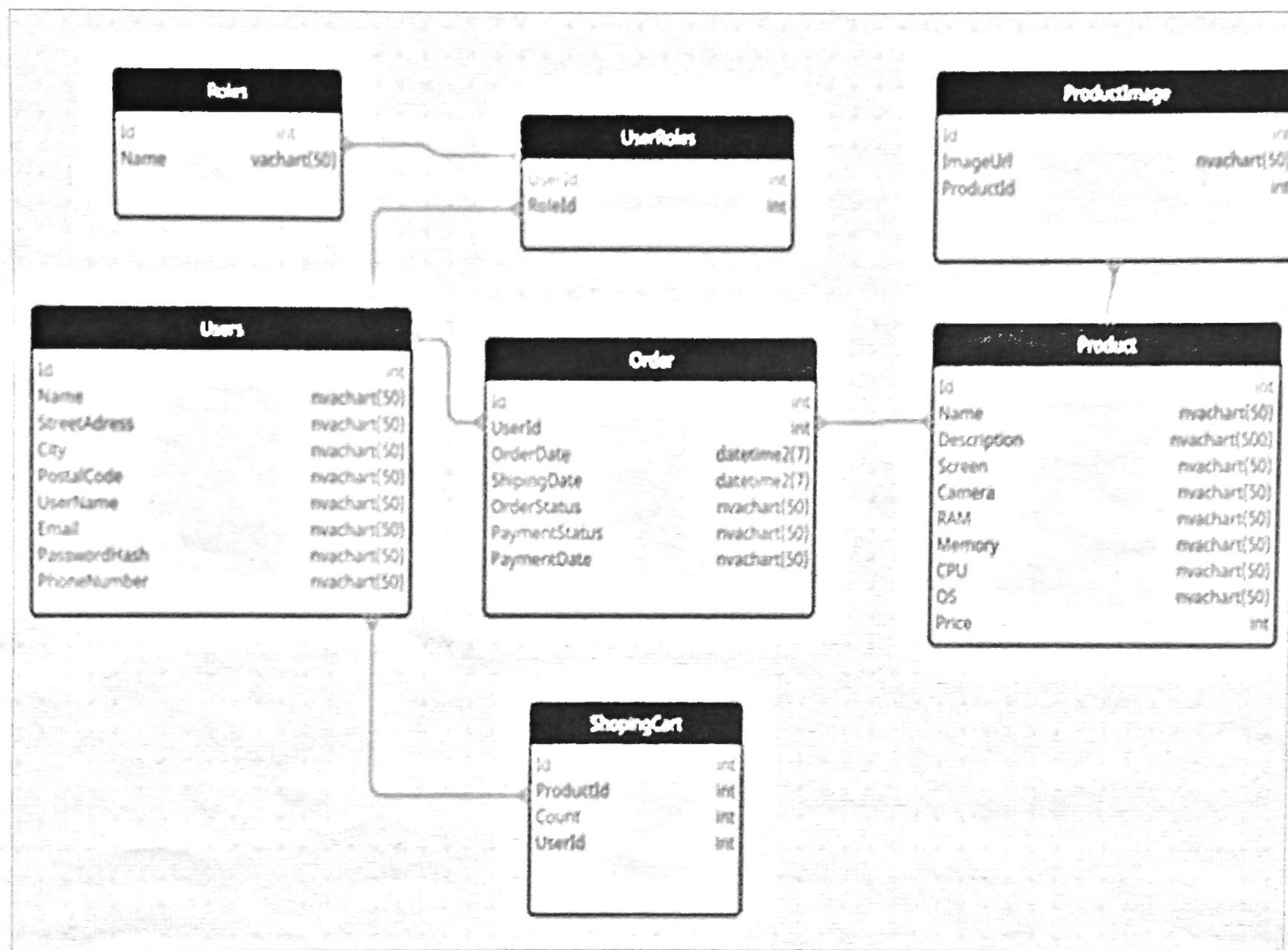
## ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проведено комплексний аналіз у галузі інтернет продажу мобільних телефонів. Визначено основні переваги та недоліки існуючого програмного забезпечення, які були враховані під час розробки проекту. Проведено аналіз технологій, методологій та практик розробки веб-додатків та обрано оптимальні для розробки проекту. В рамках проекту була створена добре організована база даних, яка зберігає інформацію про товари, замовлення, користувачів та інші необхідні дані. Код проекту написаний з урахуванням принципів модульності, розширюваності та повторного використання. Проведено тестування для виявлення та виправлення помилок.

Результатом виконаної роботи є готовий інтернет-магазин, який здатний ефективно функціонувати і задовольняти потреби як адміністраторів, так і клієнтів.

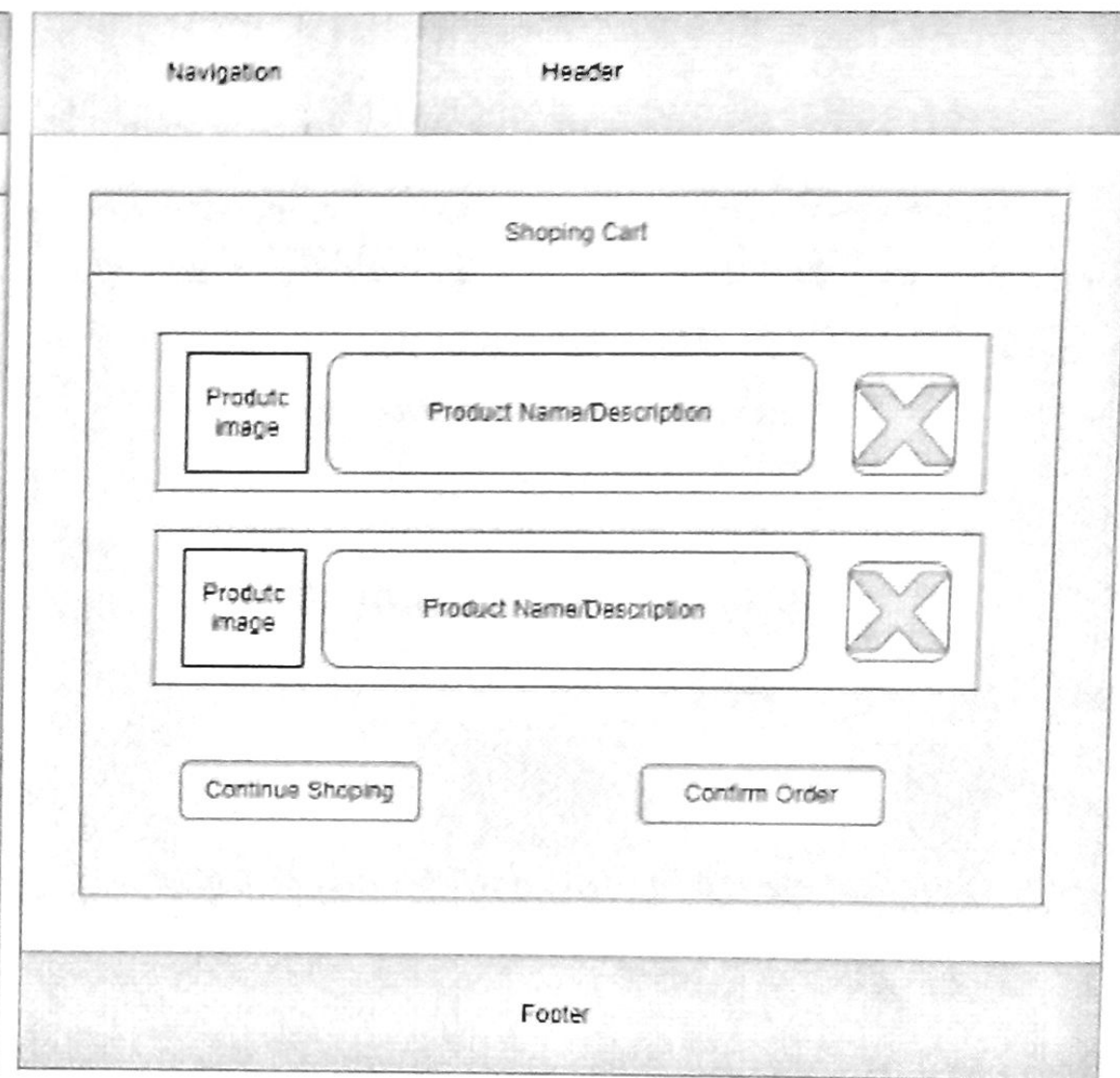
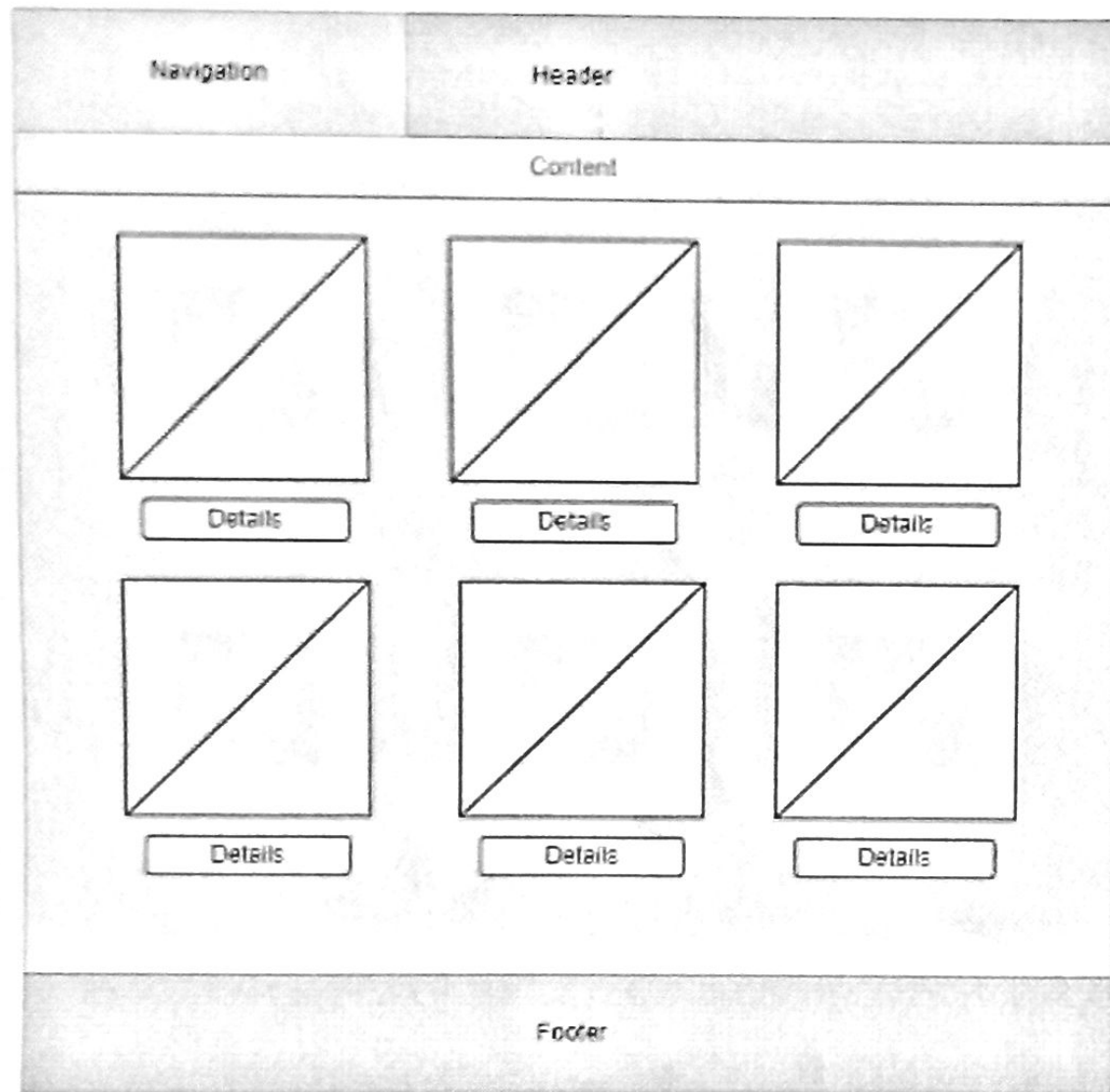


|             |                |      |        |   |               |         |         |
|-------------|----------------|------|--------|---|---------------|---------|---------|
|             |                |      |        | КвРІПЗ 190142.01.19.E8                    |               |         |         |
| Зм.         | Лек.           | М.Д. | Підпис | UML-діаграма<br>варіантів<br>використання | Листів        | Мова    | Масштаб |
| Курсант     | Виконав        | К.П. | С.С.   |   | Архив 1       | Архив 2 |         |
| Н. Бойко    | Гуменко І.В.   |      | С.С.   |   | ХНУ, ІПЗ-19-1 |         |         |
| Зав. кафедр | Васильчук Р.П. |      | С.С.   |   |               |         |         |



|           |               |          |      |      |                        |        |      |         |
|-----------|---------------|----------|------|------|------------------------|--------|------|---------|
|           |               |          |      |      | КвРІПЗ.190142.01.19.Е8 |        |      |         |
| Вид       | Лист          | № докум. | Дата | Дата | ER-діаграма бази даних | Листів | Маса | Масштаб |
| Розробка  | Тренба-ЕВ     | Прецед   | 3.06 | 3.06 |                        | 2      |      | 3       |
| Корекція  | Бадратск Л.П. |          |      |      |                        | 2      |      | 3       |
| Н. Контр. | Гурман І.В.   |          |      |      |                        |        |      |         |
| Зад. Каф. | Бадратск Л.П. |          |      |      |                        |        |      |         |

ХНУ, ІПЗ-19-1



|           |               |                    |      |                            |                        |               |      |          |
|-----------|---------------|--------------------|------|----------------------------|------------------------|---------------|------|----------|
|           |               |                    |      |                            | КвРІПЗ.190142.01.19.Е8 |               |      |          |
| Зм. Док.  | № докум.      | Підпис             | Дата | Макети сторінок застосунку |                        | Літера        | Мова | Масштаб  |
| Розробник | Грембач В.В.  | <i>[Signature]</i> |      |                            |                        |               |      |          |
| Коректор  | Бедратюк Л.П. | <i>[Signature]</i> | 1.06 |                            |                        | Архив         | 3    | Документ |
| Н. Кондо  | Гурман І.В.   | <i>[Signature]</i> | 3.06 |                            |                        | ХНУ. ІПЗ-19-1 |      |          |
| Зад. Кад. | Бедратюк Л.П. | <i>[Signature]</i> | 5.06 |                            |                        |               |      |          |

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ  
освітнього ступеня «Бакалавр»

Дипломник Трембач Владіслав Богданович

Тема Інтернет магазин для продажу мобільних телефонів

Спеціальність 121 – Інженерія програмного забезпечення

**Обсяг кваліфікаційної роботи:**

Кількість листів креслень 3; кількість сторінок записки 73

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі було досліджено і проаналізовано предметну область, визначено усі функціональні та нефункціональні вимоги. Був проведений аналіз існуючих програм на ринку, розглянуто їх переваги і недоліки, та доведено актуальність розробки нового програмного забезпечення. Розглянуто інструменти для реалізації спроектованих рішень, в результаті чого створено програмне забезпечення. Також було проведено тестування програми, за результатами якого доведено, що розроблене програмне забезпечення працює коректно та готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення. У другому розділі проведено аналіз сучасних архітектур, розглянуто їх переваги і недоліки та визначено, що система буде відповідати клієнт-серверній та п-рівневій архітектурі. У третьому розділі підготовлено всі залежності для написання коду та виконано практичну розробку програмних модулів і описано їх особливості, в результаті чого створено програмний продукт. Також виконано модульне тестування системи та проведено його у відповідності до функціональних вимог, в результаті було підтверджено коректну роботу програми.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є актуальною, оскільки на сьогодні в Україні інтернет-магазини є достатньо популярними та прибутковими. Також було застосовано новітні технології для побудови програмного продукту та актуальні архітектурні рішення.

5. Негативні сторони роботи Недостатньо. Деталь описано реалізацію програмної системи. Проведено мало сценаріїв тестування.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконано відповідно до теми кваліфікаційної роботи та подано у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у рамках тематики проектування. Графічний матеріал дає можливість наочно побачити деталі проектування системи.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Говорушенко Тетяна Олександрівна, доктор технічних наук, професор,  
зав.кафедри комп'ютерної інженерії та інформаційних систем (КІІС)  
ХНУ

“ 5 ” червне 2023 р.

(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ  
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатами звіту/звітів подібності щодо роботи, продуктованими програмно-технічним засобом (ами) перевірки текстів на плагіат:

Назва: «Інтернет-магазин для продажу мобільних телефонів»

Автор: Трембач Владіслав Богданович

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Бедратюк Леонід Петрович, д. фіз.-мат. наук, проф

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок   | Позначка про відповідність |
|---|--|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.   | <b>відповідає</b>          |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи |                            |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.      |                            |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.   |                            |
| 5 | Інше:  |                            |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

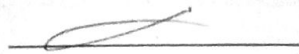
2) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

3) найбільша зафіксована схожість з джерелом Бібліотеки складає 2,8%, з джерелом з Інтернету 2,3%.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/схожості, складає 13,8% і адресується до 134 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

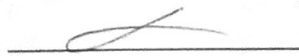
Дата 4 травня

Завідувач кафедри



Леонід БЕДРАТЮК

Гарант освітньої програми



Леонід БЕДРАТЮК

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

Завідувачу кафедри інженерії програмного  
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Трембача В.Б.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-19-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

01.06.2023

дата

Трембач

підпис

# Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 8%

|  |          |         |                             |          |
|--|----------|---------|-----------------------------|----------|
| ID: 114562<br>Назва: БКР Інтернет магазин для продажу мобільних телефонів<br>Додано в БД: 2023-06-02<br>Автора: Трембач В.Б.<br>Керівники: Бедратюк Л.П.<br>Консультанти:<br>Опоненти: | Документ |         | Сумарний збіг по Базі Даних |          |
|  | Символи  | Лексеми | Символи                     | Лексеми  |
|  | 72309    | 657     | 7809 (11%)                  | 91 (14%) |

Джерело плагиату

| ID | Опис | Наявність плагиату в документі |         |
|----|------|--------------------------------|---------|
|    |      | Символи                        | Лексеми |



Ім'я користувача:  
Кафедра ІПЗ

Дата перевірки:  
02.06.2023 12:15:15 EEST

Дата звіту:  
02.06.2023 12:16:28 EEST

ID перевірки:  
1015389791

Тип перевірки:  
Doc vs Internet + Library

ID користувача:  
100005589

Назва документа: КвР Трембач

Кількість сторінок: 74 Кількість слів: 11569 Кількість символів: 89769 Розмір файлу: 5.02 MB ID файлу: 101504441

## 13.8% Схожість

Найбільша схожість: 2.8% з джерелом з Бібліотеки (ID файлу: 1015045630)

8.28% Джерела з Інтернету

273

Сторінка 76

10.1% Джерела з Бібліотеки

72

Сторінка 78

## 0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

## 0% Вилучень

Немає вилучених джерел