Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

Дороша Олега Олександровича
Прізвище, ім'я, по батькові студента(ки)

на здобуття ступеня вищої освіти Бакалавра

Інтернет-магазин з продажу побутової техніки
Назва теми

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.200164.01.08.ПЗ

Виконав студент IV курсу, група ІПЗ-20-1 _____ Олег ДОРОШ
Підпис Ім'я, ПРІЗВИЩЕ

Керівник канд. техн. наук, доцент _____ Галина РАДЕЛЬЧУК
Науковий ступінь, звання Підпис Ім'я, ПРІЗВИЩЕ

Нормоконтролер доцент _____ Галина РАДЕЛЬЧУК
Науковий ступінь, звання Підпис Ім'я, ПРІЗВИЩЕ

До захисту допускаю:
Завідувач кафедри інженерії
програмного забезпечення _____ Леонід БЕДРАТЮК
Підпис Ім'я, ПРІЗВИЩЕ

1. 06 2024 р.

Хмельницький 2024

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення
Рівень вищої освіти Перший (бакалаврський)
Галузь знань 12 «Інформаційні технології»
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри ІПЗ
Л. П. Бедратюк
02 01 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дорошу Олегу Олександровичу
Прізвище, ім'я, по батькові студента

1. Тема роботи Інтернет-магазин з продажу побутової техніки

Керівник роботи Радельчук Галина Іванівна канд. техн. наук, доцент
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 08.03.2024 р. № 6
2. Строк подання студентом роботи на кафедру 01.06.2024 р.
3. Вихідні дані до роботи Матеріали переддипломної практики
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
Дослідження предметної області та постанока задачі
Проєктування програмного забезпечення
Програмна реалізація та тестування програмного забезпечення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)
Три креслення у форматі А3
1.Діаграма класів
2.Діаграма баз даних
3.Діаграма функціоналу магазину

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Радельчук Г. І., канд.техн. наук. доцент	13.06.24	20.06.24
Антиплагіат	Форкун Ю. В., доцент	30.05.24	06.06.24

7. Дата видачі завдання «_02_» __січня_ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Збір матеріалу за темою кваліфікаційної роботи (КвР); дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	01.12 — 30.12.2023	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення, визначення задач та вимог, розробка технічного завдання	03.01 — 31.01.2024	
3 Проєктування вебсервісу. Розробка графічної частини	02.02 — 29.02.2024	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 — 12.04.2024	
5 Написання вступу, загальних висновків, оформлення джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог стандартів	29.04 — 26.04.2024	
6 Попередній захист	Травень 2024	Згідно графіка
7 Перевірка КвР на плагіат, нормконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування пояснювальної записки	21.05.2024	
8 Підготовка до захисту та захист КвР	11.06.2024	

Студент _____ Олег ДОРОШ
Підпис Ім'я, ПРІЗВИЩЕ

Керівник роботи _____ Галина РАДЕЛЬЧУК
Підпис Ім'я, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інтернет-магазин з продажу побутової техніки»

Автор роботи: Дорош Олег Олександрович

Керівник роботи: Радельчук Галина Іванівна

Пояснювальна записка : 57с., 27 рис., 4 додатки, 30 джерел.

Графічна частина: 3 креслення у форматі А3.

ВЕБСЕРВІС, БАЗА ДАНИХ, ІНТЕРНЕТ-МАГАЗИН, ВЕБСАЙТ.

Мета кваліфікаційної роботи – розробити вебзастосунок для замовлення товарів, а саме побутової техніки за допомогою мережі Інтернет, застосувати знання та вміння, отримані у процесі навчання.

Під час виконання програмного продукту відповідно до вимог проєкту було проаналізовано вебсайти, вимоги до вебсервісу, досліджено предметну область, було знайдено функціональні та нефункціональні вимоги, зроблено архітектуру програмного забезпечення, розроблена база даних, розроблено застосунок та проведене тестування. Були використані такі технології: Postman, Bootstrap, ASP.NET Core.

В результаті виконання кваліфікаційної роботи була розроблена візуальна та сервісні частини програмного застосунку з продажу побутової техніки.

20.06.2024
(дата)

(підпис)

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			Текстові документи			
1	А4	КвРІПЗ.200164.01.08.ПЗ	Пояснювальна записка	57		
2	А4		Завдання на кваліфікаційну роботу	1		
3	А4		Анотація	1		
			Графічні документи			
4	А3	КвРІПЗ.200164.01.08.Е8	Діаграма класів	1		
5	А3	КвРІПЗ.200164.01.08. Е8	Діаграма баз даних	1		
6	А3	КвРІПЗ.200164.01.08. Е8	Діаграма фунціоналу магазину	1		

ЗМІСТ

Вступ.....8
1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....10
1.1 Аналіз та опис предметної області.....10
1.2 Аналіз існуючих подібних розробок.....14
1.3 Аналіз вимог до програмного забезпечення та розробка технічного завдання.....19
1.4 Функціональні вимоги до веб-систем інтернет-магазину:.....21
1.5 Нефункціональні вимоги до веб-системи інтернет-магазину:.....23
1.6 Вимоги до надійності веб-системи інтернет-магазину:.....24
2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ВИЗНАЧЕННЯ МЕТОДІВ І ЗАСОБІВ РЕАЛІЗАЦІЇ.....26
2.1 Проектування архітектури програмного забезпечення.....26
2.2 Проектування інтерфейсу програмного забезпечення.....31
2.3 Детальне проектування програмного забезпечення.....35
2.4 Проектування бази даних.....36
2.5 Аналіз та вибір технологій і методів реалізації програмного забезпечення.....37
3. ПРОГРАМНА РЕАЛІЗАЦІЯ.....40
3.1 Структура програмних модулів.....40
3.2 Реалізація БД.....41
3.3 Реалізація сайту.....42

3.4 Інструкція з експлуатації 47
3.5 Тестування програмного забезпечення 50
ВИСНОВКИ 54
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 55
ДОДАТОК А Технічне завдання 58
ДОДАТОК Б Програмний код 60
ДОДАТОК В Презентаційні матеріали 67

ВСТУП

Зважаючи на стрімкий розвиток технологій та торгівлі в сучасному світі, інтернет-магазини стають все більш популярними. Електронна комерція набуває все більшої затребуваності завдяки своїй зручності, доступності та широкому спектру переваг як для покупців, так і для власників магазинів.

Переваги інтернет-магазинів:

Для покупців:

– Широкий асортимент: Інтернет-магазини не обмежені фізичним простором, тому можуть пропонувати значно ширший вибір товарів, порівняно з фізичними торговими точками. Це дозволяє покупцям знаходити й купувати товари, які можуть бути недоступні у їхньому місті чи регіоні.

– Зручність та доступність: Купувати товари в інтернет-магазині можна в будь-який час доби, з будь-якого місця, де є доступ до інтернету. Це економить час та зусилля, адже не потрібно витрачати час на поїздки до магазину, пошук потрібного товару та очікування в черзі.

– Порівняння цін: Інтернет дає можливість легко порівнювати ціни на товари в різних магазинах, знаходячи найвигідніші пропозиції.

– Детальна інформація: Інтернет-магазини часто пропонують детальні описи товарів, фотографії та відео, характеристики, а також відгуки інших покупців, що допомагає зробити вдалий вибір.

– Різні способи оплати: Інтернет-магазини пропонують різні зручні способи оплати, такі як готівка при отриманні, оплата карткою, переказ на банківський рахунок тощо.

– Доставка додому: Замовлені товари можна отримати вдома, не витрачаючи час на самовивіз.

Для власників:

– Економія коштів: Відкриття інтернет-магазину потребує значно менших витрат, порівняно з відкриттям фізичного магазину, адже не потрібно витрачатися на оренду приміщення, оплату праці персоналу, обладнання тощо.

– Широке охоплення аудиторії: Інтернет-магазин доступний для покупців з будь-якої точки світу, де є доступ до інтернету, що значно розширює потенційну аудиторію.

– Зниження витрат на рекламу: Інтернет-маркетинг може бути значно ефективнішим та дешевшим, порівняно з традиційними методами реклами.

– Збір даних про клієнтів: Інтернет-магазини дають можливість збирати та аналізувати дані про поведінку клієнтів, що допомагає покращити маркетингову стратегію та пропонувати більш релевантні товари та послуги.

– Простота управління: Інтернет-магазином значно легше управляти, порівняно з фізичним магазином, адже всі процеси автоматизовані та доступні онлайн.

Зростаюча популярність інтернет-магазинів побутової техніки:

З кожним днем здійснюється все більше і більше замовлень через мережу Інтернет. Це пов'язано з низкою факторів, таких як:

– Зручність: Купувати побутову техніку онлайн значно зручніше, адже не потребує виходити з дому, витрачати час на пошук потрібної моделі в фізичних магазинах та стояти в чергах.

– Широкий вибір: Інтернет-магазини пропонують значно ширший асортимент побутової техніки, порівняно з фізичними торговими точками. Це дозволяє знайти й купити саме ту модель, яка відповідає вашим потребам та бюджету.

– Вигідні ціни: Інтернет-магазини часто пропонують більш вигідні ціни на побутову техніку, адже не мають таких витрат, як оренда приміщення та оплата праці персоналу

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз та опис предметної області

Електронний магазин — сайт, з якого можна вибрати та замовити потрібний товар чи послугу. Інтернет-магазин перш за все передбачає грошові розрахунки на відміну від сайтів, які надають послуги безкоштовно.

Інтернет-магазини завойовують шалену популярність, адже пропонують безліч переваг як для покупців, так і для власників.

Переваги для покупців:

– Цілодобова доступність: Інтернет-магазини доступні 24/7, що дозволяє купувати товари в будь-який зручний час, без потреби підлаштовуватися до режиму роботи фізичних магазинів.

– Широкий асортимент: Завдяки відсутності обмежень простору, інтернет-магазини можуть пропонувати значно ширший вибір товарів, порівняно з фізичними торговими точками.

– Економія часу: Пошук та купівля товарів в інтернеті економить час, адже не потрібно витрачати його на поїздки до магазину, пошук потрібного товару та очікування в черзі.

– Порівняння цін: Інтернет дає можливість легко порівнювати ціни на товари в різних магазинах, знаходячи найвигідніші пропозиції.

– Відгуки та описи: Інтернет-магазини часто пропонують детальні описи товарів, фотографії та відео, а також відгуки інших покупців, що допомагає зробити вдалий вибір.

– Зручні способи оплати: Інтернет-магазини пропонують різні зручні способи оплати, такі як готівка при отриманні, оплата карткою, переказ на банківський рахунок тощо.

– Доставка додому: Замовлені товари можна отримати вдома, не витрачаючи час на самовивіз.

Переваги для власників:

– Економія коштів: Відкриття інтернет-магазину потребує значно менших витрат, порівняно з відкриттям фізичного магазину, адже не потрібно витрачатися на оренду приміщення, оплату праці персоналу, обладнання тощо.

– Широке охоплення аудиторії: Інтернет-магазин доступний для покупців з будь-якої точки світу, де є доступ до інтернету, що значно розширює потенційну аудиторію.

– Зниження витрат на рекламу: Інтернет-маркетинг може бути значно ефективнішим та дешевшим, порівняно з традиційними методами реклами.

– Збір даних про клієнтів: Інтернет-магазини дають можливість збирати та аналізувати дані про поведінку клієнтів, що допомагає покращити маркетингову стратегію та пропонувати більш релевантні товари та послуги.

– Простота управління: Інтернет-магазином значно легше управляти, порівняно з фізичним магазином, адже всі процеси автоматизовані та доступні онлайн.

Заходячи в інтернет-магазин, у клієнта немає необхідності:

– Чекати менеджера: В інтернет-магазині можна самостійно знайти необхідний товар, порівняти характеристики та ціни, оформити замовлення без допомоги продавця.

– Стояти в черзі: Інтернет-магазин не має фізичних черг, адже всі покупки здійснюються онлайн.

– Слухати поради, які йому не потрібні: В інтернет-магазині клієнт може ознайомитися з інформацією про товар та зробити свій вибір без настирливих порад продавців.

– Виходити з дому: Купити товар в інтернет-магазині можна, не виходячи з дому, що особливо актуально в холодну пору року, або якщо у клієнта немає можливості самостійно відвідати магазин.

Якщо покупцеві потрібна консультація, він завжди може звернутися в онлайн підтримку, де кваліфіковані консультанти допоможуть йому з вибором товару, оформленням замовлення або вирішенням інших питань.

Аналізуючи рішення, що існують в мережежі Інтернет та відносяться до даної предметної області, було зрозуміло, що існує велике різноманіття вебдодатків, що стосуються продажу різної комп'ютерної електроніки. Інтернет-магазини такої тематики дуже поширені. Це призводить до поширеного вибору даного товару онлайн. В такому випадку важливо розуміти, як саме працює інтернет-магазин, які функції є доступними для здійснення такої діяльності, як офомлення замовлень.

Отже магазин це реалізований в мережі інтернет сайт для продажу товарів. Кількість представлених видів товарів може бути дуже великою, оскільки немає обмежень в просторі, і це призводить до збільшення різноманітності товарів. Характерними рисами інтернет-магазинів є те, що вони можуть пропонувати значно більшу кількість товарів та послуг, ніж реальні магазини і забезпечувати споживачів значно більшим обсягом інформації, необхідної для прийняття рішення про покупку. Також завдяки використанню інтернет технологій є можливим персональний підхід до споживачів з врахуванням попередніх відвідувань магазину та зроблених в ньому покупок та використання інтернет-магазину як ефективного способу маркетингових досліджень.

Процес купівлі в інтернет магазині здійснюється наступним чином:

Покупець за допомогою браузера заходить на сайт інтернет-магазину, який містить електронну вітрину, де представлений каталог товарів та необхідні елементи інтерфейсу для виконання операцій відбору та купівлі товарів;

- перегляд товарного каталогу та вибір товарів (формування кошика покупця);
- реєстрація покупця;
- підтвердження замовлення;
- оплата товару;
- доставка придбаного товару покупцеві.

Пошук товарів в інтернет-магазині може здійснюватися за допомогою каталогу або через внутрішню пошукову систему. Каталог товарів повинен містити повноцінну інформацію про товар, мати зручну структуру. Пошук необхідного товару повинен займати небагато часу. Саме за допомогою каталога можна здійснити огляд товару, ознайомитися з його характеристиками та виглядом, ціною.

У процесі перегляду і відбору товару покупець формує власний віртуальний кошик.

Корзина онлайн-магазину – це сторінка, на яку клієнт відправляє товари, щоб потім придбати їх[1]. Такий список постійно доступний покупцеві. Коли зроблено остаточний вибір товару, покупцеві слід оформити замовлення, для цього необхідно спочатку зареєструватися. Реєстрація полягає у заповненні покупцем спеціальної форми, що включає в себе інформацію про покупця, його поштову або електронну адресу, особистий пароль. Процедура реєстрації дозволяє інтернет-магазинові убезпечити себе від можливих шахрайств та полегшити процедуру покупки для покупця наступного разу.

Оскільки інтернет-магазином повинен хтось керувати, відбувається розмежування функціоналу на адміністративну та користувацьку частини. Адміністратор – це і є та ланка, що забезпечує облік товарів, спілкування з користувачем. В його обов'язки входить додавання нових товарів, їх редагування та в разі необхідності – видалення, тобто він керує наповненістю сайту. Це дуже важлива роль, оскільки від представлення товару, інформації про нього, залежить зацікавленість покупця.

Покупець – це та людина, яка довго розбиратися із функціоналом не буде, якщо щось є незрозумілим, то він вийде з сайту. Тому дуже важливо, щоб інтерфейс інтернет-магазину був продуманий до найменших деталей. Це забезпечить зручну роботу покупця, а отже і хороші враження та відгуки.

1.2 Аналіз існуючих подібних розробок

Вивчаючи доступні в Інтернеті рішення в цій тематиці, стало очевидно, що існує широкий спектр веб-додатків, орієнтованих на продаж різноманітної побутової техніки, наприклад: KVshop.com.ua[2].

Головна сторінка сайту KVshop.com.ua (рисунок 1.1) вражає багатством інформації, але при цьому залишається чітко структурованою та приємною для очей. Користувачам пропонується зручний інтерфейс, що робить навігацію по сайту інтуїтивно зрозумілою та швидкою.

Переваги сайту KVshop.com.ua:

– Ефективний пошук: Завдяки потужній системі пошуку ви зможете знайти необхідний товар за лічені секунди, використовуючи ключові слова, назву моделі або артикул.

– Чітко структурований каталог: Товари розділені за категоріями та підкатегоріями, що полегшує навігацію та пошук потрібної продукції. Ви можете легко переходити з однієї категорії до іншої, звужуючи результати пошуку за певними характеристиками, такими як бренд, ціна, колір, розмір тощо.

– Інформативний слайдер: На головній сторінці розміщено слайдер, який демонструє актуальні пропозиції, новинки, акції та розпродажі. Це дозволяє вам бути в курсі найвигідніших покупок та не пропустити цікаві акційні товари.

– Зручна система тегів: Завдяки системі тегів ви можете швидко знайти потрібний товар за ключовими словами. Цей інструмент економить ваш час та полегшує пошук продукції, особливо якщо ви знаєте точні характеристики того, що шукаєте (рисунок 1.6).

– Багатий контент: KVshop.com.ua пропонує не лише широкий асортимент товарів, але й багато корисної інформації. Ви можете знайти описи продуктів, характеристики, відгуки користувачів, порівняння моделей, поради щодо вибору

техніки та багато іншого. Це допоможе вам зробити вдалий вибір та отримати максимум користі від покупки.

– Зручні способи оплати: KVshop.com.ua пропонує різні зручні способи оплати, щоб ви могли вибрати найкращий для себе. Ви можете оплатити замовлення готівкою, карткою, через систему онлайн-оплати або оформити кредит.

– Швидка доставка: KVshop.com.ua пропонує швидку та зручну доставку замовлень по всій Україні. Ви можете вибрати доставку до відділення Нової Пошти, кур'єрську доставку до дверей або самовивіз з магазину.

– Професійна підтримка: KVshop.com.ua має кваліфіковану команду служби підтримки клієнтів, яка завжди готова допомогти вам з будь-якими питаннями. Ви можете зв'язатися з операторами підтримки по телефону, електронній пошті або через онлайн-чат.

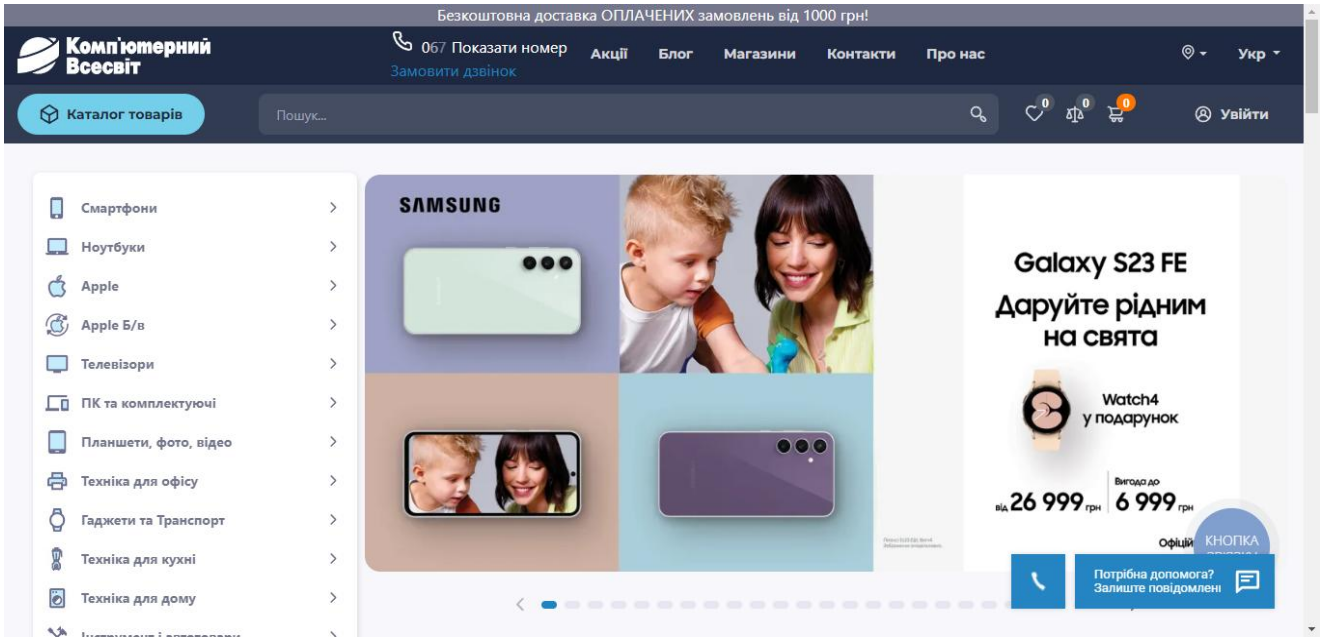


Рисунок 1.1 — Головна сторінка

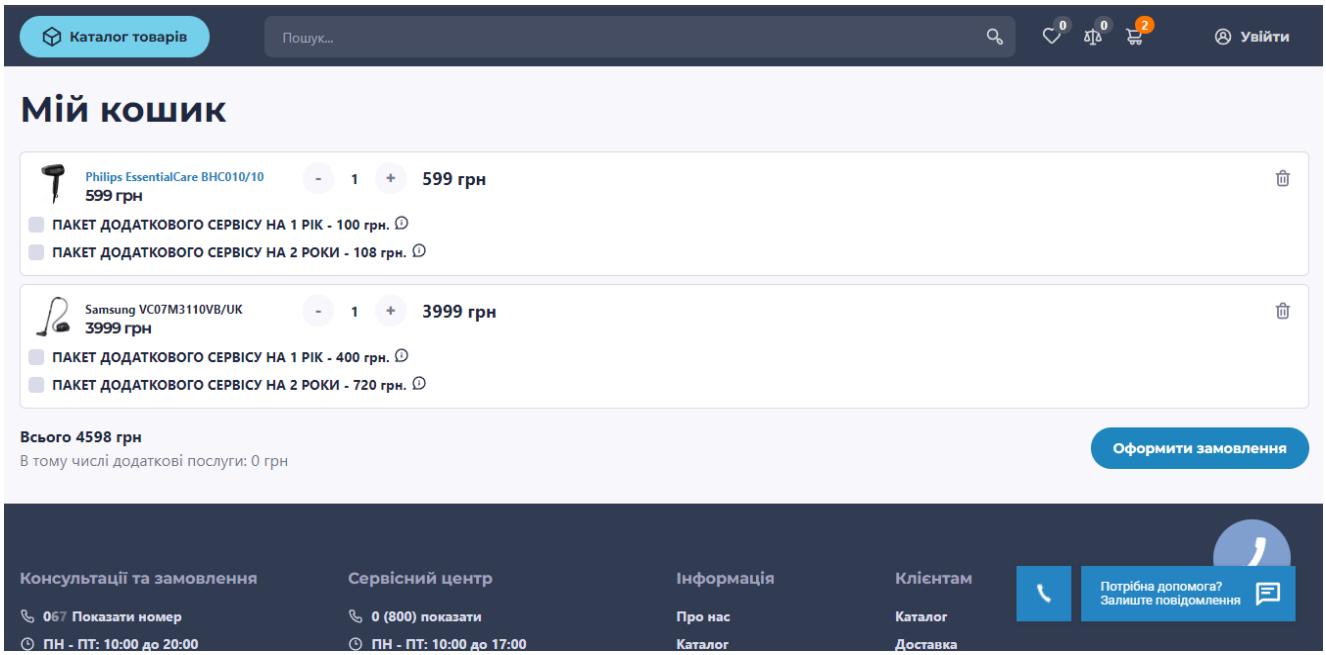


Рисунок 1.2 — Корзина магазину

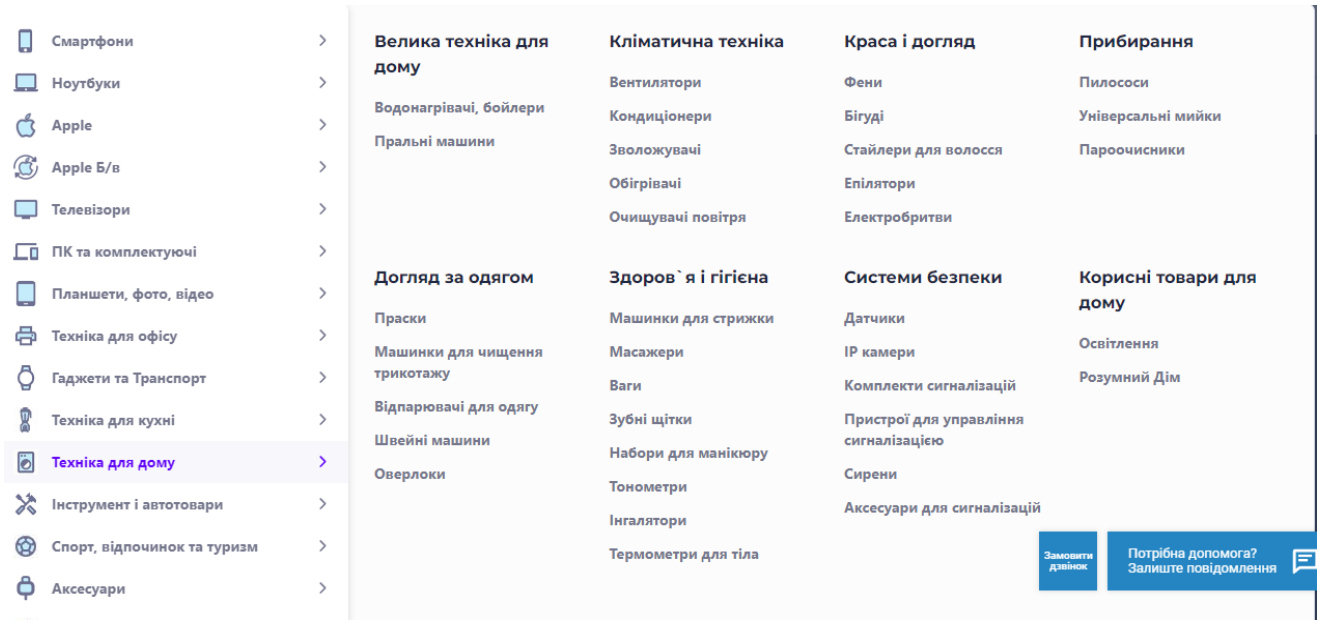


Рисунок 1.3 — Каталог товарів

У сучасному світі, де технології та онлайн-торгівля розвиваються стрімкими темпами, існує безліч інтернет-сервісів, які пропонують послуги з продажу товарів. Кожен з них має свої переваги та недоліки, тому важливо ретельно вивчити всі варіанти, перш ніж робити покупку.

Огляд Rozetka:

Одним з найвідоміших українських інтернет-магазинів є Rozetka, який посідає сьоме місце серед найвідвідуваніших сайтів країни. Цей онлайн-гіпермаркет пропонує широкий спектр товарів, від електроніки та побутової техніки до одягу, косметики та товарів для дому.

Переваги Rozetka:[3]

– Приємний та інтуїтивно зрозумілий інтерфейс: Головна сторінка Rozetka(рисунок 1.4) має лаконічний дизайн, що робить навігацію по сайту зручною та простою навіть для нових користувачів.

– Широкий асортимент товарів: Rozetka пропонує мільйони товарів від різних виробників, що дозволяє користувачам знайти все необхідне в одному місці.

– Зручний пошук: Пошукова система Rozetka дає змогу швидко знайти потрібний товар за назвою, брендом, характеристиками та ціною.

– Вигідні ціни: Rozetka часто пропонує знижки та акції на різні групи товарів, що робить покупки ще більш вигідними.

– Прості та зручні способи оплати: Rozetka пропонує різні способи оплати, включаючи готівку при отриманні, оплату карткою, переказ з банківського рахунку та оплату частинами.

– Швидка та надійна доставка: Rozetka пропонує доставку по всій Україні, а також можливість самовивозу з пункту видачі замовлень.

Недоліки Rozetka:

– Не завжди найвигідніші ціни: Іноді ціни на деякі товари в Rozetka можуть бути трохи вищими, ніж у інших інтернет-магазинах.

– Не всі товари доступні для самовивозу: Деякі товари на Rozetka можна отримати лише з доставкою.

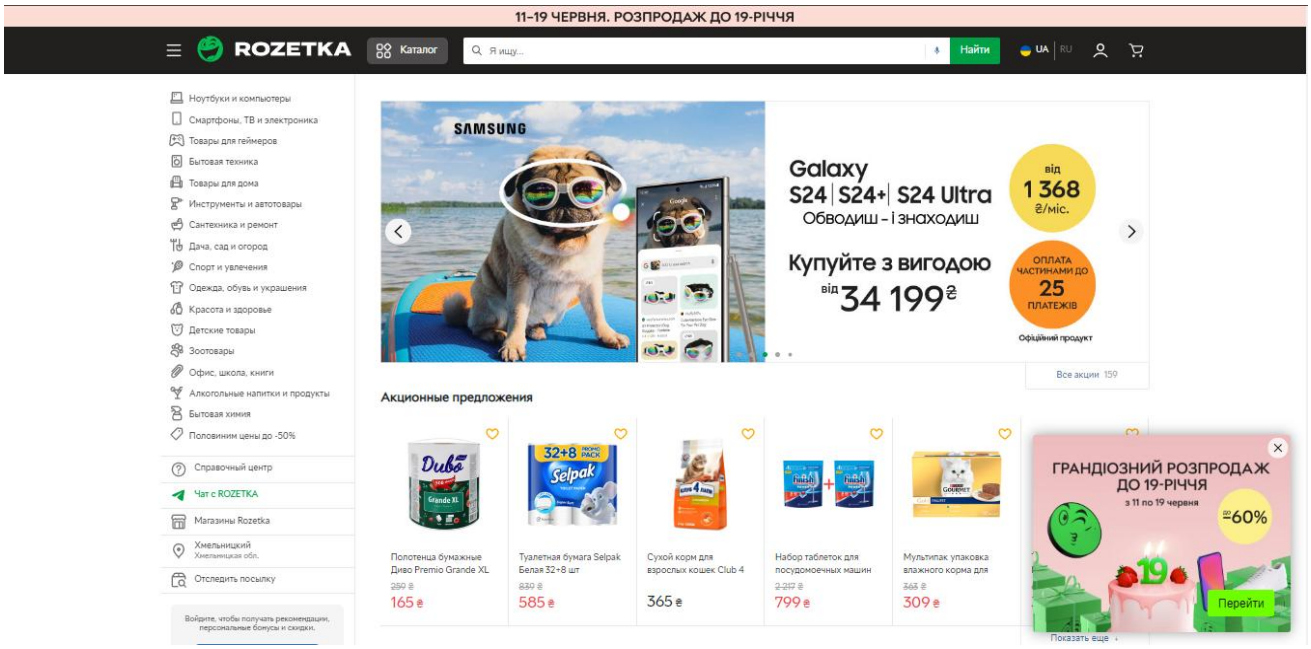


Рисунок 1.4 – Головна сторінка

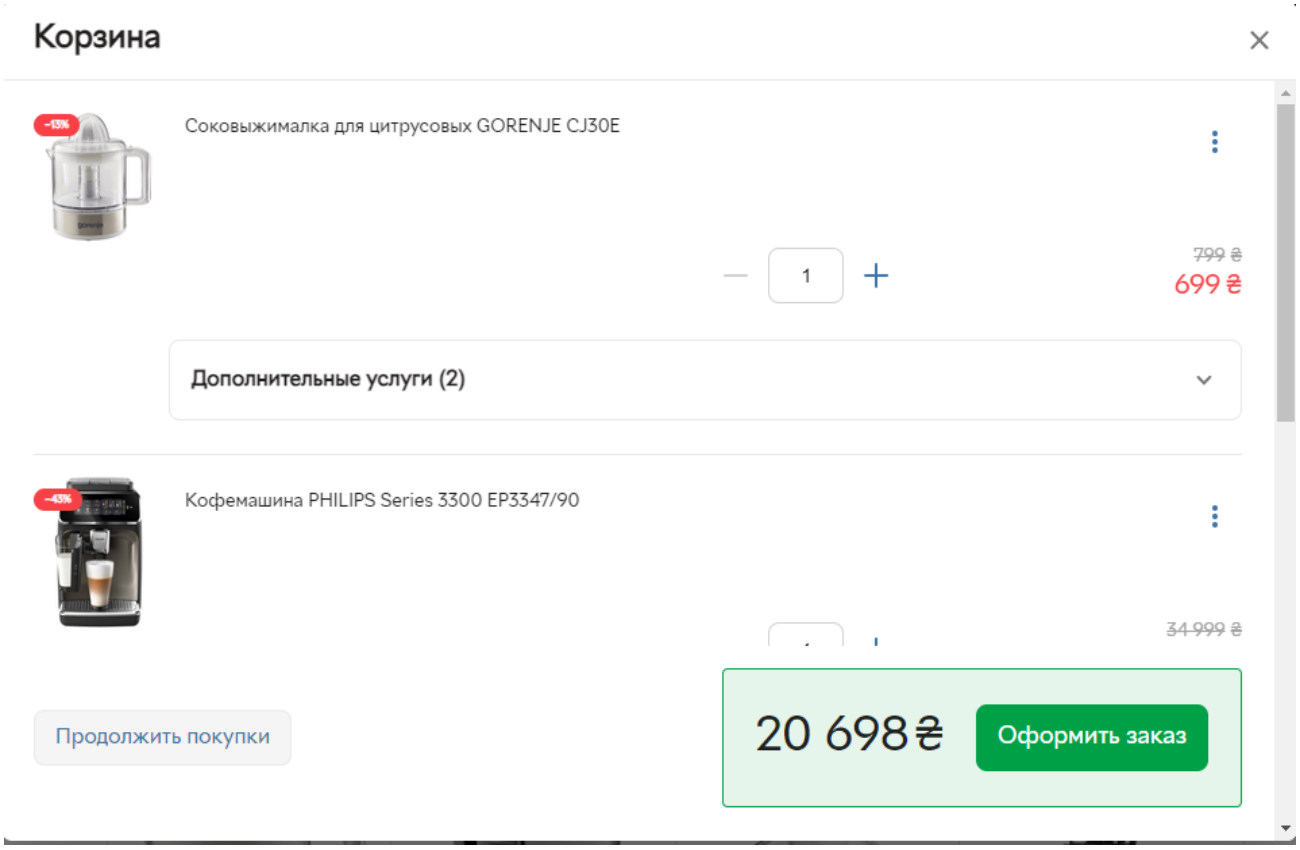


Рисунок 1.5 — Сторінка корзини

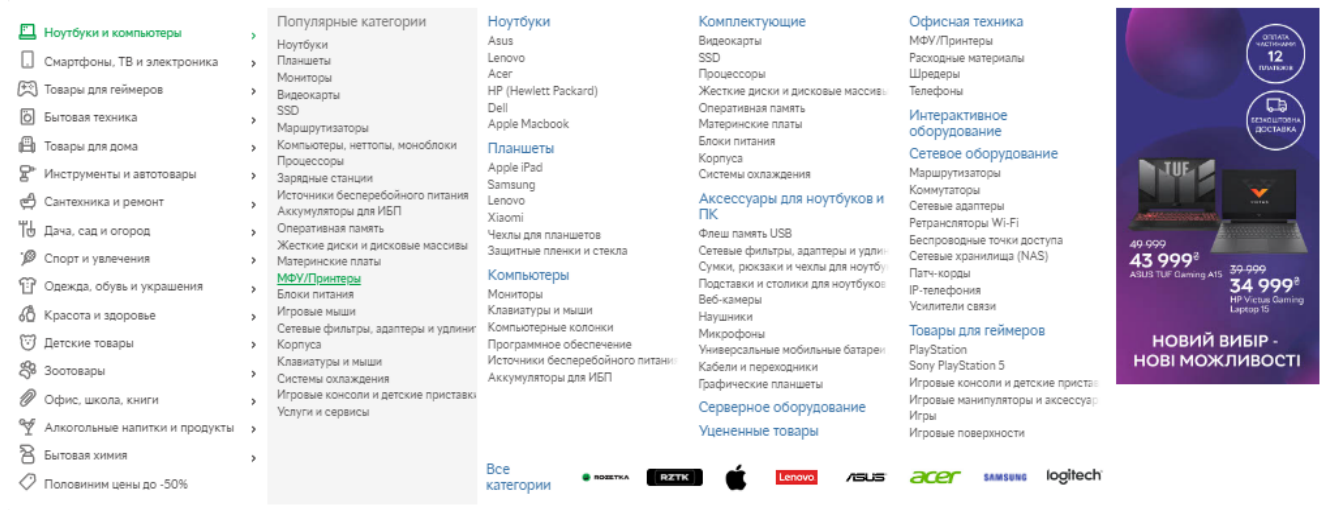


Рисунок 1.6 – Каталог товарів

Rozetka – це надійний та зручний інтернет-магазин з широким асортиментом товарів та вигідними цінами. Завдяки приємному інтерфейсу, зручному пошуку та швидкій доставці Rozetka стає чудовим вибором для онлайн-покупок в Україні.

1.3 Аналіз вимог до програмного забезпечення

Інтернет-магазин було роблено для швидкого та зручного замовлення побутової техніки.

Підставою для розробки є «Завдання на кваліфікаційну роботу».

Найменування розробки: «Інтернет магазин комп’ютерної техніки».

Користувачами програми є адміністрація магазину, що здійснює облік товарів, їх додавання та редагування а також покупці інтернет-магазину, що здійснюють замовлення на основі, доданих адміністрацією сайту товарів.

Щоб сайт функціонував було розроблено функцію замовника:

- реєстрація;
- вхід на сайт;
- меню з сортуванням;
- можливість замовити потрібний товар;
- доставка товару;

– додавання та видалення товару з кошику.

Для розробника:

– додати новий товар;

– редагувати вже створений товар;

– видалити товар.

Для того, щоб почалася робота покупця із сайтом, першим кроком є ретельне наповнення сайту контентом. Цю роботу зазвичай здійснює адміністратор сайту.

Процес наповнення сайту контентом можна розділити на два етапи:

– Додавання категорій: Спочатку адміністратор сайту створює чітку та логічну структуру категорій, до яких будуть відноситися товари, що продаються. Це робить навігацію по сайту зручною та інтуїтивно зрозумілою для покупців.

– Додавання товарів: Після створення категорій адміністратор додає до кожної з них товари. При цьому важливо заповнити всі необхідні поля, такі як назва товару, опис, характеристики, фотографії, ціна тощо. Чим більше інформації буде представлено про товар, тим краще. Це допоможе покупцям зробити вдалий вибір та уникнути розчарувань.

Важливо підкреслити, що наповнення сайту контентом - це постійний процес. Необхідно регулярно додавати нові товари, оновлювати інформацію про вже існуючі, а також видаляти застарілі позиції. Це допоможе сайту залишатися актуальним та привабливим для покупців.

Після того, як сайт буде наповнений контентом, покупці зможуть розпочати його активне використання.

Ось основні кроки, які здійснює покупець на сайті:[4]

– Перегляд товарів: Покупець може переглядати товари за категоріями, за допомогою пошуку або просто прокручуючи сторінки сайту.

– Додавання товарів до корзини: Покупець може додати до корзини будь-який товар, який йому сподобався.

– Оформлення замовлення: Коли покупець закінчить з вибором товарів, він може перейти до оформлення замовлення. На цьому етапі йому потрібно буде

вказати свої контактні дані, адресу доставки, спосіб оплати та інші необхідні відомості.

– Підтвердження замовлення: Після того, як покупець введе всю необхідну інформацію, він отримає підтвердження замовлення на свою електронну пошту.

– Оплата замовлення: Покупцеві буде запропоновано вибрати один із доступних способів оплати замовлення.

– Доставка замовлення: Після оплати замовлення воно буде доставлене за вказаною покупцем адресою.

При замовленні товару слід уникати умов про попередню оплату й здійснювати оплату за фактом отримання товару після його огляду та перевірки. Якщо товар доставляється кур'єром, слід переконатися, що він надає документ, що підтверджує придбання товару. При отриманні товару слід оглянути його та перевірити його роботу в присутності кур'єра або представника пошти[5].

Для кращого розуміння процесу роботи із сайтом, а також для демонстрації можливостей системи, у додатку А представлена діаграма використання. Ця діаграма наочно показує всі кроки, які здійснюють користувачі на сайті, від перегляду товарів до оформлення замовлення.

1.4 Функціональні вимоги до веб-систем інтернет-магазину

Для забезпечення зручного та ефективного користування інтернет-магазином, система повинна відповідати наступним функціональним вимогам:

Користувач:

– Авторизація та реєстрація: Система повинна мати можливість авторизації зареєстрованих користувачів та реєстрації нових[6].

– Кошик: Користувачі повинні мати можливість додавати товари до кошика, переглядати його вміст, міняти кількість товарів та видаляти їх.

– Каталог товарів: Каталог товарів має бути зручно структурований, з можливістю пошуку за назвою, категорією, характеристиками та іншими параметрами.

– Детальні сторінки товарів: Детальні сторінки товарів повинні містити всю необхідну інформацію про товар, включаючи фотографії, опис, характеристики, ціну та можливість покупки.

– Оформлення замовлення: Користувачі повинні мати можливість оформити замовлення, вибравши спосіб доставки, оплати та вказавши контактну інформацію.

– Пошук по сайту: На сайті має бути присутня функція пошуку, що дозволяє користувачам швидко знаходити потрібні товари.

Адміністратор:[7]

– Управління товарами: Адміністратор повинен мати можливість додавати, редагувати та видаляти товари, а також завантажувати їхні фотографії.

– Управління категоріями: Адміністратор повинен мати можливість додавати, редагувати та видаляти категорії товарів.

– Управління користувачами: Адміністратор повинен мати можливість переглядати список користувачів, редагувати їхні дані та видаляти акаунти.

– Управління замовленнями: Адміністратор повинен мати можливість переглядати список замовлень, змінювати їх статус та відправляти повідомлення користувачам.

– Статистика: Система повинна генерувати статистику про продажі, відвідування сайту та інші дані, які можуть бути корисними для адміністратора.

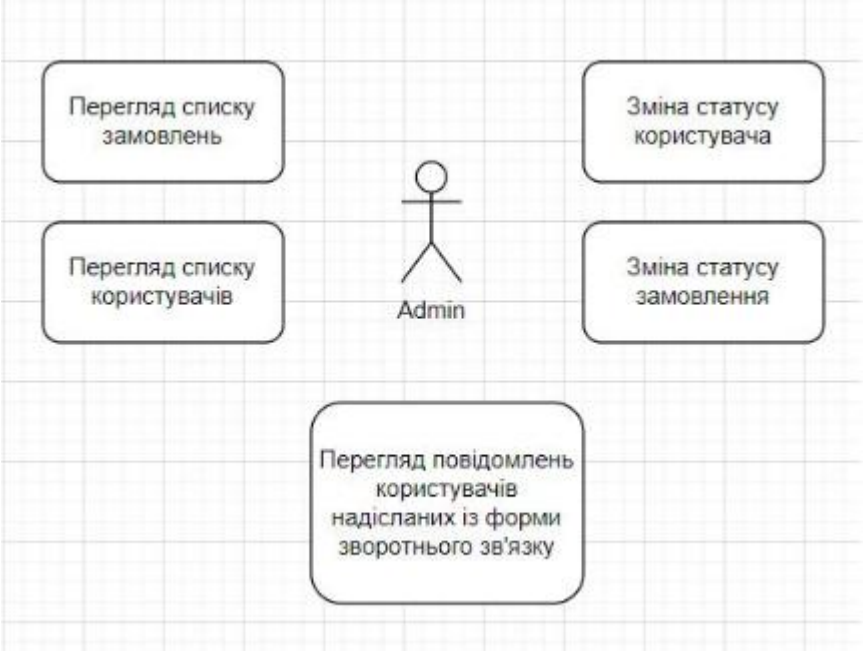


Рисунок 1.7 ─ Схема використання сайтом Адміном

1.5 Нефункціональні вимоги до веб-системи інтернет-магазину

Окрім функціональних вимог, вебсистема інтернет-магазину повинна відповідати наступним нефункціональним вимогам:

– Швидкість роботи: Система повинна швидко реагувати на запити користувачів, забезпечуючи комфортну роботу з сайтом.

– Надійність: Система повинна бути стійкою до збоїв та помилок, гарантуючи безперебійну роботу інтернет-магазину.

– Масштабованість: Система повинна бути здатною витримувати велике навантаження та обробляти велику кількість запитів одночасно.

– Користувацький інтерфейс: Інтерфейс сайту повинен бути зручним, зрозумілим та логічним, забезпечуючи легку навігацію та приємний досвід користування.

– Доступність: Сайт повинен бути доступним для користувачів з будь-якими типами пристроїв та браузерів.

– Безпека: Система повинна відповідати всім вимогам безпеки, захищаючи дані користувачів та запобігаючи шахрайству.

1.6 Вимоги до надійності веб-системи інтернет-магазину

– Відновлення після збоїв: Система повинна мати можливість автоматичного відновлення після збоїв, мінімізуючи час простою.

– Парольний захист: Доступ до системи повинен бути захищений паролем, щоб запобігти несанкціонованому доступу.

– Розмежування прав користувачів: Користувачі повинні мати різні рівні доступу до системи, відповідно до їхніх ролей.

– Захист даних: Всі дані користувачів повинні бути надійно захищені від несанкціонованого доступу, зміни та видалення.

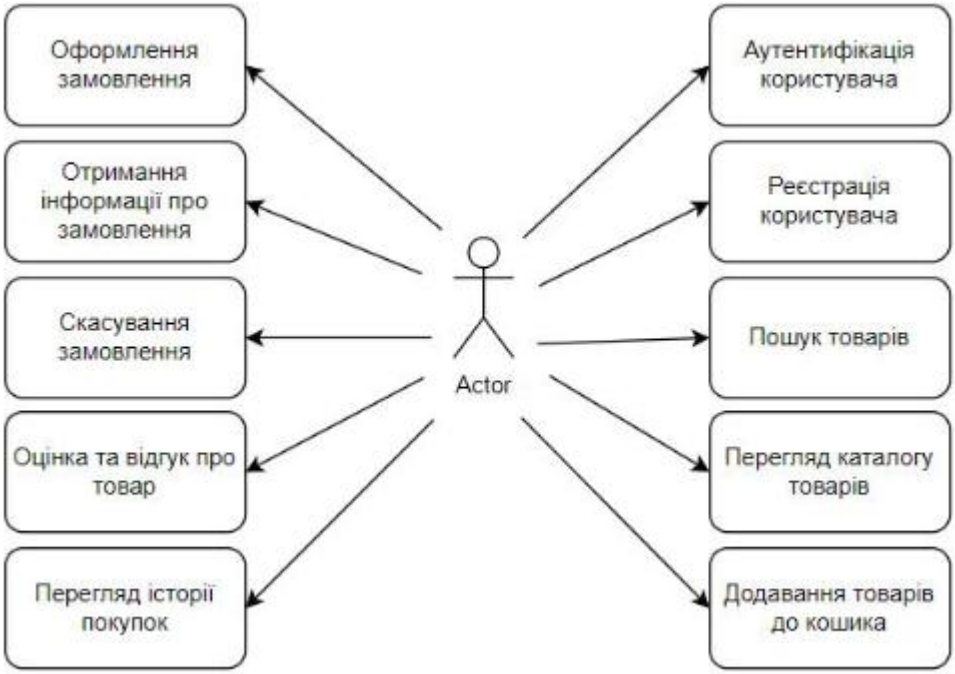


Рисунок 1.8 ─ Схема користування сайтом користувачем

Техніко-економічне обґрунтування

Інтернет-магазини користуються шаленою популярністю, це спричинено великою кількість факторів. У створені такого програмного забезпечення є переваги як для власника так і для покупця[8].

Інтернет завжди на зв'язку, і завжди під рукою. Доступ до інтернет-магазину можливий цілодобово і без вихідних. А це означає, що товари в можуть бути куплені в будь-який зручний момент.

Також інтернет безмежний. Товари можуть продаватися в будь-яку точку світу, де є доступ до мережі інтернет.

Через те що, інтернет-магазини не потребують ніяких приміщень, для огляду їх з боку користувачів, вони можуть мати необмежений асортимент. Це також сприяє економії часу, оскільки будь-який товар можна знайти в одному місці.

З боку власника великою перевагою є те, що при відкритті інтернет-магазина не потрібно наймати працівників, а витрати на оренду скорочуються в кілька разів.

Заходячи в інтернет-магазин, у клієнта немає необхідності: чекати менеджера, стояти в черзі, слухати поради, які йому не потрібні. Якщо покупцеві потрібна консультація, він завжди може звернутися в онлайн підтримку.

Також у віртуальному магазині можуть існувати різні способи оплати товару:

– готівкові при доставці кур'єром;
– поштовий або банківський переказ;
– оплата кредитною карткою;

І покупець може скористатися найоптимальнішим для себе способом.

Технічне завдання на розроблення подано у додатку А.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ВИЗНАЧЕННЯ МЕТОДІВ І ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1 Проектування архітектури програмного забезпечення

Будь-який веб-додаток – це клієнт-серверний додаток, в якому клієнтом виступає браузер, а сервером – веб-сервер.

Основна частина програми, як правило, знаходиться на стороні веб-сервера, який обробляє отримані запити відповідно до бізнес-логікою продукту і формує відповідь, що відправляється користувачеві. На цьому етапі в роботу включається браузер, саме він перетворює отриману відповідь від сервера в графічний інтерфейс, зрозумілий користувачеві.

Архітектура "клієнт-сервер" визначає загальні принципи організації взаємодії в мережі, де є сервери, вузли-постачальники деяких специфічних функцій (сервісів) і клієнти (споживачі цих функцій)[11].

Структура та функціональність Інтернет-магазинів можуть варіюватися залежно від їхніх цілей та завдань. Проте, існують базові компоненти, які є невід'ємною частиною будь-якого такого майданчика.

1. Користувацький інтерфейс:

– Візуальна складова: Це те, що бачить користувач, коли відкриває сайт. Інтерфейс повинен бути зручним, інтуїтивно зрозумілим та навігаційно простим, щоб будь-який користувач Інтернету міг легко знайти те, що шукає[9].

– Система управління контентом (CMS): Це програмне забезпечення, яке дозволяє адміністратору сайту додавати, редагувати та видаляти контент, а також керувати іншими функціями сайту[10].

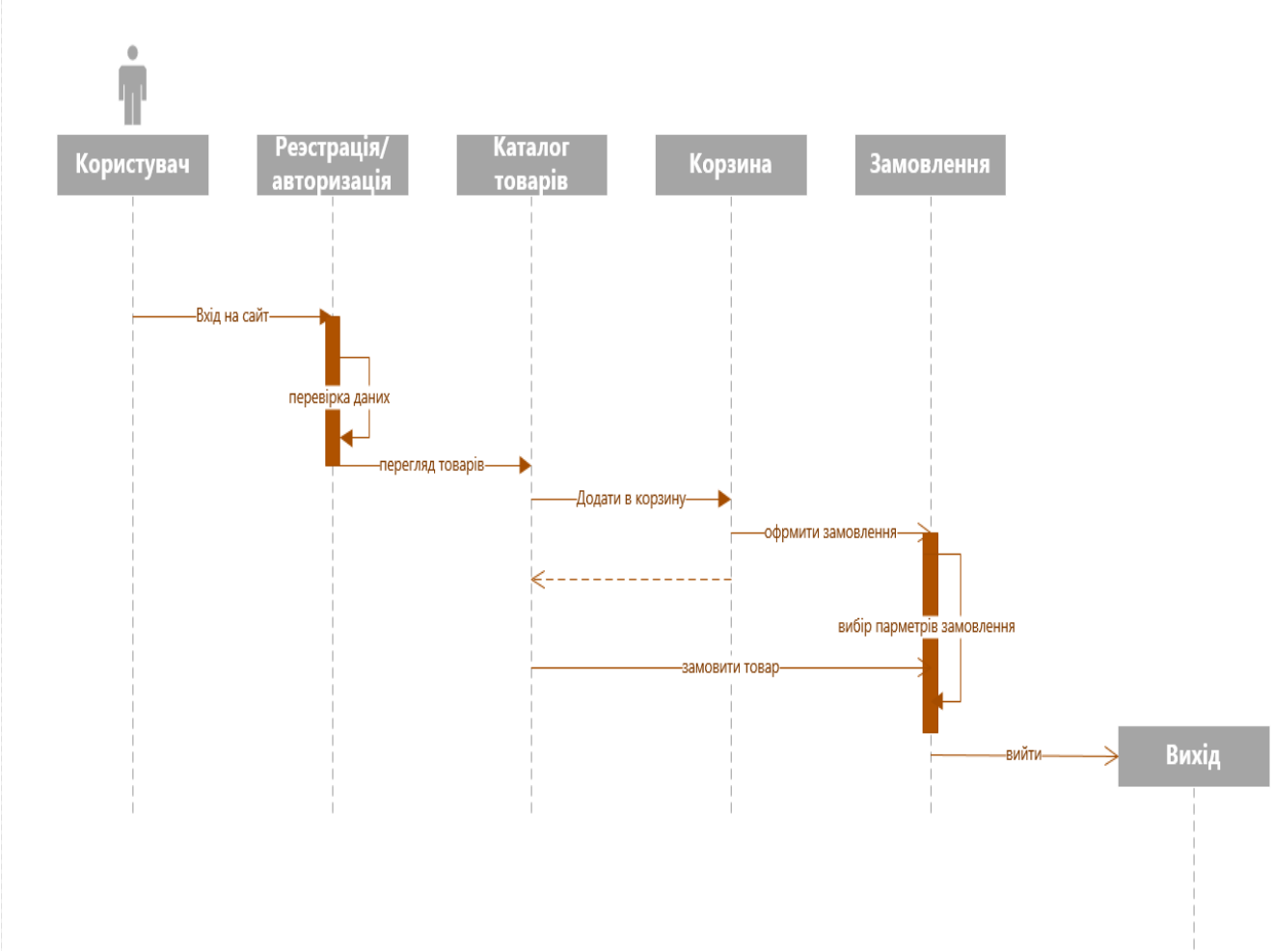


Рисунок 2.1 ─ Діаграма послідовності

2. Серверна частина:

– Логічне ядро: Це програмне забезпечення, яке розміщується на сервері та відповідає за логіку роботи сайту, обробку даних та взаємодію з базою даних[12].

– База даних: Тут зберігається вся інформація про товари, користувачів, замовлення та інші дані, необхідні для роботи сайту[13].

– Платіжний шлюз: Цей компонент відповідає за безпечну обробку платежів за товари, придбані в Інтернет-магазині[14].

– Система аналітики: Ці інструменти дозволяють адміністратору сайту збирати дані про роботу магазину, такі як кількість замовлень, обсяг продажів, поведінка користувачів, популярні категорії товарів тощ[15].

3. Додаткові компоненти:

– Система пошуку: Дозволяє користувачам швидко знаходити потрібні товари за ключовими словами.

– Система відгуків: Дає можливість користувачам ділитися своїми враженнями про товари та магазин.

– Система рекомендацій: Пропонує користувачам товари, які можуть їм сподобатися, на основі їхньої історії покупок та переглядів.

Практичні реалізації такої архітектури називаються клієнт-серверними технологіями.

Будь-який веб-додаток – це клієнт-серверний додаток, в якому клієнтом виступає браузер, а сервером – веб-сервер.

Веб-сервер – це сервер, що приймає HTTP-запити від клієнтів і видає їм HTTP-відповіді. Веб-сервером називають як програмне забезпечення, яке виконує функції веб-сервера, так і безпосередньо комп'ютер, на якому це програмне забезпечення працює. Найбільш поширеними видами ПО веб-серверів є Apache, IIS і NGINX. На веб-сервері функціонує тестоване додаток, яке може бути реалізовано з застосуванням найрізноманітніших мов програмування: PHP, Python, Ruby, Java, Perl та ін.

Для реалізації сайту було обрано архітектуру MVC.

MVC розшифровується як модель-уявлення-контролер (від англ. Model-view-controller). Це спосіб організації коду, який передбачає виділення блоків, що відповідають за вирішення різних завдань. Один блок відповідає за дані додатки, інший відповідає за зовнішній вигляд, а третій контролює роботу додатка.

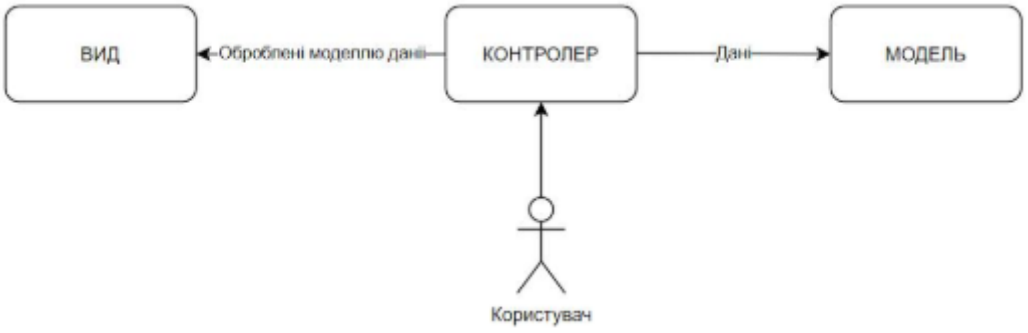


Рисунок 2.2 ─ Схема контролера MVC

Контролер керує запитами користувача одержувані у вигляді запитів HTTP GET або POST, коли користувач натискає на елементи інтерфейсу для виконання різних дій. Його основна функція – викликати і координувати дію необхідних ресурсів і об'єктів, потрібних для виконання дій, що задаються користувачем. Зазвичай контролер викликає відповідну модель для задачі і вибирає відповідний вид.

Модель – це дані і правила, які використовуються для роботи з даними, які представляють концепцію управління додатком. У будь-якому додатку вся структура моделюється як дані, які обробляються певним чином. Модель дає контролеру уявлення даних, які запросив користувача.

Модель містить найбільш важливу частину логіки застосування, логіки, яка вирішує завдання. Контролер містить в основному організаційну логіку для самого додатка.

Вид забезпечує різні способи представлення даних, які отримані з моделі. Він може бути шаблоном, який заповнюється даними. Може бути кілька різних видів, і контролер вибирає, який підходить якнайкраще для поточної ситуації.

Веб додаток зазвичай складається з набору контролерів, моделей і видів. Контролер може бути влаштований як основний, який отримує всі запити і викликає інші контролери для виконання дій в залежності від ситуації.

Перевага концепції MVC – це чіткий поділ логіки подання (інтерфейсу користувача) і логіки програми. Крім ізолювання видів від логіки додатки, концепція MVC істотно зменшує складність великих додатків. Код виходить набагато більш структурованим, і, тим самим, полегшується підтримка, тестування і повторне використання рішень[16].

Клієнт-серверна архітектура: фундамент веб-додатків

Клієнт-серверний підхід ось уже багато років залишається одним із найпоширеніших у розробці веб-додатків. Він ґрунтується на чіткому розподілі ролей: "логіка" програми розміщується на сервері, а користувач взаємодіє з нею за допомогою клієнта.

Різноманіття реалізацій:

Існує безліч варіацій клієнт-серверної архітектури, кожна з яких має свої особливості та підходить для певних задач. Ось декілька поширених моделей:

Клієнт-сервер з одним сервером: Найпростіший варіант, де всі запити обробляє один центральний сервер. Ця модель може бути економною, але не масштабується добре при зростанні навантаження.

Клієнт-сервер з декількома серверами: Розподіл навантаження між декількома серверами робить систему більш стійкою та масштабованою. Ця модель підходить для сайтів з високим трафіком.

Клієнт-сервер з кластером серверів: Об'єднання декількох серверів у кластер забезпечує ще більшу надійність та продуктивність. Ця модель використовується для сайтів з дуже високим трафіком.

Клієнт-сервер з кешуванням на стороні клієнта: Зберігання часто використовуваних даних на стороні клієнта може значно зменшити навантаження на сервер та покращити час відгуку. Ця модель добре підходить для сайтів з динамічним контентом.

Переваги клієнт-серверної архітектури:

Масштабованість: Легко додавати нові сервери для обробки зростаючого навантаження.

Надійність: Відмова одного сервера не призводить до повної зупинки роботи системи.

Безпека: Логіка програми розміщена на сервері, що ускладнює несанкціонований доступ до даних.

Простота розробки: Можна використовувати різні мови програмування та технології для розробки клієнтської та серверної частин.

Недоліки клієнт-серверної архітектури

Складність: Розробка та підтримка серверної інфраструктури може бути складнішою, ніж у випадку з однорівневими системами.

Витрати: Використання декількох серверів потребує більших витрат на обладнання та обслуговування.

Залежність від мережі: Якість роботи системи залежить від стійкості та швидкості роботи мережі.

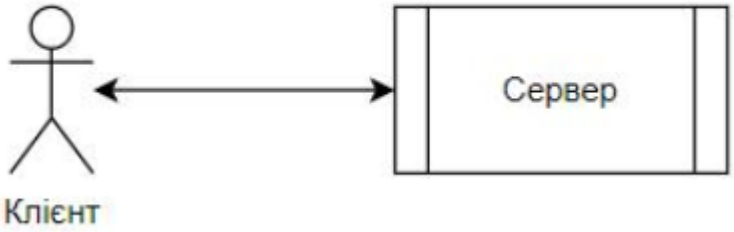


Рисунок 2.3 ─ Клієнт-серверна архітектура

2.2 Проектування інтерфейсу програмного забезпечення

Розробка інтерфейсу є дуже важливим етапом, оскільки він впливає на думку покупців про сайт, а цей фактор відповідно впливає на бажання користувачів відвідувати його знову. Саме від інтерфейсу залежить, чи зможе користувач швидко виконати покупки, чи зрозуміє покроково весь процес оформлення замовлення.

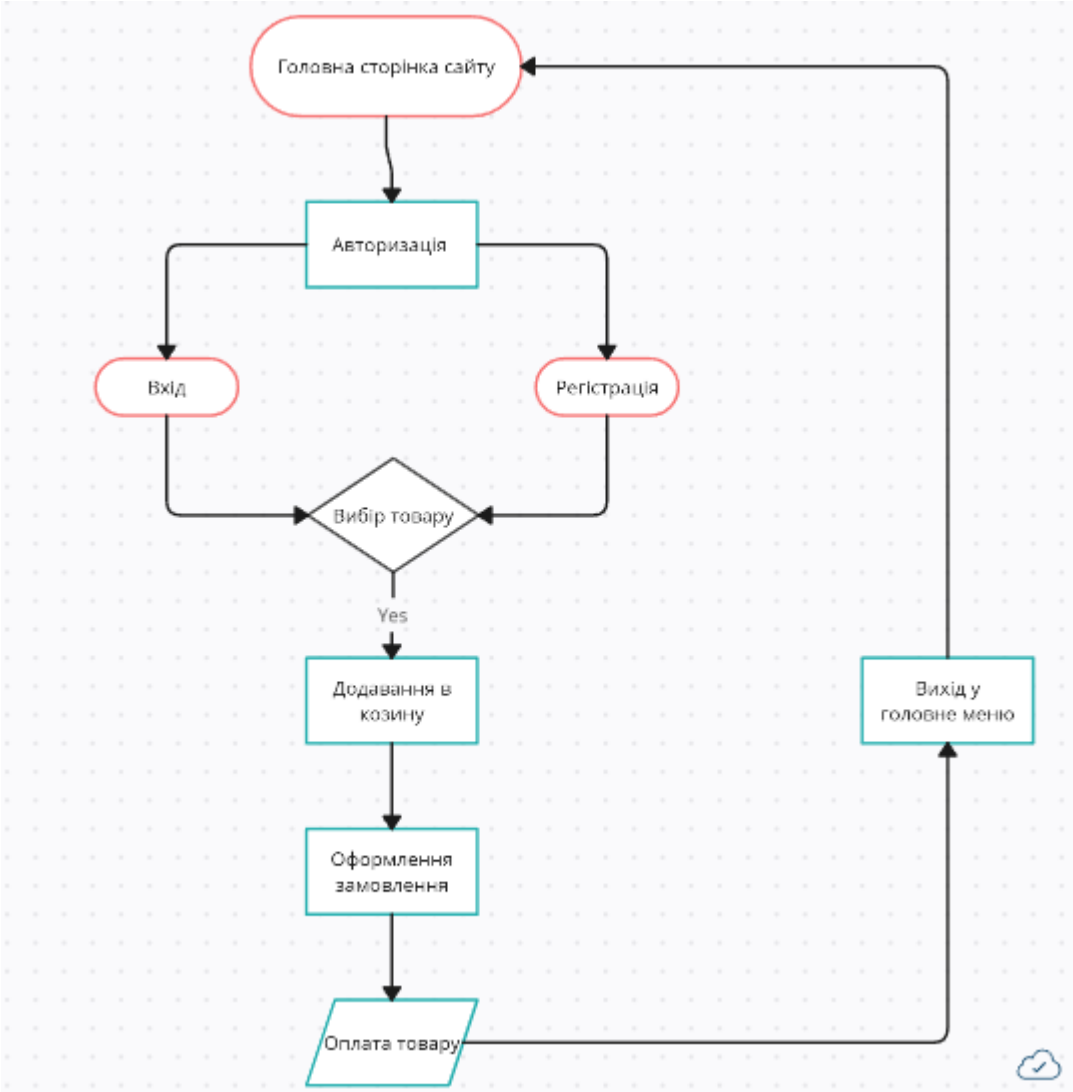


Рисунок 2.4 — Блок схема сайту

Важливою складовою будь-якої сторінки – це є зручна навігація по сторінках сайту. Для цього необхідно створити меню, яке міститиме основні посилання.

Сторінки які буде мати сайт:

– Перегляд наданих товарів;
– можливість зареєструватися;
– можливість авторизації;
– можливість додавання товару в корзину;
– можливість оформлення замовлення;
– можливість додавання товару для адміністратора;
– можливість зміни товару для адміністратора;
– можливість додавання категорії;

– сторінка переліку всіх категорій та товарів для адміністратора.

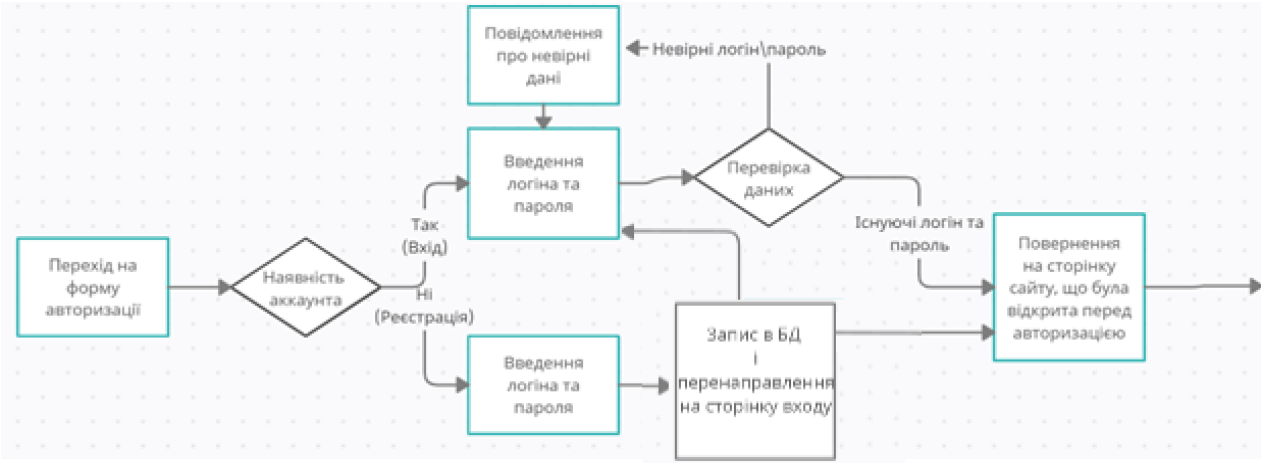


Рисунок 2.5 ─ Діаграма алгоритму авторизації

На рисунку 2.6 зображено сторінки каталогу. Тут міститься всі товари та короткий опис про кожен із них. Також міститься зображення товару, та посилання, де можна переглянути додаткову інформацію про кожен товар.

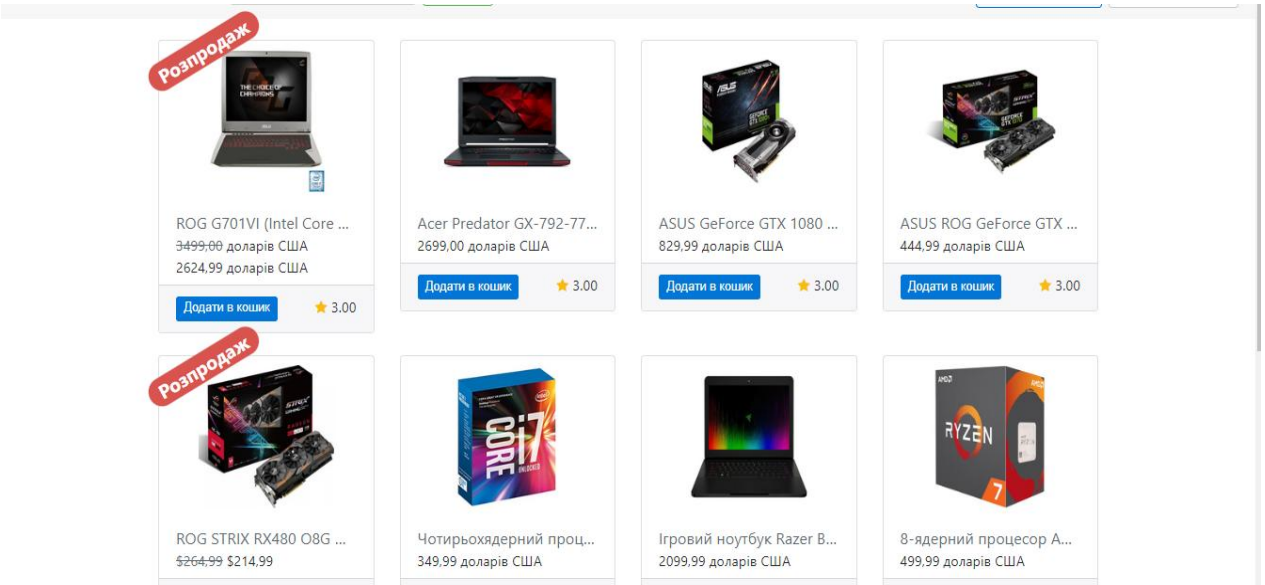


Рисунок 2.6 – сторінка каталогу

На рисунку 2.7 зображено головну сторінку сайту. Користувач може переглянути повну інформацію про товар. Та додати в корзину

Рисунок 2.7 – сторінка перегляду одного товару

Корзина користувача містить список зі всіх товарів, що були додані раніше користувачем. Тут міститься основна інформація та зображення. А також можливість переходу на сторінку оформлення замовлення (рисунок 2.8).

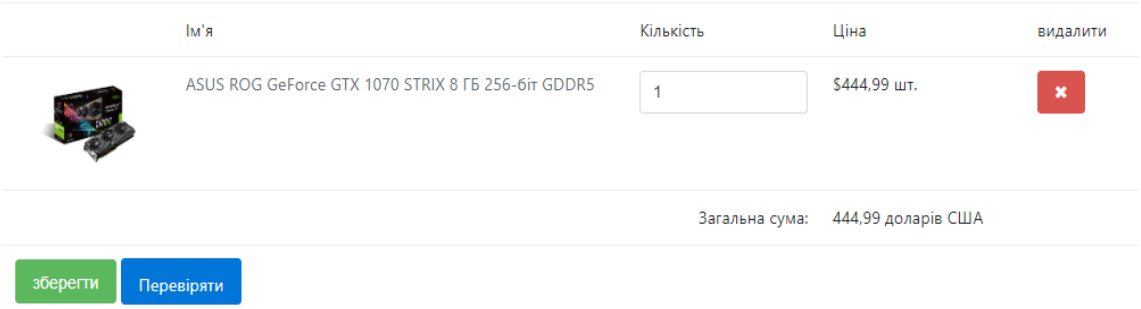


Рисунок 2.8 – сторінка із кошиком

На рисунку 2.9 показано сторінка із оформленням замовлення. Для додавання замовлення необхідно заповнити поля. Права частина сторінки – це список товарів, що було додано в замовлення.

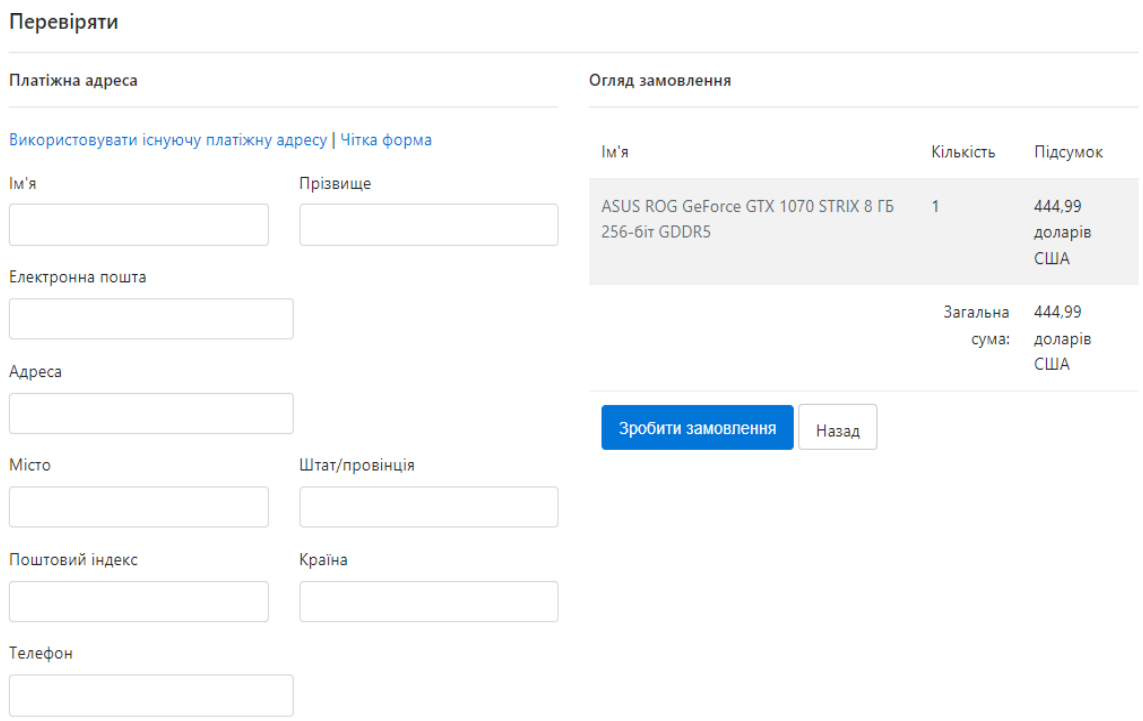


Рисунок 2.9 Оформлення замовлення

Щодо адміністративної частини, то сторінка переліку категорій та товарів повині бути реалізовані у вигляді таблиць. Тому для цих сторінок робити макети на даному етапі не є необхідністю.

2.3 Детальне проектування програмного забезпечення

При вході на сайт користувачу доступні деякі функції. Він може переглядати наявні товари та кожен окремо. На основній сторінці виводяться всі товари, а також зображення до них. На кожному товарі є посилання, за яким можна перейти, те переглянути його детальніше.

Щоб продовжити роботу із сайтом потрібно авторизуватися або зареєструватися, якщо користувач не створював попередньо акаунт. Після авторизації він може зайти, як адміністратор або покупець.

Якщо користувач зайшов як покупець, він може переглядати кошик, додавати до неї товари, та оформлювати замовлення.

Додати товар у кошик можна на сторінці перегляду цього товару.

Після чого він буде одразу доданий до кошика. Перейти до кошику можна, скориставшись головним меню.

В кошику доступне посилання на оформлення замовлення. Після натиснення цієї кнопки користувач переходить автоматично на форму оформлення замовлення, де йому необхідно вказати певні параметри. Після чого замовлення додається. Результатом цієї дії буде чек, де вказано основні дані по замовленню, а також перелік товарів, які користувач додав в нього.

Якщо ж користувач входить за ролю адміністратора, він зможе виконувати тільки дій пов’язані з наповненням сторінок контентом.

Для того, щоб переглянути наявні списки категорій та товарів, потрібно скористатися головним меню. На кожній з цих сторінок є сервіси, що дозволять відредагувати товар або видалити. При видалені товару, користувач автоматично перейде на сторінку з підтвердженням видалення товару.

2.4 Проєктування бази даних

Однією з найголовніших моментів є створення моделі бази данних, адже від неї залежатиме весь проєкт.

База даних − це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів[17].

Головною перевагою бази даних є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам, які використовуються для баз даних, можна легко знаходити необхідні дані всього за декілька секунд. Також в базі даних існує певний взаємозв'язок інформації: зміна в одному рядку може спричинити зміни в інших рядках − це допомагає працювати з інформацією простіше і швидше.

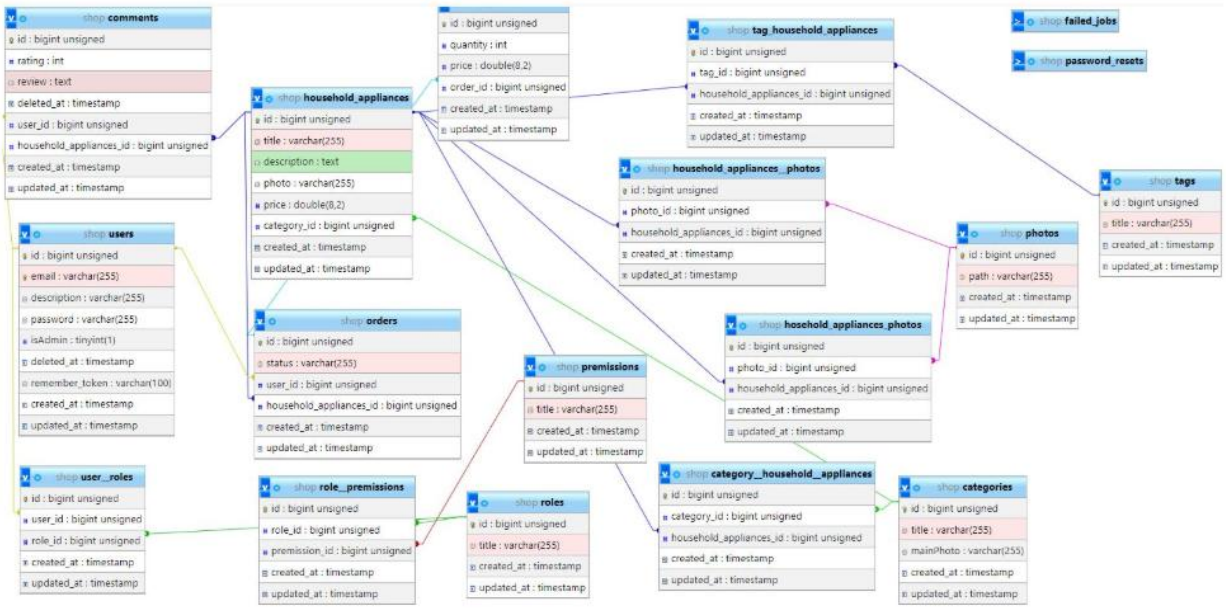


Рисунок 2.10 ─ Модель бази данних

2.5 Аналіз та вибір технологій і методів реалізації програмного забезпечення.

C# − об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML.[18]

Microsoft SQL Server − система управління базами даних, яка розробляється корпорацією Microsoft. Як сервер даних виконує головну функцію по збереженню та наданню даних у відповідь на запити інших застосунків, які можуть виконуватися як на тому ж самому сервері, так і у мережі. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних [19].

Entity Framework − це набір технологій, які підтримують розробку програмно-орієнтованих на дані програмних додатків. Вони моделюють сутності, взаємо-

зв'язки та логіку бізнес-проблем, які вони вирішують, а також повинні працювати з движками, що використовуються для зберігання та отримання даних [20].

ASP.NET − технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft.NET і розвитком старішої технології Microsoft ASP. На цей час останньою версією цієї технології є ASP.NET Core 2.0. ASP.NET зовні багато в чому зберігає схожість із старішою технологією ASP, що дозволяє розробникам відносно легко перейти на ASP.NET. У той же час внутрішній устрій ASP.NET істотно відрізняється від ASP, оскільки вона заснована на платформі. NET і, отже, використовує всі нові можливості, що надаються цією платформою [21].

HTML (англ. HyperText Markup Language − мова розмітки гіпертексту) − це мова тегів, засобами якої здійснюється розмічання веб-сторінок для мережі Інтернет. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті й передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа [22].

CSS (англ. Cascading Style Sheets – каскадні таблиці стилів) − це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів[23].

Bootstrap − це безкоштовний набір інструментів з відкритим кодом, призначений для створення веб-сайтів та веб-додатків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних веб-сайтів і веб-додатків. Bootstrap − це

клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері[24].

Висновок: в ході виконання другого розділу було проведено детальний аналіз структури веб-додатку, описано структуру та моделі бази даних. Також спроєктовано клієнтську та серверну частину додатку. Після цього розроблено дизайн і описано технології для створення сайту.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Структура програмних модулів

Було створено програмний модуль, що розподілений на три частини, а саме керування користувацькою частиною, керування адміністративою частиною, а також на частину, що пов’язана із акаунтами користувачів.

Це було реалізовано за допомогою 3-ох контролерів: AccountController, AdminController, HomeController.

AccountController – містить дії які відповдають за реєстрацію та авторизацію користувачів в системі. Ці операції здійснюються за допомогою бібліотеки ASP.NET Core Identity.

ASP.NET Core Identity:

- Це API, що підтримує функцію входу в інтерфейс користувача (UI).
- Керує користувачами, паролями, даними профілю, ролями, заявками, маркерами, підтвердженням електронною поштою тощо.

Користувачі можуть створити обліковий запис з інформацією для входу, що зберігається в Identity.

Для зручності розмежування функціоналу користувачів, було створено два окремих контролера для користувача та адміністратора.

Контролер адміністратора має такий функціонал:

- додавання, редагування, вилалення товарів;
- додавання, редагування, видалення категорій;
- додавання зображень.

Доступ до адміністративної частини обмежений.

HomeController відповідає за частину функціоналу, що здійснює покупець. У ньому реалізовано декілька дій, які доступні для всіх користувачів. Проте додавання, перегляд корзини, оформлення замовлень додавання коментарів – це ті дії, які може виконувати тільки зареєстрований користувач.

3.2 Реалізація БД

Було створено наступні таблиці :

- користувач;
- товар;
- категорія;
- кошик;
- замовлення;
- частина замовлення;
- частина кошик;
- зображення.

При роботі з кошиком замовленням виникає зв’язок багато до багатьох. Тому було реалізовано ще додаткові таблиці :частина замовлення та частина кошика. Так само для товару можна додавати декілька зображень, тому варто реалізувати окрему таблицю для зображень.

Оскільки Entity framework використовує принцип Code first, то спочатку будуються моделі, на основі яких і створюється база даних. Все це прописується в додатковому класі ApplicationContext. Просте після першого створення, база даних сама не оновлюються, якщо моделі змінюються. Щоб не втратити дані, і не перезаписувати базу даних кожного разу, було використано міграції баз даних.

Міграції виконуються програмно за допомогою інструменту міграцій. При виклику інструменту міграції з вказуванням бажаної версії схеми, інструмент автоматично застосовує або відкочує міграції в правильній послідовності аж поки не приведе базу даних до бажаного стану.

Більшість інструментів міграції стараються мінімізувати вплив змін схеми на наявні дані всередині бази. Проте, загалом збереження даних не гарантується, бо зміни схеми такі як видалення колонки таблиці можуть знищувати дані.

3.3 Реалізація сайту

Першим етапом при створенні програмного забезпечення є створення Моделей. Між моделями необхідно реалізувати зв'язки, які автоматично згенеруються в базі даних. Наприклад було реалізовано модель регестрації авторизації на сайті

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace aspCart.Web.Models.AccountViewModels
{
      public class LoginViewModel
      {
            [Required]
     [EmailAddress]
     public string Email { get; set; }

     [Required]
     [DataType(DataType.Password)]
     public string Password { get; set; }

     [Display(Name = "Remember me?")]
     public bool RememberMe { get; set; }
     }
}
```

В даному випадку встановлюється зв’язок один до одного, тобто один логін може мати один пароль.

Після створення моделей та зв’язків між ними необхідно створити контролери, в яких будуть міститися всі основні функції сайту.

Наступним кроком буде додавання нового методу в контролер та створення представлення. Наприклад функція додавання товару після форми заповнення всіх полів (рисунок 3.1):

```
using Microsoft.AspNetCore.Mvc.Rendering;
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace aspCart.Web.Areas.Admin.Models.Catalog
{
    public class ProductCreateOrUpdateModel
    {
    public ProductCreateOrUpdateModel()
    {
        Published = true;
        ActiveTab = "info";

        StockQuantity = 1000;
        MinimumStockQuantity = 0;
        NotifyForQuantityBelow = 1;
        MinimumCartQuantity = 1;
        MaximumCartQuantity = StockQuantity;
    }

    public string ActiveTab { get; set; }

    public Guid Id { get; set; }

    [Required]
    [StringLength(255, MinimumLength = 1)]
    [Display(Name = "Product Name")]
    public string Name { get; set; }
    public string Description { get; set; }

    [Required]
    [Range(0, 1000000)]
    [Display(Name = "Product Price")]
    public decimal? Price { get; set; }

    [Range(0, 1000000)]
    [Display(Name = "Special Product Price")]
    public decimal? SpecialPrice { get; set; }

    [Display(Name = "Special Price Start Date")]
            [DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:dd-
      MMM-yyyy}")]
    public DateTime? SpecialPriceStartDate { get; set; }
```

```
[Display(Name = "Special Price EndDate")]
        [DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:dd-
  MMM-yyyy}")]
public DateTime? SpecialPriceEndDate { get; set; }

[Display(Name = "Stock Quantity")]
public int? StockQuantity { get; set; }

[Display(Name = "Minimum Stock Qty")]
public int? MinimumStockQuantity { get; set; }

[Display(Name = "Notify for quantity below")]
public int? NotifyForQuantityBelow { get; set; }

[Display(Name = "Display Availability")]
public bool DisplayAvailability { get; set; }

[Display(Name = "Minimum Cart Qty")]
public int? MinimumCartQuantity { get; set; }

[Display(Name = "Maximum Cart Qty")]
public int? MaximumCartQuantity { get; set; }

[Display(Name = "SEO Url")]
[RegularExpression(@"^[a-zA-Z0-9]+(-[a-zA-Z0-9]+)*$", ErrorMessage = "URL
must only contains alphanumeric [a-z A-Z 0-9] and dash [-] e.g. abc-123-D45")]
public string SeoUrl { get; set; }

[Display(Name = "Meta Tag Title")]
public string MetaTitle { get; set; }

[Display(Name = "Meta Tag Keywords")]
public string MetaKeywords { get; set; }

[Display(Name = "Meta Tag Description")]
public string MetaDescription { get; set; }

[Display(Name = "Category")]
public List<string> CategoryIds { get; set; }

public SelectList CategorySelectList { get; set; }

public List<ImageModel> Images { get; set; }

public List<string> ImageIds { get; set; }
```

```
    public List<string> ImageSortOrder { get; set; }

    [Display(Name = "Manufacturer")]
    public List<string> ManufacturerIds { get; set; }

    public SelectList ManufacturerSelectList { get; set; }

    public List<ProductSpecificationModel> Specifications { get; set; }

    public SelectList SpecificationKeySelectList { get; set; }

    public bool Published { get; set; }
    public DateTime DateAdded { get; set; }
}
    }
```

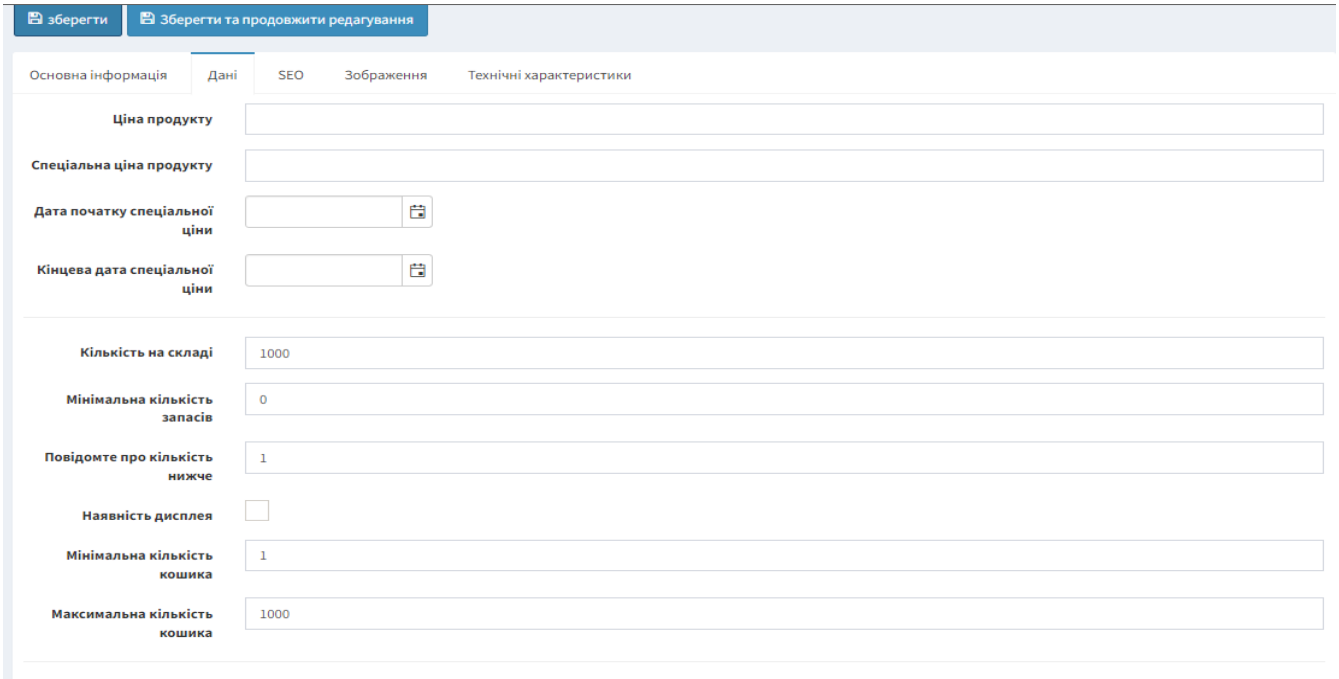


Рисунок 3.1 – Додавання нового товару

При додаванні товару список категорій автоматично генерується з раніше-доданих, тому не потрібно її вводити вручну.

Редагування та видалення продукту здійснюється наступним чином:

```
[HttpPost]
    public async Task<IActionResult> EditProduct(Product pr)
    {
        db.Products.Update(pr);
        await db.SaveChangesAsync();
        return RedirectToAction("Index");
```

```
        }

        [HttpGet]
        [ActionName("DeleteProduct")]
        public async Task<IActionResult> ConfirmProductDelete(int? id)
        {
            if (id != null)
            {
                Product product = await db.Products.FirstOrDefaultAsync(p => p.Id ==
id);
                if (product != null)
                    return View(product);
            }
            return NotFound();
        }

        [HttpPost]
        public async Task<IActionResult> DeleteProduct(int? id)
        {
            if (id != null)
            {
                Product user = await db.Products.FirstOrDefaultAsync(p => p.Id == id);
                if (user != null)
                {
                    db.Products.Remove(user);
                    await db.SaveChangesAsync();
                    return RedirectToAction("Index");
                }
            }
            return NotFound();
              }
```

Редагування здійснюється просто перезаписом об’єкту в базу даних, після чого зміни зберігаються. Для реалізації видалення необхідно 2 методи. Один з них для підтвердження

За таким же принципом розроблялися методи видалення, додавання та редагування категорій.

Ще одним цікавим моментом є додавання замовлень покупцями, оскільки одночасно в форму потрібно передати об’єкт кошика для виведення даних і об’єкт замовлення для заповнення.

З цією метою було створено окремо модель, яка призначена тільки для цієї дії, записувати через неї дані в базу даних неможливо.

Модель містить об’єкт корзини та замовлення. В результаті є можливість одночасного виводу і оформлення замовлення (рисунок 3.2)

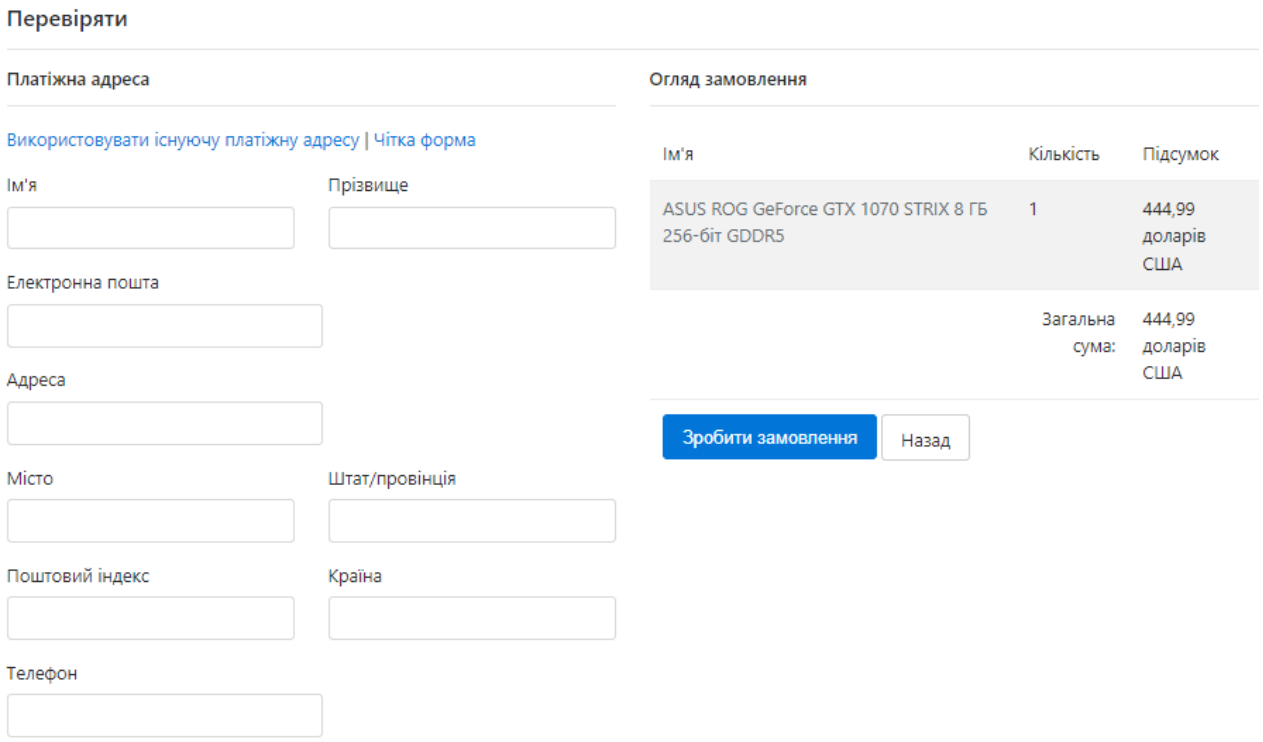


Рисунок 3.2 ─ Оформлення замовлення

Результатом реалізації сайту, є готовий продукт, який потрібно протестувати.

Код програми подано у додатку Б.

3.4 Інструкція з експлуатації

При вході на сайт користувач бачить початкову сторінку зі всіма доступними товарами. Про кожен товар виведено основну інформацію та зображення. Користувач може здійснювати пошук по товарах, за такими категоріями такі як ноутбуки, миші, процесори, відео карти.(рисунок 3.3)

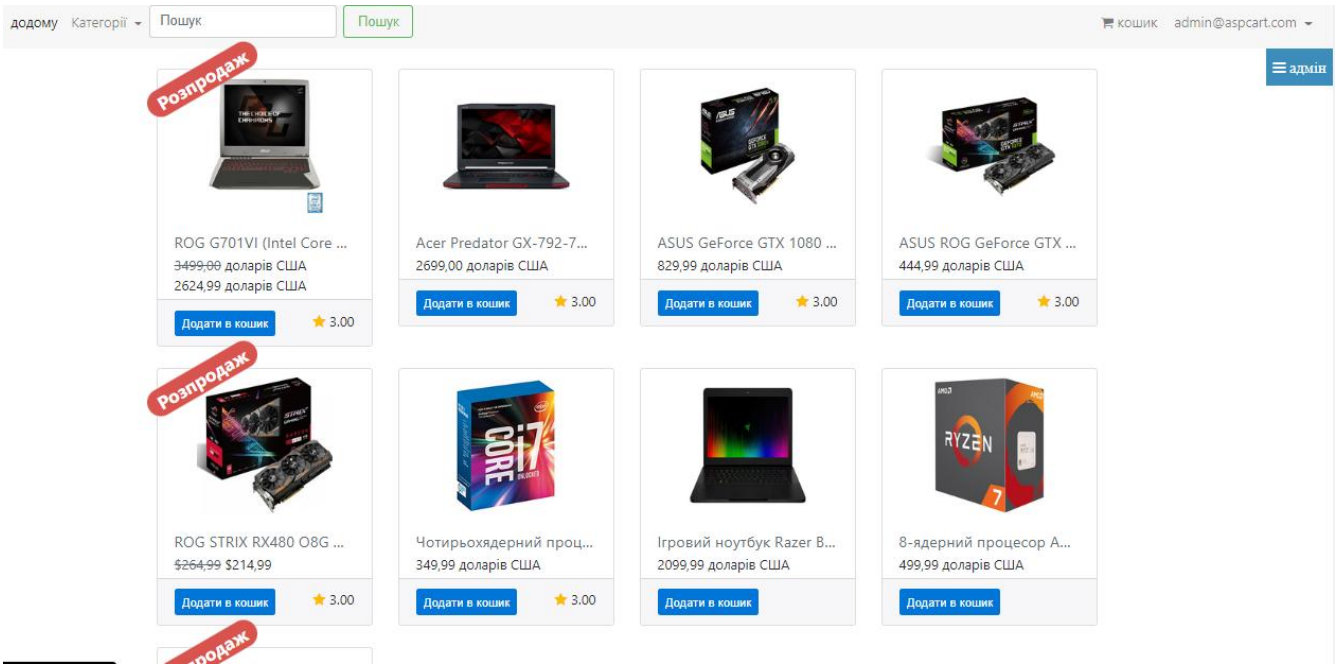


Рисунок 3.3 – Головна сторінка сайту

Покупець може оглянути кожен товар окремо (рисунок 3.4).

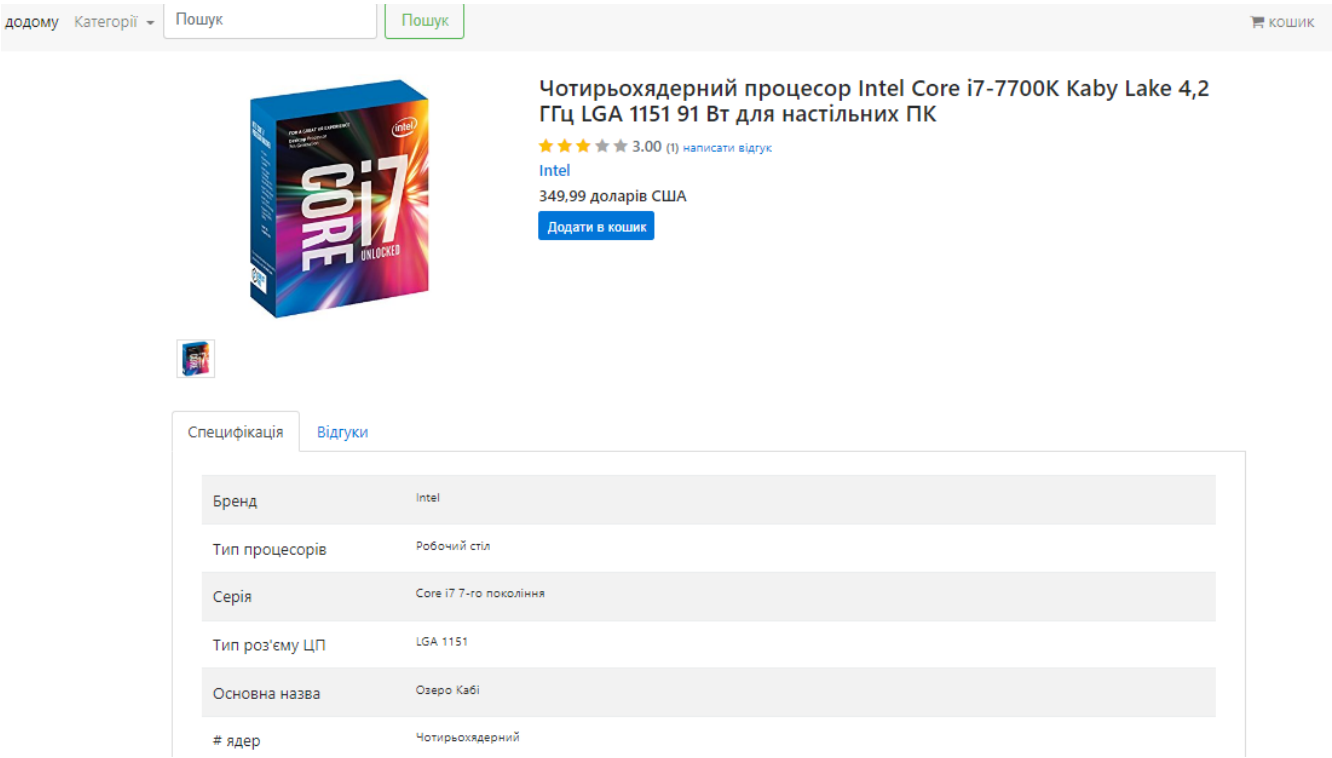


Рисунок 3.4 – Сторінка перегляду товару

Тут показано декілька зображень товарів, а також виводиться вся необхідна інформація про товар. Покупець який зареєструвався може додати товар до

кошику. Після цього він побачить кошик з переліком товарів який він обрав(рисунок 3.5). Відповідно для незареєстрованого користувача ця можливість відсутня.

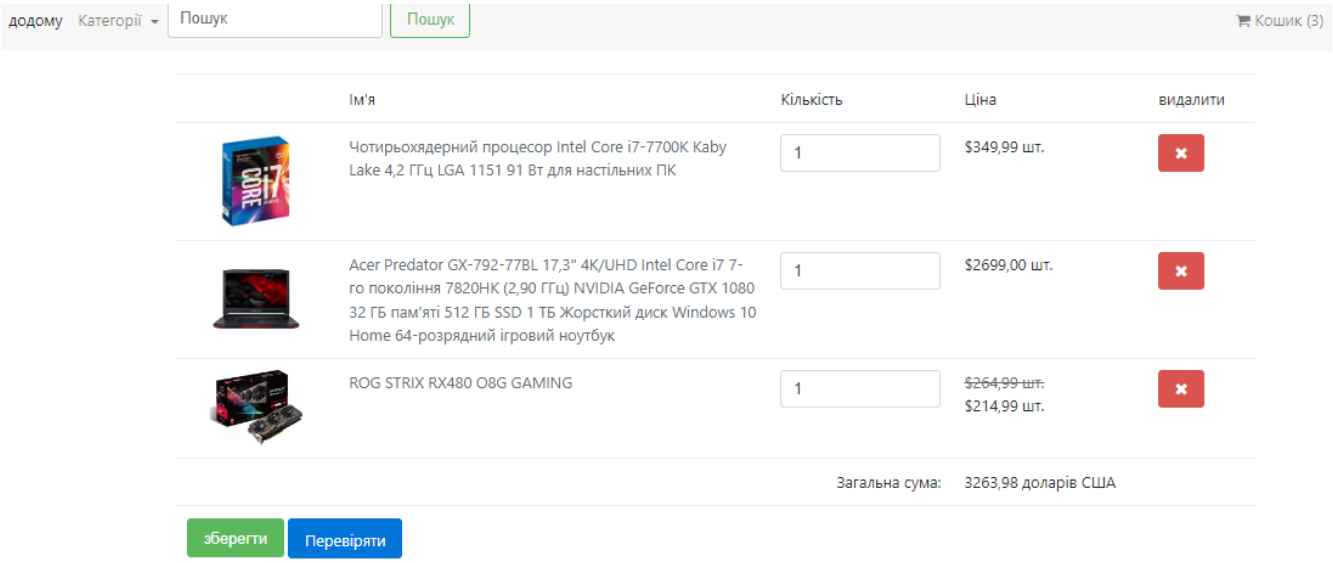


Рисунок 3.5 – Кошик користувача

Вкладка для оформлення замовлення, тут користувач вводить інформацію для доставки товару (рисунок 3.6)

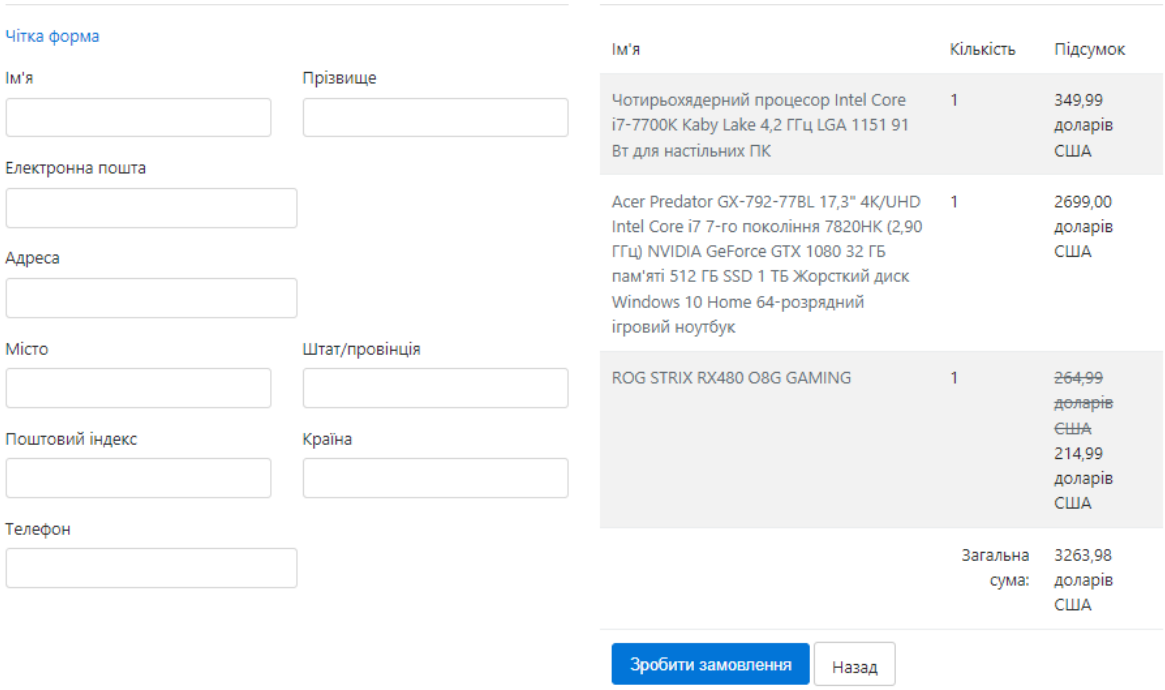


Рисунок 3.6 Оформлення замовлення

Після того як користувач ввів інформацію про доставка товару, він побачить сторінку з переліком товарів який він обрав та чек.

Висновок

У даному підрозділі було визначено структуру програмних модулів, реалізовано базу даних та зв’язки між її таблицями. Під час цього етапу розроблено програмний продукт, який реалізовано за клієнт-серверною архітектурою з використанням MVC. Також було надано рекомендії користувачам, щодо експлуатації даного сайту.

3.5 Тестування програмного забезпечення

Тестування програмного забезпечення – це обов’язковий етап перед введенням проекту в експлуатацію. Необхідно перевірити всі складові системи на коректність роботи. Перевірити чи не виникає ніяких помилок при додаванні даних, при виведенні їх сукупності користувачу.

Потрібно провести ручне тестування. Це допоможе зрозуміти які помилки можуть виникнути при введенні чи маніпуляції з даними на самому сайті.

При реєстрації та авторизації необхідно перевірити всі можливі сценарії некоректного введення даних користувачем (рисунок 3.7).

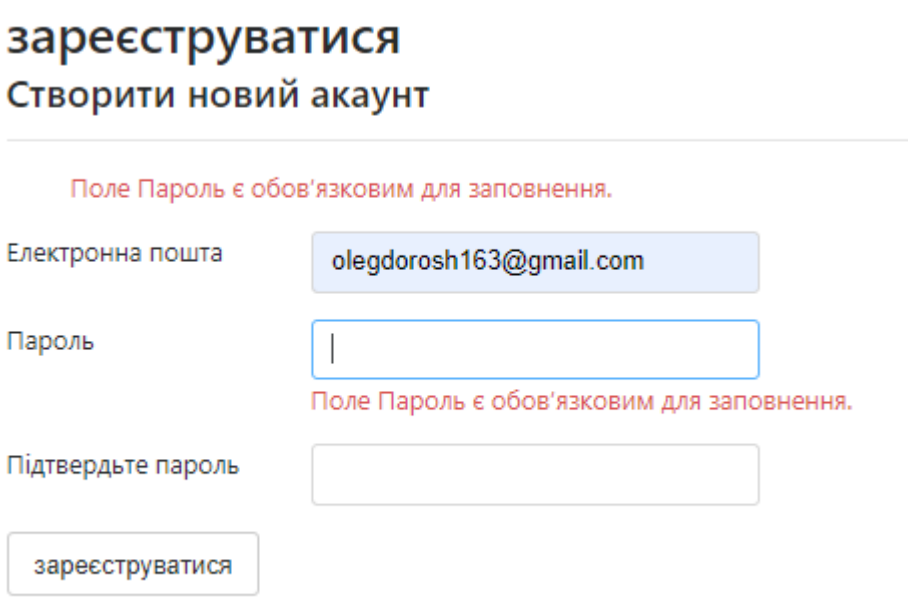


Рисунок 3.7 – Валідація даних при реєстрації

При реєстрації працює валідація, вона автоматично не допускає збереження даних, що є некоректними

При авторизації користувач проходить аутентифікацію, тобто при веденні логіну, пароль повинен співпадати із тим, що вже записаний у базі даних. Тому перевірка чи все функціонує правильно є дуже важливою, оскільки в інакшому випадку користувачу виведе помилку, якої точно не повино бути(рисунок 3.8).

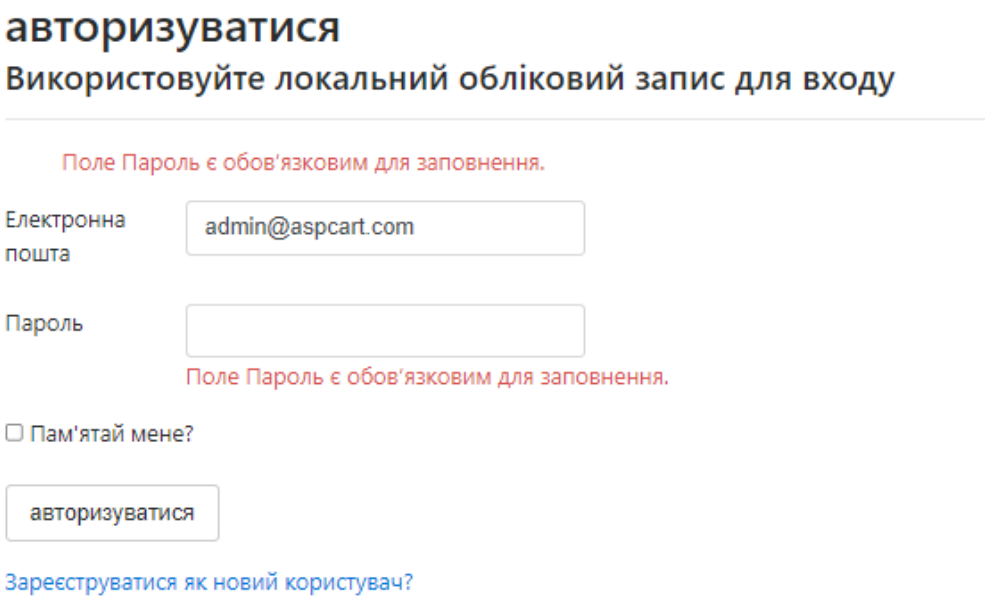


Рисунок 3.8 – Валідація даних при авторизації

Як видно з рисунка 3.8, валідація на даній сторінці присутня.

Для того, щоб здійснити модульне тестування, було додано додаткову програму у головний проєкт.

Декілька модульних тестів із цієї програми:

За допомогою цих тестів було перевірено, чи правильно зроблене показ початкової сторінки, додавання до кошику, додавання замовлення, перегляд інформації про замовлення.

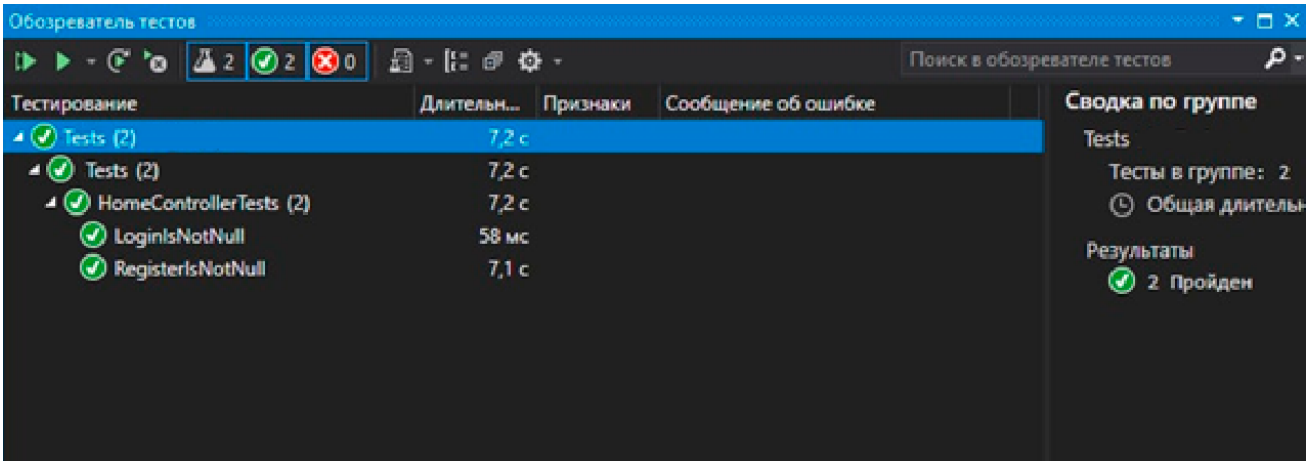


Рисунок 3.9 – Результати модульного тестування

На рисунку 3.9 зображено результати модульного тестування.

Висновок

В даному підрозділі було здійснено тестування програмного ззбезпечення. Було проведено ручне та модульне тестування, кожне з яких підтвердило готовність проекту до використання.

ВИСНОВКИ

Отже, після чотирьох етапів розробки проекту, було отримано інтернет-магазин з продажу побутової техніки. Даний вебресурс реалізує функціонал обробки контенту на сторінках та оформлення замовлень покупцями. Весь процес здійснення замовлення поділяється на декілька етапів: пошуку потрібно товару, додавання в корзину, та оформлення замовлення, із введенням необхідних даних.

Першим етапом було дослідження предметної області. Під час цього етапу було проведений комплексний аналіз усіх складових предметної області, опрацювання інформації, щодо ресурсів-аналогів. Було визначено їх недоліки та переваги, та на основі цих даних сформовано оптимальне технічне завдання. Також було визначено різного плану вимоги до вебресурсу.

Другим етапом було проектування програмного забезпечення. Цей етап є важливим і охоплює великий спектр робіт, оскільки необхідно все продумати до найменших деталей, адже на основі цих даних далі реалізується проєкт. Було спроектовано архітектуру програмного забезпечення, базу даних, інтерфейс а також проведене детальне проєктування всіх компонентів системи. Вкінці етапу було визначено що проєкт реалізовує клієнт-серверну архітектуру, за технологією MVC, мовою програмування C# та фреймворку ASP.Net Core.

Третій етап – це реалізація сайту. Було створено моделі, на основі яких будувалася база даних з використанням міграцій. Після цього створено контролери та представлення. Результатом етапу був готовий проєкт, який залишилося протестувати.

Четвертий етап проводиться перед введенням в експлуатацію, тому було проведено детальне опрацювання можливих виключень. На даному кроці використовувалися модульне та ручне тестування, що дозволило перевірити систему як з внутрішньої сторони, тобто функціонал, так і з зовнішньої сторони, користувачем якої є покупець.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Корзина онлайн магазину. URL: https://simferopil.rozrobka-sajtiv.in.ua/rozrobka-sajtiv-blog/funkcziyi-koshyka-v-internet-magazyni/ (дата звернення: 10.01.2024).
2. Головна сторінка магазину KVshop. URL: https://kvshop.com.ua (дата звернення: 11.01.2024).
3. Головна сторінка магазину Rozetka. URL: https://rozetka.com.ua (дата звернення: 11.01.2024).
4. Алгоритм замовлення товару в інтернет магазині. URL: https://www.i-law.kiev.ua/%D0%BF%D1%80%D0%B8%D0%B4%D0%B1%D0%B0%D0%BD%D0%BD%D1%8F-%D1%82%D0%BE%D0%B2%D0%B0%D1%80%D1%96%D0%B2-%D1%87%D0%B5%D1%80%D0%B5%D0%B7-%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82/ https://rozetka.com.ua (дата звернення: 12.01.2024).
5. Як правильно замовити товар в інтернеті. URL: https://spar.ua/blogs/internet-pokupki-krok-za-krokom-vid-a-do-ya (дата звернення: 12.01.2024).
6. Авторизація і реєстрація URL: https://qagroup.com.ua/publications/autentyfikatciia-i-avtoryzatciia/ (дата звернення: 14.01.2024).
7. Функції адміністратора URL: https://biznesogoliks.ru/administrator-internet-magazina/ (дата звернення: 15.01.2024).
8. Популярність інтернет магазинів URL: https://sitniks.ua/blog_post/uspishnyj-internet-magazyn-klyuchovi-strategiyi-dlya-dosyagnennya-uspihu/ (дата звернення: 15.01.2024).
9. Візуальна складова URL: https://seonomad.net/blog/dyzayn-internet-mahazynu-sekrety-efektyvnoho-vizualnoho-vyhlyadu (дата звернення: 19.01.2024).
10. Система керування контекстом URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%

BC%D0%B0_%D0%BA%D0%B5%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_%D0%B2%D0%BC%D1%96%D1%81%D1%82%D0%BE%D0%BC (дата звернення: 19.01.2024).

11. Клієнт серверна архітектура URL: https://training.qatestlab.com/blog/technical-articles/client-server-architecture/ (дата звернення: 06.02.2024).

12. Логічне ядро URL: https://rx-name.ua/blog/yak-kilkist-yader-u-servernomu-protsesori-vyznachaye-jogo-efektyvnist (дата звернення: 06.02.2024).

13. База данних URL: https://www.oracle.com/ua/database/what-is-database/ (дата звернення: 06.02.2024).

14. Платіжний шлюз URL: https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%96%D0%B6%D0%BD%D0%B8%D0%B9_%D1%88%D0%BB%D1%8E%D0%B7 (дата звернення: 06.02.2024).

15. Система Аналітики URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D0%B8%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D1%96%D1%82%D0%B8%D0%BA (дата звернення: 06.02.2024).

16. MVC URL: https://brander.ua/technologies/mvc (дата звернення: 10.02.2024).

17. База данних URL: https://www.oracle.com/ua/database/what-is-database/ (дата звернення: 14.02.2024).

18. Мова програмування C# URL: https://uk.wikipedia.org/wiki/C_Sharp (дата звернення: 16.02.2024).

19. Microsoft SQL Server URL: https://uk.wikipedia.org/wiki/Microsoft_SQL_Server (дата звернення: 17.02.2024).

20. Entity Framework URL: https://learn.microsoft.com/ru-ru/ef/ (дата звернення: 18.02.2024).

21. ASP.NET технологія URL: https://uk.wikipedia.org/wiki/ASP.NET (дата звернення: 19.02.2024).

22. HTML URL: https://ru.wikipedia.org/wiki/HTML (дата звернення: 19.02.2024).

23. CSS URL: https://ru.wikipedia.org/wiki/CSS (дата звернення: 19.02.2024).

24. Bootstrap URL: https://uk.wikipedia.org/wiki/Bootstrap (дата звернення: 19.02.2024).

25. Безпека інтернет магазину. URL: https://rau.ua/novyni/bezopasnost-internet-magazina-chto-i-pochemu-nuzhno-zashhishhat/. (дата звернення: 01.03.2023).

26. Postman URL: https://www.postman.com/

27. Сервіс-орієнтована архітектура. URL: https://aws.amazon.com/whatis/service-oriented-architecture/ (дата звернення: 03.03.2023).

28. Клієнт серверна архітектура. URL: https://quizlet.com/Тема-13-web-технології-Клієнт-серверна-архітектура-flash-cards/. (дата звернення: 04.03.2023).

29. MVC – шаблони. URL: https://brander.ua/technologies/mvc. (дата звернення: 05.03.2023).

30. Офіційний сайт MySql. URL: https://www.mysql.com/. (дата звернення:06.05.2023)

.

ДОДАТОК А

(обов’язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

1. Мета проєкту

Створення інтернет-магазину з широким асортиментом побутової техніки, зручним інтерфейсом та можливістю онлайн-замовлення з доставкою по Україні.

2. Цільова аудиторія

- Користувачі Інтернету, які шукають побутову техніку;
- Жителі України;
- Люди, які цінують зручність та економлять час;
- Користувачі, які шукають вигідні ціни та акції.

3. Функціональні вимоги

Каталог товарів:

- Широкий асортимент побутової техніки відомих брендів;
- Детальні описи та характеристики товарів;
- Якісні фотографії та відеоогляди;
- Можливість порівняння товарів;
- Наявність фільтрів та пошуку по каталогу;
- Інформація про акції та знижки.

Кошик замовлень:

- Зручне додавання та видалення товарів;
- Зміна кількості товарів;
- Розрахунок вартості замовлення з урахуванням доставки;
- Можливість збереження кошика для подальших покупок.

Замовлення та оплата:

- Проста та зрозуміла форма замовлення;
- Різні способи оплати (банківська картка, готівка при отриманні, онлайн-оплата);
- Підтвердження замовлення на email та телефон;
- Можливість відстеження статусу замовлення.

Доставка:

- Різні способи доставки (Нова Пошта, Укрпошта, самовивіз);
- Розрахунок вартості доставки;
- Терміни доставки;
- Можливість відстеження замовлення.

Особистий кабінет:

- Реєстрація та авторизація користувачів;
- Перегляд історії замовлень;
- Зміна особистих даних;
- Зберігання побажань та порівнянь.

Інші функції:

- Новини та статті про побутову техніку;
- Відгуки про товари;
- Можливість задати питання онлайн-консультанту;
- Програма лояльності для постійних клієнтів.

4. Дизайн та ергономіка

- Сучасний та лаконічний дизайн;
- Зручна навігація по сайту;
- Швидке завантаження сторінок;
- Адаптивний дизайн для мобільних пристроїв.

5. Технічні вимоги

- Надійна та безпечна платформа для інтернет-магазину;
- Висока продуктивність сайту;
- Захист персональних даних користувачів;
- SEO-оптимізація сайту.

6. Додаткові вимоги

- Можливість інтеграції з 1С;
- Можливість підключення системи онлайн-оплати;
- Наявність API для доступу до даних сайту.

7. Терміни виконання

Розробка інтернет-магазину повинна бути виконана протягом 6 місяців.

ДОДАТОК Б

(обов’язковий)

ПРОГРАМНИЙ КОД

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Threading.Tasks;

namespace aspCart.Core.Domain.Catalog
{
    public class Category
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public Guid Id { get; set; }

        [StringLength(255, MinimumLength = 1)]
        public string Name { get; set; }
        public string Description { get; set; }

        public string SeoUrl { get; set; }
        public string MetaTitle { get; set; }
        public string MetaKeywords { get; set; }
        public string MetaDescription { get; set; }

        public bool Published { get; set; }
        public DateTime DateAdded { get; set; }
        public DateTime DateModified { get; set; }

        public Guid ParentCategoryId { get; set; }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace clock_shop.Models
{
    public class Order
    {
        public int Id { get; set; }
        public double Sum{ get; set; }
        public List<OrderItem> Items { get; set; }
        public string Delivery { get; set; }
        public DateTime date { get; set; }
        public string Card { get; set; }
        public string svv { get; set; }
        public string date_card { get; set; }
        public string post_num { get; set; }

        public string UserId { get; set; }
        public User User { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Logging;
using computer_electronics_shop.Models;
using computer_electronics_shop.ViewModels;

namespace computer_electronics_shop.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;
        private ApplicationContext db;
        private readonly UserManager<User> _userManager;
        public HomeController(ApplicationContext context, UserManager<User> man)
        {
            db = context;
            _userManager = man;
        }

        public IActionResult Index()
        {
            return View(db.Products.ToList());
        }
        public IActionResult ViewProduct(int id)
        {
            return View(db.Products.Include(k=>k.Images).Include(u=>u.Category).Where(c=>c.Id==id).FirstOrDefault());
        }
        public async Task<IActionResult> AddToBasket(int id)
        {Basket basket= db.Baskets.Where(c => c.UserId == _userManager.GetUserId(User)).FirstOrDefault();
            if (basket == null) { basket = new Basket { User = await _userManager.GetUserAsync(User) }; db.Baskets.Add(basket); }
            BasketItem basketItem = new BasketItem { Product = db.Products.Where(c => c.Id == id).FirstOrDefault(), Basket = basket
};
            db.BasketItems.Add(basketItem);
            await db.SaveChangesAsync();
            return RedirectToAction("Basket");
        }
        [HttpPost]
        public async Task<IActionResult> Index(string Search)
        {

            if (!String.IsNullOrEmpty(Search))
            {
                List<Product> products = await db.Products.Include(u => u.Images).Include(c=>c.Category).Where(s =>
s.Category.Name.Contains(Search) || s.Material.Contains(Search) || s.Name.Contains(Search) ).ToListAsync();
                return View(products);
            }
            else
                return View(await db.Products.ToListAsync());
        }
        public async Task<IActionResult> Basket()
        {
            Basket basket = db.Baskets.Include(t=>t.Items).ThenInclude(c=>c.Product).Where(c => c.UserId ==
_userManager.GetUserId(User)).FirstOrDefault();
            return View(basket);
        }
        public async Task<IActionResult> AddOrder(int id)
        {
            List<String> modell = new List<string>();
            ViewBag.Delivery = modell;
            Basket basket = db.Baskets.Include(t => t.Items).ThenInclude(c => c.Product).Where(c => c.Id == id).FirstOrDefault();
```

```
            double sum = 0;
            foreach (var item in basket.Items) sum += item.Product.Price;
            OrderBasket ob = new OrderBasket();
            ViewBag.Sum = sum;
            ob.basket = basket;
            ob.order = new Order();
            return View(ob);
        }
        [HttpPost]
        public async Task<IActionResult> AddOrder(OrderBasket ob)
        {

            Basket basket = db.Baskets.Include(t => t.Items).Where(c => c.Id == ob.basket.Id).FirstOrDefault();
            Order order = ob.order;
            order.User = basket.User;
            db.Orders.Add(order);
           foreach (var ord in basket.Items)
            {
                db.OrderItems.Add(new OrderItem { BasketItem = ord, count = 1, Order = order });
            }
            await db.SaveChangesAsync();
            return RedirectToAction("OrderInform", new { id = order.Id });
        }
        public async Task<IActionResult> OrderInform(int id)
        {
            Order order= db.Orders.Include(t => t.Items).ThenInclude(c => c.BasketItem).ThenInclude(p=>p.Product).Where(c => c.Id
== id).FirstOrDefault();

            return View(order);
        }
        public IActionResult Privacy()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}



using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using computer_electronics_shop.Models;

namespace computer_electronics_shop.Controllers
{
    public class AdminController : Controller
    {
        private ApplicationContext db;
        IWebHostEnvironment _appPath;
        public AdminController(ApplicationContext context, IWebHostEnvironment appEnvironment)
        {
            db = context;
```

```
    _appPath = appEnvironment;
}
public async Task<IActionResult> Index()
{
    return View(await db.Products.ToListAsync());
}
public IActionResult AddCategory()
{
    return View();
}
public IActionResult AddProduct()
{
    List<String> modell = new List<string>();
    ViewBag.Sex = modell;
    ViewBag.Cat = db.Categories.ToList();
    return View();
}
[HttpPost]
public async Task<IActionResult> AddProduct(Product product, List<IFormFile> upFiles)
{
    string img_path = "";
    foreach(var file in upFiles )
    if (file != null)
    {
        string path = "/Img/" + file.FileName;
        using (var fileStream = new FileStream(_appPath.WebRootPath + path, FileMode.Create))
        {
            await file.CopyToAsync(fileStream);
        }
            if (img_path == "") img_path = path;
            Image image = new Image { Product = product, Path = path };
            db.Images.Add(image);

    }
    product.Img_Path = img_path;


    db.Products.Add(product);
    await db.SaveChangesAsync();
    return RedirectToAction("Index");

}
[HttpPost]
public async Task<IActionResult> AddCategory(Category product)
{
    db.Categories.Add(product);
    await db.SaveChangesAsync();
    return RedirectToAction("Index");
}

public async Task<IActionResult> EditCategory(int? id)
{

    if (id != null)
    {
            return View(await db.Categories.FirstOrDefaultAsync(p => p.Id == id));

    }
    return NotFound();
}
[HttpPost]
public async Task<IActionResult> EditCategory(Category cat)
{
    db.Categories.Update(cat);
    await db.SaveChangesAsync();
    return RedirectToAction("Index");
```

```
}

public async Task<IActionResult> AllCategory()
{
    return View(await db.Categories.ToListAsync());
}

public async Task<IActionResult> EditProduct(int? id)
{
    ViewBag.Cat = db.Categories.ToList();
    List<String> modell = new List<string>();
    ViewBag.Sex = modell;
    if (id != null)
    {
        return View(await db.Products.FirstOrDefaultAsync(p => p.Id == id));

    }
    return NotFound();
}
[HttpPost]
public async Task<IActionResult> EditProduct(Product pr)
{
    db.Products.Update(pr);
    await db.SaveChangesAsync();
    return RedirectToAction("Index");
}

[HttpGet]
[ActionName("DeleteProduct")]
public async Task<IActionResult> ConfirmProductDelete(int? id)
{
    if (id != null)
    {
        Product product = await db.Products.FirstOrDefaultAsync(p => p.Id == id);
        if (product != null)
            return View(product);
    }
    return NotFound();
}

[HttpPost]
public async Task<IActionResult> DeleteProduct(int? id)
{
    if (id != null)
    {
        Product user = await db.Products.FirstOrDefaultAsync(p => p.Id == id);
        if (user != null)
        {
            db.Products.Remove(user);
            await db.SaveChangesAsync();
            return RedirectToAction("Index");
        }
    }
    return NotFound();
}

[HttpGet]
[ActionName("DeleteCategory")]
public async Task<IActionResult> ConfirmCategoryDelete(int? id)
{
    if (id != null)
    {
        Category product = await db.Categories.FirstOrDefaultAsync(p => p.Id == id);
        if (product != null)
            return View(product);
    }
```

```
            return NotFound();
        }

        [HttpPost]
        public async Task<IActionResult> DeleteCategory(int? id)
        {
            if (id != null)
            {
                Category user = await db.Categories.FirstOrDefaultAsync(p => p.Id == id);
                if (user != null)
                {
                    db.Categories.Remove(user);
                    await db.SaveChangesAsync();
                    return RedirectToAction("Index");
                }           }
            return NotFound();
        }
    }
}
@model computer_electronics_shop.ViewModels.OrderBasket
<div class="container">
    <div class="row">
        <div class="col-sm-6 col-lg-6 col-md-6">
            <form asp-action="addorder" asp-controller="home">
                <h5>Оформити замовлення</h5>
                <div class="form-group">
                    <label asp-for="order.Sum" class="control-label">До оплати</label>
                    <input type="text" asp-for="order.Sum" class="form-control" value="@ViewBag.Sum" style=" pointer-events: none;"
/>
                </div>
                <div class="form-group">
                    <label asp-for="order.adress" class="control-label">Адреса</label>
                    <input type="text" asp-for="order.adress" class="form-control" />
                </div>
                <div class="form-group">
                    <                    <input type="text" asp-for="order.post_num" class="form-control" />
                </div>
                <div class="form-group">
                    <input type="text" asp-for="order.date" class="form-control" value="@DateTime.Now" />
                </div>
                <div class="form-group">
                    <label asp-for="order.Delivery" class="control-label">Доставка</label>
                    <select asp-for="order.Delivery" class="form-control" asp-items="@(new SelectList(ViewBag.Delivery))"></select>
                </div>
                <input type="text" asp-for="basket.Id" class="form-control" value="@Model.basket.Id" hidden />

                <div class="form-group row ">
                    <label asp-for="order.Card" class="control-label">Номер карти</label>
                    <input type="text" asp-for="order.Card" class="form-control valid"  aria-invalid="false" mask="9999-9999-9999-9999"
/>
                    <label asp-for="order.svv" class="control-label col-lg-6 col-md-6">Cvv </label>
                    <label asp-for="order.date_card" class="control-label col-lg-6 col-md-6">Дійсна до</label>
                    <input type="text" asp-for="order.svv" class="form-control valid col-lg-6 col-md-6"  aria-invalid="false" mask="999"
/>

                    <input type="text" asp-for="order.date_card" class="form-control valid col-lg-6 col-md-6" aria-invalid="false"
mask="99/99" />
                </div>
                <div class="form-group">
                    <input type="submit" value="Замовити" class="btn btn-default btn2" />
                </div>

            </form>
        </div>
        <div class="col-sm-6 col-lg-6 col-md-6">
            <p>
```

```
            <button class="btn btn-primary link1" type="button" data-toggle="collapse" data-target="#collapseExample" aria-
expanded="false" aria-controls="collapseExample">
                Показати товари замовлення
            </button>
        </p>
        <div class="collapse" id="collapseExample">


            @foreach (var item in Model.basket.Items)
            {
                <div class="row basketItem">
                    <div class="col-sm-8 col-lg-8 col-md-8 ">
                        <h4>@item.Product.Name</h4>

                        <h4>Ціна: @item.Product.Price грн.</h4>

                    </div>

                    <div class="col-sm-4 col-lg-4 col-md-4 basketImg">
                        <img src="@item.Product.Img_Path" alt="Sample Image" />

                    </div>

                </div>}


        </div>
    </div>

  </div>
</div>
@section scripts{

  <script src="https://cdn.jsdelivr.net/npm/jquery.maskedinput@1.4.1/src/jquery.maskedinput.min.js"
type="text/javascript"></script>
  <script type="text/javascript">
    $(function () {
        $('[mask]').each(function (e) {
            $(this).mask($(this).attr('mask'));
        });
    })</script>
```

ДОДАТОК В

(обов’язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

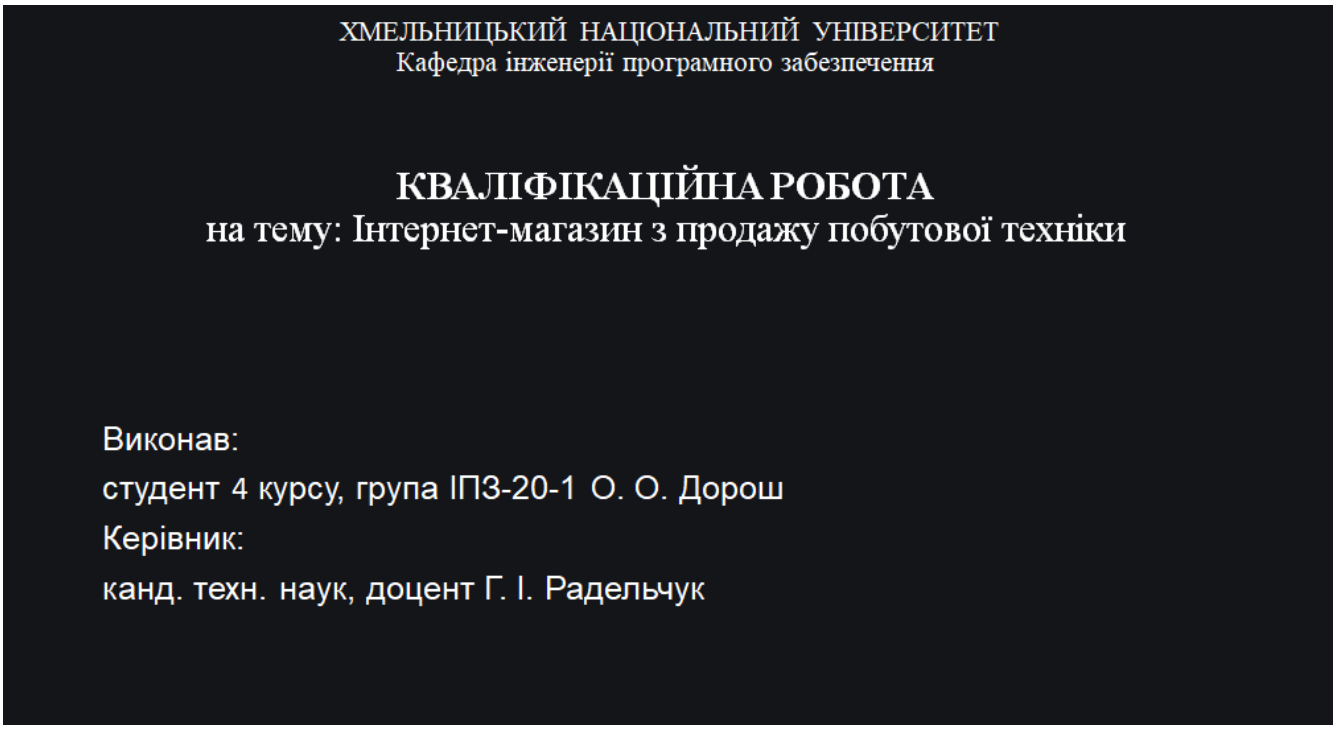


Рисунок В.1 – Слайд 1

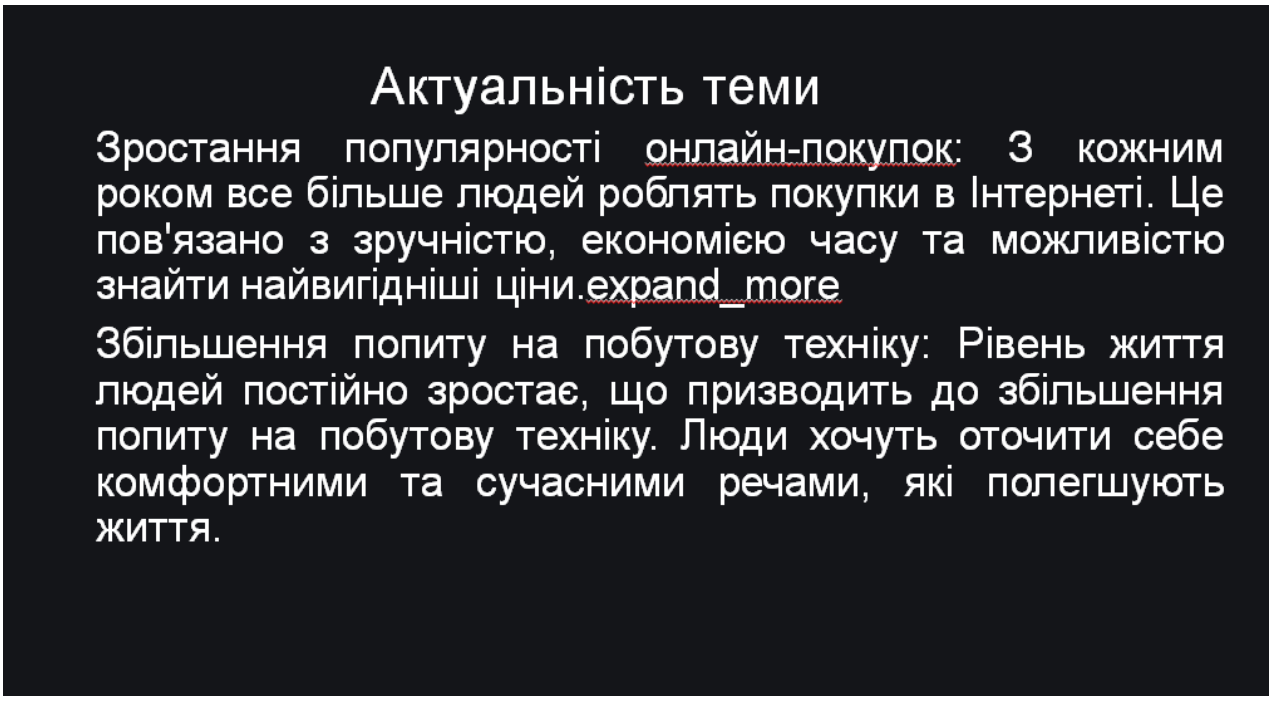


Рисунок В.2 – Слайд 2

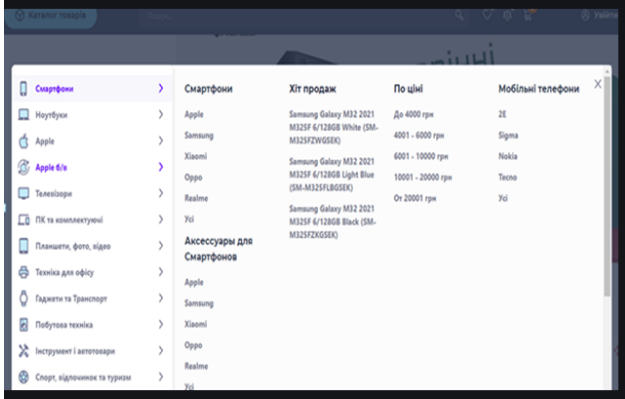
Мета кваліфікаційної роботи - створення вебзастосунку для продажу побутової техніки з використанням веб-технологій.

Для досягнення мети роботи поставлені такі завдання:

- визначити та проаналізувати особливості створення та функціонування вебзастосунків для продажу товарів;
- ознайомитися з наявними відкритими рішеннями;
- вибрати архітектурні та програмні рішення для розроблення вебзастосунку;
- розробити архітектуру вебзастосунку та програмно реалізувати його для продажу побутової техніки відповідно до обраних засобів;
- провести тестування застосунку та виправити виявлені помилки.

Рисунок В.3 – Слайд 3

Аналіз існуючих подібних розробок



Аналізуючи рішення, що існують у мережі Інтернет та відносяться до даної предметної області, було зрозуміло, що існує велике різноманіття вебзастосунків, які стосуються продажу різної електроніки, наприклад:

kvshop.com.ua (інтуїтивний інтерфейс, пошук товарів за багатьма фільтрами, яскраве виділення основних характеристик товарів тощо).

Рисунок В.4 – Слайд 4

Рисунок В.5 – Слайд 5

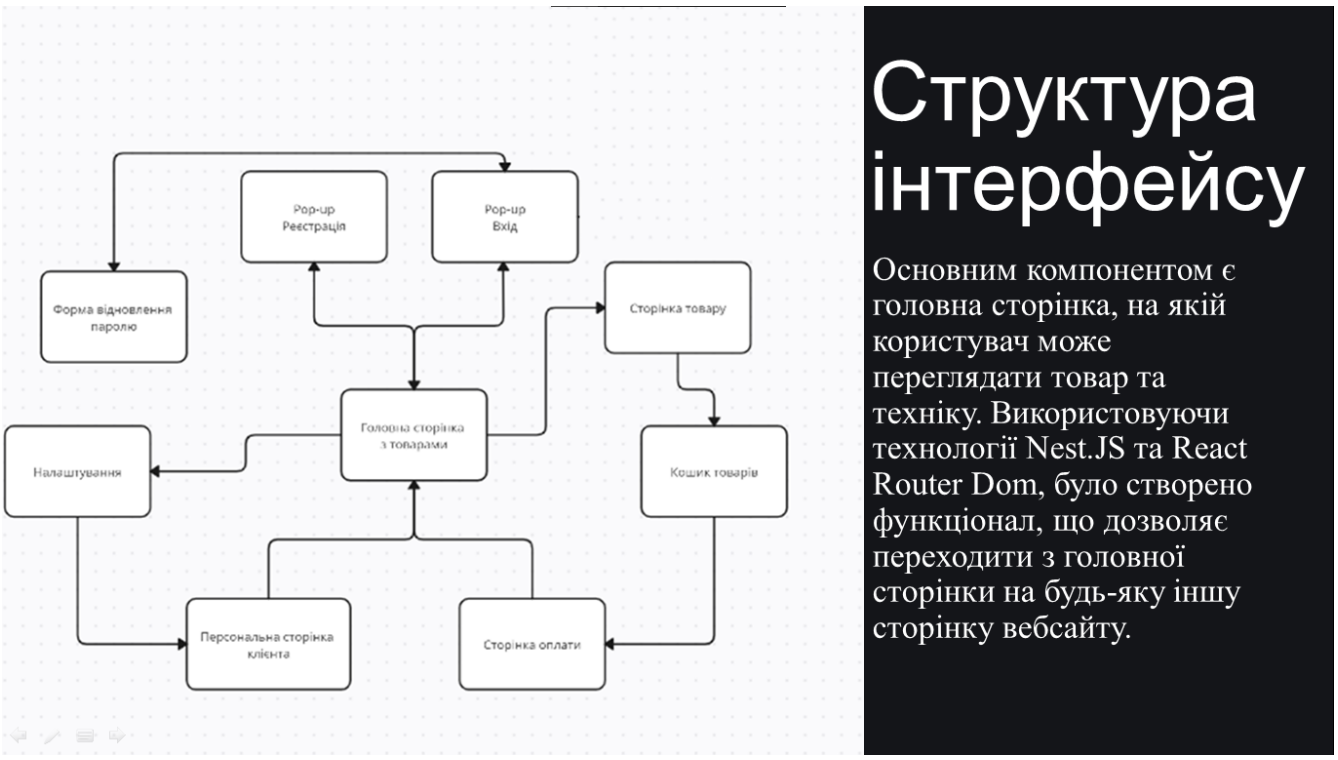
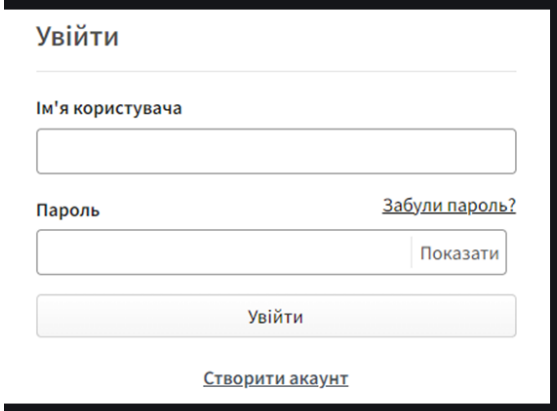


Рисунок В.6 – Слайд 6



Авторизація клієнта

Для реалізації авторизації використовується JSON Web Token (JWT). При вході користувача в систему за допомогою логіна та пароля сервер аутентифікації створює JWT і відправляє цей токен клієнту. Маючи токен, користувач при кожному запиті до API надсилає його разом із запитом. Оскільки сервер має секретний ключ для підпису токена, він перевіряє його на автентичність при отриманні запиту, таким чином підтверджуючи особу власника токена.

Рисунок В.7 – Слайд 7

Діаграма декомпозиції авторизації

Безпека даних клієнта в системі та досконалість авторизації є особливо важливими.

Для реалізації авторизації було обрано JWT - JSON Web Token. Це стандарт токена доступу на основі JSON, який використовується для передачі даних для аутентифікації в клієнт-серверних застосунках.

З використанням технологій Nest.JS було налагоджено зв’язок з базою даних таким чином, щоб користувач мав можливість управляти сласними даними для авторизації.

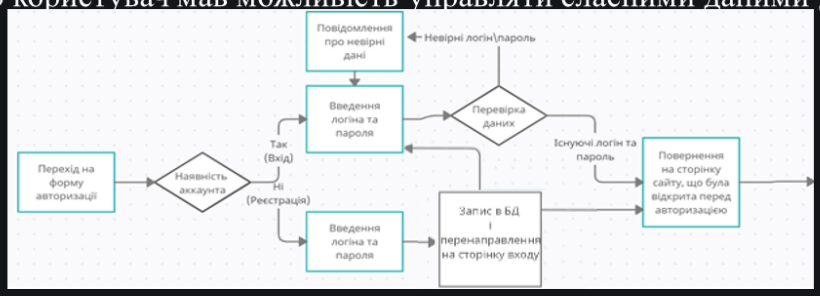


Рисунок В.8 – Слайд 8

- Розробка ПЗ проходила шляхом створення окремих модулів (сторінок сайту) та написання логіки їх взаємодії. Основна логіка серверної частини написана на TypeScript з використанням фреймворку NestJS.
- Клієнтська частина застосунку розроблена з використанням технологій ReactJS та Next.JS.
- Для роботи з базою даних використовується СКБД PosgreSQL та Prisma.

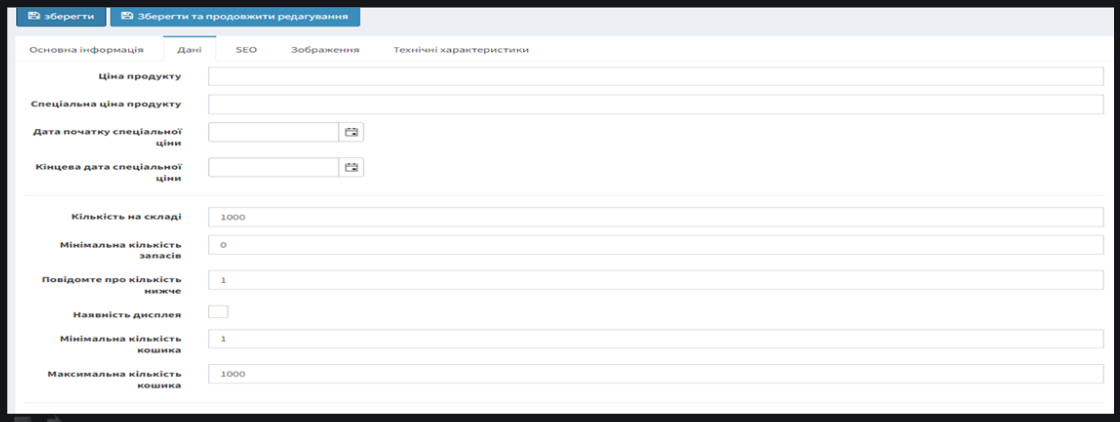


Рисунок В.9 – Слайд 9

Висновки

- При виконанні кваліфікаційної роботи на тему «Інтернет-магазин з продажу побутової техніки» було розроблено:
- архітектуру вебзастосунку, після чого спроектовано та реалізовано функціонал вебзастосунку, що виконує функції Інтернет-магазину.
- Середовище розроблення Microsoft Visual Studio 2019. Архітектура – клієнт-серверна. Платформа – Node JS. Мови – JavaScript, TypeScript . Фреймворки – Nest.JS, React.JS, Next.JS.
- Розроблений вебзастосунок відповідає поставленим завданням, виконує усі необхідні функції.

Рисунок В.10 – Слайд 10

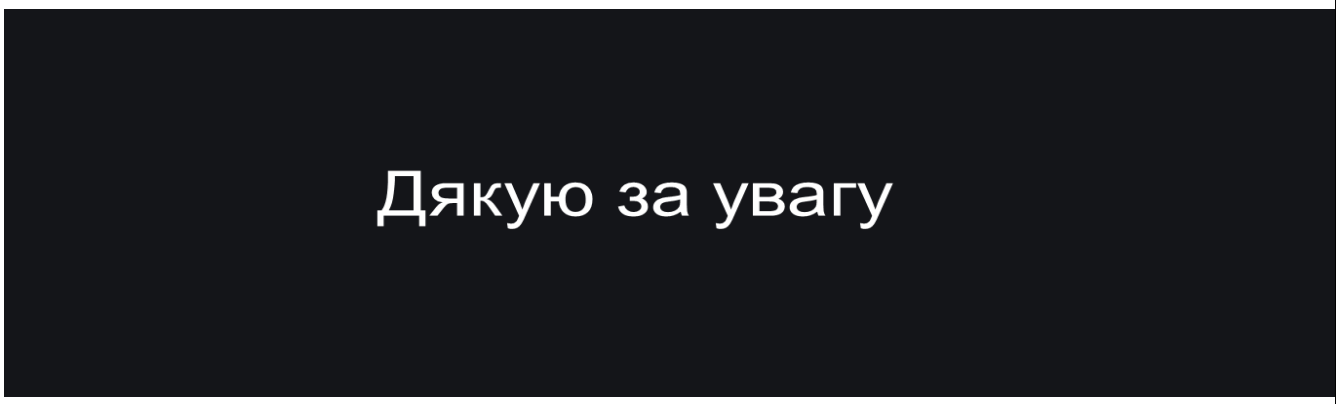
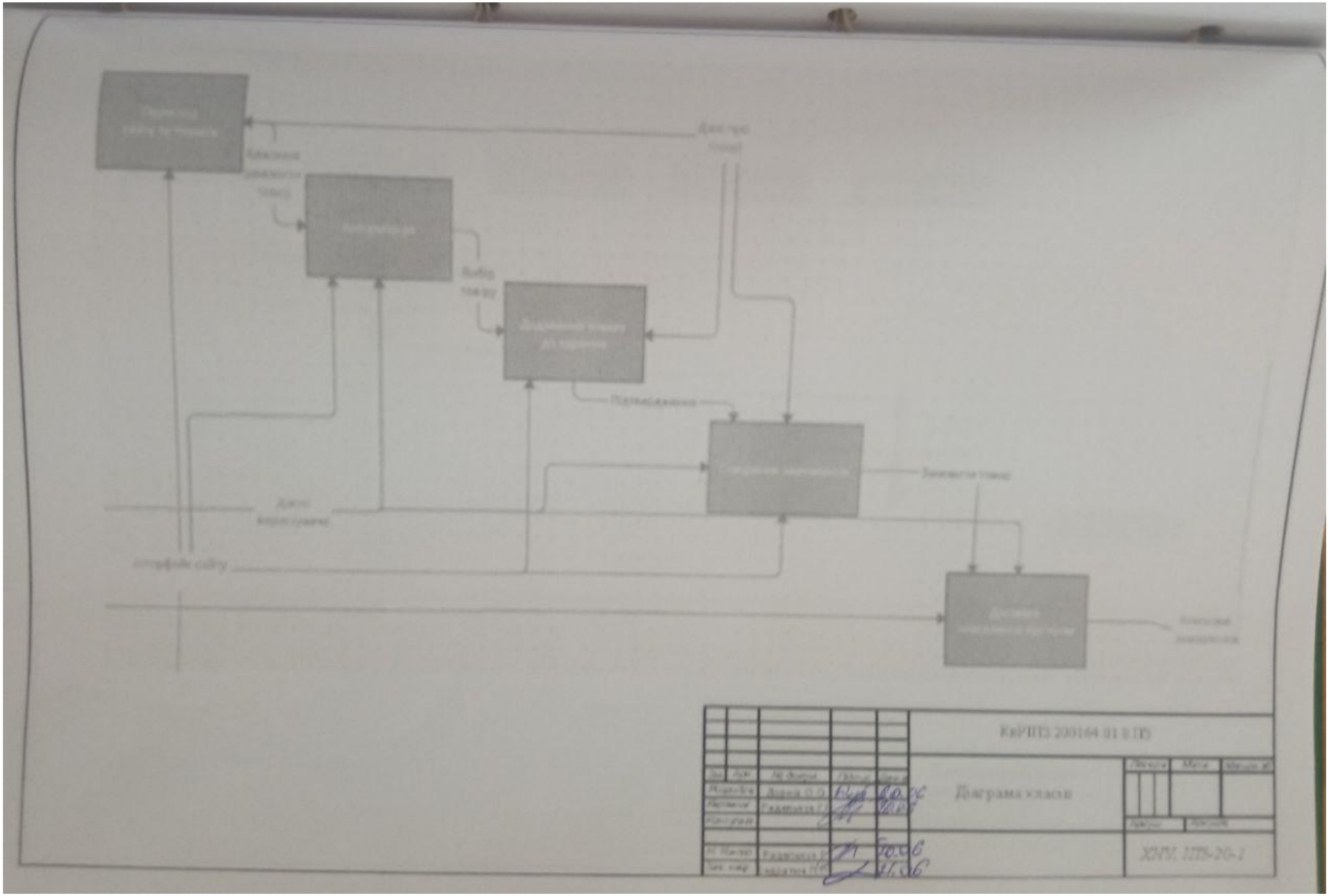
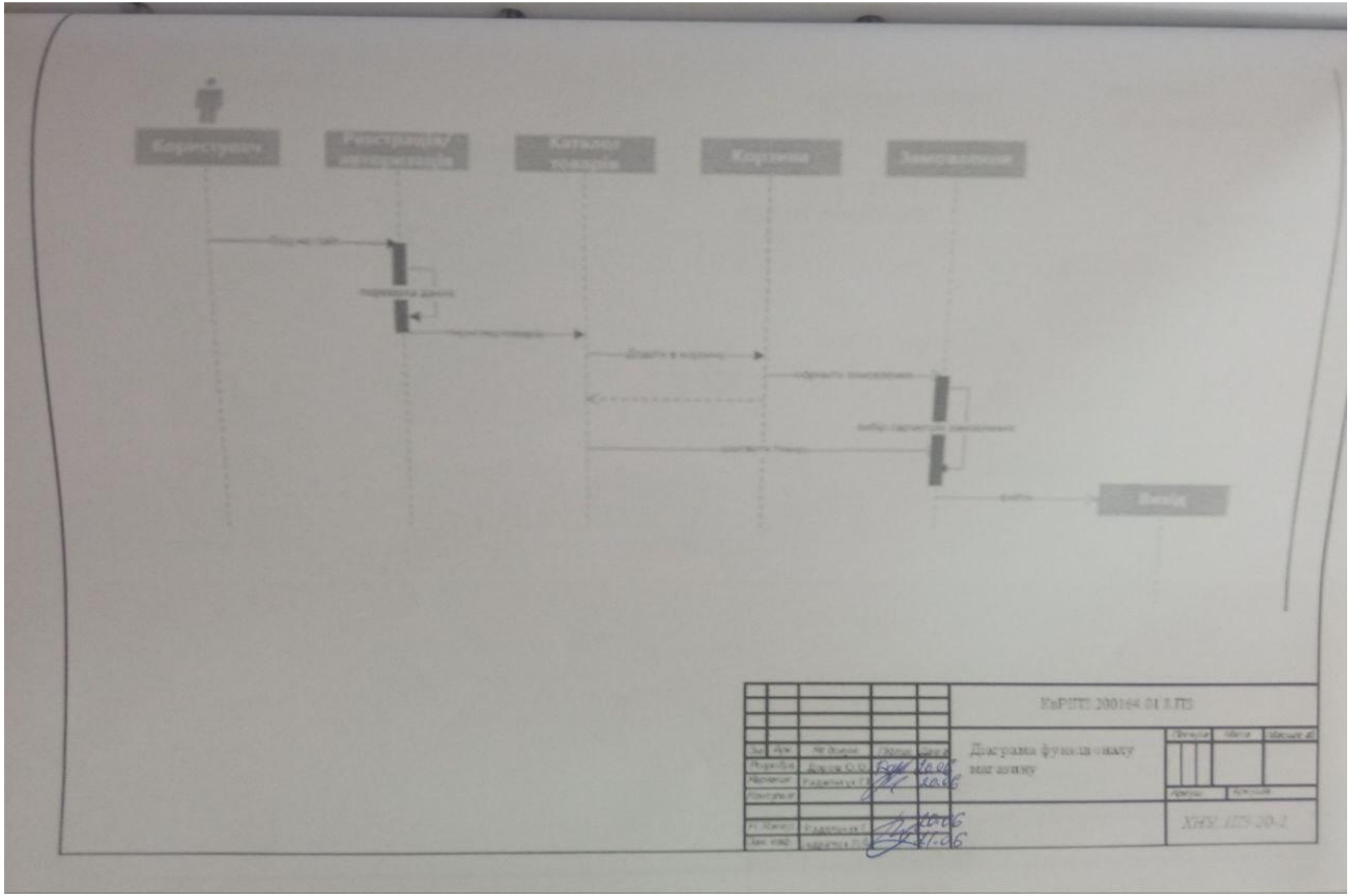
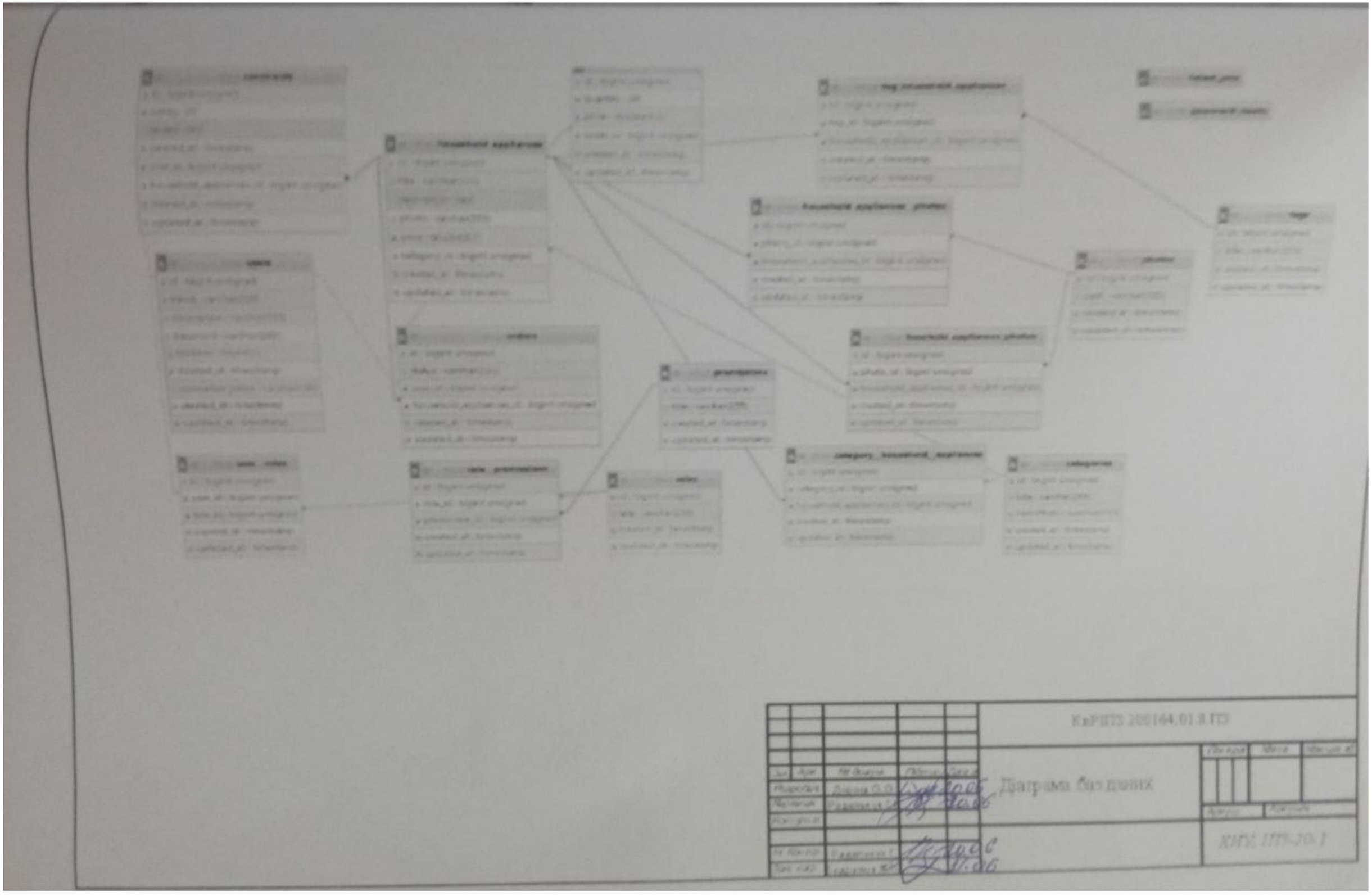


Рисунок В.11 – Слайд 11

ГРАФІЧНА ЧАСТИНА


КвРПТЗ 200164 01 8 ПЗ
Діаграма класів
ХНУ. ПТВ-20-1


Користувач
Реєстрація/ автоматизація
Каталог товарів
Корзина
Замовлення
Вихід
КвРПТЗ.200164.01.3.ПЗ
Діаграма функціоналу магазину
ХНУ.ПТЗ-20-1


КвРПТЗ 200164.01.8.ПЗ
Діаграма баз даних
КНУ ПТЗ-20-1

СУПРОВІДНІ ДОКУМЕНТИ

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

здобувача вищої освіти

Дороша О. О.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІПЗ-20-1

ЗАЯВА

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

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

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

14.06.2024

дата

підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом: 24.0%

Словники перевірки: ru_US, ru_RU, uq_UA. Шумилок в документах: 10%

ID: 151628 Назва: БКР_Інтернет-магазин з продажу побутової техніки Додано в БД: 2024-06-19 Автори: Дорош О. О. Керівники: Радельчук Г. І. канд. техн. наук, доцент Консультанти: Опоненти:	Документ		Сумарний вміст по Базі Даних	
	Сплошне	Лексичне	Сплошне	Лексичне
	50868	472	16745 (30%)	183 (39%)

Джерела плагіату

ID	Опис	Повність плагіату в документі	
		Сплошне	Лексичне
100754	Назва: Веб-ресурс «Інтернет-магазин з продажу [illegible]» Додано в БД: 2023-12-23 Автори: Н.В. Данилевич Керівники: О.М. Яцишин Консультанти: Опоненти:	13598 (24.0%)	146 (31.0%)

UNICHECK
by Turnitin

Ім'я користувача:
ІПЗ

Дата перевірки:
19.06.2024 19:13:23 EEST

Дата звіту:
19.06.2024 19:20:58 EEST

ID перевірки:
1016376103

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100012953

Назва документа: БКР_Інтернет-магазин з продажу побутової техніки_Дорош О.О_Радельчук Г.І

Кількість сторінок: 59 Кількість слів: 8311 Кількість символів: 63500 Розмір файлу: 3.49 MB ID файлу: 1016184139

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

26.3%
Схожість

Найбільша схожість: 3.39% з Інтернет-джерелом (https://vdocuments.mx/download/oe-ff-aoe-a)

23.2% Джерела з Інтернету 703 ..... Сторінка 51

14.4% Джерела з Бібліотеки 170 ..... Сторінка 71

2.97% Цитат

Цитати 12 ..... Сторінка 77

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 9 сторінок

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Здобувач Дорош Олег Олександрович

Тема Інтернет-магазин з продажу побутової техніки

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень три; кількість сторінок записки 57

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі проведено та описано дослідження предметної області щодо використання Інтернет-магазинів. Дослідження відбувалося шляхом огляду застосунків-аналогів. Згідно з отриманими даними було розроблено функціональні і нефункціональні вимоги до застосунку та доведено актуальність його розроблення. Обрана клієнт-серверна архітектура застосунку, спроєктована його структура та база даних. Для реалізації Інтернет-магазину обрано сучасні засоби та технології. Також було проведено тестування Інтернет-магазину та аналіз результатів тестування, виявлено та усунено недоліки та помилки.

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

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі до кваліфікаційної роботи описано тематику, виділено мету та актуальність роботи. У першому розділі проведено аналіз предметної області, розглянуто існуючі застосунки-аналоги та визначені функціональні і нефункціональні вимоги до Інтернет-магазину. У другому розділі обрано архітектуру застосунку, спроєктовано його структуру, базу даних, інтерфейс користувача, а також обрано необхідні технології і засоби для реалізації застосунку. У третьому розділі описано реалізацію застосунку та його тестування.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є актуальною, оскільки з кожним роком зростає кількість користувачів, які вибирають онлайн-покупки як зручний та швидкий спосіб придбання товарів. Також у проєкті використані сучасні технології розроблення ПЗ: Postman, Bootstrap, ASP.NET Core.

5. Негативні сторони роботи Реалізація Інтернет-магазину виконана не у повному обсязі. У роботі виявлено граматичні помилки.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення подано у вигляді ілюстративних рисунків, зокрема діаграм та зображень описуваних елементів. Оформлення пояснювальної записки в основному відповідає вимогам чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки подано доступно та зрозуміло, додатки і графічний матеріал дають можливість візуалізувати деталі та принципи функціонування Інтернет-магазину.

8. Інші зауваження

9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана на задовільному рівні та заслуговує на оцінку "задовільно".

РЕЦЕНЗЕНТ (прізвище, ім'я, по-батькові, посада, місце роботи) Говорущенко Тетяна Олександрівна, професор, зав. кафедри КІТС ХНУ

20 червня 2024 р.

(підпис)

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

Підтверджуємо ознайомлення з результатами звіту/звітів перевірки роботи, продукованими програмно-технічним засобом (ами), на наявність текстових збігів:

Назва кваліфікаційної роботи: «Інтернет-магазин з продажу побутової техніки»
Автор: Дорош Олег Олександрович
Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»
Спеціальність: 121 – Інженерія програмного забезпечення
Науковий керівник: Радельчук Галина Іванівна, кандидат технічних наук, доцент
Після аналізу звіту/звітів зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	
2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована.	**відповідає**
3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнуті. Робота може бути допущена до захисту після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

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

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

2) в якості запозичень системою Unicheck було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) запозичення, виявлені в тексті роботи, є фрагментарними або мають належним чином оформлені посилання.

4) виявлені модифікації тексту не впливають на відсоток схожості.

Максимальний обсяг запозичень, визначений системою Anti-Plagiarism, складає 24.0%. Обсяг запозичень, визначений системою Unicheck виявлення збігів ідентичності/схожості, складає 26.3% і адресується до 703 джерел з Інтернету і 170 джерел з бібліотеки, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 20.06.2024 р.

Завідувач кафедри _____ Леонід БЕДРАТЮК

Гарант освітньої програми _____ Леонід БЕДРАТЮК

Керівник кваліфікаційної роботи _____ Галина РАДЕЛЬЧУК