

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Web-додаток підтримки продажу портативної електроніки.

Здобувача групи ІТ-91 Кліщ Олег Юрійович

(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

(підпис)

Олег КЛІЩ
(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник доцент кафедри ІТ, к.т.н., доцент Юлія ПАРФЕНЕНКО

(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

(підпис)

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ Світлана ВАЩЕНКО

«__» _____ 2023 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Кліщ Олег Юрійович

1 Тема роботи Web-додаток підтримки продажу портативної електроніки

керівник роботи Парфененко Юлія Вікторівна, к.т.н., доцент

затверджені наказом по університету від «29» 05 2023 р. №0588-VI

2 Строк подання студентом роботи «7» червня 2023 р.

3 Вхідні дані до роботи перелік товарів портативної електроніки та їх фото.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, моделювання та проектування системи, розробка web-додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
мета і задачі дипломного проекту, порівняння аналогів, функціональні вимоги до додатку, діаграма IDEF0, декомпозиція першого рівня, діаграма варіантів використання, діаграма діяльності, засоби реалізації, база даних, демонстрація додатку (головна сторінка), демонстрація додатку (сторінка контакти), демонстрація додатку (каталог), демонстрація додатку (пошук продуктів веб-додатку), висновки

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 08.02.2023

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Дослідження предметної області	11.12.2022- 25.12.2022	
2	Аналіз програмних аналогів	26.12.2022- 05.01.2023	
3	Оформлення технічного завдання	06.01.2023- 20.01.2023	
4	Оформлення планування робіт	21.01.2023- 30.01.2023	
5	Проектування web-додатку	31.01.2023- 15.02.2023	
6	Розробка web-додатку	16.02.2023- 26.05.2023	
7	Тестування web-додатку	27.05.2023- 05.06.2023	
8	Завантаження web-додатку на хостинг	06.06.2023- 07.06.2023	
9	Оформлення пояснювальної записки	08.06.2023- 12.06.2023	

Студент

(підпис)

Олег КЛІЩ

Керівник роботи

(підпис)

к.т.н., доц. Юлія ПАРФЕНЕНКО

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки продажу портативної електроніки».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 16 найменувань, 3 додатків. Загальний обсяг роботи – 93 сторінки, у тому числі 50 сторінки основного тексту, 2 сторінки списку використаних джерел та 41 сторінки додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку підтримки продажу портативної електроніки.

У першому розділі ми дослідили та проаналізували предметну область. Для розуміння потреби створення даного програмного продукту були розглянуті останні дослідження, а також публікації з даної предметної області. Потім представили основних конкурентів для нашого проекту та детально порівняли їх між собою та безпосередньо з моїм web-додатком. Після всіх досліджень та аналізів була сформована постановка задачі.

У другому розділі виконали моделювання та проектування інформаційної системи, а саме:

- структурно-функціональне моделювання;
- проектування інформаційної системи;
- проектування моделі бази даних.

У третьому розділі показана архітектура системи та описана програмна реалізація web-додатку. Також розглянули роботу з web-додатком для авторизованих клієнтів та адміністраторів.

Ключові слова: web, MySQL, PHP, Java, Laravel, web-додаток, портативна електроніка, CSS.

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
1.1 Огляд останніх досліджень і публікацій	7
1.2 Аналіз програмних продуктів аналогів	9
1.3 Постановка задачі	14
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКУ	16
2.1 Структурно-функціональне моделювання web-додатку	16
2.2 Проектування інформаційної системи	19
2.2.1 Діаграма варіантів використання web-додатку	19
2.2.2 Структурне моделювання	20
2.3 Проектування моделі бази даних	22
3 РОЗРОБКА WEB-ДОДАТКУ	24
3.1 Архітектура програмного додатку	24
3.2 Програмна реалізація	25
3.3 Використання web-додатку покупцем	34
3.4 Адміністрування web-додатку	42
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК А	53
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ	65
ДОДАТОК В. КОД РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	78

ВСТУП

В сучасному світі портативна електроніка займає важливе місце в повсякденному житті людей. За останні десятиліття ринок портативних пристроїв досить зріс, це призводить до збільшення конкуренції між виробниками і підвищенням вимог споживачів до якості і функціональності продуктів. У цьому контексті підтримка продажу портативної електроніки стає ключовим фактором успіху на ринку.

Одним з головних викликів, що стоять перед продавцями портативної електроніки, є необхідність пропонувати клієнтам персоналізований сервіс та консультації з питань вибору та налаштування продуктів. Крім того, інтернет-магазини стають все більш популярними серед споживачів, що ставить перед продавцями додаткові виклики щодо створення ефективної онлайн-стратегії та взаємодії з клієнтами через цифрові канали.

Впровадження web-додатку є ефективною стратегією підтримки продажу портативної електроніки, яка буде орієнтована на задоволення потреб та очікувань споживачів.

Сьогодні web-додатки відіграють ключову роль в повсякденному житті людей та надають нові можливості. Вони складаються з колекції загальнодоступних web-сторінок з однаковим доменним ім'ям, що забезпечує зручність та доступність онлайн-покупок.

У порівнянні з традиційними торговими центрами, інтернет-магазини мають такі переваги, як:

- відсутність черг;
- можливість зробити покупки будь-де та в будь-який час;
- широкий асортимент товарів для порівняння та дослідження;
- доступність для покупців та продавців.

Крім того, інтернет-магазини знімають географічні обмеження та надають змогу продавати товари в інші міста, а інколи й в інші країни.

Метою бакалаврської роботи є розробка web-додатку для підтримки продажу портативної електроніки.

Для досягнення цієї мети потрібно:

- провести аналіз аналогів та виконати дослідження предметної області;
- розробити технічне завдання та виконати планування робіт;
- створити макети та реалізувати web-додаток для підтримки продажу магазину портативної електроніки;
- провести тестування web-додатку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Задача підтримки продажу портативної електроніки є актуальною у сучасному світі, де зростає популярність цих пристроїв, та стає важливою у забезпеченні конкурентоспроможності компаній, які займаються продажем таких товарів.

Маркетингові стратегії для продажу портативної електроніки є основою успіху у продажах. Однією з ефективних стратегій є використання різних каналів збуту, таких як онлайн-магазини, роздрібні магазини та власні магазини компанії. Кожен канал має свої переваги та недоліки, тому важливо аналізувати потреби та попит нашої цільової аудиторії та вибрати оптимальний канал збуту. Крім того, рекламні кампанії можуть бути використані для підвищення попиту на продукт та залучення нових клієнтів [1].

Останні дослідження та публікації показують, що успішний продаж портативної електроніки вимагає від компаній пильної уваги до клієнтів та їхніх потреб. Крім того, важливо бути уважними до ринкових тенденцій та використовувати нові технології, щоб залишатися конкурентоспроможними.

Web-додаток для електронної комерції Cooperative Store Management System, є платформою, на якій клієнти можуть здійснювати покупки онлайн, що дозволяє їм уникнути необхідності фізичного відвідування магазину.

Основна мета цієї системи полягає в тому, щоб зменшити робоче навантаження продавця і мінімізувати помилки введення записів, автоматизувавши цей процес. Клієнти отримують можливість зменшити свої витрати та заощадити час, використовуючи дану платформу. Завдяки таким перевагам клієнти можуть користуватися кращими послугами, роблячи покупки в зручний для них час з будь-якої точки світу. Це допоможе не тільки залучати нових клієнтів, але й утримувати постійних клієнтів, що сприятиме довгостроковому розвитку бізнесу [2].

На сьогоднішній час для клієнта існує два основних способи купівлі товарів. Перший – це безпосередньо прийти в магазин і придбати бажаний товар. Другий – замовити товар в інтернет магазині за допомогою web-додатків.

Порівняння цих двох способів продажу товарів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння та виділення переваг та недоліків магазинів і web-додатків

Назва	Переваги	Недоліки
Магазини	Клієнти можуть протестувати товар відразу.	Значна вартість оренди приміщення. Багато конкурентів на ринку.
Web-додатки	Клієнти можуть роздивитися будь-який товар в деталях. Відсутність черг. Оформлення замовлення в будь-якому місці. Швидке порівняння товарів аналогів.	Витрати на розробку та підтримку додатку. Значна конкуренція з іншими web-додатками. Постійна наявність інтернету для перегляду та замовлення товару.

Як бачимо, використання web-додатків є найзручнішим способом для продажу портативної електроніки.

Висновки з останніх досліджень та публікацій підтверджують важливість розуміння потреб та побажань клієнтів, а також використання ефективних маркетингових стратегій та новітніх технологій. Компанії, які враховують ці фактори, мають більші шанси на успіх у продажі портативної електроніки.

1.2 Аналіз програмних продуктів аналогів

У мережі Інтернет наявні різноманітні інтернет-каталоги, проте значна частина веб-сайтів залишається недоробленою: інтерфейс може бути не зручним або застарілим, а товари можуть бути неактуальними. Для успішного функціонування web-додатку необхідно надати користувачам повний функціонал і зручний інтерфейс. Для визначення вимог до мого розроблюваного продукту було розглянуто аналогів web-орієнтованих інформаційних систем, таких як "MobyStore", "Compservice" та "Титан".

Перший аналог, що буде розглядатися – це «Титан» [3]. Головна сторінка web-додатку представлена на рисунку 1.1.

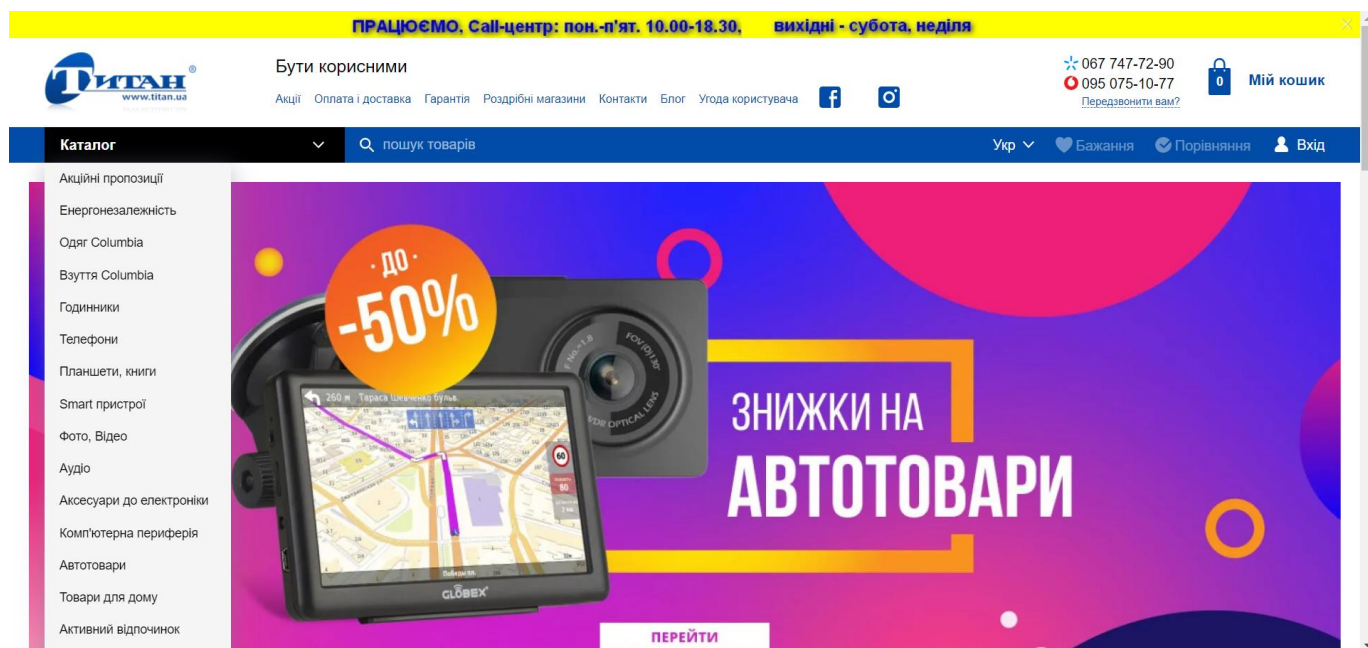


Рисунок 1.1 – Головна сторінка web-додатку «Титан»

На головній сторінці web-додатку «Титан» багато чого розміщено, але стилізація розділів та взагалі наповнення головної сторінки виконано не найкращим чином. Оформлення тексту web-додатку не сподобалося. Слід попрацювати з дизайном сайту. Також слайдер завеликий, та займає більшу частину web-додатку, що недоречно.

Для прикладу розглянемо розділ портативної електроніки, а саме електронну книгу (рис. 1.2).

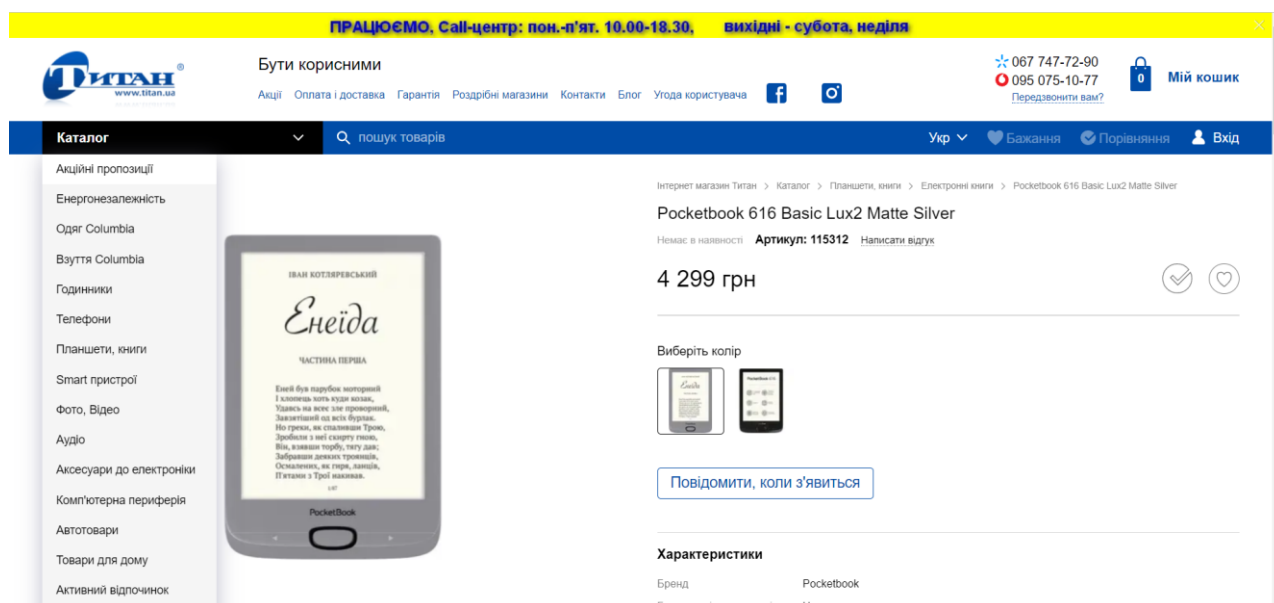


Рисунок 1.2 – Розділ портативної електроніки web-додатку «Титан»

По-перше, використані кольори тексту погано сприймаються, а по-друге, саме представлення портативної електроніки не викликає бажання її придбати.

Другий аналог, що буде розглядатися – це MobyStore [4]. Головна сторінка web-додатку представлена на рисунку 1.3.

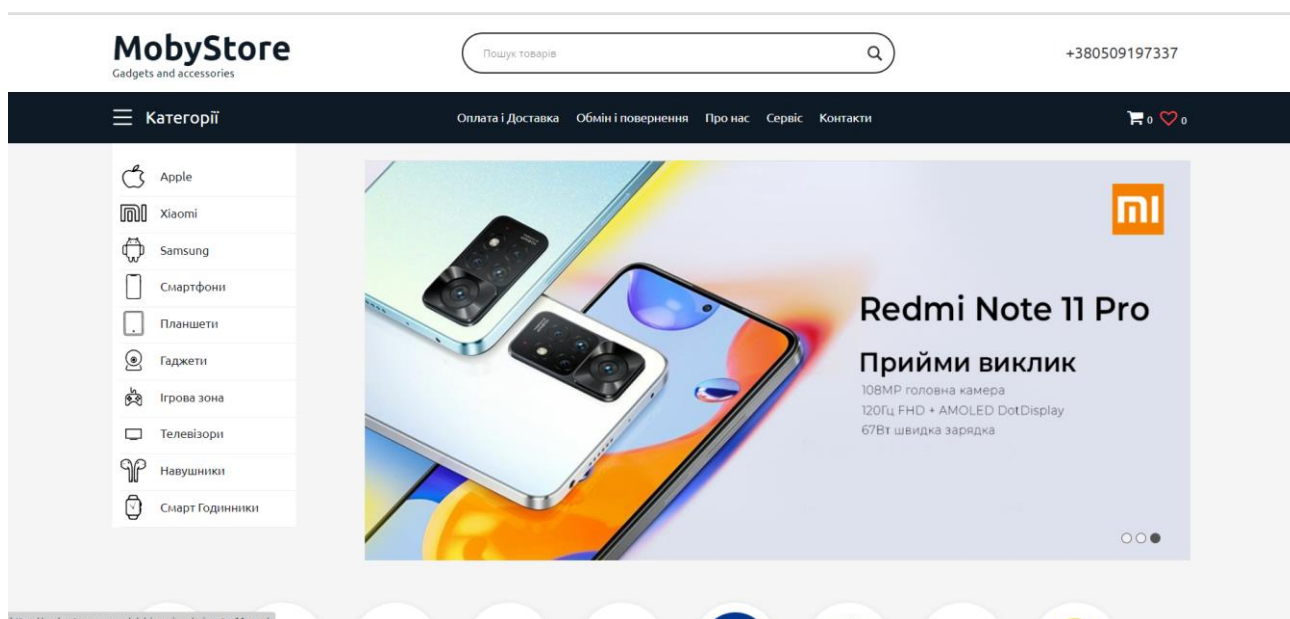


Рисунок 1.3 – Головна сторінка web-додатку «MobyStore»

На головній сторінці web-додатку «MobyStore» розміщено зручне навігаційне меню, слайдер. Web-додаток має сучасний дизайн, інформація вся згрупована, виконано умови сучасних сайтів. Під час розробки було використано темно сині та білі кольори. Цей сайт набагато кращий на відміну від попереднього.

При наведенні на категорію товарів автоматично виводиться підменю, що є досить зручною функцією (рис. 1.4).

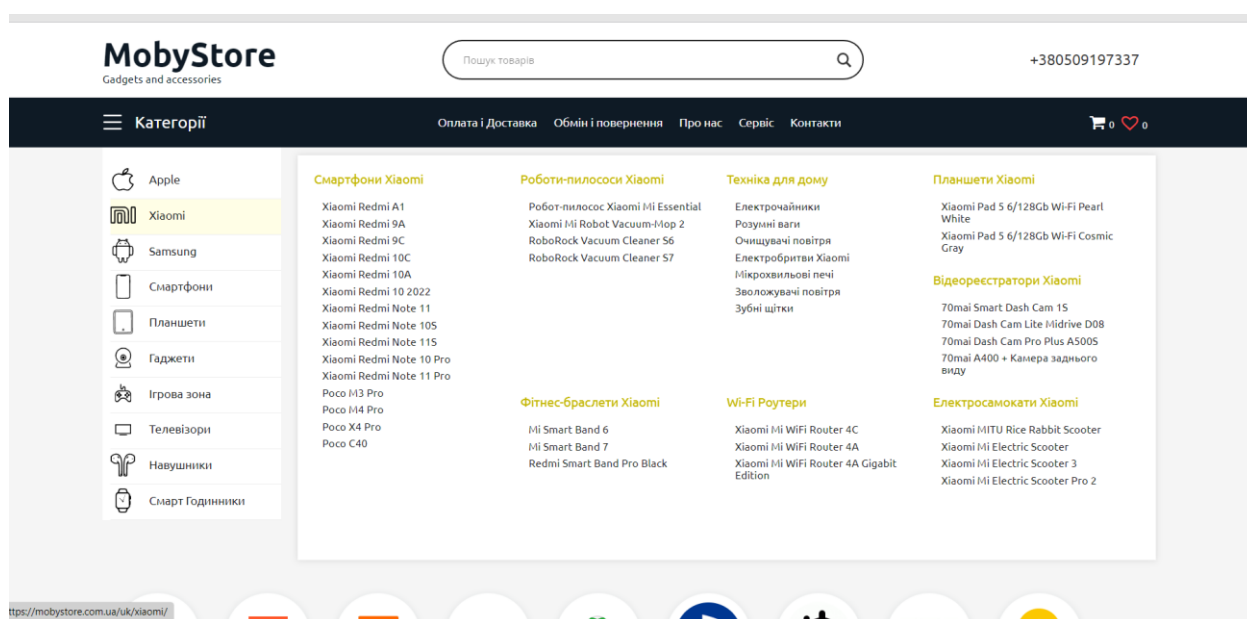


Рисунок 1.4 – Підменю категорій web-додатку «MobyStore»

Проте є й свої недоліки. Наприклад відсутня досить популярна функція «порівняння товару» (рис. 1.5). Також немає можливості змінити мову web-додатку.

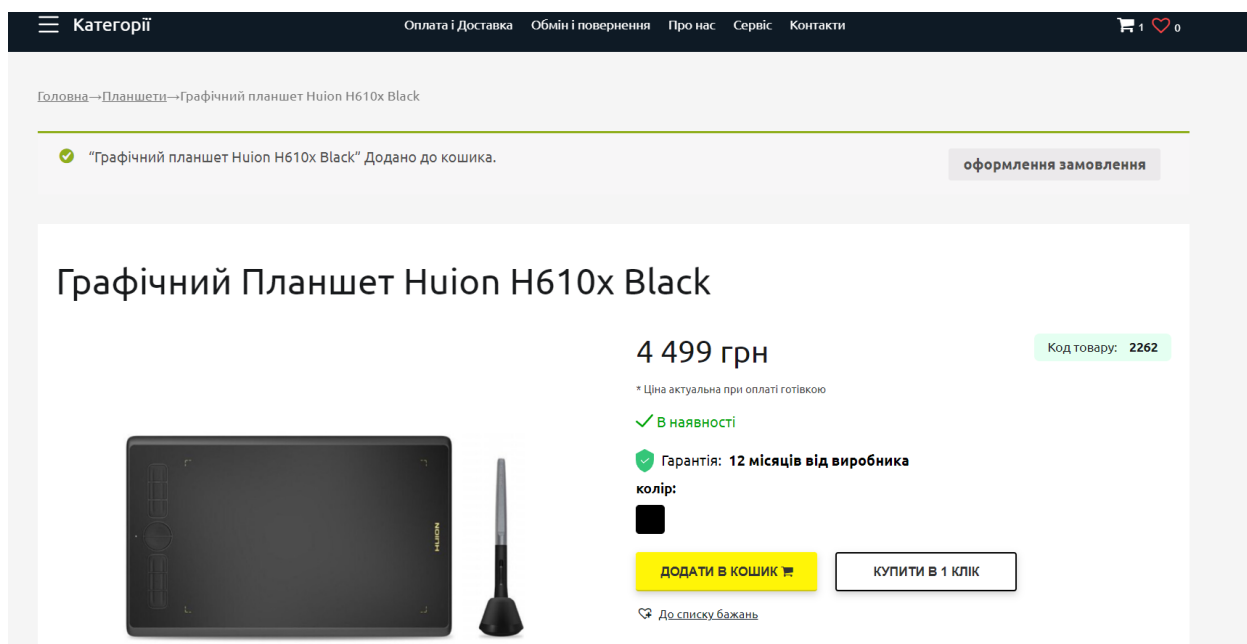


Рисунок 1.5 – Вид сторінки при переході на деякий продукт web-додатку «MobyStore»

Останній аналог – це «Compservice»[5]. Цей web-додаток є дуже популярним в Україні.

Головна сторінка web-додатку «Compservice» приємна на сприйняття (рис. 1.6). Гарно підібрані кольори. По центру ми бачимо рядок пошуку товарів, що є досить зручно.

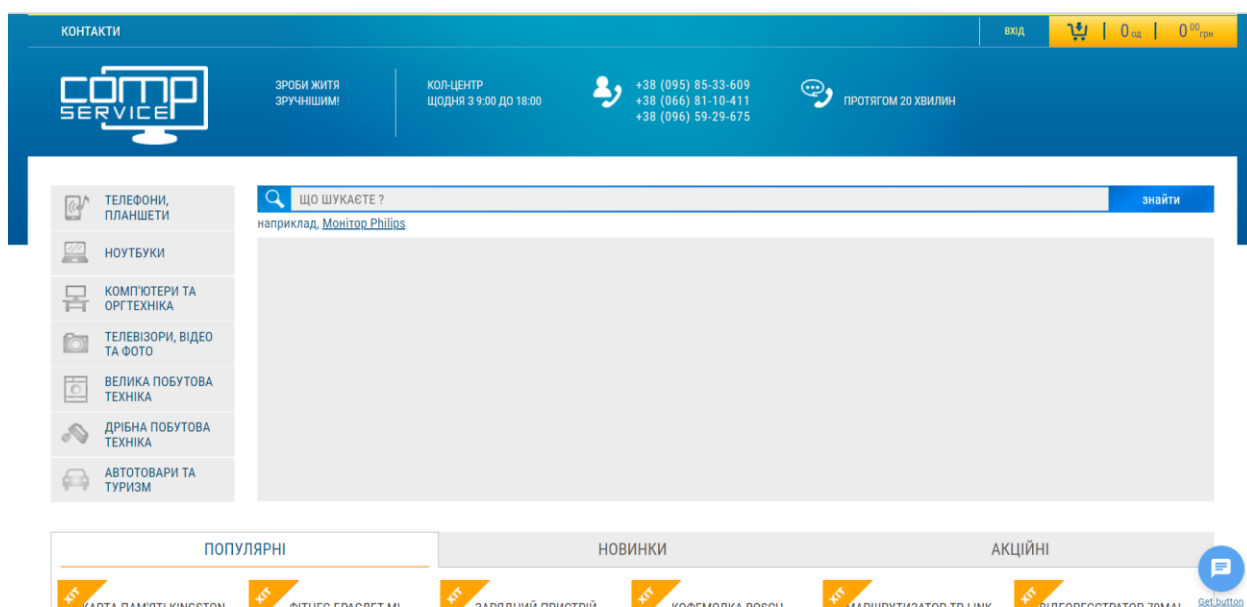


Рисунок 1.6 – Головна сторінка web-додатку «Compservice»

Але все ж таки web-додаток не ідеальний та має свої недоліки. Як бачимо, якщо в категорії товарів невеликий вибір, то залишається дуже багато зайвого, незаповненого місця, і це виглядає не досить коректно.

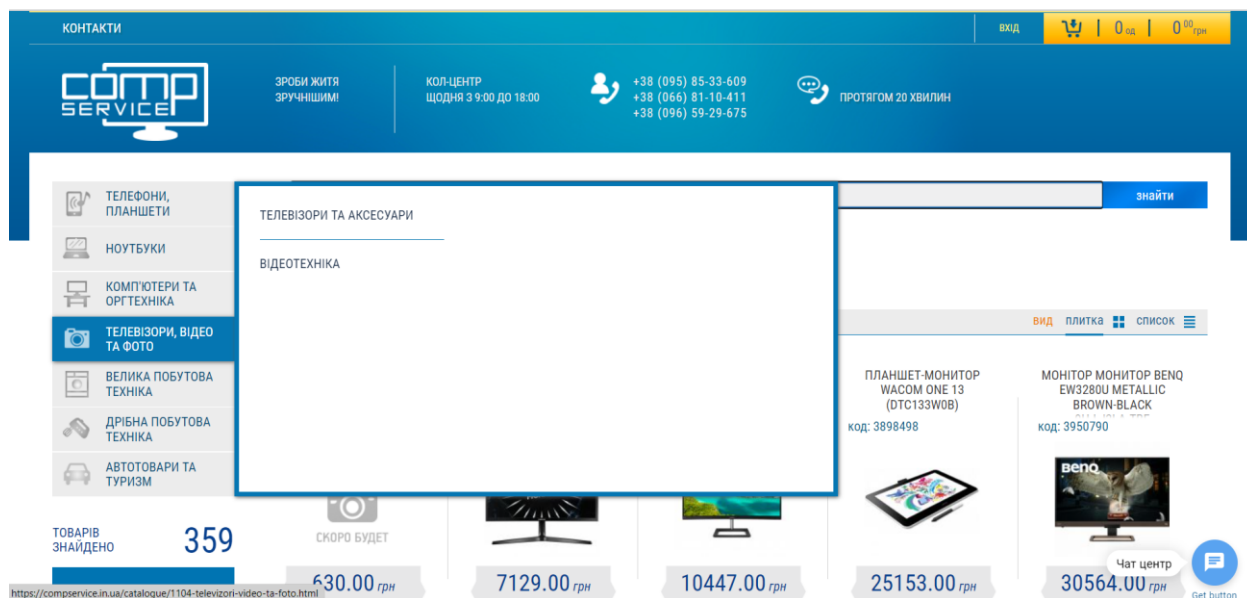


Рисунок 1.7 – Категорії web-додатку «Compservice»

Також не досить вдало підібрані кольори самих розділів та характеристик товарів (рис. 1.8). На мою думку, потрібно було використати більш виражені кольори.

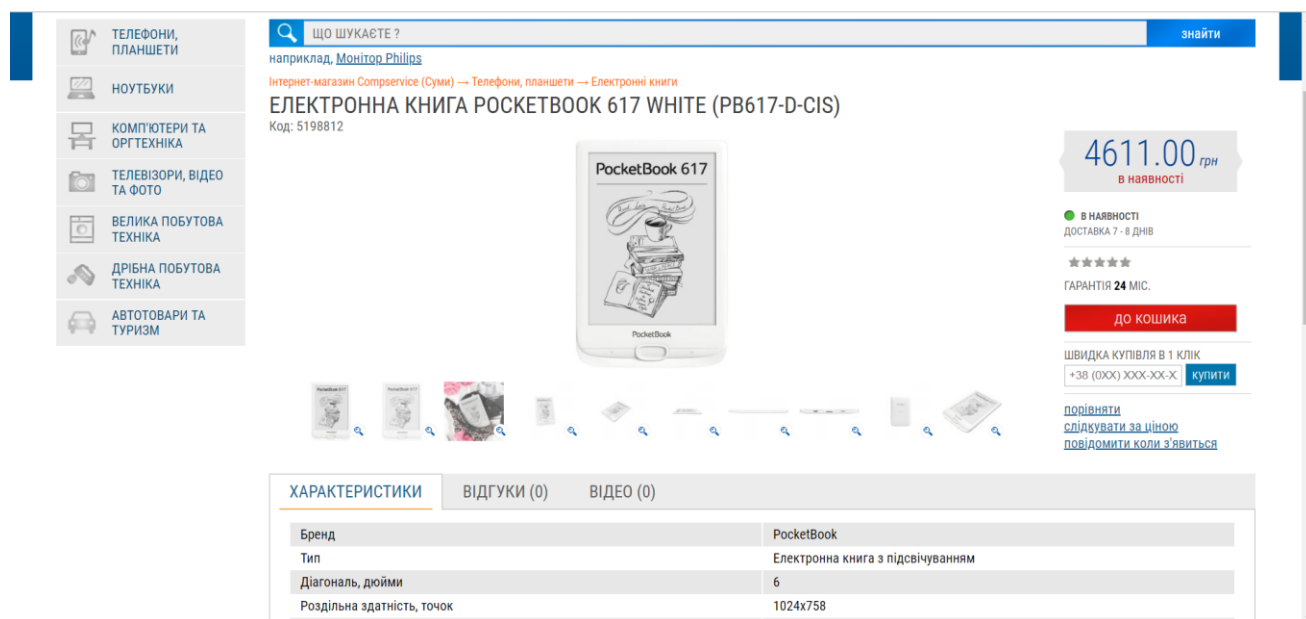


Рисунок 1.8 – Вид сторінки характеристик товару

За результатами даного аналізу web-аналогів, для продажу портативної електроніки, були визначені основні вимоги до розроблюваного продукту, які можна побачити в таблиці 1.2.

Таблиця 1.2 – Порівняння додатків аналогів з розроблюваним проектом на основі сучасних вимог

Характеристика/ Додаток	«Титан»	«MobyStore»	«Compservice»	Розроблюваний проект
Сучасний дизайн	-	+	+	+
Зручний інтерфейс	-	+	-	+
Адаптивність	+	+	+	+
Порівняння товарів	+	-	+	+
Перегляд історії пошуку товарів	-	-	-	+
Реєстрація	+	+	+	+
Перезавантаженість контентом	-	-	+	-

З таблиці 1.2 видно, що не має ідеального додатку для продажу портативної електроніки, виходячи з цього було прийнято рішення про розроблення власного web-додатку, щоб виправити всі недоліки попередніх аналогів та реалізувати всі найкращі функції, що вони мають.

1.3 Постановка задачі

Метою проекту є розробка web-додатку для підтримки продажу портативної електроніки зі зручним та легким у використанні інтерфейсом.

Використання цього додатку забезпечить можливість користувачам заощадити час на перегляд, пошук та придбання портативної електроніки. Додаток надасть можливість обрати користувачу саме той товар, який йому потрібен. Для цього у користувача буде можливість використати фільтри які значно полегшать пошук бажаного товару.

Необхідно виконати ряд завдань, щоб досягти поставленої мети:

- оцінити актуальність дослідження;
- вивчити предметну область;
- проаналізувати аналоги: усунути їх недоліки та взяти до уваги переваги;
- створити модель web-додатку;
- обрати технології для розробки;
- створити прототип web-додатку;
- реалізувати функціонал та створити структуру web-додатку;
- виконати тестування;

Всі необхідні вимоги до розроблюваного проекту прописані у технічному завданні, яке можна побачити у додатку А.

Розробка web-додатка здійснюватиметься за допомогою PHP – мова програмування та фреймворку Laravel. Створення фронтенду відбуватиметься за використання HTML для розмітки сторінок та встановлення структури, CSS і Bootstrap для стилізації, а також JavaScript для додавання динамічності. Щодо управління та зберігання даних, планується використовувати реляційну базу даних MySQL.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКУ

2.1 Структурно-функціональне моделювання web-додатку

IDEF0 – це метод графічного моделювання, для структуризації, аналізу та моделювання функціональних аспектів системи, особливо в контексті бізнес-процесів та інформаційних систем.

Основна ідея IDEF0 полягає в тому, що функції системи розбиваються на більш дрібні функції, які моделюються у вигляді блоків, відомих як функціональні блоки. Ці функціональні блоки зображаються у вигляді стрілок та прямокутників, що репрезентують вхідні та вихідні дані, а також дії, які виконуються над цими даними.

IDEF0 є потужним інструментом для аналізу та моделювання функціональних аспектів систем. Він допомагає розбити складні процеси на більш зрозумілі блоки, аналізувати їх взаємозв'язки та виявляти можливі покращення [14].

Функціональне моделювання бізнес-процесу підтримки продажу портативної електроніки за допомогою web-додатку в нотації IDEF0 можна побачити на рисунку 2.1.

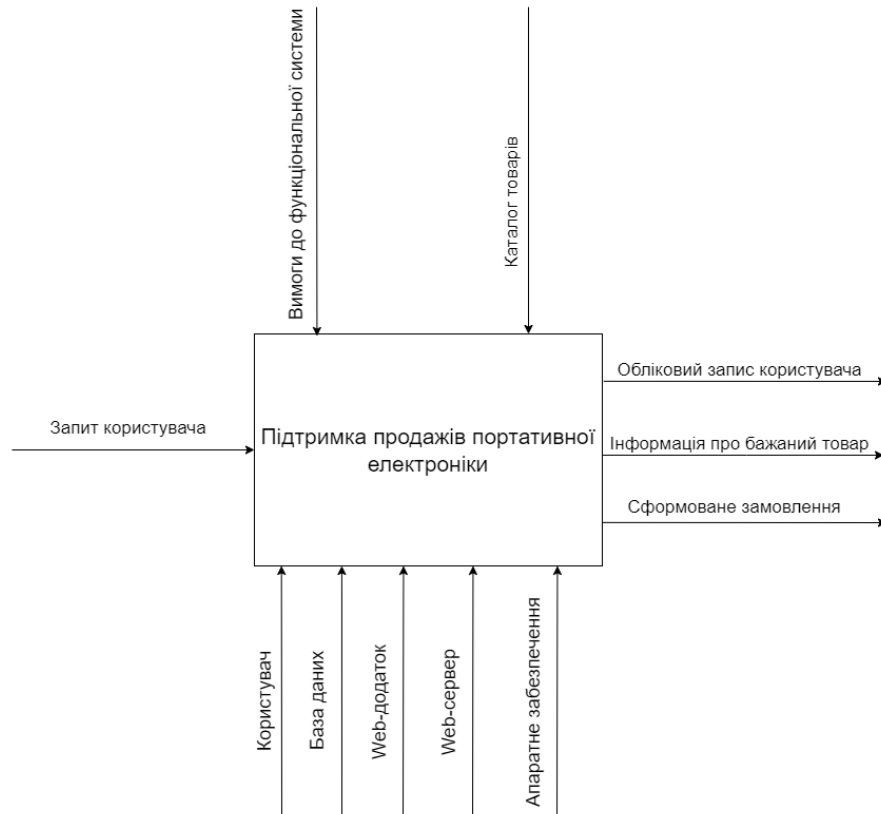


Рисунок 2.1 – Контекстна діаграма IDEF0

Взаємодія з програмним продуктом починається, якщо до програми надійшов певний запит. Це може бути запит на пошук інформації або авторизації чи реєстрації від користувача.

В свою чергу web-додаток повинен забезпечити користувача каталогом відповідних товарів та надати гарну функціональність, щоб він з легкістю обрав бажаний товар.

Для роботи з web-додатком потрібен безпосередньо користувач, база даних та програмне забезпечення.

Кінцевим результатом роботи може бути створений обліковий запис користувача, сформоване замовлення або просто інформація про бажаний товар.

Діаграма декомпозиція IDEF0 використовується для більшої деталізації внутрішніх процесів (рис. 2.2).

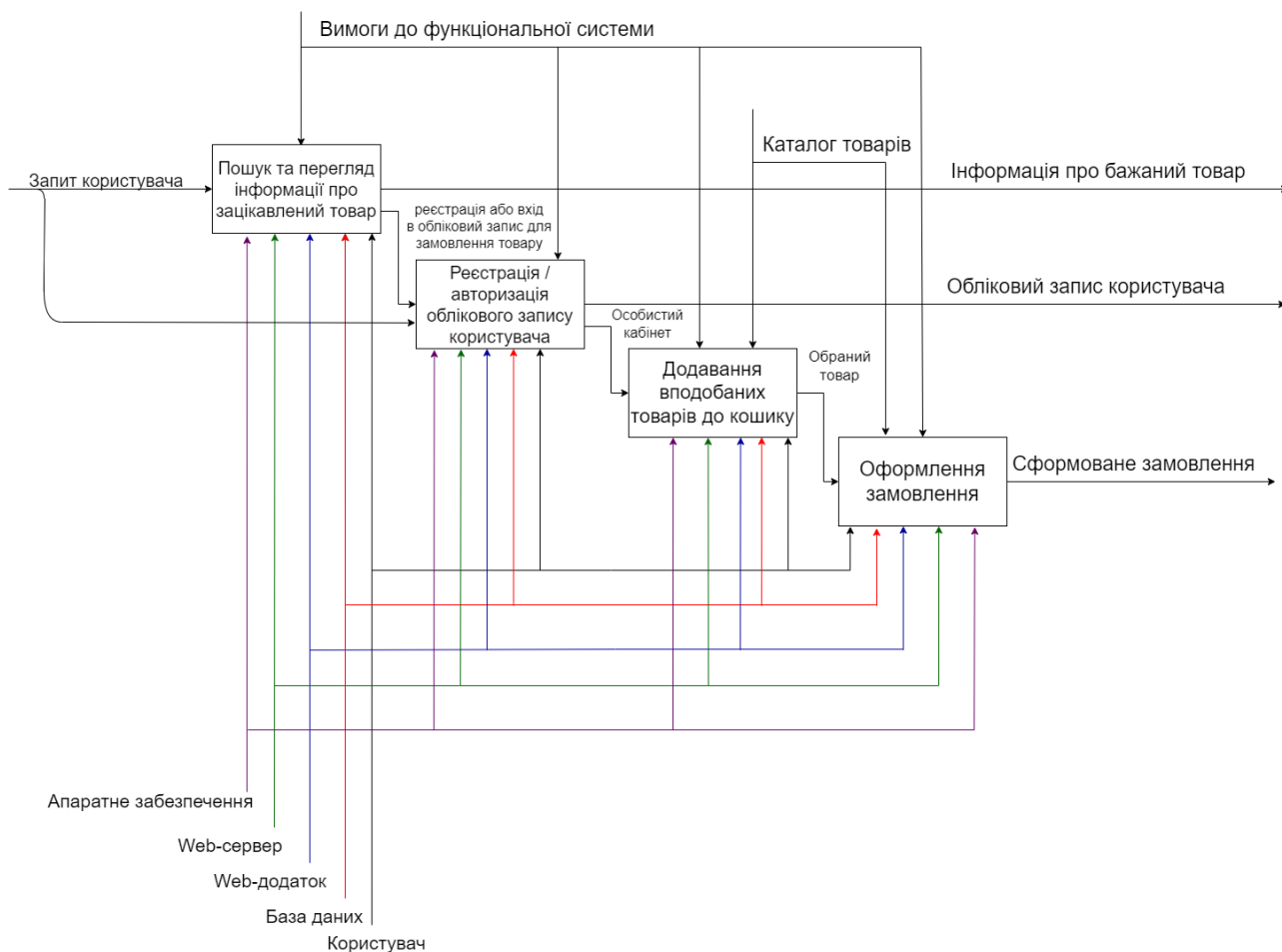


Рисунок 2.2 – Декомпозиція контекстної діаграми IDEF0

Після проведеної декомпозиції IDEF0 було виділено чотири основні функціональні блоки, а саме:

- пошук та перегляд інформації про зацікавлений товар;
- реєстрація або авторизація облікового запису користувача;
- додавання вподобаних товарів до кошику;
- оформлення замовлення.

2.2 Проектування інформаційної системи

2.2.1 Діаграма варіантів використання web-додатку

Use-case діаграма є графічним інструментом моделювання, що використовується в області розробки програмного забезпечення для візуалізації функціональності системи з точки зору її користувачів або акторів.

Використання цієї діаграми дозволяє ідентифікувати основні функціональні можливості системи, взаємодію між акторами та системою, а також послідовність подій у процесі взаємодії.

Основною перевагою використання use-case діаграми є її простота та зрозумілість, що дозволяє команді проекту легко спілкуватися та вирішувати питання щодо функціональних вимог системи.

Діаграма варіантів використання web-додатку зображена на рисунку 2.3.

У системі існує 3 типи акторів:

- Гість – користувач, що може авторизуватися або зареєструватися для більш широкого функціоналу додатку, або якщо він не бажає цього робити просто переглянути та здійснити пошук товарів.
- Зареєстрований користувач – клієнт, що може виконувати пошук товарів за фільтрами, порівнювати їх додавати товар до кошика, а потім оформити замовлення.
- Адміністратор – користувач, що має особливі права доступу і може додавати, редагувати та видаляти товари у каталозі, а також повинен обробляти замовлення з кошику.

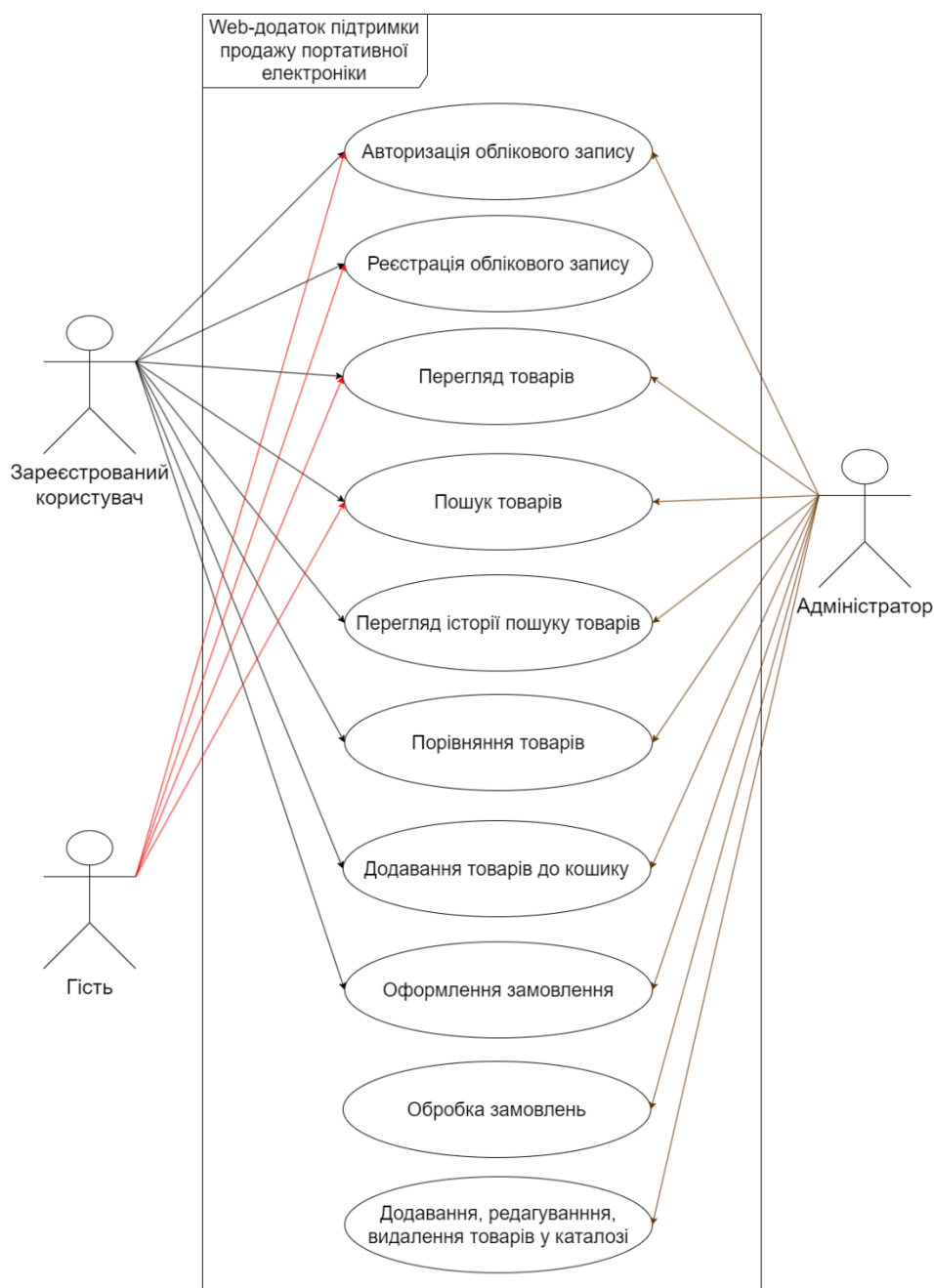


Рисунок 2.3 – Діаграма варіантів використання

2.2.2 Структурне моделювання

Структурне моделювання – це взаємодія між базою даних та безпосередньо веб-додатком. UML діаграма діяльності, що показує основні операції, які залежать від БД, зображена на рисунку 2.4.

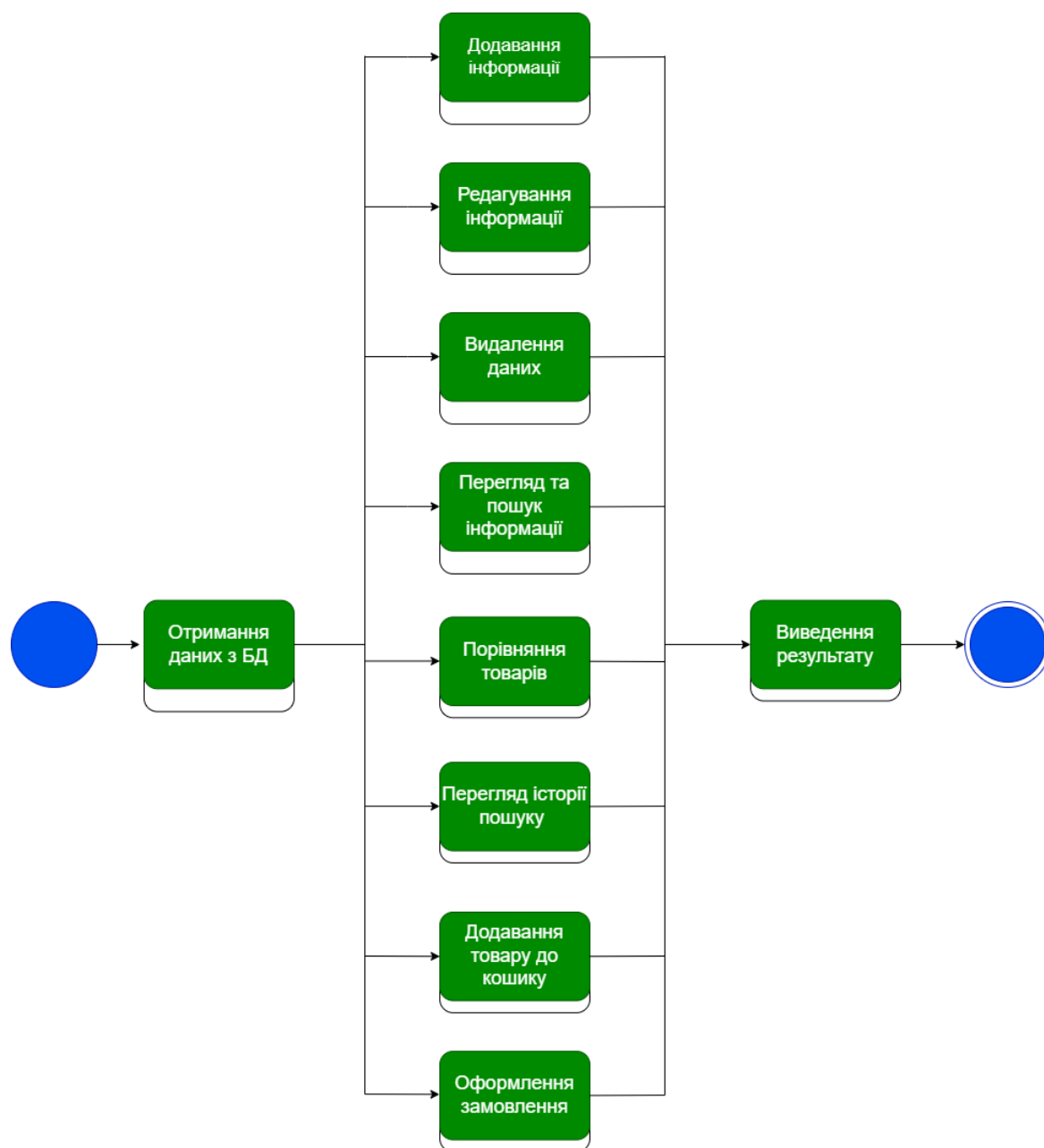


Рисунок 2.4 – Діаграма діяльності з підтримки продажу портативної електроніки

2.3 Проектування моделі бази даних

Спочатку було проведено вибір системи управління базами даних, яка повинна задовольняти такі вимоги, як робота на різних платформах, шифрування даних під час передачі, швидкодія та зручність в експлуатації. Після розгляду цих вимог було прийнято рішення вибрати MySQL як систему управління реляційними базами даних.

По-перше, MySQL є відомою своєю швидкодією та надійністю, що робить її привабливою для веб-додатків з великим обсягом даних. Крім того, вона підтримує масштабованість, дозволяючи легко розширювати та вдосконалювати базу даних зі зростанням потреб.

По-друге, вона є дуже популярною та широко використовується в галузі веб-розробки. Це означає, що є велика спільнота розробників, яка забезпечує багато ресурсів, підтримку та оновлення. [15].

База даних «shop.sql» містить такі таблиці:

- users: таблиця, яка зберігає в собі дані авторизації, а саме пошту та пароль. Роль користувача на сайті(адміністратор або звичайний користувач);
- cabinets: таблиця, яка зберігає в собі дані для відправлення замовлення(ім'я, прізвище, дата народження, номер телефону, місто в яке потрібно відправити товар, пошта відправлення);
- baskets: таблиця, яка зберігає в собі дані товарів, що були додані в кошик;
- histories: таблиця, яка зберігає в собі дані переглянутих товарів;
- migrations: таблиця, яка поєднує між собою Laravel та Mysql;
- products: таблиця, яка зберігає загальну про товари;
- categories: таблиця, яка зберігає категорії до яких належать товари;
- compares: таблиця, яка зберігає товари які необхідно порівняти;
- orders: таблиця з даними про оформлене замовлення;
- options: таблиця з описом характеристики товару;
- options_products: таблиця, яка зберігає детальну інформацію про товари;
- brands: таблиця, яка зберігає назву бренду товарів.

Фізична модель бази даних показана на рисунку 2.5

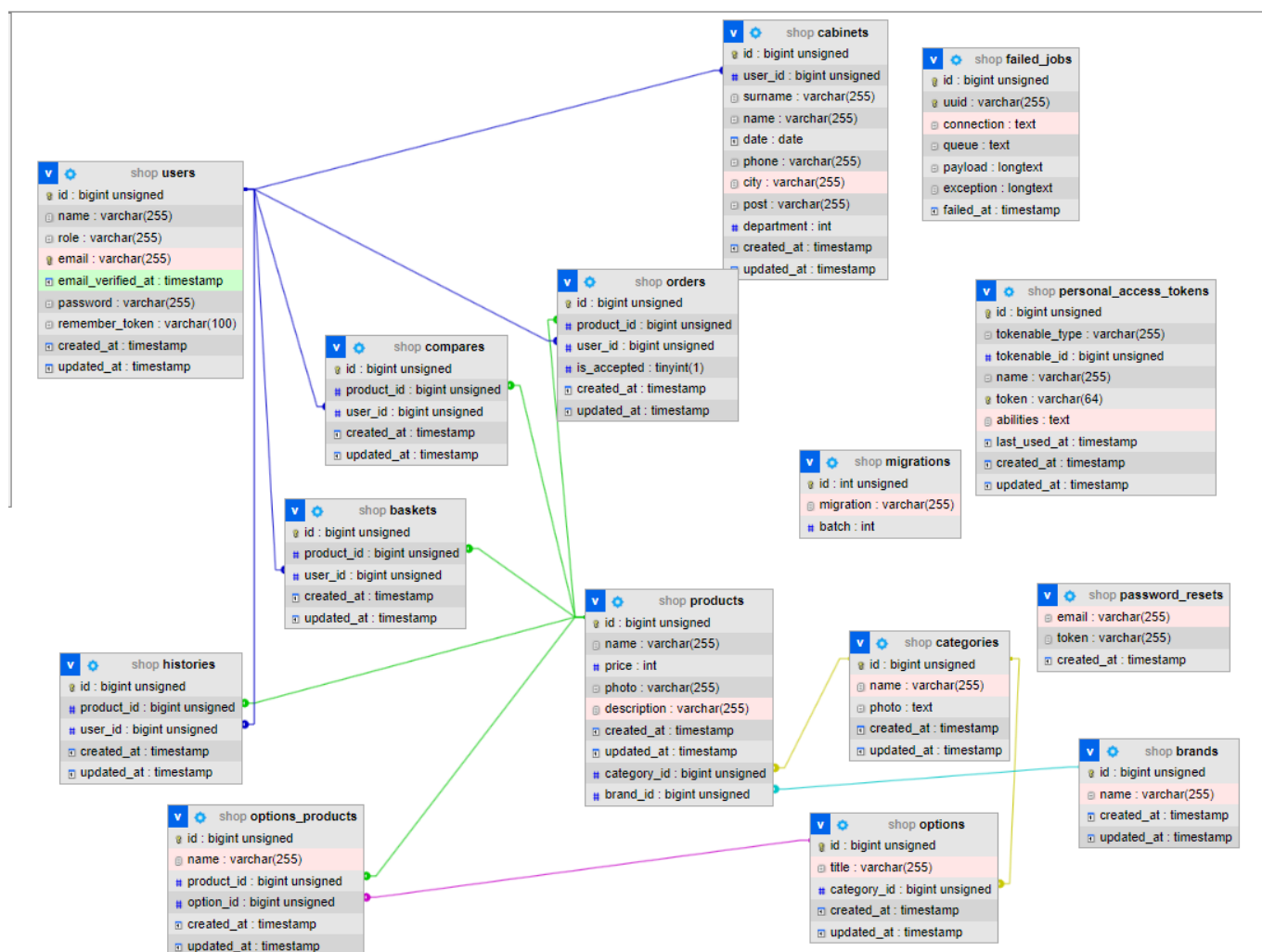


Рисунок 2.5 – Фізична модель бази даних web-додатку

3 РОЗРОБКА WEB-ДОДАТКУ

3.1 Архітектура програмного додатку

Для розробки web-додатку було використано Model-View-Controller (MVC) патерн, діаграму взаємодії компонентів якого показано на рис. 3.1.

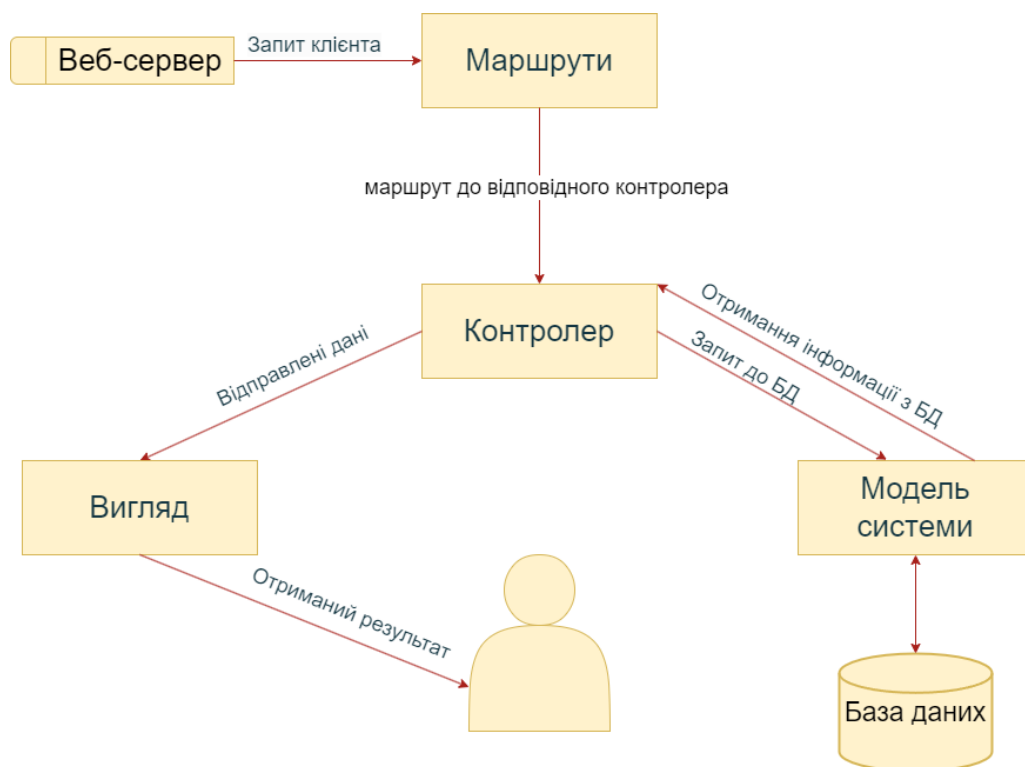


Рисунок 3.1 – Діаграма MVC

Складовими MVC є:

- model – це ядро функціональності додатку, де здійснюються складні операції (обробка, сортування, збереження даних, взаємодія з базою даних, генерація веб-сторінок та інші), після чого результат передається до представлення (View)
- view – це інтерфейс, який забезпечує взаємодію користувача з додатком та відображення його візуальної складової (наприклад, веб-сторінки). Основна роль представлення полягає в зчитуванні введеної користувачем інформації і передачі її до відповідного методу або функції контролера. Також представлення відстежує дії

користувача, такі як натискання кнопок або оновлення сторінки, і викликає відповідні методи або функції контролера для їх обробки

– `controller` – цей компонент виконує роль зв'язку між `model` та `view`. Він приймає запити, взаємодіє з моделлю для отримання необхідних даних та відправляє їх до відповідного представлення для відображення результату користувачу. Контролер є посередником, що забезпечує розподіл завдань між моделлю та представленням, що сприяє збереженню чистоти та організованості коду в програмі [16].

3.2 Програмна реалізація

Створення користувальницького інтерфейсу веб-додатку виконано з використанням інтегрованого середовища розробки PhpStorm. Для розмітки основи макету було використано гіпертекстову розмітку HTML, а для задання візуального відображення сторінок використано мову стилів CSS. Макет включає три блочні елементи: верхню частину (`header`), основний контент (`body`) та нижню частину (`footer`). Ця структура сторінки спрощує і прискорює процес верстання сторінок додатку.

Так як я використовув MVC патерн то на рисунках 3.2 – 3.4 зображені основні контролери (`controller`) моделі (`model`) та інтерфейс (`view`) відповідно.

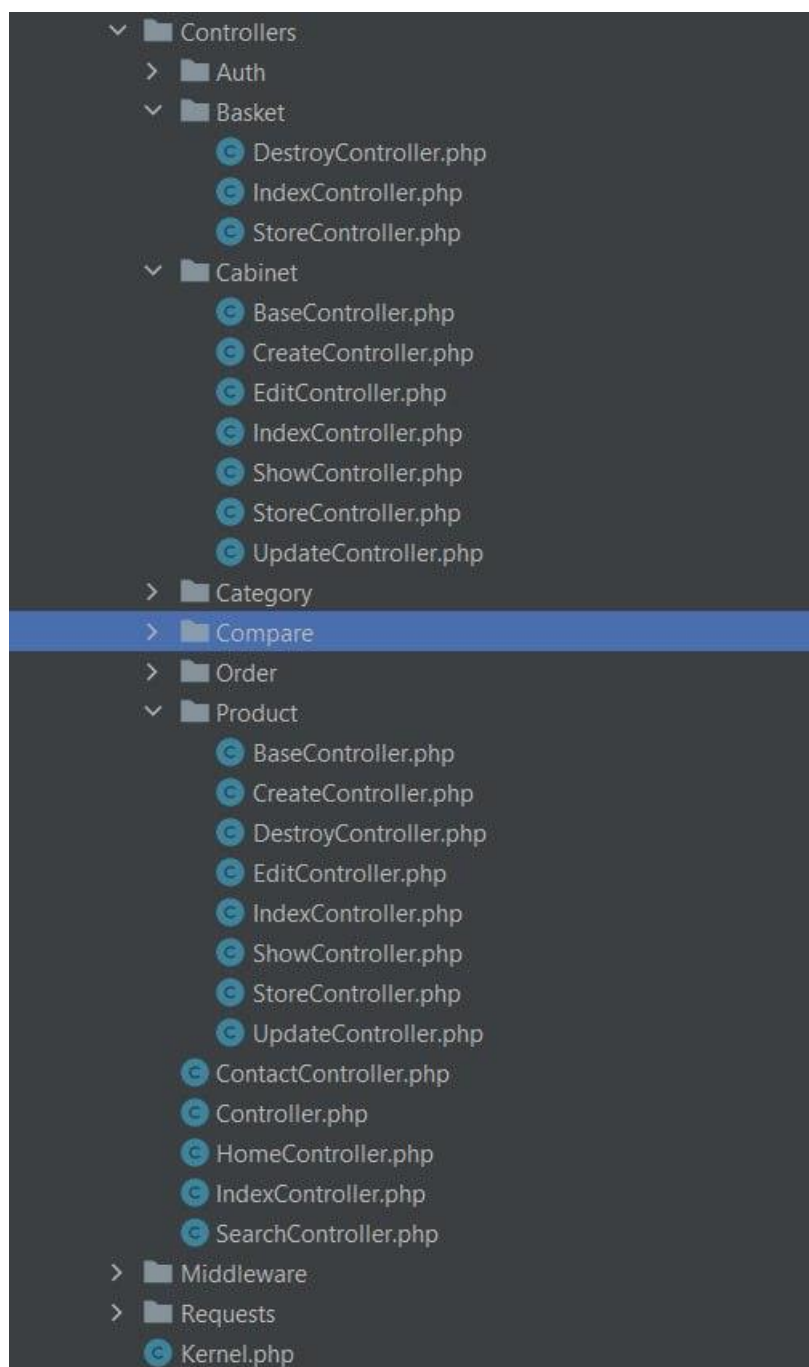


Рисунок 3.2 – Контролери веб-додатку

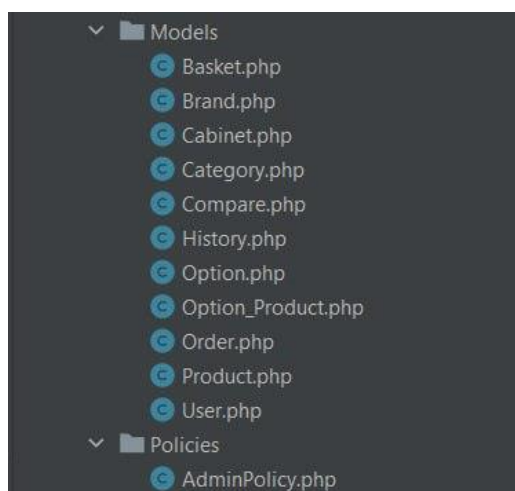


Рисунок 3.3 – Моделі веб-додатку

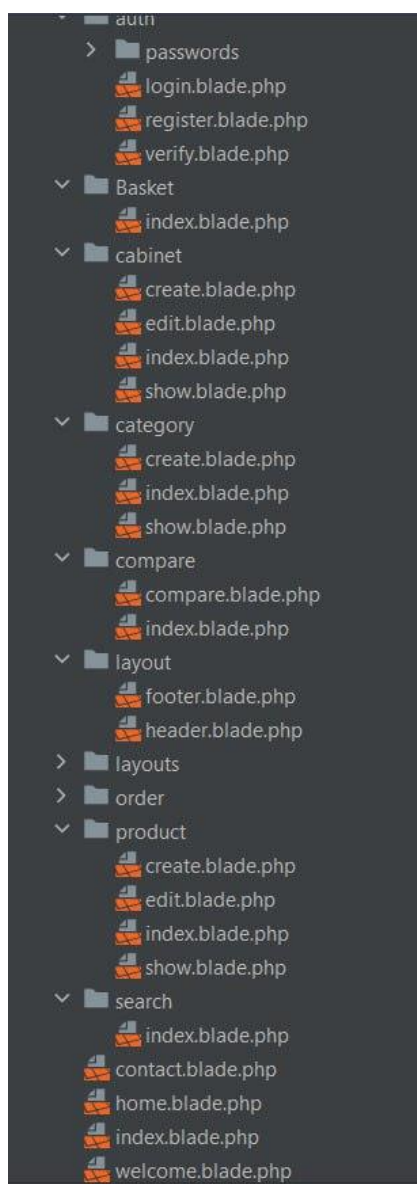


Рисунок 3.4 – Інтерфейс веб-додатку

BaseController.php – контролер, що слугує для перевизначення базового контролеру

IndexController.php – файл який передає необхідні дані для виведення на екран основної інформації підрозділу сайту.

EditController.php – контролер, який направляє на сторінку для зміни певної інформації на сайті.

UpdateController.php – контролер, що за допомогою моделі замінює інформацію в базі даних.

CreateController.php - контролер, який направляє на сторінку для додавання певної інформації на сайті.

StoreController.php - контролер, що за допомогою моделі додає інформацію в базу даних.

DestroyController.php - контролер, що за допомогою моделі видаляє інформацію з бази даних.

index.blade.php – головна сторінка певного підрозділу сайту.

create.blade.php – форма додавання певної інформації на сайт.

edit.blade.php – форма редагування певної інформації на сайті.

show.blade.php – сторінка для відображення певного атрибуту сайту.

Моделі потрібні для з'єднання з базою даних, щоб створювати, зчитувати, оновлювати та видаляти інформацію в базі даних.

Розміри блоків макету сторінок не задані, а можуть варіюватися залежно від потреб та вимог проекту. В CSS файлі ширина цих блоків встановлюється відсотками від ширини екрану. При стандартному налаштуванні, ширина додатку складає 1080 пікселів, проте за допомогою медіа-запитів у CSS стилі адаптуються до ширини використаного пристрою. Код деякого медіа-запиту:

```
@media screen and (max-width: 1440px) {
    .wrapper_category{
        grid-template-columns: repeat(4, 1fr);
    }
}
```

```

.input_search{
  margin-left: 2%;
  width: 90%;
}

.btn_search{
  width: 150%;
}

h1{
  font-size: 45px;
}

h2{
  font-size: 35px;
}

.hot_img{
  width: 80%;
  height: 60%;
}
}

```

У складі основних елементів стилізації можна виділити використання єдиного шрифту з назвою «Trebuchet». Основний колір фону сторінок білий та чорний. Нижче наведено кілька прикладів CSS-стилів для оформлення основних блоків сторінок:

```

*{
  text-align: center;
  color: black;
  font-family: "Trebuchet MS";
}

a{
  text-decoration: none;
}

body{
  background-image: url("../images/back.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
  padding: 1% 5%;
}

hr{
  border: 2px solid black;
}

```

```

}

input{
  border-radius: 8px;
  outline: none;
}

```

Для того, щоб зберегти контент web-додатку і реалізувати програмно-апаратну частину, була розроблена база даних з назвою «shop.sql», яка розміщена на локальному сервері. У розробці використовувався програмний комплекс Open Server Panel, зокрема базовий додаток PhpMyAdmin.

Структуру бази даних можна побачити на рисунках 3.5 - 3.16.

Таблиця	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> baskets	☆ [іконки]	0	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
<input type="checkbox"/> brands	☆ [іконки]	2	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> cabinets	☆ [іконки]	1	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
<input type="checkbox"/> categories	☆ [іконки]	6	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> compares	☆ [іконки]	2	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
<input type="checkbox"/> failed_jobs	☆ [іконки]	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> histories	☆ [іконки]	75	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
<input type="checkbox"/> migrations	☆ [іконки]	14	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> options	☆ [іконки]	7	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
<input type="checkbox"/> options_products	☆ [іконки]	14	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
<input type="checkbox"/> orders	☆ [іконки]	0	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
<input type="checkbox"/> password_resets	☆ [іконки]	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> personal_access_tokens	☆ [іконки]	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
<input type="checkbox"/> products	☆ [іконки]	2	InnoDB	utf8mb4_unicode_ci	48.0 КиБ	-
<input type="checkbox"/> users	☆ [іконки]	3	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-

Рисунок 3.5 – Всі таблиці бази даних «shop.sql»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 product_id 🔑	bigint		UNSIGNED	Да	<i>NULL</i>			Ещё ▾
<input type="checkbox"/>	3 user_id 🔑	bigint		UNSIGNED	Да	<i>NULL</i>			Ещё ▾
<input type="checkbox"/>	4 created_at	timestamp			Да	<i>NULL</i>			Ещё ▾
<input type="checkbox"/>	5 updated_at	timestamp			Да	<i>NULL</i>			Ещё ▾

Рисунок 3.6 – Структура таблиці «baskets»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 name	varchar(255)	<i>utf8mb4_unicode_ci</i>		Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	3 created_at	timestamp			Да	<i>NULL</i>			Ещё ▾
<input type="checkbox"/>	4 updated_at	timestamp			Да	<i>NULL</i>			Ещё ▾

Рисунок 3.7 – Структура таблиці «brands»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	<i>Нет</i>		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 user_id 🔑	bigint		UNSIGNED	Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	3 surname	varchar(255)	<i>utf8mb4_unicode_ci</i>		Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	4 name	varchar(255)	<i>utf8mb4_unicode_ci</i>		Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	5 date	date			Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	6 phone	varchar(255)	<i>utf8mb4_unicode_ci</i>		Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	7 city	varchar(255)	<i>utf8mb4_unicode_ci</i>		Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	8 post	varchar(255)	<i>utf8mb4_unicode_ci</i>		Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	9 department	int			Нет	<i>Нет</i>			Ещё ▾
<input type="checkbox"/>	10 created_at	timestamp			Да	<i>NULL</i>			Ещё ▾
<input type="checkbox"/>	11 updated_at	timestamp			Да	<i>NULL</i>			Ещё ▾

Рисунок 3.8 – Структура таблиці «cabinets»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	3 photo	text	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	4 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	5 updated_at	timestamp			Да	NULL			Ещё ▾

Рисунок 3.9 – Структура таблиці «categories»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 product_id 🔑	bigint		UNSIGNED	Нет	Нет			Ещё ▾
<input type="checkbox"/>	3 user_id 🔑	bigint		UNSIGNED	Нет	Нет			Ещё ▾
<input type="checkbox"/>	4 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	5 updated_at	timestamp			Да	NULL			Ещё ▾

Рисунок 3.10 – Структура таблиці «compares»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 product_id 🔑	bigint		UNSIGNED	Нет	Нет			Ещё ▾
<input type="checkbox"/>	3 user_id 🔑	bigint		UNSIGNED	Нет	Нет			Ещё ▾
<input type="checkbox"/>	4 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	5 updated_at	timestamp			Да	NULL			Ещё ▾

Рисунок 3.11 – Структура таблиці «histories»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	3 category_id 🔑	bigint		UNSIGNED	Нет	Нет			Ещё ▾
<input type="checkbox"/>	4 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	5 updated_at	timestamp			Да	NULL			Ещё ▾

Рисунок 3.12 – Структура таблиці «option»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	3 product_id 🔑	bigint		UNSIGNED	Нет	Нет			Ещё ▾
<input type="checkbox"/>	4 option_id 🔑	bigint		UNSIGNED	Нет	Нет			Ещё ▾
<input type="checkbox"/>	5 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	6 updated_at	timestamp			Да	NULL			Ещё ▾

Рисунок 3.13 – Структура таблиці «option_product»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 product_id 🔑	bigint		UNSIGNED	Да	NULL			Ещё ▾
<input type="checkbox"/>	3 user_id 🔑	bigint		UNSIGNED	Да	NULL			Ещё ▾
<input type="checkbox"/>	4 is_accepted	tinyint(1)			Нет	0			Ещё ▾
<input type="checkbox"/>	5 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	6 updated_at	timestamp			Да	NULL			Ещё ▾

Рисунок 3.14 – Структура таблиці «orders»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	3 price	int			Нет	Нет			Ещё ▾
<input type="checkbox"/>	4 photo	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	5 description	varchar(255)	utf8mb4_unicode_ci		Да	NULL			Ещё ▾
<input type="checkbox"/>	6 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	7 updated_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	8 category_id 🔑	bigint		UNSIGNED	Да	NULL			Ещё ▾
<input type="checkbox"/>	9 brand_id 🔑	bigint		UNSIGNED	Да	NULL			Ещё ▾

Рисунок 3.15 – Структура таблиці «products»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id 🔑	bigint		UNSIGNED	Нет	Нет		AUTO_INCREMENT	Ещё ▾
<input type="checkbox"/>	2 name	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	3 role	varchar(255)	utf8mb4_unicode_ci		Да	NULL			Ещё ▾
<input type="checkbox"/>	4 email 📧	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	5 email_verified_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	6 password	varchar(255)	utf8mb4_unicode_ci		Нет	Нет			Ещё ▾
<input type="checkbox"/>	7 remember_token	varchar(100)	utf8mb4_unicode_ci		Да	NULL			Ещё ▾
<input type="checkbox"/>	8 created_at	timestamp			Да	NULL			Ещё ▾
<input type="checkbox"/>	9 updated_at	timestamp			Да	NULL			Ещё ▾

Рисунок 3.16 – Структура таблиці «users»

3.3 Використання web-додатку покупцем

На початку роботи з web-додатком користувач опиняється на початковій сторінці, це головна сторінка програмного продукту. (рис. 3.17). У верхній частині сторінки розташоване основне навігаційне меню, яке дозволяє переходити до інших розділів, таких як пошук товарів, вхід або реєстрація, а також логотип магазину. Логотип розміщений у лівій частині та виконує свою функцію посилання на головну сторінку, забезпечуючи можливість повернення користувача на неї з будь-якого розділу додатку. Під ним розташовується блок контенту, де знаходиться слайдер з тематичними фотографіями, а також фото товару, що дає зрозуміти, що магазин орієнтується на продаж портативної електроніки. На нижньому блоці сторінки присутня коротка інформація про магазин, зокрема навігаційне меню, офіційні представники та назва магазину.

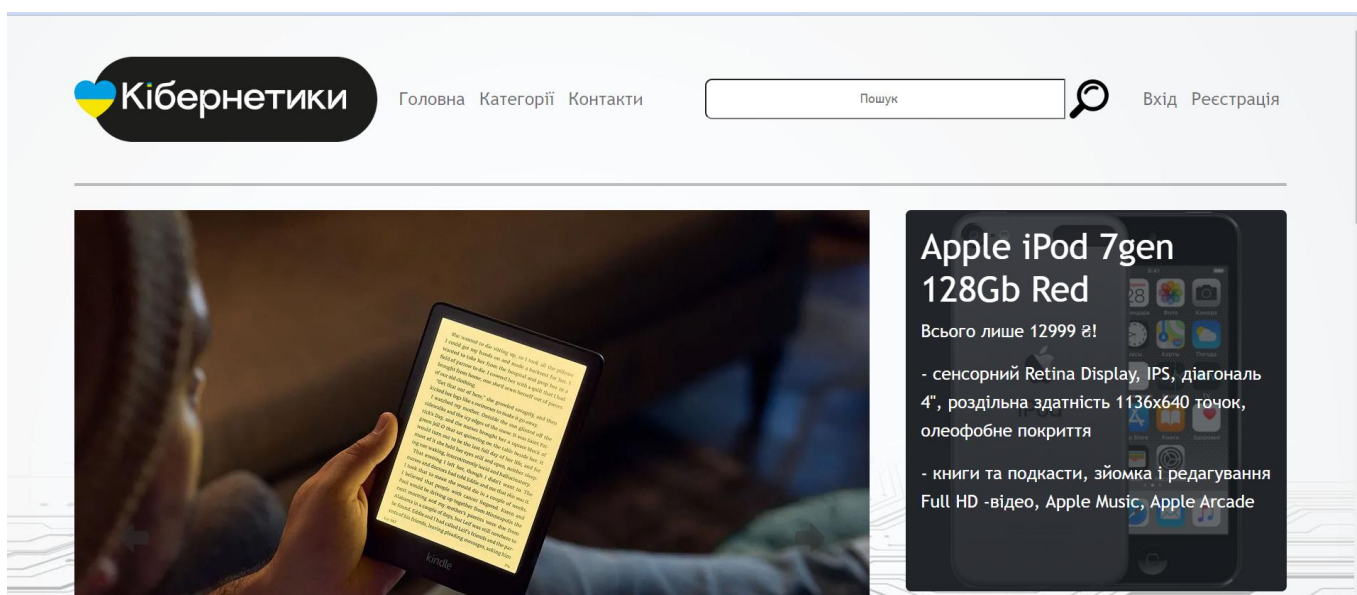


Рисунок 3.17 – Головна сторінка веб-додатку

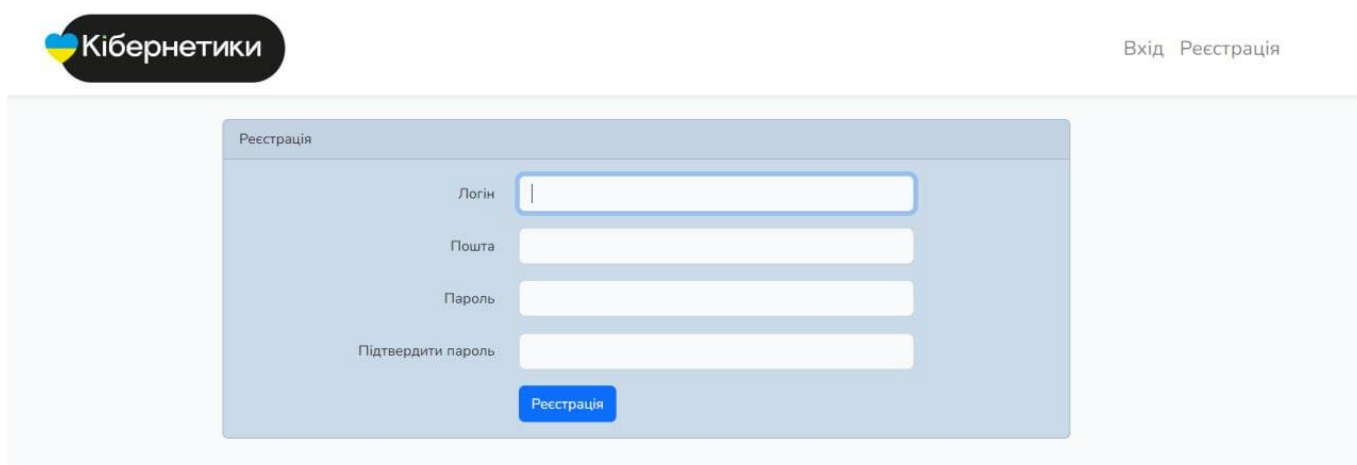


Рисунок 3.18 – Сторінка реєстрації

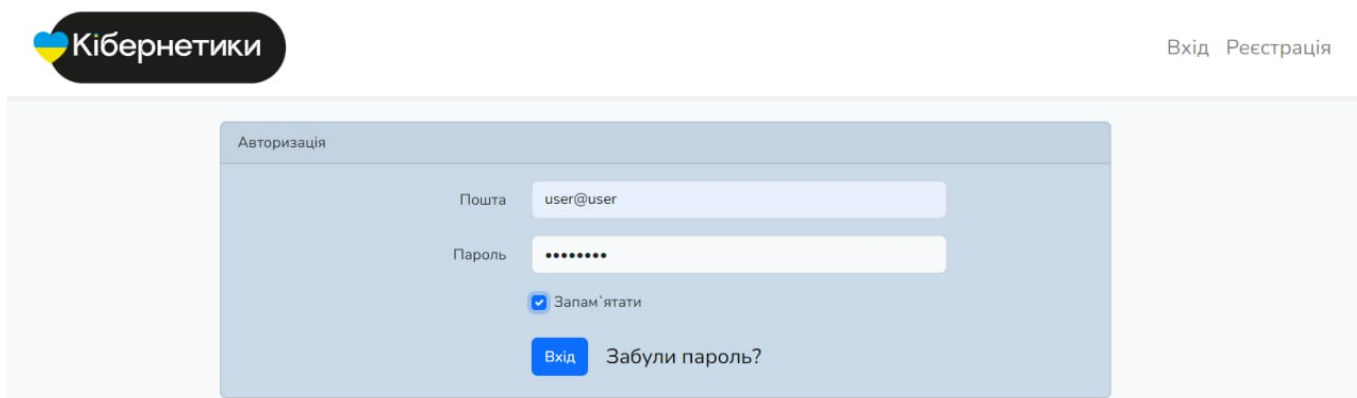


Рисунок 3.19 – Сторінка Авторизації

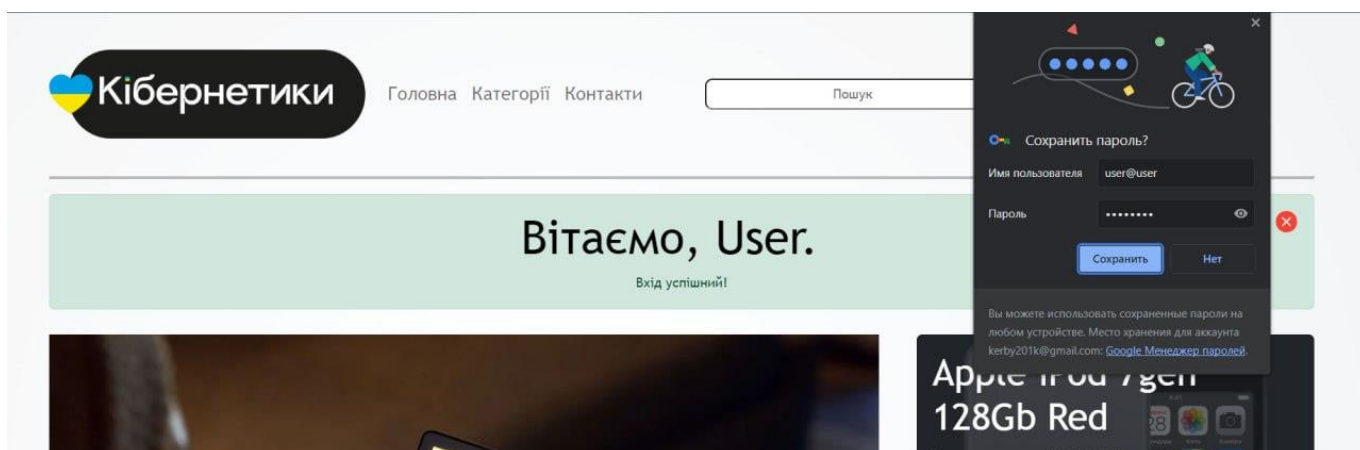


Рисунок 3.20 – Привітання користувача

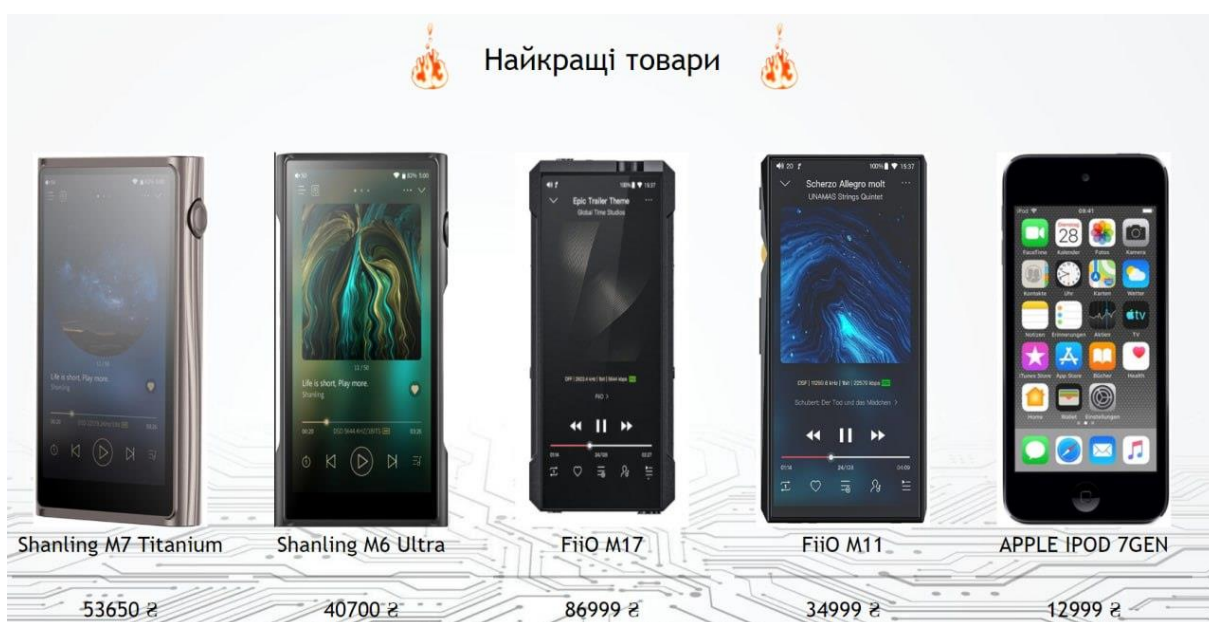


Рисунок 3.21 – Продовження головної сторінки веб-додатку

Нижня частина сторінки відображає інформацію про магазин, зокрема навігаційне меню, офіційних представників та назву магазину. Цей блок макету використовується на всіх сторінках додатку.



Рисунок 3.22 – Footer веб-додатку

Щоб переглянути каталог категорій доступних товарів магазину, користувачу необхідно перейти на сторінку «Категорії» (рис. 3.23).



Рисунок 3.23 – Категорії веб-додатку

Для прикладу відкриємо категорію «MP3-плеєри». Бачимо товари, які доступні клієнту для замовлення.



Рисунок 3.24 – MP3-плеєри веб-додатку

Якщо натиснути на кнопку «Сортування», відкриється перелік фільтрів, за якими можна підібрати бажаний товар.

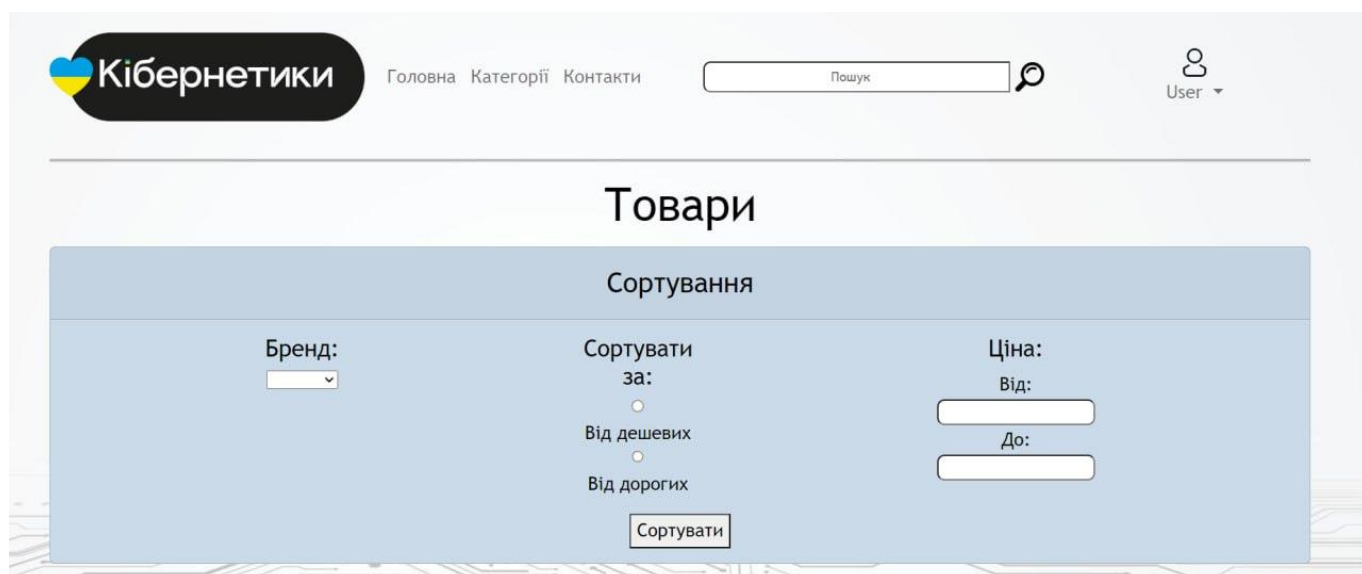


Рисунок 3.25 – Сортування пошуку

Якщо відкрити детальну інформація про товар, з'являться дві опції. Перша – це додавання товару до кошика, де потім можна буде підтвердити замовлення. Друга – додати товар до порівняння.



Рисунок 3.26 – Характеристики товару MP3-плеєр


Характеристики	
Дисплей	Сенсорний Retina Display, IPS, діагональ 4", роздільна здатність 1136x640, олеофобне покриття
Память	128 ГБ
Наушники	mini-jack 3,5 mm
Акумулятор	1043 mAh
Формати	HEVC, H.264, MPEG4, JPEG, BMP, GIF, TIFF, PSD, PNG, Dolby Digital (AC? 3), Dolby Digital Plus (E? AC? 3) i Audible (формати 2, 3, 4, Audible Enhanced Audio, AAX i AAX +)
Розмір	123.4x58.6x6.1
Вага	88 г

Рисунок 3.27 – Продовження характеристики товару MP3-плеєр

Контакти

Адреса:
вулиця Воскресенська, 13, Суми, Сумська область, 40000

Номер телефону:
[+380664295977](tel:+380664295977)



Ми на карті

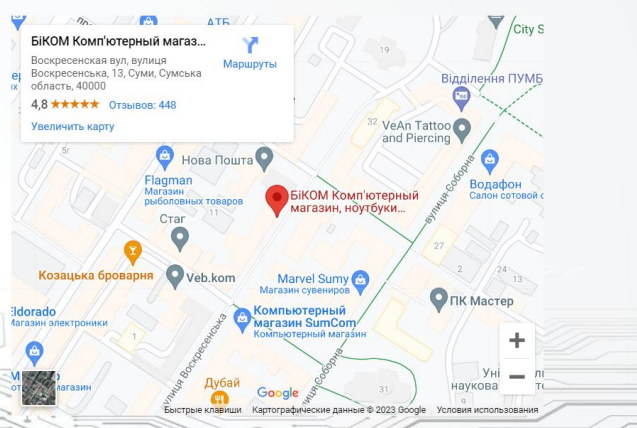


Рисунок 3.28 – Контакти веб-додатку

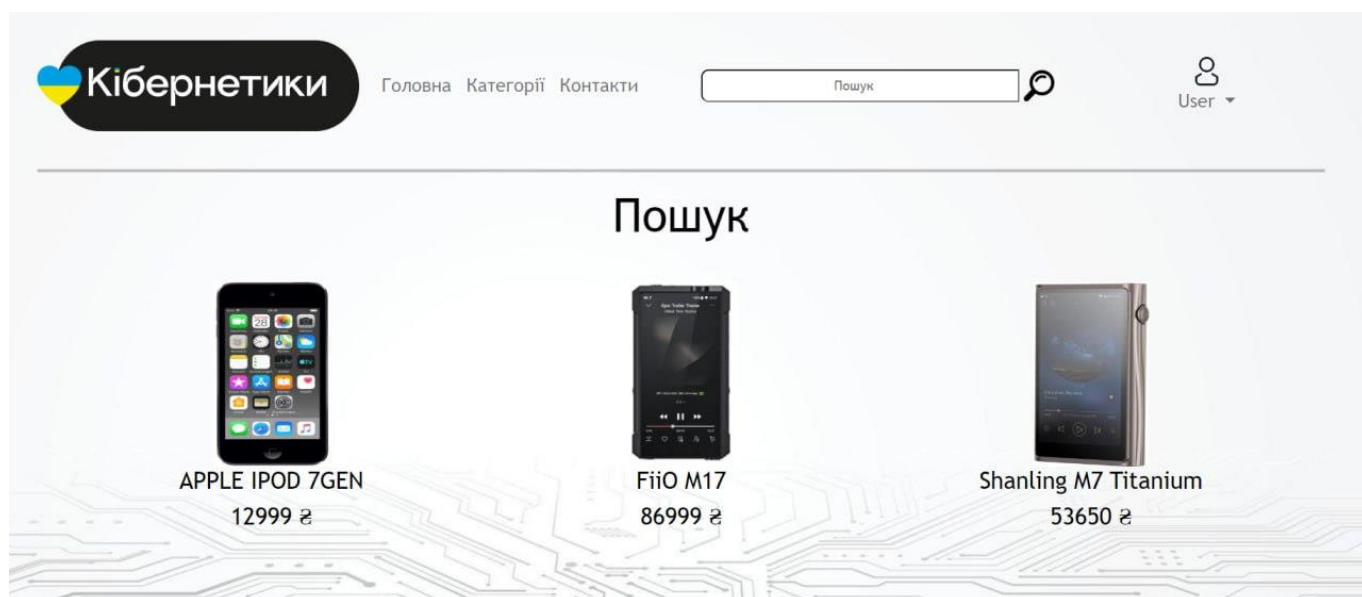


Рисунок 3.29 – Реалізація пошуку веб-додатку



Рисунок 3.30 – Історія останніх переглядів

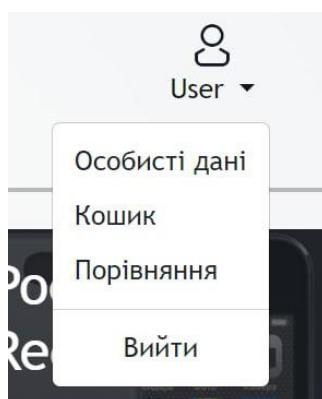


Рисунок 3.31 – Панель користувача



Рисунок 3.32 – Вигляд кошика веб-додатку

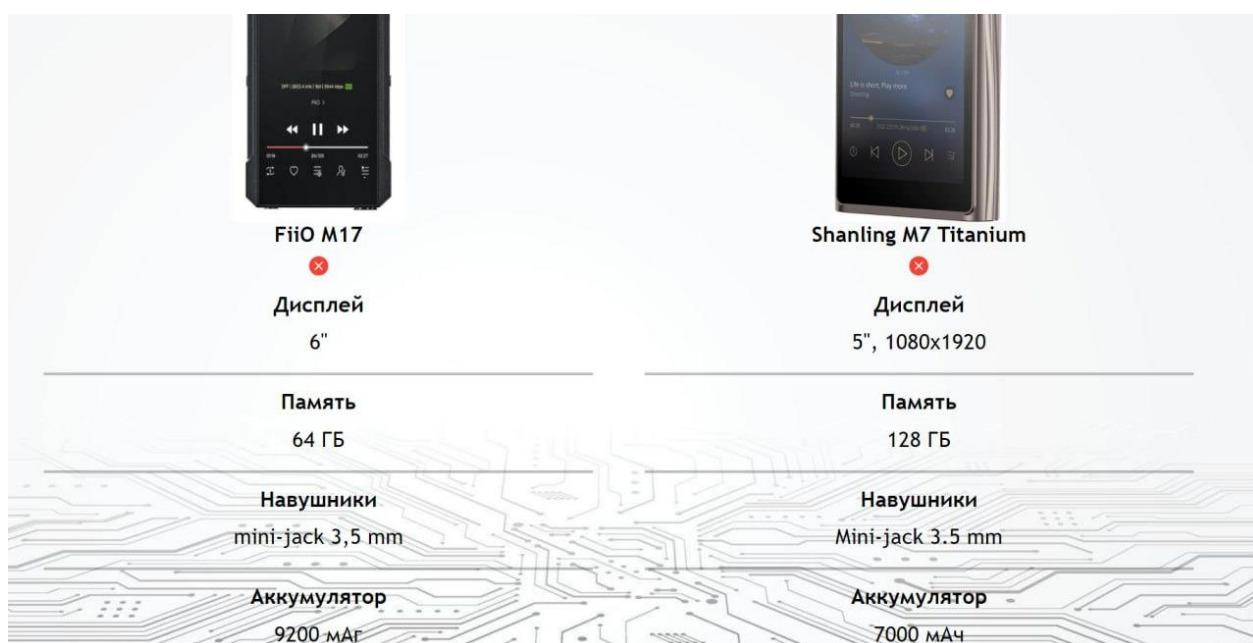


Рисунок 3.33 – Порівняння двох товарів

Особистий кабінет
Особисті дані не знайдено!
[Додати інформацію](#)

Рисунок 3.34 – Особистий кабінет до заповнення даними

Додати інформацію

Ім'я
Анастасія

Прізвище
Мирна

Дата народження
04.07.2007

Номер телефону
+38(050)841-76-81


Місто
Житомир

Пошта, якою користуєтесь:
 Нова пошта УкрПошта Джастин


Номер відділення
65

[Додати інформацію](#)

Рисунок 3.35 – Особистий кабінет під час заповнення даними



[Головна](#) [Категорії](#) [Контакти](#)


User ▾

Особистий кабінет

Анастасія Мирна
2007-07-04
+38(050)841-76-81
Житомир
Нова пошта №65

[Редагувати дані](#)

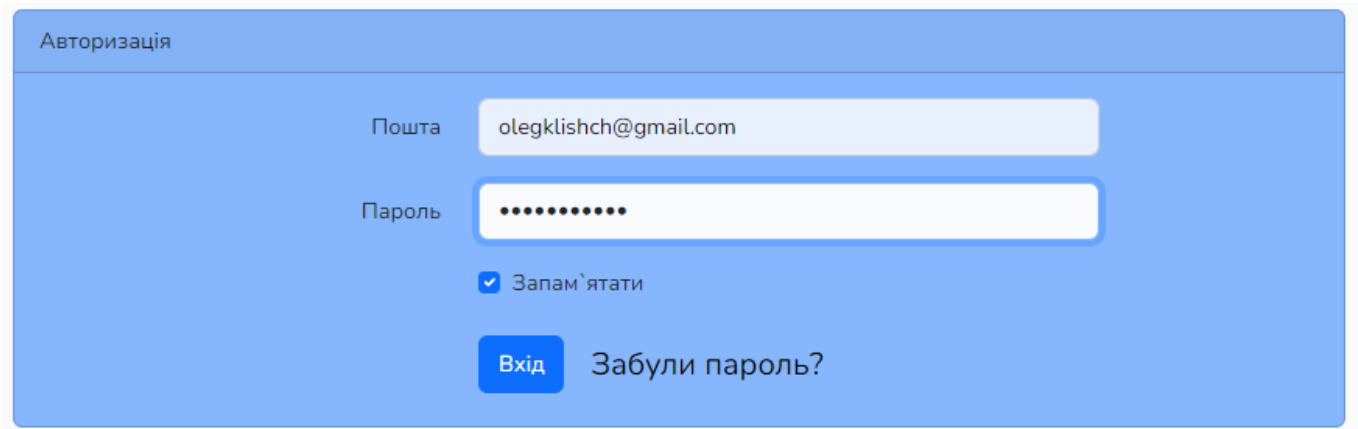
[Успішні замовлення](#)

Рисунок 3.36 – Особистий кабінет з заповненими даними

3.4 Адміністрування web-додатку

Всі сторінки та попередній функціонал залишаються такими ж, як і для клієнта, але додається додатковий функціонал, який доступний тільки адміністратору.

Вікно авторизації для адміністратора є спільним з клієнтом, тому ніяких змін немає. Пошта та пароль для адміністратора заздалегідь прописані в базі даних, тому ми не можемо зареєструватися як адміністратор.



The screenshot shows a blue-themed authorization window titled "Авторизація". It contains two input fields: "Пошта" (Email) with the value "olegkishch@gmail.com" and "Пароль" (Password) with masked characters. Below the password field is a checked checkbox labeled "Запам'ятати" (Remember me). At the bottom, there is a blue "Вхід" (Login) button and a link "Забули пароль?" (Forgot password?).

Рисунок 3.37 – Авторизація у ролі адміна

Після авторизації з'являється іконка та надпис Admin у правому верхньому куті, це демонструє, що ми увійшли в додаток у ролі адміністратора.

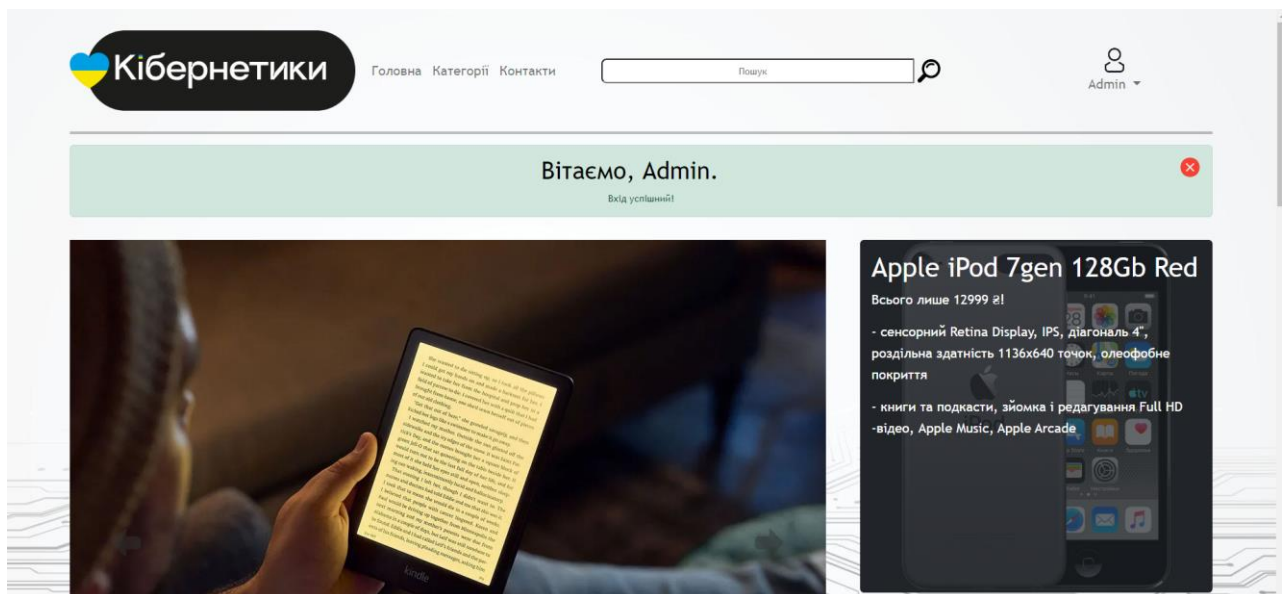


Рисунок 3.38 – Головна сторінка у ролі адміністратора

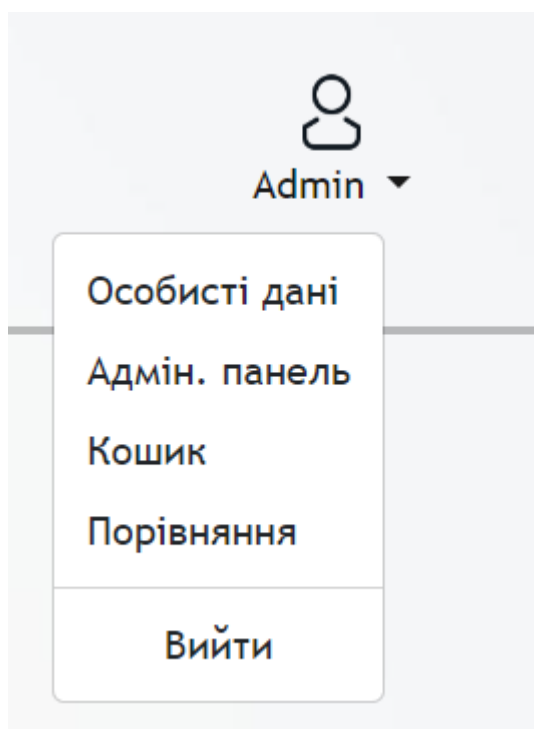


Рисунок 3.39 – Можливості адміністратора

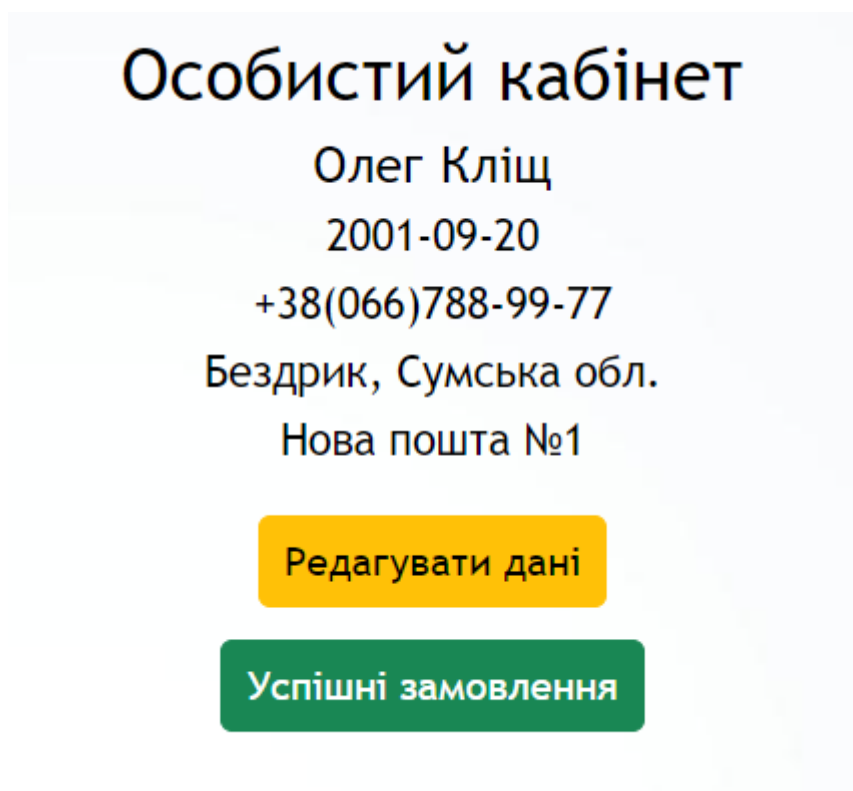


Рисунок 3.40 – Особистий кабінет адміністратора

Після оформлення замовлення клієнтом, його потрібно підтвердити адміністратору. Для цього він повинен натиснути на замовника та зателефонувати для уточнення деталей.

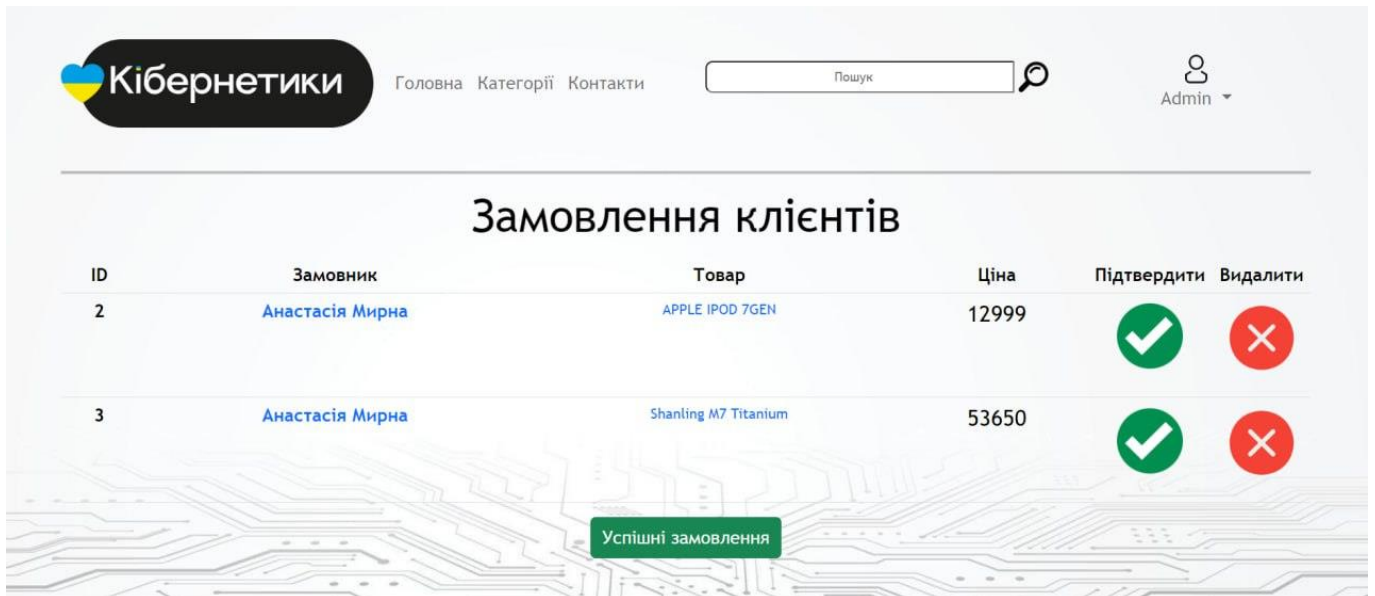


Рисунок 3.41 – Адміністративна панель

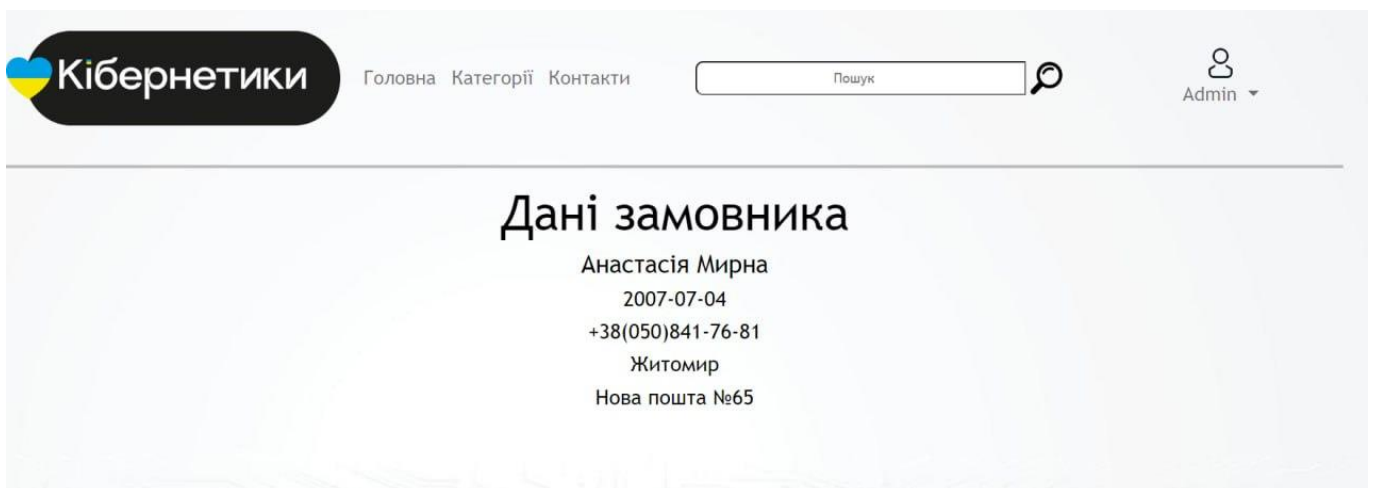
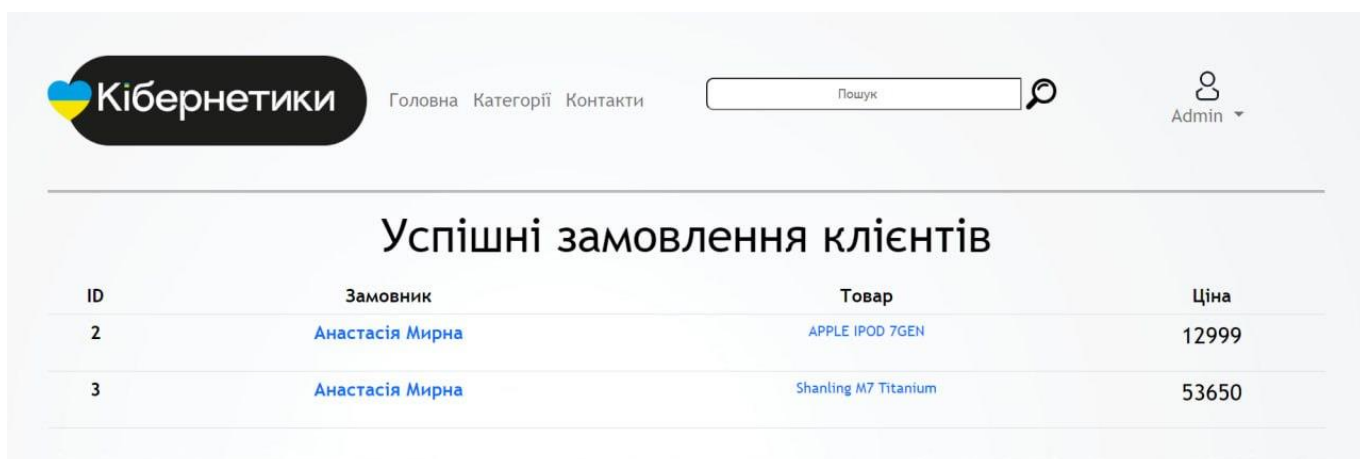


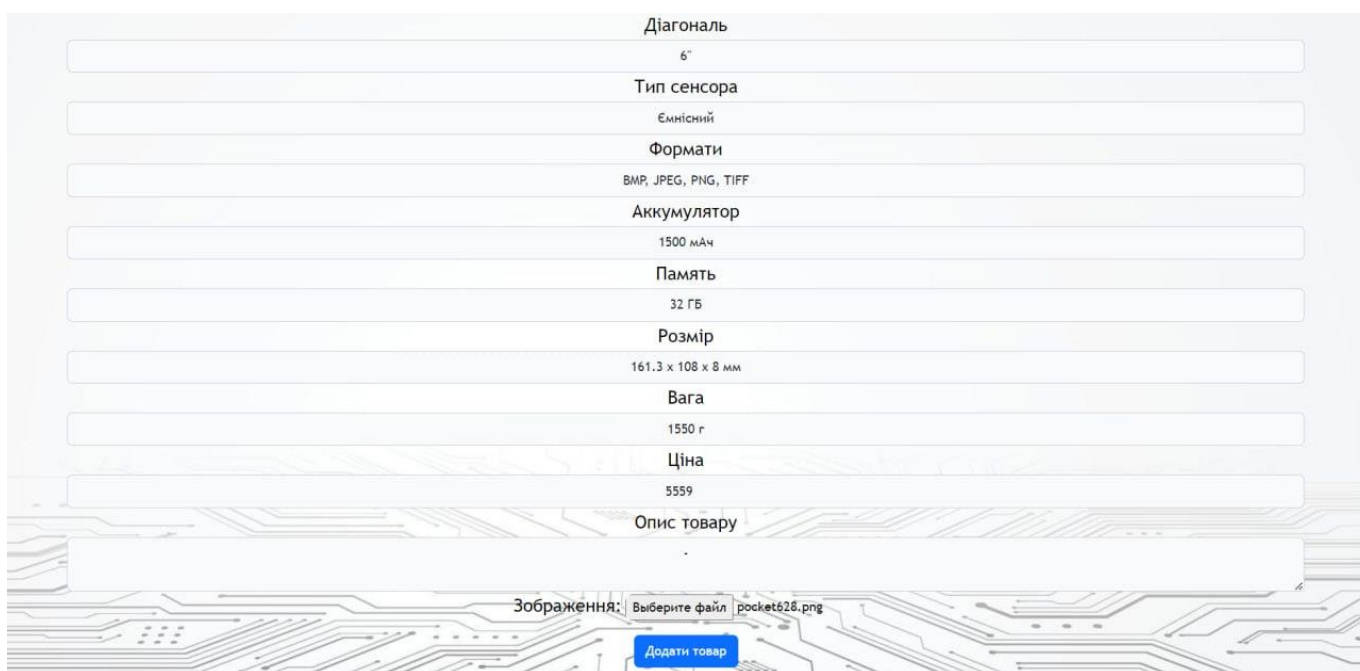
Рисунок 3.42 – Дані замовника



ID	Замовник	Товар	Ціна
2	Анастасія Мирна	APPLE IPOD 7GEN	12999
3	Анастасія Мирна	Shanling M7 Titanium	53650

Рисунок 3.43 – Підтверджені замовлення адміністратором

Також адміністратор може додавати, редагувати та видаляти інформацію про товари.



Діагональ
6"

Тип сенсора
Ємнісний

Формати
BMP, JPEG, PNG, TIFF

Акумулятор
1500 мАч

Память
32 ГБ

Розмір
161.3 x 108 x 8 мм

Вага
1550 г

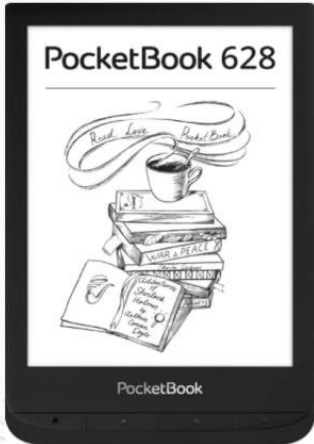
Ціна
5559

Опис товару

Зображення: rocket626.png

Рисунок 3.44 – Додавання товару

PocketBook 628



PocketBook 628

Ціна: 5559 ₴

[В кошик](#)

[В порівняння](#)

Характеристики

[Редагувати](#)

Рисунок 3.45 – Доданий товар

PocketBook 628



PocketBook 628

Ціна: 15999 ₴

[В кошик](#)

[В порівняння](#)

Характеристики

Рисунок 3.46 – Відредагований товар по ціні

The screenshot shows the 'Кібернетики' website header with a search bar and an 'Admin' dropdown. Below is a form for adding a category. The 'Назва категорії' (Category Name) field contains 'Смарт-годинники' (Smartwatches). The 'Зображення:' (Image) field has a file selection button and the text 'Виберіть файл smart.png'. A blue 'Додати категорію' (Add Category) button is at the bottom.

Рисунок 3.47 – Можливість додавати нову категорію



Рисунок 3.48 – Додана категорія



Рисунок 3.49 – Наявні товари MP3-плеєрів перед видаленням



Рисунок 3.50 – Товари після видалення

ВИСНОВКИ

Під час створення web-додатку продукту був проведений аналіз і порівняння існуючих web-додатків для продажу між собою. У результаті було знайдено достатньо переконливих аргументів, що підтверджують необхідність розробки web-додатку підтримки продажу портативної електроніки.

Аналіз програмних продуктів конкурентів продемонстрував їх сильні та слабкі сторони. Завдяки цьому були зроблені висновки, які дозволили уникнути повторення їх помилок та використати найбільш ефективні функції, що були реалізовані у аналогічних web-додатків. Розроблений програмний продукт відрізняється від конкурентів завдяки сучасному дизайну, взаємодії з користувачем та додатковому функціоналу, що робить його більш привабливим для користувачів, які зацікавлені в придбанні портативної електроніки. Отримані результати після порівняння аналогів дозволили стверджувати, що створення такого продукту на сьогоднішній день є актуальним та доцільним. Шляхом проведеного аналізу було сформовано головні вимоги до ефективної роботи системи, її зовнішнього вигляду, безпеки та надійності. Під час постановки задачі було визначено ряд завдань, які необхідні для реалізації поставленої мети, а також основні функціональні вимоги. На етапі моделювання та проектування системи була розроблена контекстна діаграма на основі стандарту IDEF0 для відображення загального функціоналу web-додатку. Згодом вона була розбита на частини, щоб проілюструвати складні бізнес-процеси та системні операції. Щоб продемонструвати потенційне використання web-додатку, була розроблена діаграма варіантів використання, яка відображає користувачів web-додатку та їхні взаємозв'язки. Крім того, було використано СУБД MySQL для створення бази даних для зберігання необхідної інформації з детальним описом її сутності та структури.

Розроблений web-додаток має на меті заощадити час покупців та надати максимально зручний та інтуїтивно зрозумілий інтерфейс програми для перегляду та замовлення портативної електроніки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Principles-of-Marketing – Kotler-Armstrong [Електронний ресурс] – Доступ до ресурсу: <https://nit-edu.org/wp-content/uploads/2021/09/Principles-of-Marketing-Kotler-Armstrong.pdf> (Дата звернення: 01.05.23)
2. Design and Development of E-Commerce Web Application for Cooperative Store: [Електронний ресурс] – Доступ до ресурсу: https://www.researchgate.net/publication/358927961_Design_and_Development_of_E-Commerce_Web_Application_for_Cooperative_Store (Дата звернення: 07.05.23)
3. Web-додаток «Тинан» [Електронний ресурс] – Доступ до ресурсу: <https://titan.ua/> (Дата звернення: 02.05.23)
4. Web-додаток «MobyStore» [Електронний ресурс] – Доступ до ресурсу: <https://mobystore.com.ua/uk/> (Дата звернення: 03.05.23)
5. Web-додаток «Compservice» [Електронний ресурс] – Доступ до ресурсу: <https://compservice.in.ua/> (Дата звернення: 04.05.23)
6. Структура web-додатку [Електронний ресурс] – Доступ до ресурсу: https://stud.com.ua/97612/informatika/struktura_dodatku (Дата звернення: 04.05.23)
7. WBS структура проекту [Електронний ресурс] – Доступ до ресурсу: <https://studfile.net/preview/5680685/page:12/> (Дата звернення: 04.05.23)
8. Application of it-based web on online store [Електронний ресурс] – Доступ до ресурсу: https://www.researchgate.net/publication/327899146_Application_of_IT-Based_Web_on_Online_store (Дата звернення: 07.05.23)
9. Технології створення Web-ресурсів для реалізації Internet торгівлі: [Електронний ресурс] – Доступ до ресурсу: <https://www.sworld.com.ua/konferbe12/24.pdf> (Дата звернення: 07.05.23)
10. Як створити веб-додаток: типи, переваги, принцип роботи [Електронний ресурс] – Доступ до ресурсу: <https://wezom.com.ua/ua/blog/kak-sozdat-veb-prilozhenie> (Дата звернення: 08.05.23)

11. Навіщо бізнесу потрібен сайт ? [Електронний ресурс] – Доступ до ресурсу: <https://ag.marketing/blog/navishcho-biznesu-potriben-sayt/> (Дата звернення: 08.05.23)
12. Складові системи планування робіт [Електронний ресурс] – Доступ до ресурсу: <https://buklib.net/books/23851/> (Дата звернення: 10.05.23)
13. Ризики проекту [Електронний ресурс] – Доступ до ресурсу: <https://skillsetter.io/blog/risk-management-ua> (Дата звернення: 10.05.23)
14. Основні переваги IDEF0 [Електронний ресурс] – Доступ до ресурсу: https://er.chdtu.edu.ua/bitstream/ChSTU/2786/1/%D0%A0%D0%91%D0%9F_%D0%9F%D0%BE%D1%81_%D0%94%D0%B0%D0%BD%D1%87%D0%B5%D0%BD%D0%BA%D0%BE.pdf#page=93&zoom=100,109,677 (Дата звернення: 15.05.23)
15. Основні переваги субд MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5607354/page:3/> (дата звернення: 20.05.2023)
16. Розділяй та володарюй: що таке патерни MVC і MVP, та як їх використовувати [Електронний ресурс] – Режим доступу до ресурсу: <https://highload.today/uk/blogs/shho-take-mvc-ta-mvp-patterni/> (дата звернення: 21.05.2023).

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Web-додаток підтримки продажу портативної електроніки»

ПОГОДЖЕНО:

Доцент кафедри інформаційних технологій

_____ Парфененко Ю.В.

Студент групи ІТ-91

_____ Кліщ О.Ю.

Суми 2023
1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ WEB-ДОДАТКУ

1.1 Призначення web-додатку

Web-додаток призначений для перегляду портативної електроніки та її замовлення.

1.2 Мета створення web-додатку

Мета проекту – інформаційна підтримка процесу продажу за рахунок створення web-додатку для скорочення часу вибору продукту.

1.3 Цільова аудиторія

Цільовою аудиторією даного проекту є клієнти web-додатку, які зацікавлені у придбанні товарів за найвигіднішою ціною та економії свого часу на відвідування спеціалізованих магазинів з мобільною технікою. Також до цільової аудиторії можна віднести власника магазину, який зацікавлений у створенні web-додатку для збільшення продажів.

2 ВИМОГИ ДО WEB-ДОДАТКУ

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури й функціонування web-додатку

Web-додаток повинен бути реалізований за допомогою web-інструментів та забезпечувати визначений набір функціональних можливостей.

Кінцевий продукт даного проекту має бути представлений web-сайтом, який містить всю необхідну інформацію стосовно товару певної категорії та інтерфейсу в цілому.

2.1.2 Вимоги до персоналу

Персонал закладу не повинен мати особливих технічних навичок для роботи з web-каталогом і його підтримкою. Єдиною вимогою є наявність навичок користування персональним комп'ютером та web-браузером, а також мати мінімальні знання з керування панеллю адміністрування: додавати, редагувати та видаляти товари.

2.1.3 Вимоги до збереження інформації

Уся інформація надана на web-сайті повинна зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL на сервері.

2.1.4 Вимоги до розмежування доступу

Розроблюваний web-додаток має бути загальнодоступним у мережі інтернет. Права доступу до інформації розмежовані за групами користувачів: адміністратор, відвідувач та клієнт. Адміністратор має необмежений доступ до даних з правами перегляду, додавання, редагування та видалення. Доступ до адміністративної панелі надається за спеціальним логіном та паролем.

Відвідувач web-сайту може переглядати інформацію на web-сторінках, робити пошук товару, порівнювати їх та робити замовлення. Адміністратор, в свою чергу може управляти асортиментом (додавання, оновлення та видалення). Також передбачена сторінка для підтвердження замовлення.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

До структури web-додатку входять усі його web-сторінки, які є загальнодоступними, та адміністративна панель для редагування змісту каталогу.

Перелік сторінок web-додатку наступний:

- «Головна» сторінка містить навігаційне меню, яке закріплене на кожній сторінці web-додатку, карусель із зображеннями акційних пропозицій та оголошень, найпопулярніші товари, адреса, та адміністративну панель навігації;
- призначенням сторінки «Контакти» є демонстрація способів зв'язку з адміністрацією та прикріплена Google-карта з міткою місця розташування закладу;
- «Категорії» надають можливість користувачу обрати бажаний товар за доступною категорією.

2.2.2 Навігаційне меню

Для зручної навігації повинно бути створене меню, що забезпечить швидке переміщення користувача по всім доступним сторінкам web-сайту. Меню має бути закріплене і розташовуватися зверху (у шапці) на кожній сторінці.

2.2.3 Управління контентом

Управління контентом web-каталогу має здійснюватися за допомогою адміністративної панелі. Усе інформаційне наповнення має міститися у базі даних. Графічні матеріали та інформацію для наповнення буде обрано самостійно.

2.2.4 Дизайн web-додатку

Дизайн web-каталогу має бути виконаний у сучасному стилі. Оскільки розробляється каталог портативної електроніки, то акцент був поставлений на візуалізації інтерфейсу. Кольорова схема буде використана в світлих тонах, шрифт – Trebuchet, також буде розроблена адаптація для мобільних пристроїв.

Види і розміри шрифтів повинні бути комфортними для перегляду. Інформаційні блоки, графічні матеріали та інші елементи web-сторінок повинні мати зручне і логічне розташування. Шаблон майбутнього програмного продукту зображено на рисунку А.1



Рисунок А.1 – Схема головної сторінки

2.2.5 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.2.

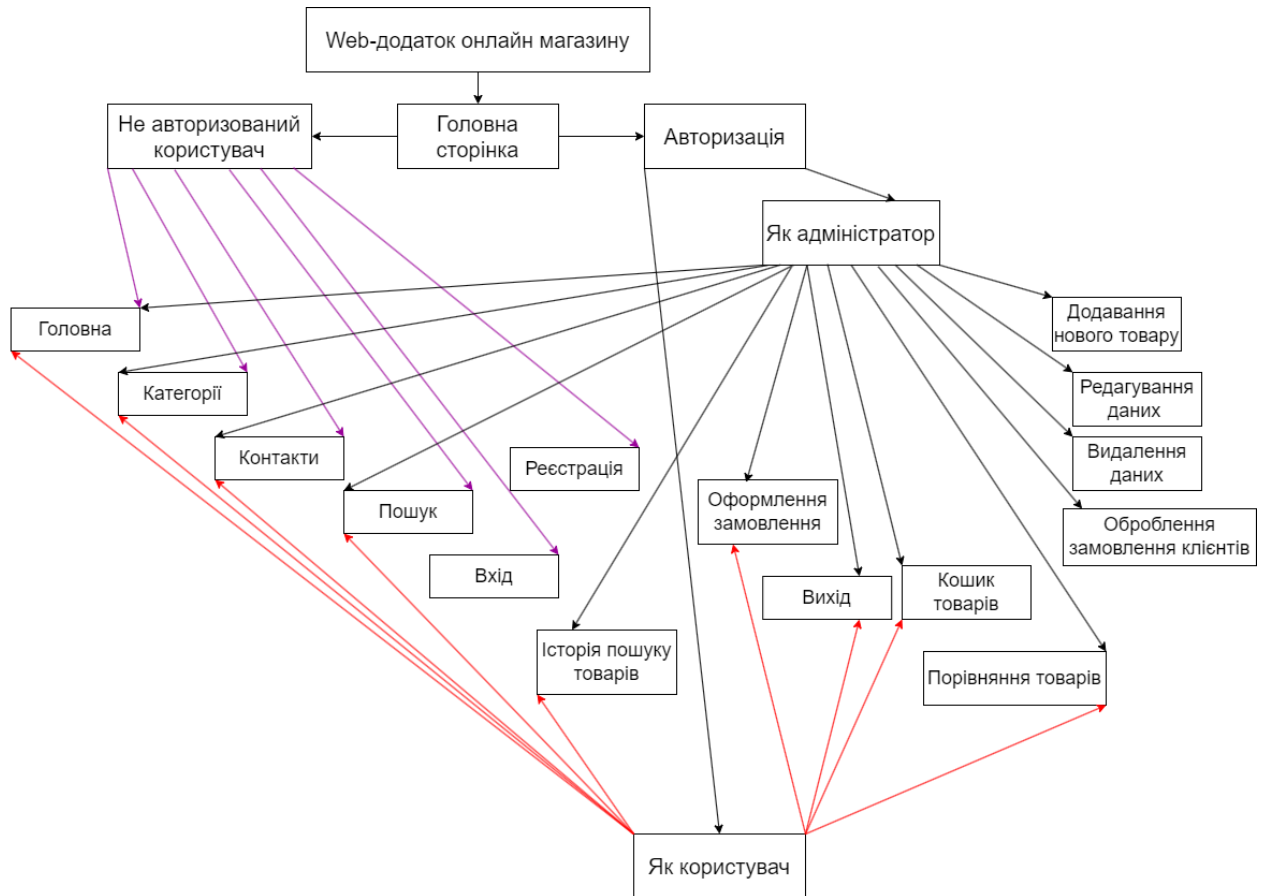


Рисунок А.2 – Система навігації

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Перегляд продукції каталогу	Клієнт, Відвідувач, Адміністратор
UN-02	Замовлення продукції	Клієнт, Адміністратор

Таблиця А.2 – Продовження таблиці А.1

UN-03	Перегляд опису товару	Клієнт, Відвідувач, Адміністратор
UN-04	Перегляд акцій та оголошень	Клієнт, Відвідувач, Адміністратор
UN-05	Перегляд інформації про web-каталог	Клієнт, Відвідувач, Адміністратор
UN-06	Перегляд контактів	Клієнт, Відвідувач, Адміністратор
UN-07	Редагування даних	Адміністратор
UN-08	Додавання інформації	Адміністратор
UN-9	Видалення даних	Адміністратор
UN-10	Опрацювання замовлення клієнтів	Адміністратор
UN-11	Порівняння товарів	Клієнт, Адміністратор
UN-12	Пошук товарів	Клієнт, Відвідувач, Адміністратор
UN-13	Історія пошуку товарів	Клієнт, Адміністратор
UN-13	Додавання товару до кошика	Клієнт, Адміністратор

2.3.2 Функціональні вимоги

Проаналізувавши потреби користувачів та персоналу закладу було визначено наступні вимоги:

- наявність реєстрації та авторизації клієнтів;
- пошук інформації про товари на web-сайті;
- можливість оформлення онлайн-замовлень за допомогою форми;
- наявність повної інформації про бажаний продукт;
- наявність панелі адміністратора для додавання, редагування, видалення інформації з web-каталогу, а також опрацювання замовлень клієнтів;
- сортування продуктів за фільтрами;
- порівняння товарів однакового типу;
- історія перегляду товарів;
- додавання товарів до кошику
- перегляд контактів

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.3.-А.4.

Таблиця А.3 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність модуля авторизації/реєстрації	М	Надає можливість клієнту авторизуватися або зареєструватися
SR-02	База даних із замовленнями	М	Надає можливість користувачу додавати товари до кошика
SR-03	База даних з товарами	М	Відповідає за заповнення web-додатку контентом
SR-04	База даних з цінами на товари	М	Надає можливість подальшого відображення цін на всі товари
SR-05	Зберігання даних про акції в окремій таблиці бази даних	С	Відповідає за відображення акцій та пропозицій на товари портативної електроніки
SR-06	Наявність модуля зворотного зв'язку	С	Надає можливість клієнту зв'язатися з адміністратором інтернет-магазину
SR-07	Панель адміністратора	М	Відповідає за подальше наповнення та редагування контенту адміністратором, а також обробленні замовлень клієнтів
SR-08	Наявність модуля пошуку товарів за фільтрами	М	Відповідає за пошук товарів в інтернет магазині за доступними фільтрами: ціною, від дешевих до дорогих, за популярністю

Таблиця А.4 – Продовженн таблиці А.3

SR-09	Наявність модулю порівняння товарів	S	Відповідає за порівняння товарів за їх характеристиками
SR-10	Наявність модулю збереження історії пошуку	C	Відповідає за зберігання історії пошуку товару
SR-11	Наявність модулю оформлення замовлень	M	Відповідає за оформлення замовлення користувачем

Умовні позначення в таблиці А.3-А.4:

Must have (M) – вимоги, які повинні бути реалізовані в системі;

Should have (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Could have (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбувається з використанням:

- PHP 8.0
- Laravel
- MySQL 8.0
- OpenServer
- HTML
- CSS
- Bootstrap
- JS
- PhpStor

2.4.2 Вимоги до лінгвістичного забезпечення

Весь текст у web-каталогу має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Для забезпечення стабільної роботи web-додатку web-браузер має бути Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище. Операційна система Windows XP і вище.

3 СКЛАД І ЗМІСТ РОБІТ ЗІ СТВОРЕННЯ WEB-ДОДАТКУ ПОРТАТИВНОЇ ЕЛЕКТРОНІКИ

Детальний опис етапів створення web-каталогу наведено в таблиці А.5.

Таблиця А.5 – Етапи створення web- каталогу

№	Склад і зміст робіт	Строк розробки
1	Розробка шаблону web-каталогу	2 дні
2	Задання верстки сторінок web-каталогу	14 днів
3	Розробка модулю оформлення замовлень	7 днів
4	Розробка модулю авторизації/реєстрації	7 днів
5	Розробка модулю порівняння товарів	3 дні
6	Розробка модулю зворотнього зв'язку	4 дні
7	Розробка модулю збереження історії пошуку товару	2 дні
8	Розробка панелі адміністратора	5 днів
9	Розробка бази даних	8 днів
10	Наповнення контентом web-каталогу	6 днів
11	Beta-тестування	5 днів
12	Alpha-тестування	3 днів
13	Розміщення на хостингу	1 день
14	Перевірка працездатності	2 дні
15	Написання супровідної документації	5 днів
16	Реліз web-каталогу	1 день
	Загальна тривалість робіт	75 дні

4 ВИМОГИ ДО СКЛАДУ Й ЗМІСТУ РОБІТ ІЗ ВЕДЕННЯ WEB-ДОДАТКУ В ЕКСПЛУАТАЦІЮ

Для того, щоб web-додатком могли користуватися клієнти та відвідувачі необхідно розмістити його у мережі Інтернет, тому необхідно придбати доменне ім'я та місце на хостингу. На хостинг переноситься web-додаток і наповнення бази даних з подальшою їх доробкою. Для коректного переносу web-додатку на хостинг необхідно, щоб параметри хостинга відповідали вимогам, зазначеним у ТЗ.

ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ

Портативна електроніка все більше набуває популярності. Майже всі прагнуть купити собі найкращий товар за свою ціну. Багато людей не знають як правильно обирати портативну електроніку або не знають сервіси де це можна зробити.

Web-додатки мають багато можливостей і становлять ефективне рішення для різних завдань. Інтернет-каталоги можуть зберігати значну кількість інформації за якою не потрібно звертатися у магазин. Користувач може переглянути детальну інформацію про товар з будь-якого місця та обрати саме той, що задовольняє його потребам.

Web-додаток має бути розроблений у встановлений графіком час та з використанням наявних ресурсів. Цей продукт буде актуальним тільки на території України, оскільки доставка портативної електроніки в інші країни дорога та проблематична.

Результат деталізації даного проекту методом SMART розміщено у таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Specific (конкретна)	Створення web-додатку підтримки продажу портативної електроніки «PortableElectronics»
Measurable (вимірювана)	Забрати 2-5% частки ринку збуту портативної електроніки, стабільний прибуток 10000\$ на місяць
Achievable (досяжна, узгоджена)	Мета досяжна, тому що є технічне завдання, воно завержене, є людські ресурси, постачальники, фінансові ресурси, великий попит на портативну електроніку та чіткий план виконання проекту.
Relevant (актуальна)	Для стабільного прибутку, можливості подальшого розширення магазину та підвищення ефективності продажу портативної електроніки.

Таблиця Б.2 – Продовження таблиці Б.1

Time-framed (обмежена в часі)	Є конкретний термін – середини червня
-------------------------------------	---------------------------------------

WBS – це графічне подання згрупованих елементів проекту у вигляді пакета робіт, які ієрархічно пов'язані з продуктом проекту [7]. Верхній рівень WBS відображає продукт проекту, який повинен відповідати меті проекту. На другому рівні відображаються дії або основні етапи, необхідні для досягнення цілей проекту.

Розбивка цих етапів продовжується до елементарних робіт, які мають один чіткий результат – окреме виконання певної дії. На всі елементарні роботи призначений один відповідальний. WBS-структура представляє склад роботи та обчислює тривалість виконання кожної дії.

Кількість рівнів WBS може відрізнитися залежно від проекту. Для верхнього рівня зазвичай використовують іменники для позначення елементів, а на нижньому рівні - віддієслівні іменники. Елементарні роботи можуть бути ще більш деталізовані, але це вимагає спеціалізованих знань.

Важливо розуміти, що елементарні роботи нижнього рівня WBS не є деталізованими з погляду управління проектами, але насправді вони можуть бути ще більш дрібними і можуть бути розділені на окремі операції. Проте для цього потрібні спеціалізовані технологічні знання, які можуть бути отримані на місці виконання роботи.

Діаграму WBS представлено на рисунку Б.1

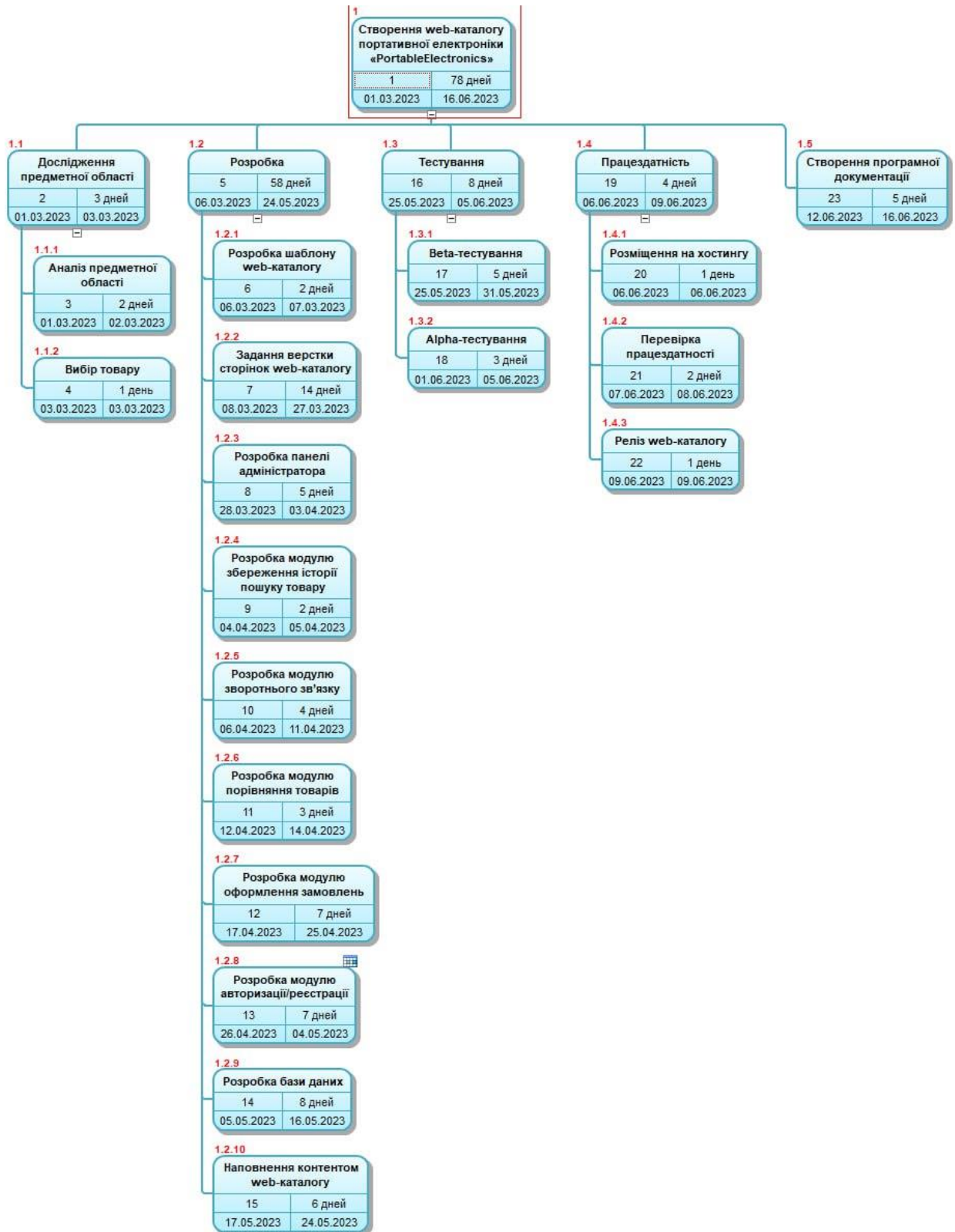


Рисунок Б.1 – Планування змісту структури роботи за допомогою діаграми WBS

Після розбиття процесів на складові етапи, наступним кроком є розробка графічної структури, що відображає учасників або відповідальних осіб, які

приймають участь у проекті, так звану організаційну структуру виконавців (OBS). Відповідальні особи в цій структурі - співробітники, які забезпечують організацію та виконання елементарних робіт, які вказані у WBS. Кожна з цих елементарних робіт може бути розглянута як окремий проект. На рисунку Б.2 показана організаційна структура для планування проекту.

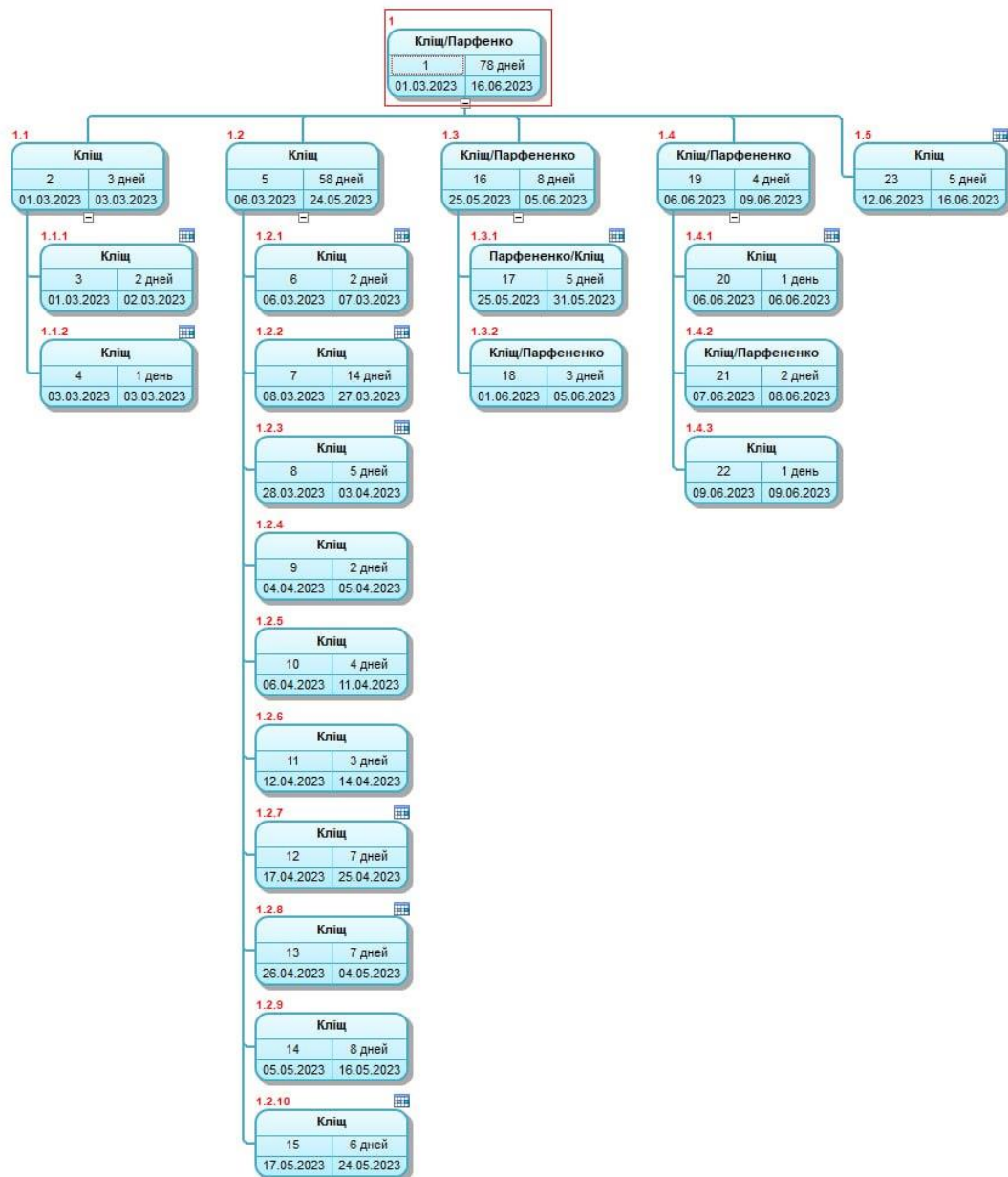


Рисунок Б.2 – OBS-структура проекту

Один з ключових етапів планування проекту - побудова календарного графіку (діаграми Ганта), який показує розклад виконання робіт з урахуванням реальних дат. Це дозволяє отримати точне уявлення про терміни виконання процесів з врахуванням

обмежень в ресурсах, вихідних днів та свят. Календарний графік проекту можна побачити на рисунках Б.3-Б.10.

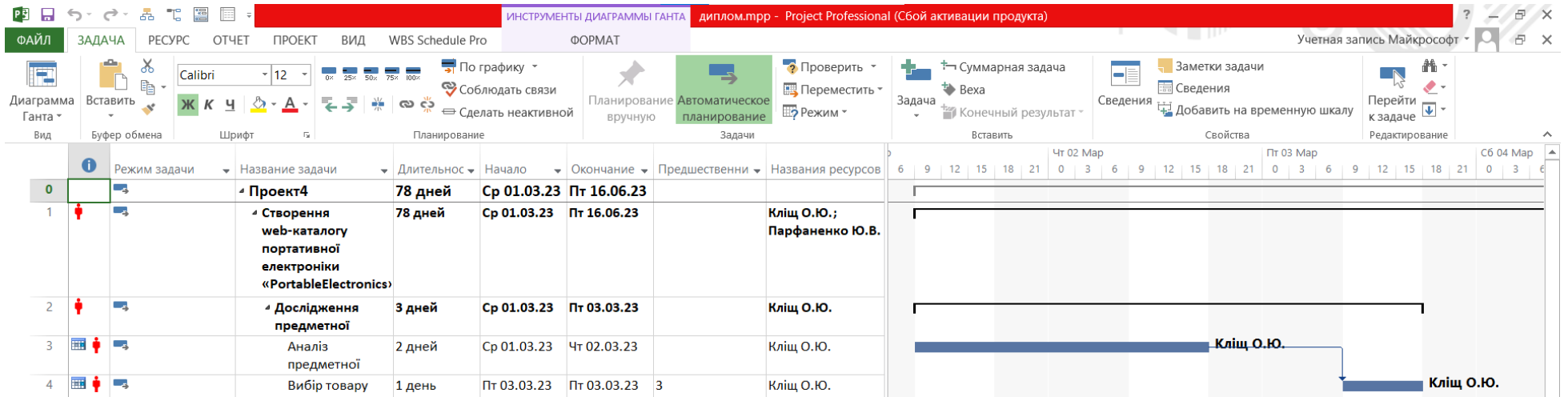


Рисунок Б.3 – Загальний вигляд діаграми всього проекту, частина 1

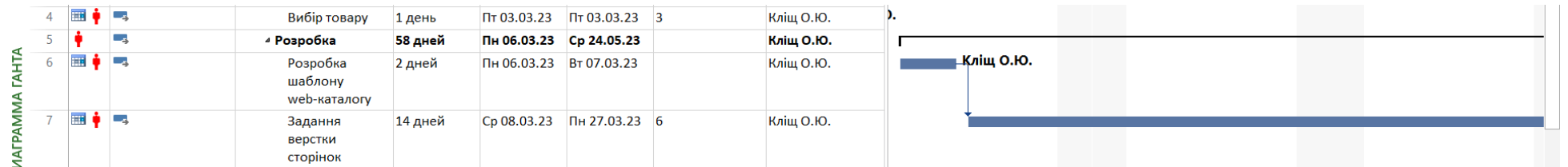


Рисунок Б.4 – Загальний вигляд діаграми всього проекту, частина 2



Рисунок Б.5 – Загальний вигляд діаграми всього проекту, частина 3

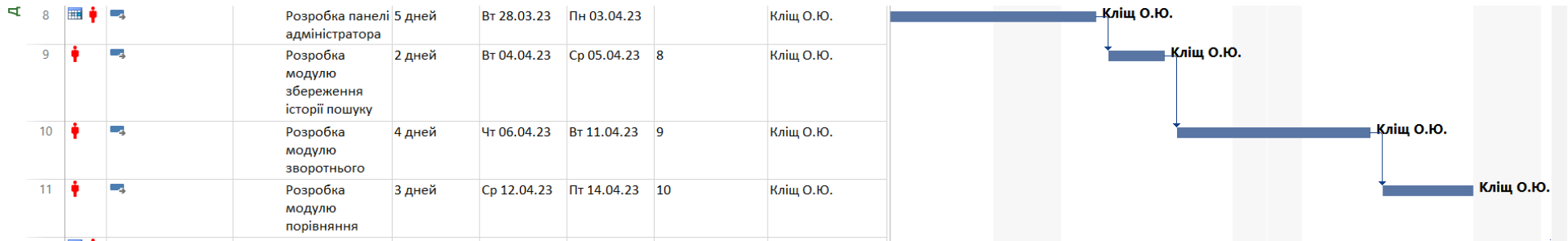


Рисунок Б.6 – Загальний вигляд діаграми всього проекту, частина 4

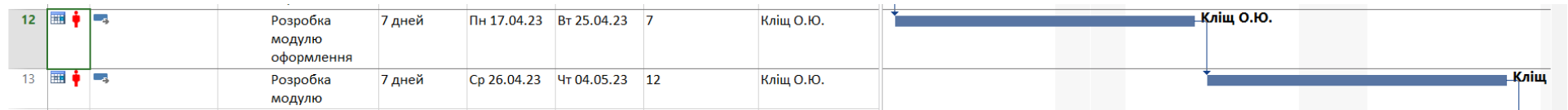


Рисунок Б.7 – Загальний вигляд діаграми всього проекту, частина 5



Рисунок Б.8 – Загальний вигляд діаграми всього проекту, частина 6

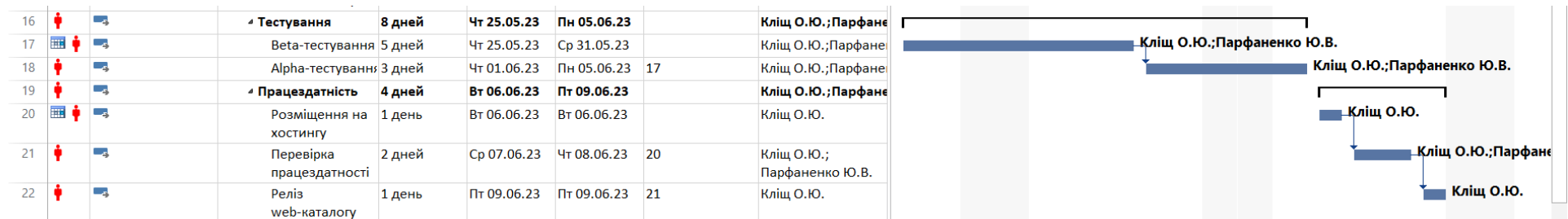


Рисунок Б.9 – Загальний вигляд діаграми всього проекту, частина 7

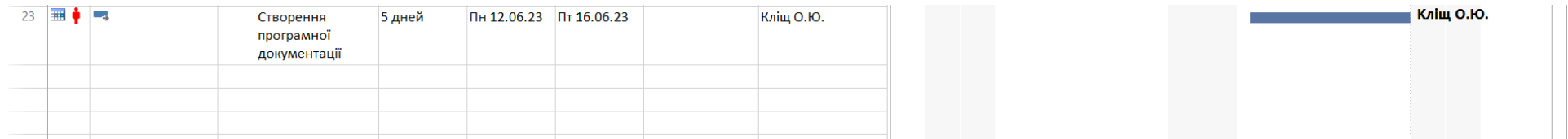


Рисунок Б.10 – Загальний вигляд діаграми всього проекту, части

Під час проведення якісної оцінки ризиків необхідно виділити ті, які потребують негайної уваги. Реакція на кожний ризик буде залежати від його значущості. Після цього, наступним кроком є кількісна оцінка ризиків. Якісна та кількісна оцінки можуть виконуватися окремо або одночасно, залежно від рівня забезпечення проекту. Шкала класифікації ризиків за їх впливом на проект та ймовірністю виникнення подана в таблиці Б.3.

Таблиця Б.3 – Шкала класифікації ризиків за ймовірністю виникнення та величиною впливу на проект

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Щоб зменшити негативний вплив ризиків на проект, необхідно здійснити планування заходів по їхньому усуненню. Це включає визначення ефективності заходів та оцінку наслідків, які можуть впливати на проект. Оцінка проводиться згідно з показниками, які наведені в таблиці Б.3. У результаті планування була створена матриця ймовірності виникнення ризиків та їхнього впливу, яку можна побачити в таблицях Б.4.-Б.5. Зеленим кольором позначають прийнятні ризики, жовтим - виправдані, а червоним - неприпустимі.

Таблиця Б.4 – Матриця ймовірності та впливу

Ймовірність ризику(Й)	Вплив загрози(ризику)				
	Дуже малий	Малий	Середній	Великий	Дуже великий
	0,05	0,1	0,2	0,4	0,8
0,9			R1(0,18)		

0,9			R1(0,18)		
-----	--	--	----------	--	--

0,7			R3(0,16)		
0,5					R7(0,4)
0,3		R8(0,02)		R2(0,12) R6(0,15) R9(0,08)	
0,1				R4(0,04), R5(0,04)	R10(0,08)

Таблиця Б.5 – Продовження таблиці Б.4

Таблиця Б.5 містить класифікацію ризиків за рівнем, відповідно до отриманого значення індексу. У таблицях Б.6.-Б.7. надано опис ризиків та відповідні стратегії реагування на кожен з них.

Таблиця Б.6 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$0,005 \leq R \leq 0,05$	4,8,5
2	Виправдані	$0,05 < R \leq 0,14$	2,6,9,3
3	Недопустимі	$0,14 < R \leq 0,72$	1,7,10

Таблиця Б.7 – Стратегії реагування на ризики

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив	Ранг ризику	План А	Тип стратегії	План Б
RS_1	Відкритий	Частина роботи яка викликає складність виконання	Малий	Великий	0,18	Отримати знання по ходу виконання	Ухилення	Найняти працівника
RS_2	Відкритий	Технічні збої(відсутність електроенергії, доступу до Інтернету)	Малий	Середній	0,12	Зателефонувати до служби	Ухилення	Використати техніку в іншому місці
RS_3	Відкритий	Зміна ТЗ	Середній	Середній	0,16	Договоритись про зміни	Зменшення	Внести зміни в ТЗ

Таблиця Б.8 – Продовження таблиці Б.7

RS_4	Відкритий	Погане самопочуття виконавця	Малий	Середній	0,04	Вилікуватись	Попередження	Наняти людей на період хвороби
RS_5	Відкритий	Недостатня конфігурація техніки	Малий	Середній	0,04	Взяти в оренду це обладнання	Ухилення	Докупити необхідне обладнання
RS_6	Відкритий	Знаходження помилок під час тестування	Середній	Середній	0,04	Зробити відповідні зміни, внести їх в ТЗ	Ухилення	Обговорити з замовником
RS_7	Відкритий	Затримка розробки по календарному графіку	Середній	Великий	0,15	Пришвидшити роботу за рахунок інших робіт	Попередження	Обговорити з замовником строки здачі

Таблиця Б.9 – Продовження таблиці Б.8

RS_8	Відкритий	Помилки при написанні документації	Малий	Середній	0,4	Зробити відповідні зміни в ТЗ	Зменшення	
RS_9	Відкритий	Некоректна робота хостингу	Малий	Середній	0,02	Звернутись до служби підтримки хостингу	Ухилення	Зміна хостингу
RS_10	Відкритий	Втрата файлів	Малий	Великий	0,08	Робити резервні копії, використати саме їх	Попередження	Відновити втрачені файли за допомогою утиліт

ДОДАТОК В. КОД РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

web.php

```

Route::get('/', 'IndexController@index')->name('index');

Route::get('/login', 'HomeController@index');

Route::get('/search', 'SearchController@index')->name('search.index');

Route::get('/contact', 'ContactController@index')->name('contact');

Route::group(['namespace' => 'Cabinet'], function () {
    Route::get('/cabinet', 'IndexController')->name('cabinet.index');
    Route::get('/cabinet/create', 'CreateController')->name('cabinet.create');
    Route::post('/cabinet', 'StoreController')->name('cabinet.store');
    Route::get('/cabinet/{user}', 'ShowController')->name('cabinet.show');
    Route::get('/cabinet/{cabinet}/edit', 'EditController')->name('cabinet.edit');
    Route::patch('/cabinet/{cabinet}', 'UpdateController')->name('cabinet.update');
});

Route::group(['namespace' => 'Category'], function() {
    Route::get('/category', 'IndexController')->name('category.index');
    Route::get('/category/create', 'CreateController')->name('category.create');
    Route::post('/category', 'StoreController')->name('category.store');
    Route::get('/category/{category}', 'ShowController')->name('category.show');
});

Route::group(['namespace' => 'Product'], function () {
    Route::get('/product', 'IndexController')->name('product.index');
    Route::get('/product/create/{category}', 'CreateController')->name('product.create');
    Route::post('/product', 'StoreController')->name('product.store');
    Route::get('/product/{product}', 'ShowController')->name('product.show');
    Route::get('/product/{product}/edit', 'EditController')->name('product.edit');
    Route::patch('/product/{product}', 'UpdateController')->name('product.update');
    Route::delete('/product/{product}', 'DestroyController')->name('product.delete');
});

Route::group(['namespace' => 'Basket'], function () {
    Route::get('/basket', 'IndexController')->name('basket.index');
    Route::post('/basket', 'StoreController')->name('basket.store');
    Route::delete('/basket/{basket}', 'DestroyController')->name('basket.delete');
});

Route::group(['namespace' => 'Order'], function () {
    Route::get('/order', 'IndexController')->name('order.index');
    Route::post('/order', 'StoreController')->name('order.store');
    Route::delete('/order/{order}', 'DestroyController')->name('order.delete');
});

```

```

Route::patch('/order/{order}','UpdateController')->name('order.update');
Route::get('/order/complete','CompleteController')->name('order.complete');

});

Route::group(['namespace' => 'Compare'], function (){
    Route::get('/compare','IndexController')->name('compare.index');
    Route::post('/compare','StoreController')->name('compare.store');
    Route::delete('/compare/{compare}','DestroyController')->name('compare.delete');
});

Auth::routes();

Route::get('/home',[App\Http\Controllers\HomeController::class,'index']->name('home'));

```

header.blade.php

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="{{ asset('css/app.css') }}">
    <link rel="stylesheet" href="{{ asset('css/style.css') }}">
    <link rel="icon" href="{{ asset('images/slider/flag.jpeg') }}">
    <title>Кибернетики</title>
</head>
<body>
    <div>
        <nav class="navbar navbar-expand-lg navbar-transparent bg-transparent">
            <a class="navbar-brand" href="{{ route('index') }}"></a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item active">
                        <a class="nav-link" href="{{ route('index') }}">Головна</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{ route('category.index') }}">Категорії</a>

```



```

    </li>
    <li class="nav-item">
      <a class="nav-link" href="{{ route('contact') }}">Контакти</a>
    </li>
  {{--
    <li class="nav-item">--}}
  {{--
    <a class="nav-link" href="{{ route('product.index') }}">Товари</a>--}}
  {{--
    </li>--}}
</ul>
<form class="form-inline input-group" action="{{ route('search.index') }}">
  @csrf
  <div class="input-group m-lg-5">
    <input type="search" id="search" class="input_search" name="search"
placeholder="Пошук" aria-label="Search">
    <div class="input-group-append">
      <button class="btn_search" type="submit"></button>
    </div>
  </div>
</form>

<ul class="navbar-nav ms-auto">
  <!-- Authentication Links -->
  @guest
    @if (Route::has('login'))
      <li class="nav-item">
        <a class="nav-link" href="{{ route('login') }}">{{ ('Вхід') }}</a>
      </li>
    @endif

    @if (Route::has('register'))
      <li class="nav-item">
        <a class="nav-link" href="{{ route('register') }}">{{ ('Реєстрація') }}</a>
      </li>
    @endif
  @else
    <li class="nav-link dropdown">
      <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>
        <br>
        {{ Auth::user()->name }}
      </a>
      <div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="{{ route('cabinet.index') }}">Особисті
дані</a>

        @can('view', auth()->user())
          <a class="dropdown-item" href="{{ route('order.index') }}">Адмін.
панель</a>
        @endcan
        <a class="dropdown-item" href="{{ route('basket.index') }}">Кошик</a>
        <a
class="dropdown-item"
href="{{ route('compare.index') }}">Порівняння</a>

```

```

<div class="dropdown-divider"></div>
<div>
  <a class="dropdown-item" href="{{ route('logout') }}"
    onclick="event.preventDefault();
    document.getElementById('logout-form').submit();">
    {{ __('Выйти') }}
  </a>

  <form id="logout-form" action="{{ route('logout') }}" method="POST"
class="d-none">
    @csrf
  </form>
</div>
</div>
</li>
@endguest
</ul>
</div>
</nav>
</div>
<hr>
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkVYIK3UENzmM7KChRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
crossorigin="anonymous"></script>
  @yield('layout/header')
</body>
</html>

```

index.blade.php

```

@extends('layout/header')
@section('layout/header')
  @if(!empty($successMsg))
    <div class="alert alert-success" role="alert">
      
      <h1>Битачмо, {{ \Illuminate\Support\Facades\Auth::user()->name }}.</h1>
      {{ $successMsg }}
    </div>
  @endif

  <div class="wrapper_slider">
    <div id="carouselExampleIndicators" class="carousel slide box-1" data-ride="carousel">
      <div class="carousel-inner">

```

```

        <div class="carousel-item active">
            
        </div>
        <div class="carousel-item">
            
        </div>
        <div class="carousel-item">
            
            <div class="carousel-caption d-none d-md-block">
                <h3 class="text-white">Слава Україні!</h3>
                <h5 class="text-white">Віримо в Україну! Віримо в ЗСУ!</h5>
            </div>
        </div>
        </div>
        <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-
slide="prev">
            
        </a>
        <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-
slide="next">
            
        </a>
    </div>
    <div class="box-2">
        <a href="{{route('product.show', 26)}}">
            <div class="card bg-dark text-white">
                <img alt="Card
image">
                <div class="card-img-overlay">
                    <h5 class="card-title text-white">Apple iPod 7gen 128Gb Red</h5>
                    <p class="card-text text-white">Всього лише 12999 €!</p>
                    <p class="card-text text-white"> - сенсорний Retina Display, IPS, діагональ 4",
роздільна здатність 1136x640 точок, олеофобне покриття</p>
                    <p class="card-text text-white"> - книги та подкасти, зйомка і редагування Full
HD -відео, Apple Music, Apple Arcade</p>
                </div>
            </div>
        </a>
    </div>
    <div class="box-3">
        <div class="card bg-black">
            
            <div class="card-img-overlay">
                <h5 class="card-title text-center text-white">Акція!</h5>
                <p class="card-text text-white">При першому замовленні - знижка 10%!</p>
            </div>
        </div>
    </div>

```

```

        <p class="card-text text-white">Акція розповсюджується на замовлення до 5000
грн.</p>
    </div>
</div>
</div>
</div>
<div>
    <h2>
        
        Найкращі товари
        
    </h2>
    <div class="wrapper_index_products">
        @foreach($products as $product)
            <div>
                <a href="{{ route('product.show',$product->id) }}">
                    
                    <p>{{ $product->name }}</p>
                    <hr>
                    <p>{{ $product->price }} ₪</p>
                </a>
            </div>
        @endforeach
    </div>
</div>

@if(isset($history) && !$history->isempty())
    <div>
        <h2>Історія останніх переглядів</h2>
        <div class="wrapper_history">
            @foreach($history as $history)
                <p hidden="visible">{{ $product = \App\Models\Product::find($history-
>product_id) }}</p>
                <div>
                    <a href="{{ route('product.show',$product->id) }}">
                        
                        <p>{{ $product->name }}</p>
                    </a>
                </div>
            @endforeach
        </div>
    </div>
@endif
@include('layout/footer')
@endsection

```

```

<!doctype html>
<html lang="en">
<head>
  <link rel="stylesheet" href="{{ asset('css/style.css') }}">
</head>
<body>
<hr>
  <div class="wrapper_footer">
    <div class="item1">
      <h3>Навігація</h3>
      <h5><a class="nav-link" href="{{ route('index') }}">Головна</a></h5>
      <h5><a class="nav-link" href="{{ route('category.index') }}">Категорії</a></h5>
      <h5><a class="nav-link" href="{{ route('contact') }}">Контакти</a></h5>
    </div>

    <div class="item2">
      <h3>Офіційні представники</h3>
      
      
      
      
      
      
      
    </div>

    <div class="item3">
      <h3>© Всі права захищені 2023</h3>
      
    </div>
  </div>
  @yield('layout/footer')
</body>
</html>

```

login.blade.php

```

@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header">{{ ('Авторизація') }}</div>

        <div class="card-body">
          <form method="POST" action="{{ route('login') }}">
            @csrf

            <div class="row mb-3">
              <label for="email" class="col-md-4 col-form-label text-md-end">{{ ('Пошта')
            }}</label>

              <div class="col-md-6">
                <input id="email" type="email" class="form-control @error('email') is-invalid
                @enderror" name="email" value="{{ old('email') }}" required autocomplete="email" autofocus>

                @error('email')
                  <span class="invalid-feedback" role="alert">
                    <strong>Невірна пошта або пароль!</strong>
                  </span>
                @enderror
              </div>
            </div>

            <div class="row mb-3">
              <label for="password" class="col-md-4 col-form-label text-md-end">{{ ('Пароль')
            }}</label>

              <div class="col-md-6">
                <input id="password" type="password" class="form-control @error('password') is-
                invalid @enderror" name="password" required autocomplete="current-password">

                @error('password')
                  <span class="invalid-feedback" role="alert">
                    <strong>Невірний пароль!</strong>
                  </span>
                @enderror
              </div>
            </div>

            <div class="row mb-3">
              <div class="col-md-6 offset-md-4">
                <div class="form-check">

```



```

@endif
@if(!empty($successMsgCompare))
  <div class="alert alert-success" role="alert">
    
    <h1>{{ $successMsgCompare }}</h1>
    <p><a href="{{ route('compare.index') }}">Перейти до порівняння!</a> </p>
  </div>
@endif
<p hidden="hidden">{{ $options_product = \App\Models\Option_Product::query()-
>where('product_id', '=', $product->id)->get() }}</p>
<div class="container">
  <div class="row">
    <div class="col-sm">
      
    </div>
    <div class="col-sm">
      <h3>{{ $product->name }}</h3>
      <h4 class="mb-4 mt-4">Ціна: {{ $product->price }} ₪</h4>
      <form action="{{ route('basket.store') }}" method="post">
        @csrf
        <input name="product_id" value="{{ $product->id }}" hidden="hidden">
        <input name="user_id" value="{{ Auth::id() }}" hidden="hidden">
        <button type="submit" class="btn btn-success">В кошик</button>
      </form>
      <form action="{{ route('compare.store') }}" method="post">
        @csrf
        <input name="product_id" value="{{ $product->id }}" hidden="hidden">
        <input name="user_id" value="{{ Auth::id() }}" hidden="hidden">
        <button type="submit" class="btn btn-warning mt-3 mb-3">В порівняння</button>
      </form>
    </div>
  </div>
</div>
{{ --<h2>Характеристики</h2>
<p>{{ $product->description }}</p>-- }}

<div id="accordion">
  <div class="card">
    <div class="card-header" id="headingOne">
      <h5 class="mb-0">
        <button class="btn btn-link" data-toggle="collapse" data-target="#collapseOne" aria-
expanded="true" aria-controls="collapseOne">
          Характеристики
        </button>
      </h5>
    </div>
    <div id="collapseOne" class="collapse show" aria-labelledby="headingOne" data-
parent="#accordion">
      <div class="card-body">
        <table class="character">

```



```

{{--      todo: ВЫРОВНЯТЬ ПОЛЯ--}}
@foreach($options as $option)
    <tr>
        <td class="td_char">
            <p>{{ $option->title }}</p>
        </td>
        @foreach($options_product as $option_product)
            @if($option->id == $option_product->option_id)
                <td class="td_char">
                    <p>{{ $option_product->name }}</p><br>
                </td>
            @endif
        @endforeach
    </tr>
@endforeach
</table>
</div>
</div>
</div>
</div>

@can('view', auth()->user())
    <a href="{{ route('product.edit', $product->id) }}"><button type="button" class="btn btn-success mt-3 active">Редагувати</button></a>
<form action="{{ route('product.delete', $product->id) }}" method="post">
    @csrf
    @method('delete')
    <button type="submit" class="btn btn-danger mt-3 active">Видалити</button>
</form>
@endcan

<a href="{{ url()->previous() }}"><button type="button" class="btn btn-warning mt-3 active">Назад</button></a>
</body>
</html>

@endsection

```

contact.blade.php

```

@extends('layout/header')
@section('layout/header')

<h1>Контакти</h1>
<div class="wrapper_product">
    <div>
        <h2>Адреса:</h2>
        <h4>вулиця Воскресенська, 13, Суми, Сумська область, 40000</h4>
        <br>
        <h2>Номер телефону:</h2>
    </div>

```

```

    <h4><a href="tel:+380664295977">+380664295977</a></h4>
    
</div>
<div>
    <h2>Ми на карті</h2>
    <iframe
                                                class="mt-3"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d1257.8605439183227!2d34.798775
87585685!3d50.91037671944615!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x41290189e
78c4ddb%3A0xa70f9e79c6d03fc2!2z0JHRltCa0J7QnCDQmtC-
0LzQvyfRjtGC0LXRgNC90YvQuSDQvNCw0LPQsNC30LjQvSwg0L3QvtGD0YLQsdGD0LrQuCwg0
YHQvNCw0YDRgtGE0L7QvdC4LCDQv9GA0LjQvdGC0LXRgNCwINC30LDQv9GA0LDQstC60LA
g0YDQtdC80L7QvdGCLdGB0LXRgNCy0ZbRgQ!5e0!3m2!1sru!2sua!4v1685100207243!5m2!1sru!2s
ua" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-
referrer-when-downgrade"></iframe>
</div>
</div>

    @include('layout/footer')
@endsection

```

Product\ShowController.php

```

<?php

namespace App\Http\Controllers\Product;

use App\Http\Controllers\Category\BaseController;
use App\Models\History;
use App\Models\Option;
use App\Models\Product;
use Illuminate\Support\Facades\Auth;

class ShowController extends BaseController
{
    public function __invoke(Product $product)
    {
        if(Auth::user()){
            $data = ['product_id' => $product->id, 'user_id' => Auth::id()];
            $request = \Illuminate\Http\Request::create('/product/{product}','POST',$data);
            $data = $request->validate([
                'product_id' => 'required|integer',
                'user_id' => 'required|integer',
            ]);
            History::create($data);
        }

        $options = Option::query()->where('category_id' , '=', $product->category_id)->get();
        return view('product.show',compact('product','options'));
    }
}

```

Product\StoreRequest.php

```

<?php

namespace App\Http\Requests\Product;

use Illuminate\Foundation\Http\FormRequest;

class StoreRequest extends FormRequest
{
    /
    * Determine if the user is authorized to make this request.
    *
    * @return bool
    */
    public function authorize()
    {
        return true;
    }

    /
    * Get the validation rules that apply to the request.
    *
    * @return array
    */
    public function rules()
    {
        return [
            'name' => 'required|string',
            'price' => 'required|integer',
            'photo' => 'required|image|mimes:jpeg,png,jpg,gif,svg',
            'description' => 'string',
            'brand_id' => 'required|integer',
            'category_id' => 'required|integer',
        ];
    }
}

```

Product\UpdateController.php

```

<?php

namespace App\Http\Controllers\Product;

use App\Http\Requests\Product\UpdateRequest;
use App\Models\Option_Product;
use App\Models\Product;

class UpdateController extends BaseController
{
    public function __invoke(Product $product, UpdateRequest $request)
    {

```

```

//      todo: обновление инфы
$data = $request->validated();
$this->service->update($data, $product);

$names = $request->input('names');
foreach ($names as $key => $value) {
    $data = ['name' => $value, 'product_id' => $product->id, 'option_id' => $key];
    $request = \Illuminate\Http\Request::create('/product', 'POST', $data);
    $data = $request->validate([
        'name' => 'required|string',
        'product_id' => 'required|integer',
        'option_id' => 'required|integer',
    ]);
    $option_product = Option_Product::find($key);
    $option_product->update($data);
}
return redirect()->route('product.show', $product->id);
}
}

```

Product\StoreController.php

```
<?php
```

```

namespace App\Http\Controllers\Product;

use App\Http\Requests\Product\StoreRequest;
use App\Models\Option_Product;
use App\Models\Product;
use Illuminate\Http\Request;

class StoreController extends BaseController
{
    public function __invoke(StoreRequest $request)
    {
        $data = $request->validated();
        $this->service->store($data);
        $product = Product::latest()->first();

        $names = $request->input('names');
        foreach ($names as $key => $value) {
            $data = ['name' => $value, 'product_id' => $product->id, 'option_id' => $key];
            $request = \Illuminate\Http\Request::create('/product', 'POST', $data);
            $data = $request->validate([
                'name' => 'required|string',
                'product_id' => 'required|integer',
                'option_id' => 'required|integer',
            ]);
            Option_Product::create($data);
        }
    }
}

```

```

    }
    return redirect()->route('product.show', $product->id);
}
}

```

Cabinet.php

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class Cabinet extends Model
```

```
{
    use HasFactory;
    protected $table = 'cabinets';
    protected $guarded = false;
    protected $fillable = ['user_id', 'name', 'surname', 'date', 'phone', 'city', 'post', 'department'];
}
```

Product.php

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class Product extends Model
```

```
{
    use HasFactory;
    protected $table = 'products';
    protected $guarded = false;
    protected $fillable = ['name', 'brand_id', 'price', 'photo', 'description', 'category_id'];
}
```

.env

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
```

```
DB_PORT=3306  
DB_DATABASE=shop  
DB_USERNAME=root  
DB_PASSWORD=
```