

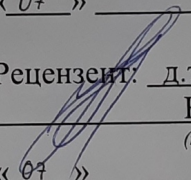
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

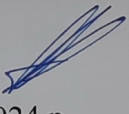
Бакалаврська дипломна робота на тему:

Розробка інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них

Виконав: студент 4 курсу, групи ЗКН-206
спеціальності 122 Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)
 Щерба О. О.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН
 Барабан С. В.
(прізвище та ініціали)
« 07 » 06 2024 р.

Рецензент: д.т.н., професор кафедри КСУ
 Юхимчук М. С.
(прізвище та ініціали)
« 07 » 06 2024 р.

Допущено до захисту Зав.
кафедри Яровий А. А. 
« 07 » 06 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН проф., д.т.н.

Яровий А. А.

« 07 » 03 2024 р

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Щербі Олегу Олеговичу

1. Тема роботи: Розробка інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них

Керівник роботи: к.т.н., доц., доц. кафедри КН Барабан С. В.

Затверджені наказом вищого навчального закладу "ч" 03 2024 року № 80

2. Термін подання студентом роботи 27.05.2024 р.

3. Вихідні дані до роботи:

Мови програмування JavaScript, PHP.

Середовище розробки Visual Studio code;

Зрозумілий та простий інтерфейс користувача.

4. Зміст текстової частини (перелік питань, які потрібно розробити): Вступ; Обґрунтування доцільності розробки інтернет-магазину продажу для продажу контактних лінз, окулярів та аксесуарів до них; Проектування інтернет-магазину продажу для продажу контактних лінз, окулярів та аксесуарів до них; Програмна реалізація інтернет-магазину продажу для продажу контактних лінз, окулярів та аксесуарів до них; Висновки; Список джерел; Додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): схема алгоритму; вигляд інтерфейсу користувача; схема архітектури клієнт-серверної взаємодії; діаграма варіантів використання.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	к.т.н., доц. кафедри КН Барабан С. В.		

7. Дата видачі завдання 07.03.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Приміт
		початок	закінчення	
1	Аналіз предметної області інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них.	06.05.24	09.05.24	
2	Проектування інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них.	10.05.24	13.05.24	
3	Програмна реалізація інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них.	14.05.24	17.05.24	
4	Розробка інструкції користувача.	18.05.24	21.05.24	
5	Підбиття підсумків та аналіз отриманих результатів.	22.05.24	25.05.24	
6	Оформлення матеріалів до захисту БКР.	26.05.24	06.06.24	

Студент 
(підпис)

Керівник роботи 
(підпис)

Щерба О. О.
(прізвище та ініціали)

Барабан С. В.
(прізвище та ініціали)

АНОТАЦІЯ

Щерба О. О. Розробка інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них. Бакалаврська дипломна робота складається з 97 сторінок формату А4 на яких є 45 рисунка, список використаних джерел містить 20 найменувань.

В даній бакалаврській дипломній роботі проаналізовано актуальність створення інтернет-магазину для продажу окулярів, лінз та аксесуарів для них, досліджено існуючі рішення та їх переваги і недоліки. Наведено проектування структури, архітектури та діаграм варіантів використання програмного модуля. Обґрунтовано вибір технологій для розробки клієнтської та серверної частин веб-додатку, зокрема JavaScript з використанням фреймворку Vue.js для фронтенду та PHP для бекенду. Представлено схеми алгоритмів роботи основних компонентів, обґрунтовано вибір середовища розробки Visual Studio Code. Описано процес розробки клієнтської та серверної частин додатку із прикладами коду та проведено тестування розробленого програмного продукту.

Ключові слова: зір, лінзи, окуляри, інтернет-магазин, веб-додаток.

ABSTRACT

Shcherba O. O. Development of an Online Store for Selling Contact Lenses, Glasses, and Accessories. The bachelor's thesis consists of 97 A4 pages, featuring 45 illustrations, and a list of references containing 20 sources.

This bachelor's thesis analyzes the relevance of creating an online store for selling glasses, lenses, and accessories for them. It examines existing solutions, their advantages, and disadvantages. The thesis presents the design of the structure, architecture, and use case diagrams of the software module. The choice of technologies for developing the client and server parts of the web application is justified, specifically JavaScript with the Vue.js framework for the frontend and PHP for the backend. It includes workflow diagrams of the main components and justifies the choice of Visual Studio Code as the development environment. The development process of the client and server parts of the application is described with code examples, and the developed software product is tested.

Keywords: web application, vision, lenses, glasses, online store.

ЗМІСТ

ВСТУП.....	3
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ПРОДАЖУ КОНТАКТНИХ ЛІНЗ, ОКУЛЯРІВ ТА АКСЕСУАРІВ ДО НИХ ...	5
1.1 Аналіз розробки інтернет-магазину	5
1.2 Огляд систем-аналогів	7
1.3 Формування вимог та постановка задачі на дослідження.....	13
1.4 Висновок до розділу 1	15
2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ПРОДАЖУ КОНТАКТНИХ ЛІНЗ, ОКУЛЯРІВ ТА АКСЕСУАРІВ ДО НИХ.....	16
2.1 Розробка структури інтернет-магазину.....	16
2.2 Вибір архітектури інтернет-магазину	18
2.3 Розробка UML-діаграм варіантів використання.....	20
2.4 Висновок до розділу 2	22
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ПРОДАЖУ КОНТАКТНИХ ЛІНЗ, ОКУЛЯРІВ ТА АКСЕСУАРІВ ДО НИХ.....	23
3.1 Обґрунтування вибору інструментів технічної реалізації	23
3.2 Розробка схеми алгоритму роботи інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них	27
3.3 Обґрунтування вибору середовища розробки	31
3.4 Розробка клієнтської та серверної частин інтернет-магазину	36
3.5 Тестування розробленого програмного продукту	45
3.6 Висновок до розділу 3	55
ВИСНОВКИ	56
СПИСОК ДЖЕРЕЛ	57
ДОДАТКИ	59
ДОДАТОК А ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ.....	60
ДОДАТОК Б ІНСТРУКЦІЯ КОРИСТУВАЧА	61
ДОДАТОК В ЛІСТИНГ ПРОГРАМИ.....	68
ДОДАТОК Г ГРАФІЧНА ЧАСТИНА.....	75

ВСТУП

Актуальність досліджень. Зір – один з найважливіших сенсорів людини, який дає можливість бачити світ у всій його красі. Але він може погіршуватися з віком, внаслідок травм, захворювань або через надмірне навантаження на очі.

Згідно зі статистикою Всесвітньої організації охорони здоров'я, близько 2,2 мільярдів людей у світі мають порушення зору, які можна було б виправити за допомогою окулярів або контактних лінз [1]. З кожним роком використання комп'ютерів, смартфонів та інших пристроїв у повсякденному житті значно збільшується, вони стають невід'ємним атрибутом нашого життя, але це благо також збільшує навантаження на очі, що призводить до поширення проблем із зором [2].

В Україні, за даними МОЗ, близько 40% населення мають проблеми із зором [3]. При цьому, медичні операції на очах, незважаючи на свою ефективність, залишаються дорогими та ризикованими. Тому, все більшу популярність здобувають лінзи та окуляри, як доступні та зручні засоби корекції.

Недоліків зору може бути багато, починаючи від простої далекозорості або короткозорості до більш складних проблем, таких як катаракта, глаукома чи деградація сітківки [4]. У більшості випадках, особливо коли мова йде про простіші дефекти, окуляри або контактні лінзи можуть значно полегшити життя людини, забезпечуючи йому зручність і комфорт.

Сьогодні суспільство переживає зростання онлайн-торгівлі, включаючи магазини для окулярів, лінз та аксесуарів до них. Цей тренд викликаний зручністю і доступністю інтернет-покупок.

Інтернет-магазин, який легко масштабується зі зростанням бізнесу та обсягів даних - допоможе підтримувати ефективність та конкурентоспроможність бізнесу в умовах сучасного ринку.

Отже, з огляду на поширеність проблем зі зором і зростання популярності онлайн-шопінгу, розробка інтернет-магазину для окулярів, лінз та аксесуарів до них є доцільною та актуальною в нинішній час.

Метою дослідження є розширення функціональних можливостей інтернет-магазину для окулярів, лінз та аксесуарів до них.

Об'єкт дослідження є процес розробки інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них.

Предметом дослідження є програмне забезпечення інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них.

Задачі дослідження:

1. Здійснити аналіз сучасних методів та технологій для розробки веб-додатків;
2. Проаналізувати системи-аналоги;
3. Розробити структуру та архітектуру веб-додатка інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них;
4. Проаналізувати та обґрунтувати вибір інструментів для розробки;
5. Виконати програмну реалізацію серверної та клієнтської частини інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них;
6. Виконати тестування розробленого веб-додатку та провести аналіз отриманих результатів.

Публікації: за основними результатами досліджень опубліковано тези доповіді на науково-технічній конференції [5].

Апробація роботи: робота апробувалась на конференції: LIII Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024), доповідь «Аналіз передумов розробки інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів для них» [5].

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ПРОДАЖУ КОНТАКТНИХ ЛІНЗ, ОКУЛЯРІВ ТА АКСЕСУАРІВ ДО НИХ

1.1 Аналіз розробки інтернет-магазину

Створення якісного веб-додатку, зокрема інтернет-магазину, потребує ретельного планування та виконання певних етапів [6]. Починається все з ретельного аналізу вимог і потреб майбутніх користувачів. На цьому етапі необхідно зібрати максимально повну інформацію про цільову аудиторію, визначити ключові функції, які мають бути реалізовані, а також сформулювати чіткі критерії успішності проекту. Для ефективного управління та координації робочого процесу зазвичай використовують спеціалізовані інструменти, такі як Jira, Trello чи Asana.

Наступним кроком є ретельний вибір технологій, необхідних для втілення задуму в життя. Виходячи з визначених функціональних вимог, нам потрібно вибрати оптимальну архітектуру майбутнього веб-додатку та відповідний стек технологій для front-end та back-end розробки. На цьому етапі дуже корисними можуть бути програми для створення діаграм, схем архітектури та прототипів, такі як draw.io чи Lucidchart. Вони допоможуть наочно візуалізувати структуру та полегшити розуміння програмного продукту.

Після визначення технічних деталей, починається робота над дизайном інтерфейсу користувача. Для цього використовуються спеціалізовані програми, такі як Sketch, Adobe XD, Figma або універсальні графічні редактори на кшталт Adobe Photoshop. Сучасні інструменти дозволяють створювати інтерактивні прототипи майбутнього продукту для тестування та відпрацювання зручності користувацького досвіду.

Після завершення етапу дизайну розпочинається безпосередня розробка веб-додатку, яка умовно поділяється на створення front-end (клієнтської) та back-end (серверної) частин.

Фронтенд відповідає за все, що бачить та з чим взаємодіє користувач у своєму веб-браузері. Для його розробки використовується мова HTML для розмітки структури сторінок, CSS для стилізації візуальних елементів, а також JavaScript разом із сучасними фреймворками, такими як React, Vue чи Svelte, для забезпечення динамічного функціоналу та взаємодії з сервером. Для цього потрібно написати код, який максимально точно відповідатиме розробленим на попередньому етапі схемам роботи та функціонування.

Бекенд, у свою чергу, відповідає за обробку запитів від клієнтських додатків, виконання бізнес-логіки, зберігання та отримання даних із бази даних, формування відповідей тощо. Для цього використовуються такі мови програмування, як Python, Java, C#, PHP, а також фреймворки, що спрощують розробку, наприклад ASP.NET Core, Django, Ruby on Rails. Він має бути надійним, безпечним та масштабованим.

Важливою складовою більшості сучасних веб-додатків є база даних, в якій зберігається інформація про користувачів, товари, замовлення, контент тощо. В залежності від специфіки веб-додатку можуть використовуватись реляційні бази даних, такі як MySQL, PostgreSQL, MS SQL Server, або нереляційні, наприклад MongoDB чи Firebase. Правильна організація та структурування даних має критичне значення для забезпечення продуктивності та цілісності системи.

Для розширення функціоналу інтернет-магазину можуть інтегруватися сторонні сервіси, програмні інтерфейси (API) та бібліотеки. Наприклад, дуже важливою складовою е-комерс проекту є підключення платіжних шлюзів для забезпечення можливості оплати товарів. Також часто застосовуються геолокаційні сервіси, інтеграція з певними поштовими операторами для відстеження відправлень, сервіси розсилок та повідомлень і багато іншого.

Після того, як основний функціонал веб-додатку буде розроблено, розпочинаються етапи тестування та перевірки якості коду. На цьому етапі використовуються різноманітні методи та інструменти для виявлення та виправлення помилок, дефектів та недоліків у системі. Проводяться одиничні

тести окремих модулів, інтеграційні тести для перевірки коректної роботи компонентів у поєднанні, а також тестування в максимально наближених до реальних умов експлуатації. Популярними засобами автоматизованого тестування веб-додатків є Selenium та Cypress.

Після виправлення критичних проблем, виявлених на етапі тестування, веб-додаток розгортається на сервері, де налаштовуються і встановлюються додаткові бібліотеки та компоненти, виконується підключення до бази даних тощо. Для спрощення процесів розгортання та масштабування додатку часто використовуються хмарні платформи, такі як AWS, Microsoft Azure чи Google Cloud.

Однак розробка програмного забезпечення не завершується випуском першої фінальної версії продукту. Надалі потрібно здійснювати постійний моніторинг роботи системи, виправляти виявлені проблеми, оновлювати функціонал відповідно до мінливих потреб користувачів та ринку, оптимізувати продуктивність та безпеку. Для полегшення цього процесу важливо з самого початку організувати чітку систему виявлення та реагування на помилки, налагодити ефективну комунікацію з користувачами для отримання зворотнього зв'язку.

У сучасному світі інтернет-технологій життєвий цикл веб-додатку є безперервним процесом постійного вдосконалення та розвитку. Лише за умови дотримання найкращих практик розробки, тестування та підтримки можна створити дійсно якісний, конкурентоспроможний та затребуваний продукт, який здатний забезпечити потреби бізнесу та задовольнити вимоги користувачів.

1.2 Огляд систем-аналогів

Аналіз існуючих інтернет-магазинів для продажу контактних лінз, окулярів та аксесуарів до них дозволяє оцінити їх функціональність, зовнішній вигляд та конкурентні переваги. На українському ринку доступно багато інтернет-магазинів цього типу, але кожен з них має свої особливості,

тому проведемо аналіз існуючих, а саме Люксоптика [7], Оптика Люксор [8], kupit-ochki.com.ua [9], LINZA.ua [10].

Люксоптика – один з найбільших українських магазинів окулярів, лінз та аксесуарів до них. Вони займаються продажем окулярів з 1994 року та мають багато звичайних магазинів. Інтернет-магазин був створений у 2009, тому переважний дохід йде саме з офлайн продажів, через що робиться менший акцент на веб-ресурс та його підтримку. На головній сторінці веб-додатку (рис. 1.1) можна знайти великий банер зі знижками, а також посилання на їх YouTube канал, де вони надають корисну інформацію про очі та зір. Нижче можна побачити основну інформацію про компанію та топ товарів за продажами, а саме 3 моделі контактних лінз. Це відбувається через те що на сайті наразі немає розділу з окулярами для зору. Якщо перейти на сторінку, де повинні бути окуляри для зору, ми побачимо напис «Цей розділ наразі в розробці» (рис. 1.2). Тобто, на сьогоднішній момент замовити окуляри для зору через веб-ресурс не вийде. Також на сайті доступні відгуки покупців, в яких також є зауваження до сайту, які потрібно врахувати та виправити. Наприклад, в останньому відгуку користувач скаржиться на поганий пошук товарів на сайті (рис. 1.3).

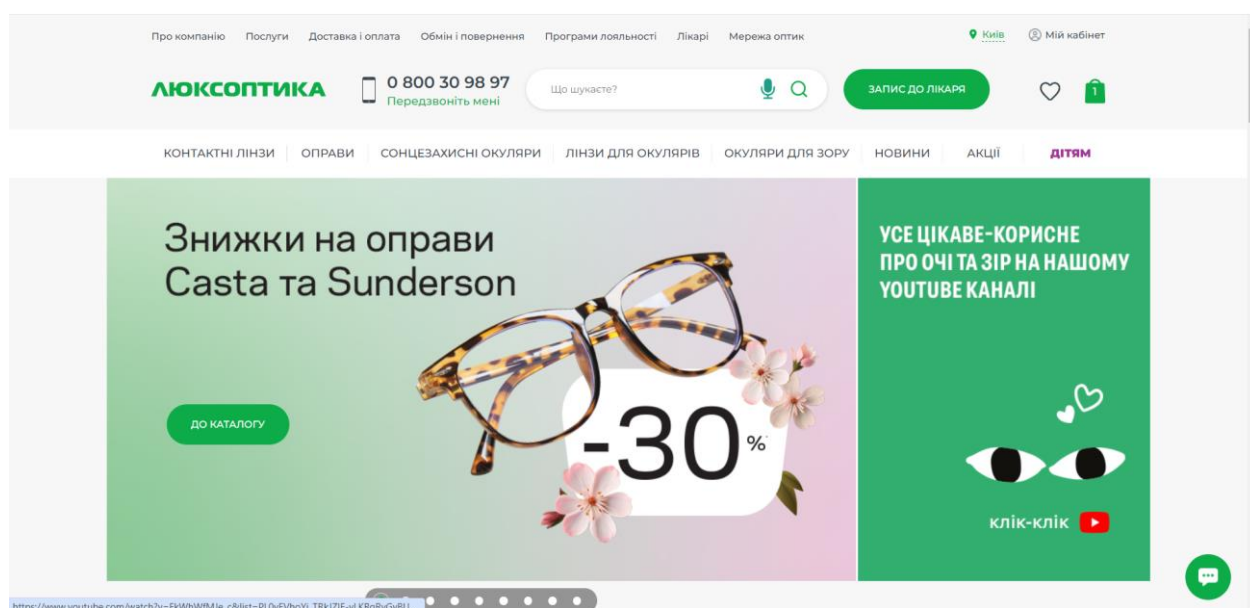


Рисунок 1.1 – Головна сторінка інтернет-магазину Люксоптика

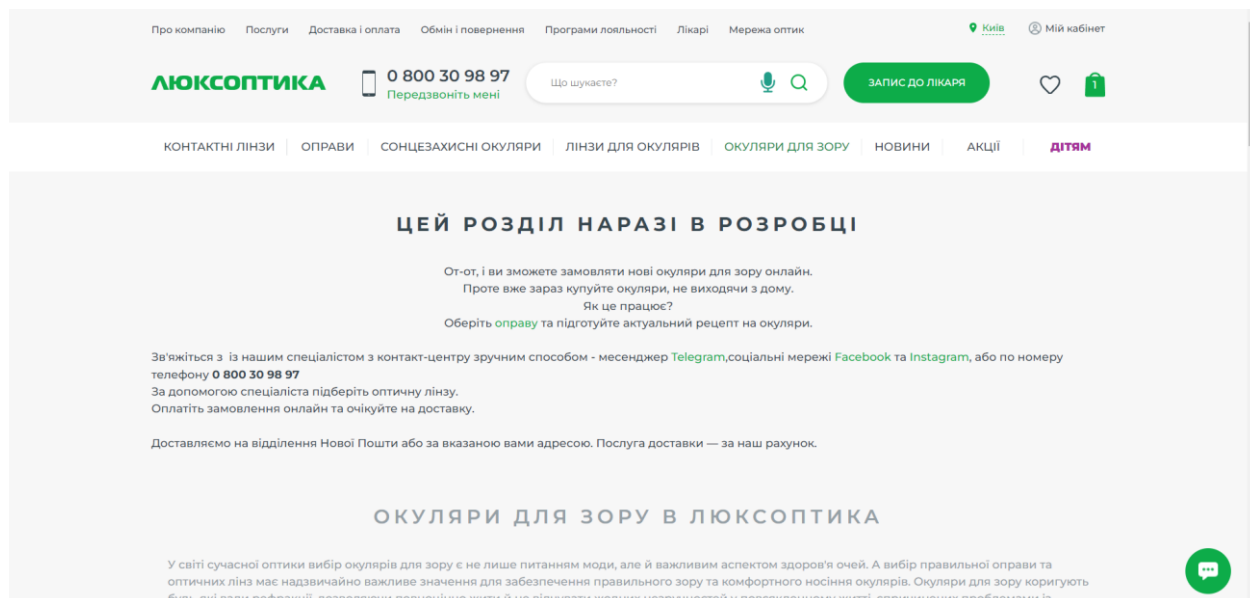


Рисунок 1.2 – Сторінка окулярів для зору в інтернет-магазині Люксоптика

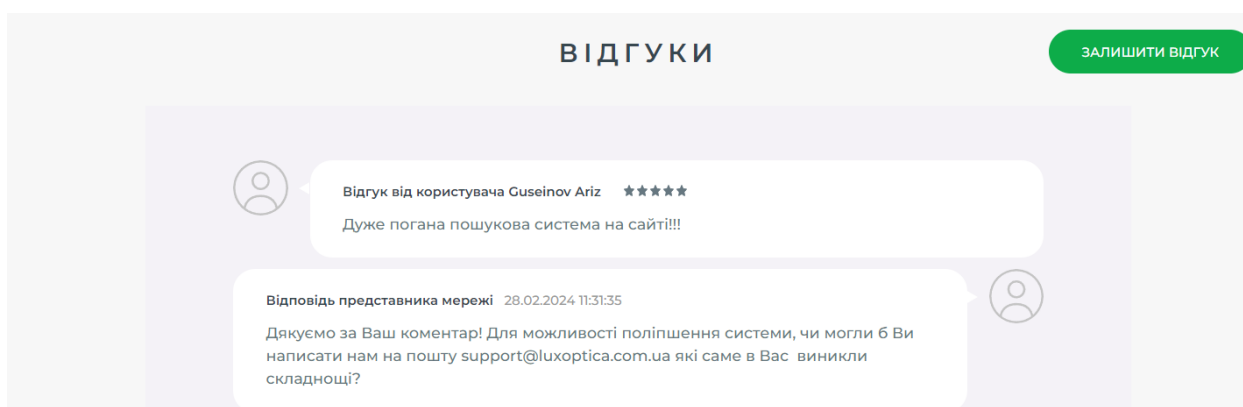


Рисунок 1.3 – Сторінка з відгуками в інтернет-магазині Люксоптика

Оптика Люксор також являється великим гравцем на ринку. Заснована ще в 1998 році та має більше десятка офлайн магазинів у різних містах України. На головній сторінці веб-додатку (рис. 1.4) можна знайти великий банер з акціями та рекламними пропозиціями. Також можна помітити, що він досить великий, через що його зручно читати людям з вадами зору. Нижче на головній ми можемо побачити вибрані товари, а саме сонцезахисні окуляри, оправы або контактні лінзи. Спустившись ще трішки нижче можна побачити послуги, які вони надають. Веб-додаток оптика Люксор не пропонує конструктор, а тільки саму оправу без варіанту вставлення необхідних лінз.

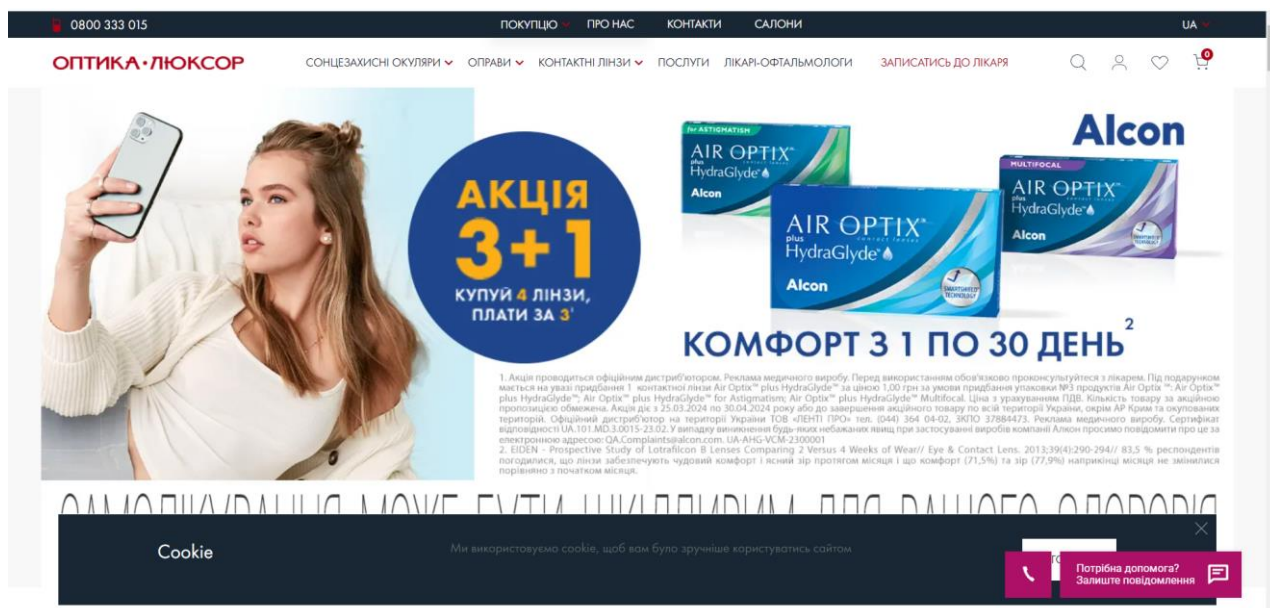


Рисунок 1.4 – Головна сторінка інтернет-магазину Оптика Люксор

Kupit-ochki.com.ua - це молодий інтернет-магазин, який стрімко набуває популярності. Основним напрямком діяльності цього магазину є роздрібна торгівля товарами широкого вжитку. Наразі на їхньому сайті представлено п'ять основних категорій продукції. Провідною товарною групою є окуляри для корекції зору. Магазин був заснований 13 лютого 2020 року. На відміну від традиційних торгових точок, це виключно інтернет-магазин без звичайних відділень. Структура головної сторінки сайту є типовою для подібних ресурсів і включає банер та блоки з рекомендованими та популярними товарами.

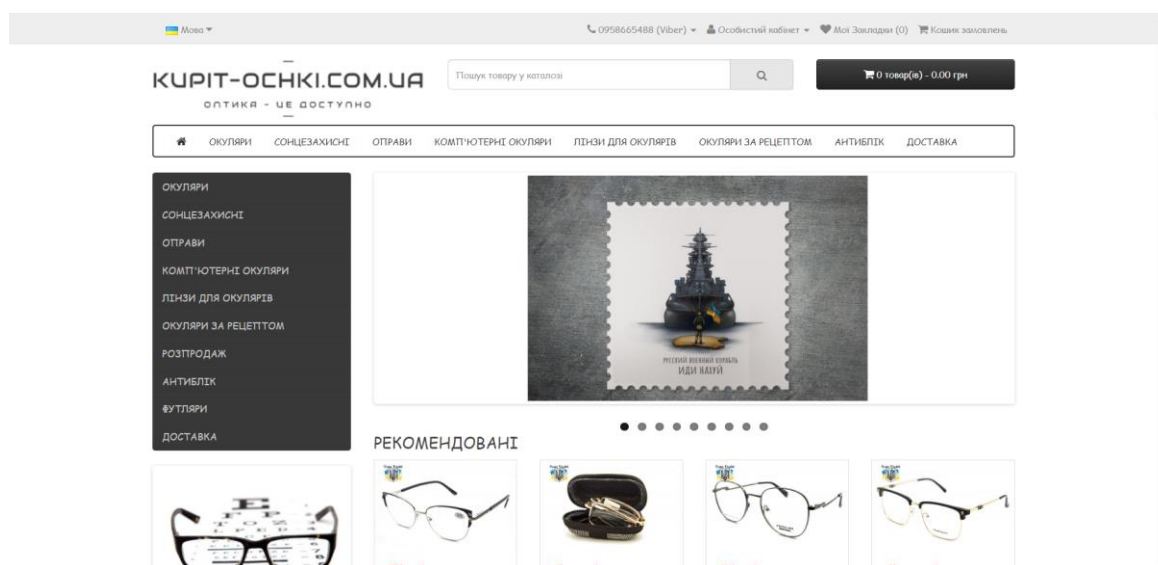


Рисунок 1.5 – Головна сторінка інтернет-магазину kupit-ochki.com.ua

Linza.com.ua є провідним інтернет-магазином в Україні з продажу товарів для контактної корекції зору. Вони пропонують великий вибір м'яких контактних лінз та засобів по догляду за ними. В їхньому асортименті представлено понад 70 видів лінз, 30 видів розчинів і пероксидних систем очищення, 15 видів зволожуючих крапель для очей, ферментні таблетки, аксесуари, дорожні набори. Місією Linza.com.ua є надання клієнтам максимально зручного та якісного сервісу з придбання засобів контактної корекції зору для задоволення їхніх індивідуальних потреб. Вони прагнуть пропонувати широкий асортимент продукції від провідних світових виробників для забезпечення комфортного носіння контактних лінз та ретельного догляду за ними. Завдяки налагодженим партнерським стосункам з постачальниками, Linza.com.ua можуть запропонувати вигідні ціни та оперативну доставку в будь-який регіон України. Як і попередній магазин це суто онлайн магазин, без звичайних точок продажів. На головній сторінці ми також бачимо великий банер, а під ним акційні товари. Також на ній можна знайти статті корисні для догляду за лінзами, про корекцію та профілактику зору і багато чого іншого цікавого. Також можна знайти оправы для окулярів, але без конструктора.

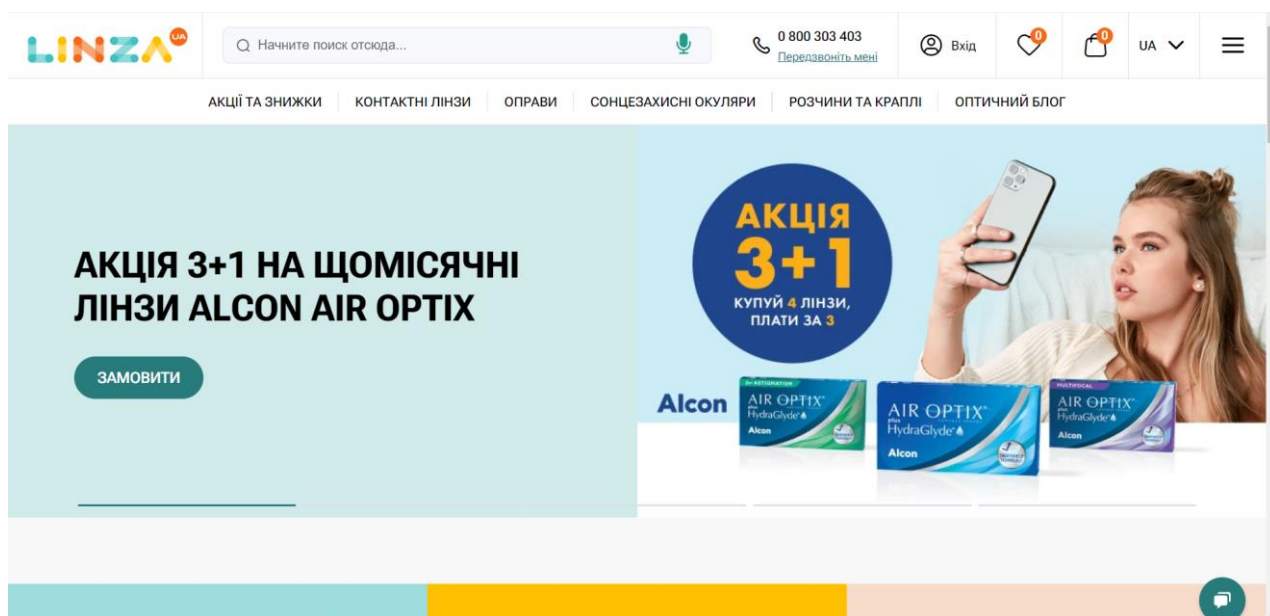


Рисунок 1.6 – Головна сторінка інтернет-магазину LINZA.ua

Для більшої зручності підсумуємо порівняльні характеристики проаналізованих інтернет-магазинів окулярів, лінз та аксесуарів до них. Після аналізу було визначено такі критерії:

- Адаптивність: це можливість автоматично підлаштовуватися під різні розміри екранів та пристроїв, забезпечуючи зручний перегляд і використання як на комп'ютерах, так і на мобільних пристроях.
- Наявність контактних лінз: передбачає, що на сайті представлений асортимент контактних лінз різних типів та від різних виробників для задоволення потреб клієнтів.
- Наявність окулярів: пропонує вибір оправ та готових окулярів для корекції зору.
- Наявність конструктора (опцій для лінз): це функціонал, який дозволяє користувачам самостійно сконфігурувати окуляри згідно індивідуальних потреб, обираючи тип лінз, діоптрії тощо.
- Фільтри/Сортування: це наявність інструментів фільтрації та сортування товарів за різними критеріями, такими як ціна, бренд, категорія, полегшує пошук потрібних позицій на сайті.

Таблиця 1 – Порівняльна характеристика популярних додатків для створення правильного раціону

	Люксоптика	Оптика Люксор	kupit-ochki.com.ua	LINZA.ua
Адаптивність	+	+	+	+
Наявність контактних лінз	+	+	-	+
Наявність Окулярів	+	+	+	+
Наявність конструктору (опцій для лінз)	-	-	+	-
Фільтри/Сортування	+	+	+	+

На основі аналізу даних таблиці 1 видно, що більшість інтернет-магазинів мають такі переваги:

- наявність адаптивності під різні екрани;
- наявність вибору контактних лінз;
- наявність оправ, окулярів;
- наявність фільтрів та сортування.

Крім того, kurit-ochki.com.ua має конструктор, який дозволяє зробити замовлення з урахуванням діоптрій та іншими побажаннями, але він не має наявності вибору контактних лінз. Також Люксоптика в найближчий час може отримати конструктор, адже зараз в тому розділі напис «Цей розділ наразі в розробці».

Після проведеного аналізу, можна зробити висновок, що майбутній веб-ресурс має включати всі ключові функції конкурентів, такі як адаптивність, наявність окулярів і лінз, фільтри та сортування. Крім того, для підвищення конкурентоспроможності, необхідно розробити зручний та інтуїтивно зрозумілий конструктор окулярів з можливістю вибору лінз за рецептом. Також, важливим аспектом є створення сучасного і привабливого дизайну з акцентом на доступність та легкість використання, особливо для користувачів.

1.3 Формування вимог та постановка задачі на дослідження

Інтернет-магазини стали невід'ємною частиною сучасного світу електронної комерції, це сталося завдяки технологічному прогресу та зростанню популярності онлайн покупок. Вони пропонують певні переваги як для покупців, так і для бізнесу, роблячи процес купівлі товарів та послуг зручнішим та ефективнішим.

Однією з головних переваг інтернет-магазинів є їхня доступність з будь-якої точки світу. Не важливо де знаходиться покупець, він може отримати доступ до широкого асортименту товарів з власного дому чи офісу. Це дозволяє економити час, який раніше витрачався на походи по традиційних

магазинах. Крім того, веб-магазини часто пропонують зручні інструменти пошуку, фільтрації та порівняння товарів, полегшуючи процес вибору.

Важливою перевагою також є кросплатформність веб-ресурсів. Покупці можуть заходити в інтернет-магазин зі смартфона, планшета чи комп'ютера, що робить покупки максимально зручним у будь-який час та з будь-якого місця. Проте веб-магазини не позбавлені і певних недоліків, таких як технічні збої чи відсутність можливості випробувати товар перед покупкою, особливо коли йдеться про підбір необхідних лінз.

Однак, незважаючи на деякі недоліки, переваги інтернет-магазинів значно перевищують, завдяки чому вони продовжують зберігати свою актуальність та популярність серед споживачів по всьому світу. Зокрема, розробка інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них є досить перспективною метою.

Плануючи створення такого веб-ресурсу, важливо врахувати низку вимог та функціональних можливостей, необхідних для задоволення потреб клієнтів:

- Надійна система реєстрації та авторизації для захисту облікових записів користувачів;
- Зручний адаптивний інтерфейс для доступу з різних пристроїв;
- Каталог товарів з функціями пошуку та сортування за різними критеріями;
- Можливість замовлення окулярів за індивідуальним рецептом;
- Наявність детальної інформації про товари, включно з фото та описами.

Реалізувати весь цей функціонал допоможе ретельне планування архітектури веб-додатку, використання сучасних технологій для back-end та front-end розробки, а також створення ефективної бази даних для зберігання та обробки інформації. Важливим є також використання принципів user experience (UX) для забезпечення максимальної зручності та інтуїтивності у використанні ресурсу.

Тому розробка веб-ресурсу для продажу контактних лінз, та аксесуарів

до них є перспективним та актуальним завданням, успішна реалізація якого дозволить задовільнити зростаючий попит споживачів у цій сфері та створити конкурентоспроможний продукт на ринку електронної комерції.

1.4 Висновок до розділу 1

У цьому розділі було проведено аналіз існуючих інтернет-магазинів з продажу окулярів, контактних лінз та супутніх товарів. Розглянуто провідні платформи на ринку, такі як Люксоптика, Оптика Люксор, kupit-ochki.com.ua та LINZA.ua. Під час аналізу були виявлені їхні сильні та слабкі сторони, а також основний функціонал, який вони пропонують.

Більшість досліджених веб-ресурсів мають адаптивний дизайн, можливість вибору контактних лінз та окулярів, фільтри та сортування товарів. Однак лише один аналог надає функцію конструктора для підбору окулярів з урахуванням індивідуальних параметрів та діоптрій. Саме ця можливість є однією з ключових переваг перед конкурентами.

Під час аналізу були сформовані основні вимоги до майбутнього веб-ресурсу, серед яких: надійна система авторизації, адаптивний інтерфейс, каталог товарів з пошуком та фільтрами, детальна інформація про продукцію, можливість формувати замовлення.

Розробка такого багатофункціонального веб-ресурсу для продажу контактних лінз, окулярів та аксесуарів до них є актуальним завданням, оскільки дозволить задовольнити зростаючий попит споживачів у цій сфері та створити конкурентоспроможний продукт на ринку електронної комерції.

2 ПРОЕКТУВАННЯ ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ПРОДАЖУ КОНТАКТНИХ ЛІНЗ, ОКУЛЯРІВ ТА АКСЕСУАРІВ ДО НИХ

2.1 Розробка структури інтернет-магазину

Розробка великих веб-додатків, таких як інтернет-магазин, часто вимагає розробки багатьох функцій та складних модулів. Через що зберігати весь код в одному файлі незручно та непрактично, оскільки це ускладнює розуміння коду та робить неможливою одночасну роботу кількох розробників. Для вирішення цієї проблеми використовується підхід модульного програмування.

Модульне програмування полягає у розбитті великої задачі на менші частини, які називаються модулями [11]. Кожен з них реалізується в окремому файлі, що дозволяє зосередитися на конкретній функціональності. Завдяки тому, що кожен модуль робить певну річ, можна не тримати в голові весь веб-додаток, а тільки певну його частину, що значно спрощує процес розробки та тестування.

Також однією з головних переваг модульного програмування є переінженерія. Тобто модулі можна використовувати повторно в різних частинах програми або навіть у інших проектах. Завдяки меншому розміру порівняно з повноцінною програмою, модуль легше перевірити та протестувати, що знову ж прискорює процес розробки.

Оновлення та модифікація веб-додатку також стає простішою завдяки модульному програмуванню. Нові або оновлені модулі можуть бути використані в існуючій програмі без необхідності змінювати всю структуру.

Для інтернет-магазину оптики будуть реалізовані такі модулі, як сторінка авторизації, сторінка реєстрації, сторінка каталог сонцезахисних окулярів, сторінка каталог оправ, сторінка каталог аксесуарів, сторінка окремого товару, сторінка створення нового товару, сторінка профілю користувача, сторінка створення замовлення та сторінка перегляду замовлення. Кожен модуль має свої власні компоненти, які відповідають за графічне оформлення, розмітку, функціональність клієнтської та серверної

частини програми.

Така модульна структура дозволяє ефективно розвивати та підтримувати веб-додаток інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них, зберігаючи його організованим та легким для розширення.

Структура програмного модуля зображена на рисунку 2.1:



Рисунок 2.1 – Загальна структурна схема функціонування системи

На сторінці авторизації користувач зможе ввести свої облікові дані (логін/електронну пошту та пароль) для входу в систему або перейти на сторінку реєстрації, якщо ще немає облікового запису. Сторінка реєстрації дозволить створити новий обліковий запис шляхом введення необхідних даних (ім'я, прізвище, електронна пошта, пароль тощо) та підтвердити реєстрацію для переходу до авторизації.

На сторінці каталогу сонцезахисних окулярів користувач зможе переглядати список доступних моделей, застосовувати фільтри та сортування для пошуку потрібних окулярів за критеріями (виробник, ціна тощо), а також переходити на сторінку окремого товару для більш детальної інформації. Сторінки каталогів оправ та аксесуарів матимуть аналогічну функціональність з фільтрами, сортуванням та переходом на сторінку товару.

На сторінці окремого товару буде представлена детальна інформація про обраний товар (опис, характеристики, фото, ціна), можливість додати товар до кошику, а також перейти до оформлення замовлення.

На сторінці створення нового товару (для адміністраторів) можна буде

завантажувати фото, вводити опис, характеристики, ціну для нового товару та зберігати його для публікації в каталозі.

Сторінка профілю користувача дозволить переглядати особисті дані облікового запису, переглядати його замовлення.

На сторінці створення замовлення користувач зможе оглянути вміст кошику із вибраними товарами, ввести адресу доставки та способи оплати, а потім оформити та підтвердити замовлення.

Сторінка перегляду замовлення надасть можливість переглянути деталі вже оформленого замовлення (список товарів, сума, адреса доставки тощо) та відстежувати його статус.

Така модульна структура із даними функціями на кожній сторінці забезпечить інтуїтивний користувацький інтерфейс, пошук необхідного товару, вибору та оформлення замовлень, окуляри, лінзи та аксесуари в інтернет-магазині.

2.2 Вибір архітектури інтернет-магазину

Для розробки веб-додатку інтернет-магазину буде використана клієнт-серверна архітектура, яка є однією з найпоширеніших архітектурних моделей для веб-застосунків. Ця модель передбачає розподіл функціональності між клієнтською та серверною частинами [12].

Клієнтська частина (фронтенд) - це інтерфейс, який взаємодіє з користувачем у веб-браузері. Вона відповідає за відображення веб-сторінок, обробку вводу даних користувача та відправку запитів на сервер. Для розробки фронтенду зазвичай використовуються мови HTML, CSS та JavaScript, а також фреймворки, такі як React, Angular або Vue.js.

Серверна частина (бекенд) - це основна логіка додатку, яка обробляє запити від клієнтів, взаємодіє з базою даних, виконує необхідні обчислення та формує відповіді. Бекенд розробляється за допомогою мов програмування, таких як PHP, Python, Ruby, Java або Node.js, а також використовуються

різноманітні фреймворки та бібліотеки.

Взаємодія між клієнтом і сервером відбувається за допомогою протоколу HTTP. Клієнт відправляє HTTP-запит на сервер, а сервер обробляє цей запит та повертає HTTP-відповідь назад клієнту.

Крім того, для зберігання даних, таких як інформація про товари, користувачів та замовлення, використовується база даних. Найпоширенішими типами баз даних для веб-додатків є реляційні бази даних та нереляційні бази даних.

Архітектура веб-додатку інтернет-магазину зображена на рисунку 2.1:



Рисунок 2.2 – Архітектура клієнт-серверної взаємодії

На діаграмі показано, що клієнт (веб-браузер) відправляє HTTP-запит на сервер (веб-сервер). Сервер обробляє цей запит, взаємодіє з базою даних за необхідності та формує HTTP-відповідь, яку відправляє назад клієнту.

Ця архітектура забезпечує розподіл обов'язків між клієнтською та серверною частинами, що дозволяє ефективно масштабувати додаток, покращити простоту розуміння та підвищити надійність при змінах.

2.3 Розробка UML-діаграм варіантів використання

При аналізі предметної області було встановлено такі класи користувачів: гість, користувач, редактор та адміністратор.

Давайте розглянемо варіанти використання у вигляді діаграм, які відображають як різні класи користувачів будуть користуватись веб-додатком.

Гостю доступні до перегляду сторінки каталогів товарів, сторінка товарів, сторінка авторизації та сторінка реєстрації. (рис. 2.3)

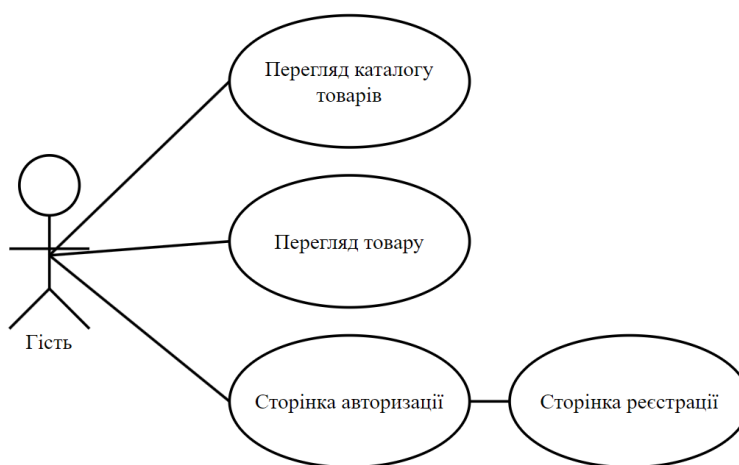


Рисунок 2.3 – Діаграма варіантів використання для класу «Гість»

Користувачеві доступні ті ж сторінки, що і гостю, але він також може переглядати сторінку профіля користувача, створення замовлення та сторінку перегляду замовлення.

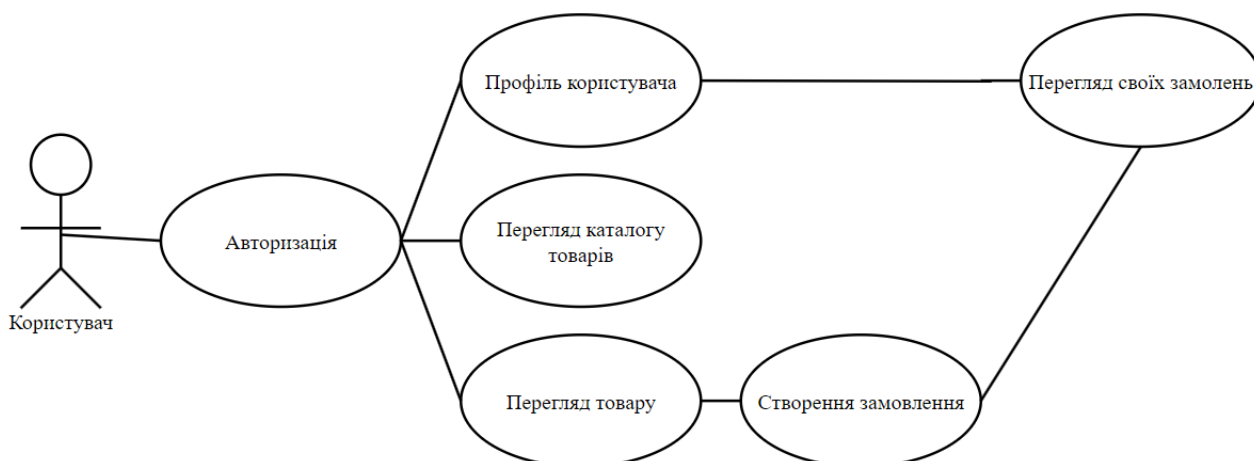


Рисунок 2.4 – Діаграма варіантів використання для класу «Користувач»

Редактор може переглядати всі доступні користувачу сторінки і добавляти, видаляти товари, а також переглядати всі замовлення.

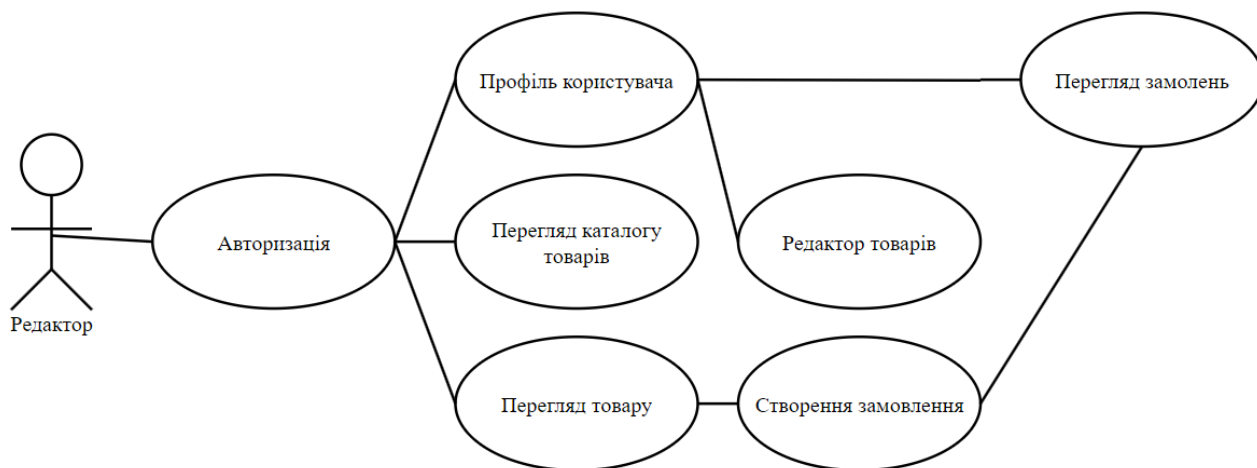


Рисунок 2.5 – Діаграма варіантів використання для класу «Редактор»

Адміністратор може робити все, що і редактор, але також змінювати права користувачів.

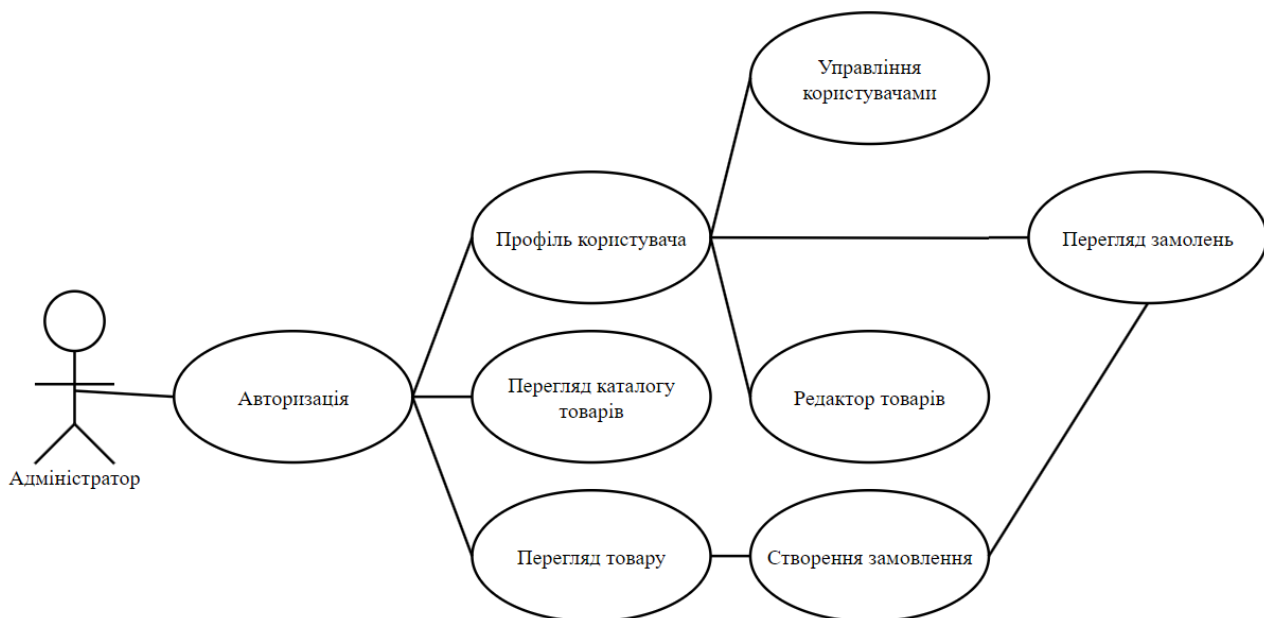


Рисунок 2.6 – Діаграма варіантів використання для класу «Адміністратор»

Таким чином різні класи будуть взаємодіяти з веб-додатком.

2.4 Висновок до розділу 2

У цьому розділі було розроблено структуру програмного модуля інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них, архітектуру клієнт-серверної взаємодії веб-додатку та UML-діаграми варіантів використання для різних класів користувачів.

Структура програмного модуля складається з багатьох окремих сторінок та компонентів, таких як сторінка авторизації, сторінка реєстрації, сторінка каталог сонцезахисних окулярів, сторінка каталог оправ, сторінка каталог аксесуарів, сторінка окремого товару, сторінка створення нового товару, сторінка профілю користувача, сторінка створення замовлення та сторінка перегляду замовлення. Така модульна структура забезпечує організованість, зрозумілість та легкість розширення веб-додатку.

Для інтернет-магазину було обрано клієнт-серверну архітектуру, яка передбачає розподіл функціональності між клієнтською (фронтенд) та серверною (бекенд) частинами. Фронтенд відповідає за відображення веб-сторінок та взаємодію з користувачем, а бекенд обробляє запити від клієнтів, взаємодіє з базою даних та формує відповіді. Така архітектура забезпечує ефективність, масштабованість та надійність веб-додатку.

Також були розроблені UML-діаграми варіантів використання для різних класів користувачів: гість, користувач, редактор та адміністратор. Ці діаграми наочно демонструють функціональні можливості та привілеї кожного класу в межах веб-додатку, що полегшує розуміння вимог та допомагає у розробці.

Отже, розроблена структура, архітектура та діаграми варіантів використання створюють міцне підґрунтя для ефективної реалізації функціонального веб-додатку, який зможе задовільнити потреби різних категорій користувачів та забезпечити високу масштабованість.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ПРОДАЖУ КОНТАКТНИХ ЛІНЗ, ОКУЛЯРІВ ТА АКСЕСУАРІВ ДО НИХ

3.1 Обґрунтування вибору інструментів технічної реалізації

Для розробки фронтенду, або як її ще називають клієнтської частини, використовують багато мов програмування. Вона може бути написана як програма для комп'ютера з використанням C++, C# чи Java, як програма для смартфона з операційною системою Android, або під смартфон на iOS використовуючи Swift. Тобто, є безліч способів реалізації клієнтської частини. Саме тому потрібно обрати найбільш універсальний спосіб, щоб наш інтернет-магазин працював усюди без зайвих витрат на розробку і підтримку під різні пристрої. Тому було звернено увагу на JavaScript.

JavaScript — це інтерпретована високорівнева мова програмування, яка підтримує імперативний, функціональний і подієво-орієнтовані стилі. Мова Програмування з динамічною типізацією. На синтаксис мови програмування JavaScript вплинули мови C/C++ і Java. Мова програмування офіційно представлена була ще 1995 року. Свої позиції стрімко JS зайняла у зв'язку з розвитком AJAX технології. Коли зробила можливим оновлення інформації на сторінці без перезавантаження браузера. [13]

Однією з особливостей JavaScript є те, що вона виконується у веб-браузері, а вони в свою чергу є на всіх платформах. Тому веб-додатки, написані на JavaScript, здатні працювати на різних пристроях та операційних системах без необхідності додаткової адаптації чи компіляції. Це робить JavaScript ідеальним вибором для створення кросплатформених веб-додатків, таких як інтернет-магазин для окулярів, лінз та аксесуарів до них.

Також JavaScript має ряд переваг перед вище переліченими мовами, які роблять її ідеальним вибором для розробки веб-додатків:

- Кросплатформеність: JavaScript може виконуватися в будь-якому сучасному веб-браузері, незалежно від операційної системи чи пристрою.

- Велика спільнота та екосистема: JavaScript має величезну спільноту розробників, що значно полегшує навчання та вирішення проблем. Крім того, існує велика кількість готових бібліотек та фреймворків, які можна використовувати для прискорення розробки.
- Безперервний розвиток: JavaScript постійно оновлюється та розвивається, додаючи нові функції та можливості.

Таким чином, завдяки своїй кросплатформеності, великій спільноті та постійному розвитку, JavaScript є ідеальним вибором для розробки клієнтської частини веб-додатку інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них. Це забезпечить сумісність з різними пристроями та операційними системами, а також дозволить використовувати сучасні фреймворки та бібліотеки для прискорення розробки та підвищення якості коду.

Також важливим моментом є вибір бібліотек та фреймворків для реалізації клієнтської частини. Популярними виборами є React, Vue та Angular. Пропоную розібрати кожного з них детальніше.

React - це бібліотека JavaScript для створення користувацьких інтерфейсів, розроблена Meta (раніше Facebook) [14]. Вона використовує декларативний підхід до створення UI, що полегшує розробку складних інтерфейсів шляхом розбиття їх на невеликі компоненти. React має переваги у вигляді високої продуктивності завдяки віртуальному DOM та легкості інтеграції з іншими бібліотеками. Однак його недоліком є те, що розробникам доводиться стикатися з певними складнощами через застарілі концепції, наприклад `useEffect` тощо, які можуть бути складними для розуміння та обслуговування. Щоб вирішити цю проблему, розробники React працюють над створенням React Compiler, але його реліз затримується.

Angular - це повнофункціональний фреймворк JavaScript для створення веб-додатків, розроблений компанією Google [15]. Він забезпечує комплексне рішення для розробки клієнтської частини додатків з акцентом на модульність, продуктивність та масштабованість. Angular має потужні інструменти для

маршрутизації, управління станом, форм та ін'єкції залежностей. Однак він має досить крутий нахил кривої навчання та може бути складним для початківців. Крім того, Angular може бути дещо повільнішим у розробці порівняно з іншими фреймворками через свою комплексність.

Vue - це прогресивний фреймворк JavaScript для створення користувацьких інтерфейсів [16]. Він поєднує в собі переваги бібліотек та повноцінних фреймворків, що робить його гнучким і зручним у використанні. Vue має зрозумілий синтаксис, добре масштабується та забезпечує високу продуктивність, а також пропонує вбудовані інструменти для маршрутизації, управління станом та створення компонентів. Його перевагами є його невелика крива навчання, простота інтеграції та активна спільнота розробників. Саме Vue я вибрав для реалізації клієнтської частини інтернет-магазину завдяки його зручності, продуктивності та швидкості розробки.

Крім нього я вибрав ще додаткові інструменти, щоб прискорити та полегшити процес розробки клієнтської частини, а саме Tailwind CSS, Vite, Axios та Pinia.

- Tailwind CSS - це CSS-фреймворк, що дозволяє швидко створювати адаптивні інтерфейси. Він забезпечує модульність коду, гнучкість у налаштуванні дизайну без редагування CSS-файлів і зручні інструменти для створення адаптивного дизайну під різні пристрої.
- Vite - це сучасний інструмент для збірки JavaScript-додатків. Його перевагами є миттєвий запуск сервера розробки без попередньої компіляції, можливість миттєвого оновлення модулів без перезавантаження сторінки та оптимізована збірка, що забезпечує високу продуктивність.
- Axios - це бібліотека для здійснення HTTP-запитів у JavaScript-додатках. Вона пропонує інтуїтивний API для HTTP-запитів, зручне управління асинхронними операціями через обробку відповідей та гнучкість у налаштуванні базового URL, тайм-аутів і заголовків запитів.
- Pinia - це сучасний менеджер стану для Vue.js. Вона забезпечує простоту

та зрозумілість API для управління станом додатку, підтримку TypeScript для безпеки коду та інтеграцію з Vue Devtools для зручного налагодження стану додатку під час розробки.

Також потрібно вибрати мову для бекенду, існує кілька популярних варіантів. Розглянемо їх детально.

Python - це універсальна мова програмування, яка широко використовується для створення веб-додатків. Вона має чистий і зрозумілий синтаксис, що робить її легкою для вивчення та підтримки [17]. Python пропонує потужні фреймворки для розробки веб-додатків, такі як Django та Flask. Однак він може бути повільнішим порівняно з деякими іншими мовами програмування.

Node.js - це середовище виконання JavaScript на стороні сервера. Воно дозволяє використовувати JavaScript як для фронтенду, так і для бекенду, що спрощує розробку та інтеграцію. Node.js відомий своєю високою продуктивністю та асинхронною моделлю, що робить його ідеальним вибором для створення масштабованих додатків. Проте він може бути менш підходящим для ресурсномістких завдань порівняно з деякими іншими мовами.

Java - широко використовується для розробки корпоративних додатків. Вона пропонує багато фреймворків для веб-розробки, таких як Spring та JSF, також відома своєю надійністю, безпекою та кросплатформеністю. Однак вона більш складна для вивчення, та більш часоємна для написання веб-додатків.

PHP - це одна з найпопулярніших мов програмування для розробки веб-додатків. Вона має велику спільноту розробників, безліч бібліотек та фреймворків. PHP часто використовується для створення динамічних веб-сайтів та веб-додатків завдяки своїй простоті та інтеграції з веб-серверами.

Після ретельного аналізу я вирішив обрати PHP для розробки бекенду інтернет-магазину. Ця мова має ряд переваг, які роблять її відмінним вибором для цього проекту:

- Простота у вивченні та використанні. PHP має відносно простий і

зрозумілий синтаксис, що полегшує розробку та супровід веб-додатків, особливо для невеликих команд або індивідуальних розробників.

- Інтеграція з веб-серверами. PHP тісно інтегрований з популярними веб-серверами, такими як Apache та Nginx, що спрощує розгортання та налаштування веб-додатків.
- Підтримка різних баз даних. PHP має вбудовану підтримку різних систем управління базами даних, включаючи MySQL, PostgreSQL та SQLite, що забезпечує гнучкість у виборі рішення для зберігання даних.
- Доступність хостингу. Велика кількість провайдерів хостингу пропонує підтримку PHP, що робить розгортання веб-додатків досить простим і недорогим.
- Велика спільнота та документація. PHP має велику та активну спільноту розробників, яка створює численні ресурси, такі як документація, форуми, блоги та навчальні матеріали. Це робить вирішення проблем та отримання підтримки набагато простішим.
- Масштабованість. PHP добре підходить як для невеликих проєктів, так і для великих масштабованих систем. Багато відомих веб-сайтів, таких як Facebook, Wikipedia та WordPress, були створені за допомогою PHP.

3.2 Розробка схеми алгоритму роботи інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них

Для ефективного функціонування інтернет-магазину необхідно розробити чіткий алгоритм його роботи. Цей алгоритм можна представити у вигляді блок-схем, які наочно демонструють послідовність дій та логіку прийняття рішень на кожному етапі.

Давайте розглянемо алгоритм авторизації користувача (рис. 3.1):

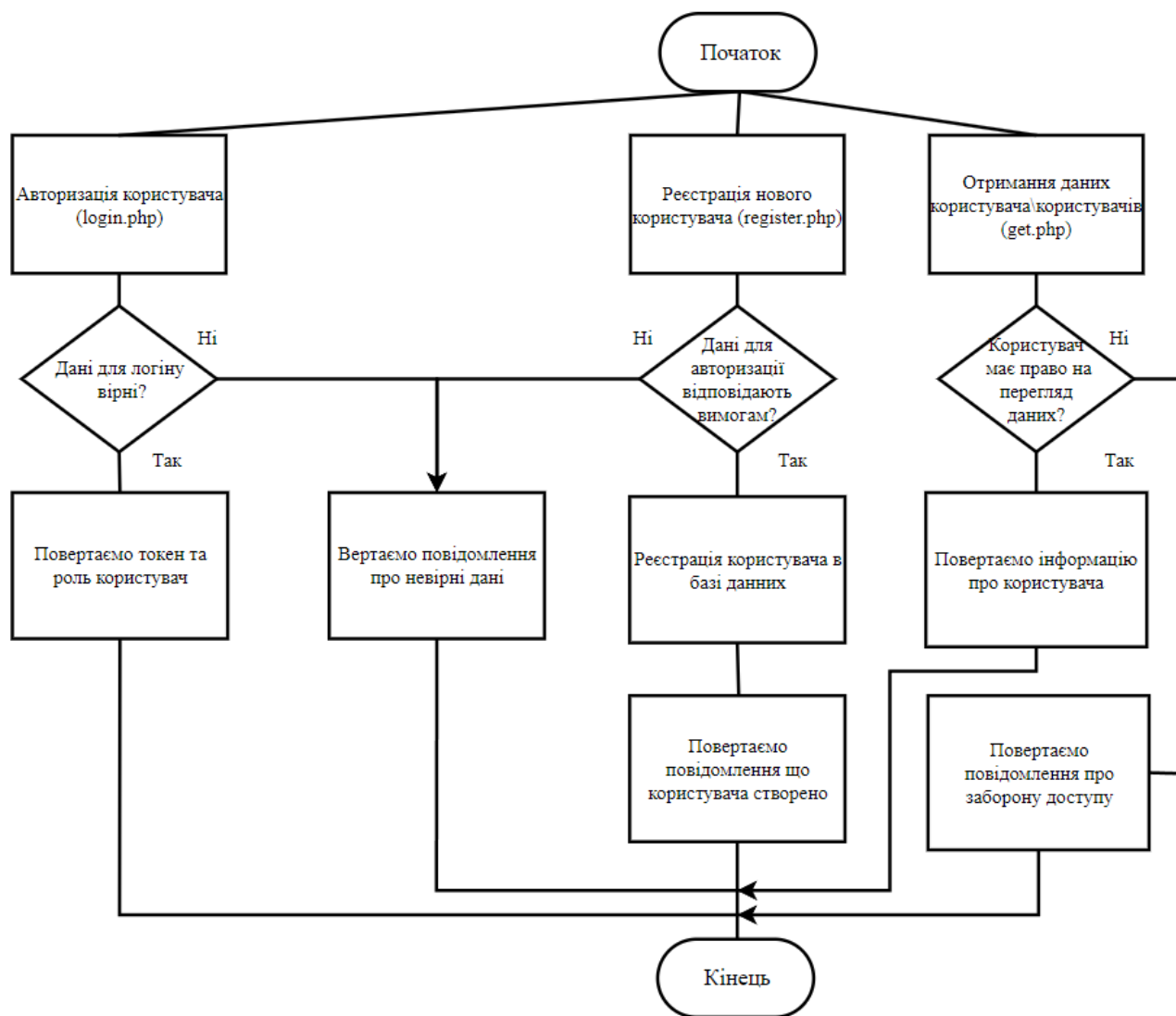


Рисунок 3.1 – Алгоритм авторизації користувача

Дана блок-схема демонструє процес обробки різних запитів для дій з користувачами. Початковий етап схеми передбачає вибір дії користувача: авторизація (login.php), реєстрація (register.php) або отримання даних користувача (get.php). Якщо користувач обирає авторизацію, система перевіряє вірність введених даних. У разі успіху повертається токен та роль користувача, інакше - повідомлення про невірні дані. При реєстрації нового користувача система перевіряє, чи відповідають дані вимогам. Якщо так, користувача реєструють у базі даних і повертають повідомлення про створення користувача, інакше повертається повідомлення про невірні дані. При отриманні даних користувача система перевіряє права доступу: якщо користувач має право перегляду, повертається інформація про користувача,

інакше - повідомлення про заборону доступу.

Алгоритм по якому будуть працювати товари не менш важливий ніж алгоритм авторизації чи реєстрації, адже користувачі впершу чергу будуть взаємодіяти з ними. Тому розглянемо його (рис. 3.2):

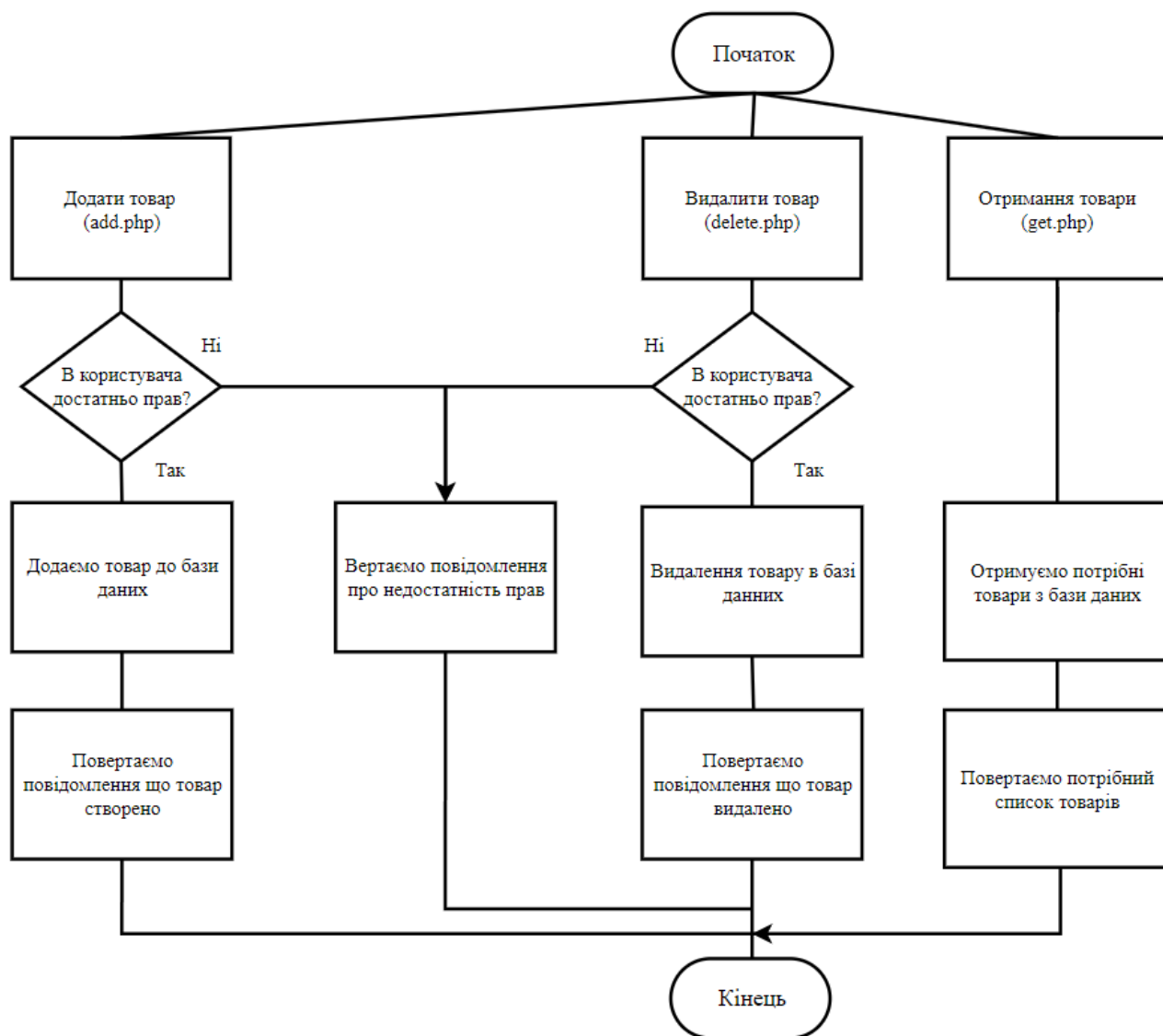


Рисунок 3.2 – Алгоритм роботи товарів

Процес починається з визначення операції, яку потрібно виконати над товарами: додати новий товар (add.php), видалити існуючий товар (delete.php) або отримати список товарів (get.php). Якщо користувач обирає додавання товару, система перевіряє наявність достатніх прав. У разі успіху товар додається до бази даних, після чого повертається повідомлення про створення

товару; якщо прав недостатньо, повертається повідомлення про їх недостатність. При видаленні товару система також перевіряє права користувача: якщо права достатні, товар видаляється з бази даних і повертається повідомлення про видалення товару; у разі недостатності прав повертається відповідне повідомлення. Для отримання товарів система отримує потрібні товари з бази даних і повертає необхідний список товарів. Кінцевий етап схеми позначає завершення процесу обробки запиту.

Тепер розглянемо алгоритм роботи з замовленнями (рис. 3.3).

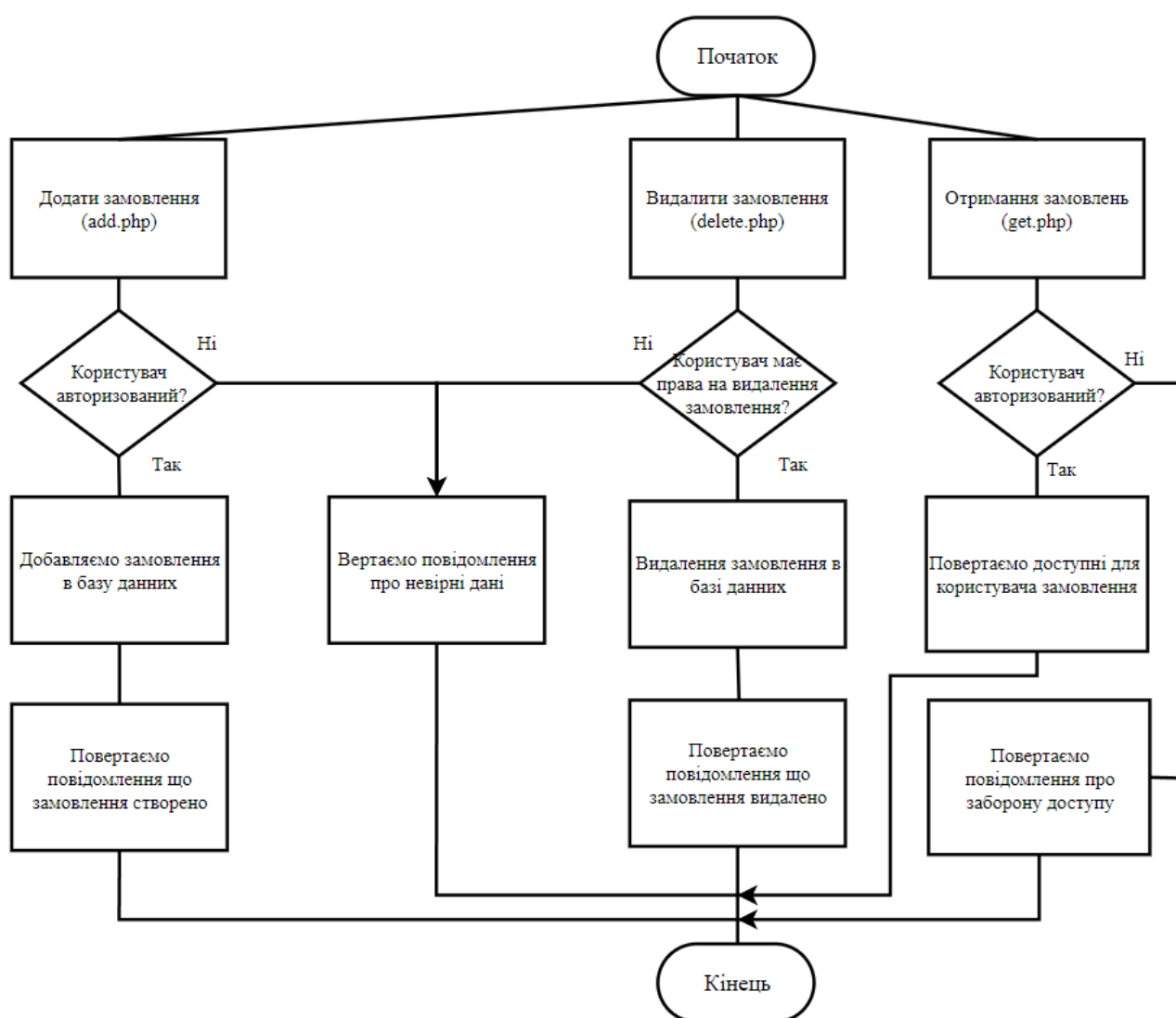


Рисунок 3.3 – Алгоритм роботи замовлень

Ця блок-схема демонструє процес обробки запитів для управління замовленнями в системі. Початок схеми передбачає вибір дії користувача:

додавання замовлення (add.php), видалення замовлення (delete.php) або отримання замовлень (get.php). Якщо користувач обирає додавання замовлення, система перевіряє, чи авторизований користувач. У разі успіху замовлення додається до бази даних і повертається повідомлення про створення замовлення; якщо користувач не авторизований, повертається повідомлення про невірні дані. При видаленні замовлення система перевіряє, чи має користувач права на видалення замовлення. Якщо права достатні, замовлення видаляється з бази даних і повертається повідомлення про видалення замовлення; якщо прав недостатньо, повертається повідомлення про недостатність прав. Для отримання замовлень система перевіряє авторизацію користувача: якщо користувач авторизований, повертається доступний список замовлень; якщо не авторизований, повертається повідомлення про заборону доступу. Кінцевий етап схеми позначає завершення процесу обробки запиту.

3.3 Обґрунтування вибору середовища розробки

Вибір відповідного середовища розробки (IDE) є важливим кроком у процесі створення веб-додатку, оскільки воно забезпечує зручний та продуктивний робочий простір для написання коду, налагодження, тестування та розгортання програми. На ринку існує велика кількість IDE, кожна з яких має свої переваги та недоліки. Ми розглянемо три популярні IDE: Visual Studio Code, WebStorm та Sublime Text, та обґрунтуємо вибір Visual Studio Code для розробки інтернет-магазину.

Visual Studio Code (VS Code) - це безкоштовний, крос-платформний редактор вихідного коду, розроблений компанією Microsoft. Він має відкритий вихідний код і підтримує численні мови програмування, включаючи JavaScript, TypeScript, PHP, Python та багато інших. VS Code пропонує зручний та налаштовуваний інтерфейс, що робить його привабливим для розробників різних рівнів досвіду [18]. Зовнішній вигляд зображено на рисунку 3.4.

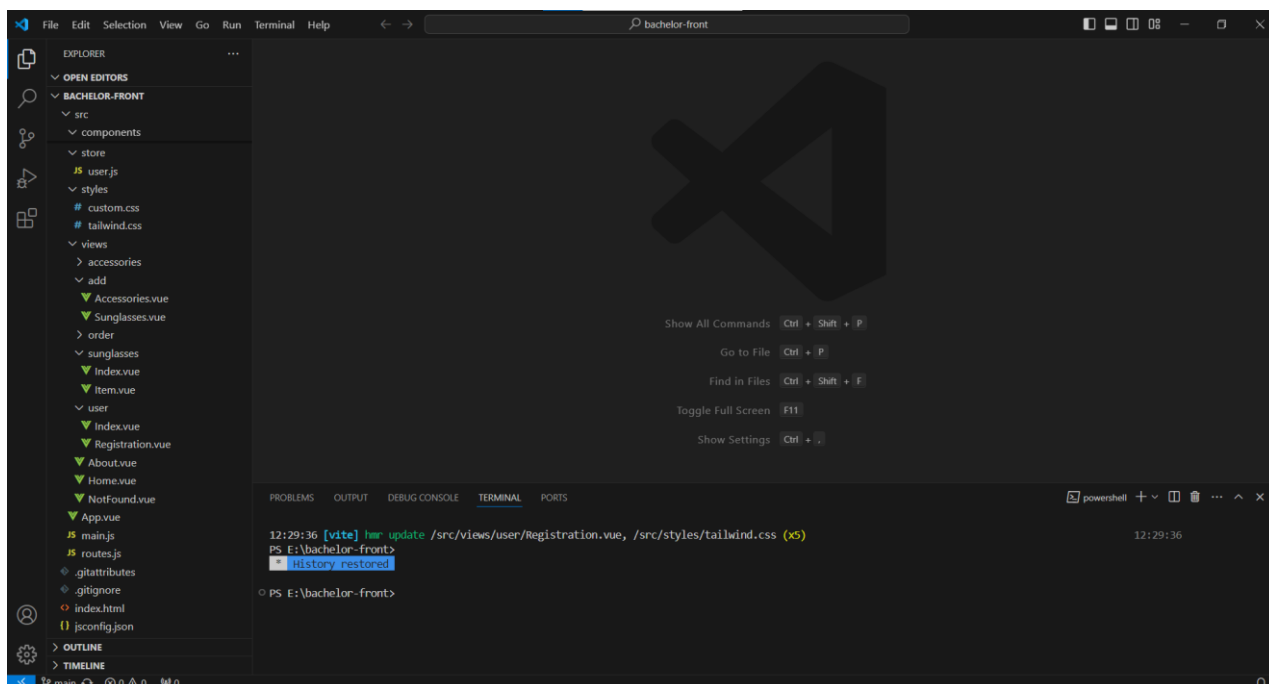


Рисунок 3.4 – Вигляд Visual Studio Code

Переваги Visual Studio Code:

- Легкість у використанні та зручний інтерфейс: VS Code має інтуїтивно зрозумілий інтерфейс з можливістю налаштування, що дозволяє розробникам персоналізувати своє робоче середовище відповідно до своїх потреб та вподобань.
- Підтримка великої кількості мов та фреймворків: VS Code підтримує широкий спектр мов програмування, фреймворків та бібліотек, що робить його ідеальним вибором для розробки веб-додатків з використанням різноманітних технологій, таких як JavaScript (React, Vue.js, Node.js) та PHP.
- Інтеграція з інструментами контролю версій: VS Code має вбудовану підтримку популярних систем контролю версій, таких як Git, що спрощує процес спільної роботи над проектами.
- Потужні можливості налагодження: VS Code пропонує зручні інструменти для налагодження коду, включаючи точки зупинки, перегляд змінних, крок-за-кроком виконання та інші функції, що допомагають виявляти та усувати помилки.

- Розширення та інтеграції: Завдяки великій кількості розширень та інтеграцій, користувачі можуть розширити функціональність VS Code відповідно до своїх потреб, додавши підтримку нових мов, фреймворків, інструментів та плагінів.
- Крос-платформеність: VS Code доступний для Windows, macOS та Linux, що забезпечує гнучкість та зручність для розробників, які працюють на різних операційних системах.

WebStorm - це потужне комерційне середовище розробки, розроблене компанією JetBrains. Воно орієнтоване на веб-розробку і пропонує всебічну підтримку для JavaScript, TypeScript, React, Angular, Vue.js, Node.js та інших веб-технологій [19]. Зовнішній вигляд зображено на рисунку 3.5.

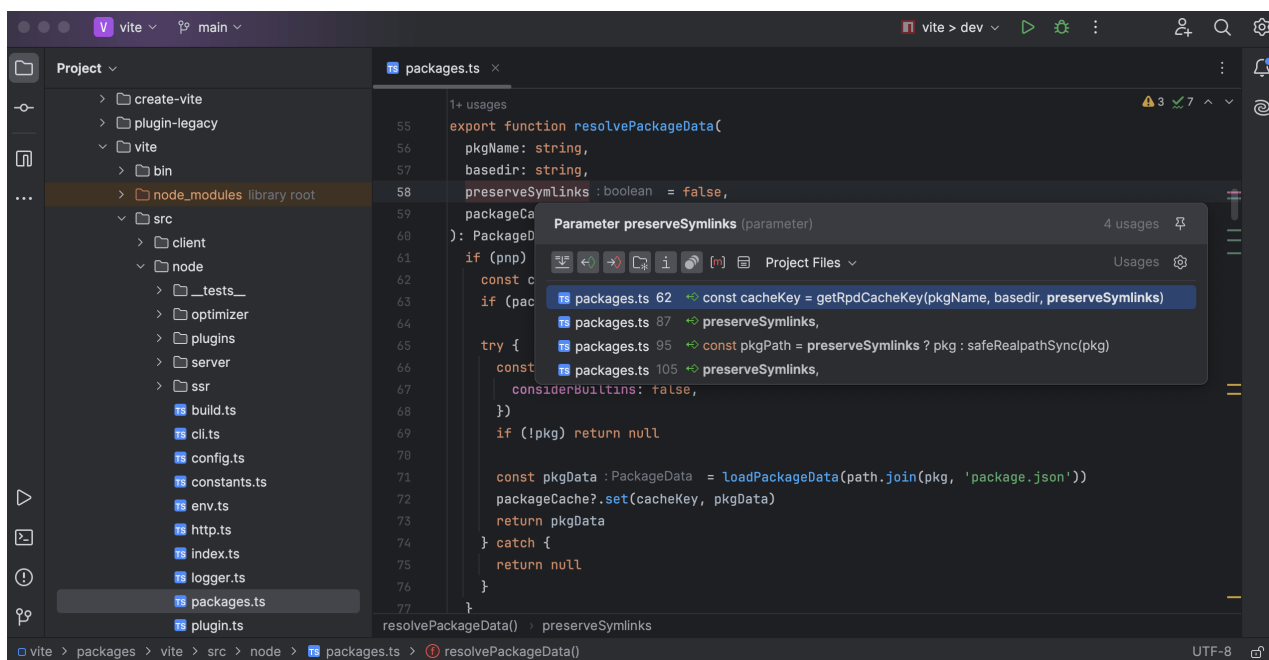


Рисунок 3.5 – Вигляд WebStorm

Переваги WebStorm:

- Інтелектуальне автозаповнення та перевірка коду: WebStorm має потужні функції автозаповнення та перевірки коду, що допомагає запобігати помилкам та підвищувати продуктивність розробки.
- Інтегроване середовище для налагодження: WebStorm пропонує вбудовані інструменти для налагодження та профілювання коду, що

полегшує виявлення та усунення помилок.

- Підтримка популярних фреймворків та бібліотек: WebStorm має вбудовану підтримку для багатьох популярних веб-фреймворків та бібліотек, таких як React, Angular, Vue.js, Node.js та Redux.
- Підтримка TypeScript: WebStorm надає повну підтримку для TypeScript, включаючи автозаповнення, рефакторинг та налагодження, що робить його ідеальним інструментом для розробників, які використовують цей мову.
- Можливості для рефакторингу: WebStorm надає широкий спектр інструментів для рефакторингу коду, що дозволяє легко та безпечно вносити зміни в структуру проекту.
- Інтеграція з інструментами розробки: WebStorm інтегрується з різними інструментами розробки, такими як системи контролю версій (Git, SVN), менеджери завдань (Jira, Trello) та інструменти безперервної інтеграції.
- Потужні можливості налагодження та тестування: WebStorm пропонує зручні інструменти для налагодження та тестування коду, включаючи юніт-тестування, автоматизоване тестування інтерфейсу та інтеграцію з тестовими фреймворками.

Проте WebStorm є комерційним продуктом, що може бути дорогим для деяких розробників або невеликих команд.

Sublime Text - це крос-платформений текстовий редактор вихідного коду з акцентом на продуктивність та зручність використання [20]. Він має широкі можливості налаштування та підтримує численні мови програмування, включаючи JavaScript, PHP, Python та інші. Зовнішній вигляд зображено на рисунку 3.6.

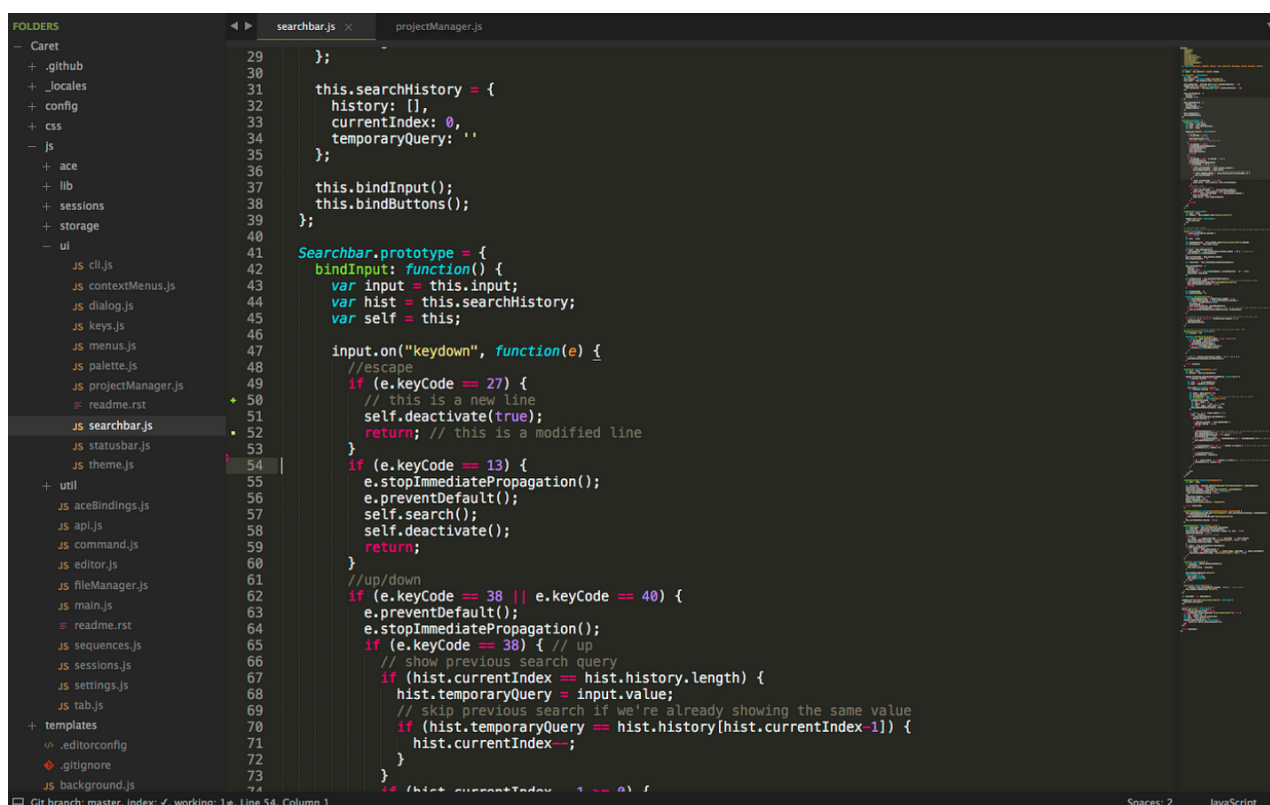


Рисунок 3.6 – Вигляд Sublime Text

Переваги Sublime Text:

- Швидкість та продуктивність: Sublime Text відомий своєю високою швидкістю та продуктивністю, що забезпечує плавну роботу навіть з великими файлами вихідного коду.
- Налаштовуваність та розширення: Sublime Text пропонує широкі можливості налаштування, підтримуючи велику кількість користувацьких налаштувань та розширень, які можна встановлювати та налаштовувати відповідно до своїх потреб.
- Крос-платформеність: Sublime Text доступний для Windows, macOS та Linux, що забезпечує гнучкість та зручність для розробників, які працюють на різних операційних системах.
- Зручний інтерфейс та функціональність: Sublime Text пропонує зручний інтерфейс та корисні функції, такі як відкриття декількох файлів одночасно, швидкий пошук, заміна тексту та багато іншого.
- Підтримка плагінів спільноти: Sublime Text має активну спільноту

розробників, які створюють та підтримують плагіни, що розширюють функціональність редактора.

Однак Sublime Text не пропонує вбудованих можливостей для налагодження та тестування коду, що може бути недоліком для деяких розробників.

Після ретельного аналізу переваг та недоліків кожного з розглянутих середовищ розробки, я вирішив обрати Visual Studio Code (VS Code) для розробки інтернет-магазину.

3.4 Розробка клієнтської та серверної частин інтернет-магазину

На серверній частині, як вже було вирішено в минулих розділах, буде використовуватися PHP для обробки запитів від клієнтської частини. А також взаємодія з базою даних та реалізація основної бізнес-логіки додатку буде саме тут. Для початку було створено файл `main.php`, який містить основні функції та налаштування, такі як підключення до бази даних та визначення ролей користувачів:

```
<?php
define('DB_PATH', realpath(dirname(__FILE__) . '/../database/index.db'));
// Підключення до БД
$db = new SQLite3(DB_PATH);
// Основні ролі
$roles = array(
    'user' => 0,
    'editor' => 1,
    'admin' => 2
);
function getDB() {
    global $db;
    return $db;
```

```

}
function getUserRole($token) {
    // ... логіка для отримання ролі користувача за токеном
}
function getUserID($token) {
    // ... логіка для отримання ID користувача за токеном
}

```

Як видно цей файл слугує для визначення константи `DB_PATH` для шляху до файлу бази даних SQLite, створювання з'єднання з базою даних за допомогою класу `SQLite3`, визначення масиву `$roles` з основними ролями користувачів, а також містить допоміжні функції `getDB()`, `getUserRole()` та `getUserID()` для взаємодії з базою даних та обробки даних користувачів.

Це необхідно, щоб часто використовувані функції та головний код не повторювалися в кожному файлі. Для прикладу розглянемо файл `get_user.php`, що обробляє GET-запити для отримання даних про користувача за його токеном:

```

<?php
require '../core/main.php';
if ($_SERVER['REQUEST_METHOD'] === 'GET') {
    // ... логіка для перевірки токена та отримання ролі через getUserRole()
    if ($roleId === false) {
        http_response_code(401);
        echo json_encode(['error' => 'Невірний токен']);
        exit;
    }
    $db = getDB();
    $stmt = $db->prepare('SELECT UserID, Username, Email, FullName, RoleID
FROM Users WHERE Token = :token');
    $stmt->bindValue(':token', $token, SQLITE3_TEXT);
    $result = $stmt->execute();
}

```

```

$userData = $result->fetchArray(SQLITE3_ASSOC);
if ($userData) {
    echo json_encode($userData);
} else {
    http_response_code(404);
    echo json_encode(['error' => 'Користувач не знайдений']);
}
}

```

Цей файл спочатку перевіряє, чи надійшов GET-запит, а потім перевіряє наявність токена в заголовках запиту. Якщо токен присутній, функція `getUserRole()` з файлу `main.php` використовується для перевірки дійсності токена та отримання ролі користувача. Якщо токен дійсний, виконується SQL-запит для отримання даних про користувача з таблиці `Users` за його токеном. Результат запиту відправляється у форматі JSON як відповідь на запит.

Також пропоную розглянути файл `register.php`, який відповідає за обробку запитів для реєстрації нових користувачів в системі. Ось як він працює:

```

<?php
require '../core/main.php';

function register($username, $password, $email, $fullName) {
    $db = getDB();
    $stmt = $db->prepare('
        INSERT INTO Users (
            Username,
            Password,
            Email,
            FullName,
            Token,
            RoleID
        ) VALUES (

```

```

        :username,
        :password,
        :email,
        :fullName,
        :token,
        0
    )
);
$stmt->bindValue(':username', $username, SQLITE3_TEXT);
$stmt->bindValue(':password', password_hash($password,
PASSWORD_DEFAULT), SQLITE3_TEXT);
$stmt->bindValue(':email', $email, SQLITE3_TEXT);
$stmt->bindValue(':fullName', $fullName, SQLITE3_TEXT);
$stmt->bindValue(':token', bin2hex(random_bytes(32)), SQLITE3_TEXT);
$stmt->execute();
return $db->lastInsertRowID();
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = json_decode(file_get_contents('php://input'), true);
    // Перевірка, чи заповнені всі необхідні поля
    if (!isset($data['username']) || !isset($data['password']) || !isset($data['email']) ||
!isset($data['fullName'])) {
        http_response_code(400);
        echo json_encode(['error' => 'Не всі необхідні поля заповнені']);
        return;
    }
    // Перевірка унікальності імені користувача та електронної пошти
    $db = getDB();
    $stmt = $db->prepare('SELECT COUNT(*) FROM Users WHERE Username =
:username OR Email = :email');

```

```

$stmt->bindValue(':username', $data['username'], SQLITE3_TEXT);
$stmt->bindValue(':email', $data['email'], SQLITE3_TEXT);
$result = $stmt->execute();
$count = $result->fetchArray(SQLITE3_NUM)[0];
if ($count > 0) {
    http_response_code(400);
    echo json_encode(['error' => 'Ім'я користувача або електронна пошта вже
зайняті']);
    return;
}
// Реєстрація нового користувача
$userId = register($data['username'], $data['password'], $data['email'],
$data['fullName']);
if ($userId) {
    echo json_encode(['message' => 'Користувач успішно зареєстрований',
'userId' => $userId]);
} else {
    http_response_code(500);
    echo json_encode(['error' => 'Сталася помилка під час реєстрації']);
}
}

```

Як видно, цей файл спочатку перевіряє, чи надійшов POST-запит, якщо так, він отримує дані з тіла запиту у форматі JSON (username, password, email, fullName). Далі виконується перевірка, чи всі необхідні поля заповнені. Якщо ні, повертається помилка 400. Потім здійснюється перевірка унікальності імені користувача та електронної пошти в базі даних. Якщо такі вже існують, повертається помилка 400 з повідомленням про зайняті дані.

Якщо всі перевірки пройдені успішно, викликається функція register(), яка вставляє нового користувача в базу даних. Ця функція хешує пароль за допомогою password_hash() для безпечного зберігання, генерує унікальний

токен для користувача за допомогою `bin2hex(random_bytes(32))` та присвоює роль користувача за замовчуванням (`RoleID = 0` для звичайного користувача).

Якщо реєстрація пройшла успішно, повертається JSON-відповідь з повідомленням про успіх та ідентифікатором новоствореного користувача (`userId`). Якщо сталася помилка під час реєстрації, повертається помилка 500.

Такий підхід забезпечує безпечну реєстрацію нових користувачів з перевіркою унікальності даних, хешуванням паролів та генерацією токенів. Це є важливим аспектом для захисту даних користувачів та забезпечення безпеки веб-додатку.

Такий підхід дозволяє чітко розділити функції на файли, зробити роботу зручною з серверною частиною та забезпечує безпечну обробку даних з взаємодією з базою даних на бекенді.

Тепер розглянемо клієнтську частину. Для її розробки було обрано `Vue.js` – прогресивний JavaScript фреймворк для створення користувацьких інтерфейсів. Він поєднує в собі переваги бібліотек та повноцінних фреймворків, що робить його гнучким і зручним у використанні.

Структура клієнтської частини складається з наступних основних компонентів:

- `App.vue` - головний компонент, який відповідає за відображення загального макету додатку та навігацію.
- `Routes.js` - файл для налаштування маршрутизації додатку та визначення шляхів для різних сторінок.
- `Main.js` - файл для ініціалізації додатку, та служить точкою входу.
- `views` - папка з `Vue`-компонентами, які відповідають за відображення окремих сторінок, таких як головна сторінка, сторінка каталогу, сторінка товару, сторінка авторизації тощо.
- `components` - папка з `Vue`-компонентами, які використовуються для створення повторюваних елементів інтерфейсу, таких як навігаційне меню, футер, картка товару тощо.
- `store` - модулі для управління станом додатку за допомогою `Pinia`, які

відповідають за зберігання та оновлення даних.

- styles - папка для зберігання CSS стилів.

Розглянемо код з main.js для розуміння основи функціонування додатку.

Спочатку необхідно імпортувати необхідні модулі, щоб потім правильно їх налаштувати:

```
import { createApp } from 'vue'
import axios from "axios";
import '@/styles/tailwind.css'
import '@/styles/custom.css'
import App from '@/App.vue'
import { routes } from '@/routes.js'
import { createRouter, createWebHistory } from 'vue-router'
import { createPinia } from 'pinia'
```

Код вище імпортує необхідні модулі та компоненти для роботи додатку, такі як Vue.js, Axios, стилі, головний компонент, маршрути, маршрутизатор та менеджер стану Pinia.

```
const app = createApp(App)
// Базовий url
axios.defaults.baseURL = "/t/src"
// Максимальний час запиту
axios.defaults.timeout = 9999;
```

Тут створюється новий екземпляр Vue-додатку, а також налаштовуються базова URL-адреса та тайм-аут для HTTP-запитів через Axios.

```
const router = createRouter({
  history: createWebHistory(),
  routes,
  scrollBehavior(to, from, savedPosition) {
    if (savedPosition) {
      return savedPosition
    } else {
```

```

    return { top: 0 }
  }
},
})

```

Далі створюємо новий екземпляр маршрутизатора Vue Router з певними налаштуваннями, такими як використання HTML5 History API та з певною поведінкою прокручування сторінки.

```

router.beforeEach((to, from, next) => {
  next();
  // Зімення scroll visibility
  document.body.style.overflowY = to.meta.showScroll ? 'scroll' : 'visible';
  // Title & Description сторінки
  document.title = to.meta.title+" - магазин";
  // OG
  if(navigator.userAgent.toLowerCase().indexOf('prerender') !== -1) {
    document.querySelector('meta[property="og:title"]').setAttribute("content",
to.meta.title+" - магазин");
    document.querySelector('meta[property="og:description"]').setAttribute("content",
to.meta.description);
    document.querySelector('meta[name="description"]').setAttribute("content",
to.meta.description);
    document.querySelector('meta[property="og:url"]').setAttribute("content",
window.location);
    to.meta.image
    document.querySelector('meta[property="og:image"]').setAttribute("content",
window.location.protocol+'//'+window.location.hostname+to.meta.image);
  }
})

```

Цей кусок коду визначає функцію, яка виконується перед кожним переходом між маршрутами. Вона змінює видимість прокручування сторінки,

встановлює заголовок вікна браузера та оновлює метатеги для соціальних мереж.

```
app.use(createPinia())
```

```
app.use(router)
```

```
app.mount('#app')
```

Залишилось підключити Pinia та маршрутизатор до додатку та змонтувати Vue-додаток до елемента з ідентифікатором #app на сторінці.

Тепер розглянемо файл app.vue, він відповідає за загальний макет та відображення сторінок веб-додатку. Для початку проаналізуємо шаблон:

```
<template>
```

```
<div class="bg-gray-200 dark:bg-gray-800 transition-colors min-h-screen">
```

```
<header>
```

```
<Navbar/>
```

```
</header>
```

```
<main class="mx-auto max-w-7xl px-4 sm:px-6 lg:px-8 pt-18 sm:pt-20 lg:pt-24">
```

```
<router-view v-slot="{ Component }">
```

```
<keep-alive :max="5" exclude="notAlive">
```

```
<component :is="Component" />
```

```
</keep-alive>
```

```
</router-view>
```

```
</main>
```

```
</div>
```

```
</template>
```

Тут можна побачити розмітку головного компонента, де відображається компонент Navbar, а в основній частині - компонент, який відповідає поточному маршруту. Компоненти, що були відображені раніше, зберігаються в кеші keep-alive для прискорення завантаження.

```
<script setup>
```

```
import NavbarLoader from '@components/NavbarLoader.vue'
```

```
import { defineAsyncComponent } from 'vue'
```

```
const Navbar = defineAsyncComponent({
  loader: () => import('@components/Navbar.vue'),
  loadingComponent: NavbarLoader,
  delay: 0,
})
</script>
```

Цей скрипт імпортує необхідні компоненти та визначає асинхронне завантаження компонента `Navbar.vue` та під час завантаження буде показуватися компонент-заповнювач `NavbarLoader` для зменшення основного бандлу.

Таким чином, код `main.js` встановлює основні налаштування додатку, такі як створення екземпляра `Vue`, налаштування `Axios`, маршрутизатора, підключення `Pinia` та монтування додатку. А код `App.vue` визначає загальну структуру додатку, де відображається компонент навігації, а в основній частині - компоненти, що відповідають поточному маршруту, з використанням кешування для прискорення завантаження.

3.5 Тестування розробленого програмного продукту

Тестування є невід'ємною частиною процесу розробки програмного забезпечення, оскільки воно допомагає виявляти та виправити помилки, забезпечити відповідність вимогам. Для інтернет-магазину необхідно провести ретельне тестування як клієнтської, так і серверної частин, щоб забезпечити якість та стабільність роботи.

Плануючи веб-додаток було визначено вимоги до програмного продукту, а саме:

- Надійна система реєстрації та авторизації для захисту облікових записів користувачів.
- Зручний адаптивний інтерфейс для доступу з різних пристроїв.
- Каталог товарів з функціями пошуку та сортування за різними

критеріями.

- Наявність детальної інформації про товари, включно з фото та описами.

Почнемо розглядати з першого пункту, а саме системи реєстрації\авторизації користувача. Ці сторінки повинні мати простий та інтуїтивно зрозумілий дизайн, щоб користувачі могли легко розібратися і швидко пройти процес. Важливо, щоб форма реєстрації містила мінімальну кількість полів, була адаптивною, забезпечуючи зручний доступ як з настільних комп'ютерів, так і з мобільних пристроїв. Важливо також забезпечити візуальний зворотний зв'язок, наприклад, підсвічування вибраного поля. Зовнішній вигляд сторінки авторизації та реєстрації зображено на рисунку 3.7 та 3.8 відповідно:

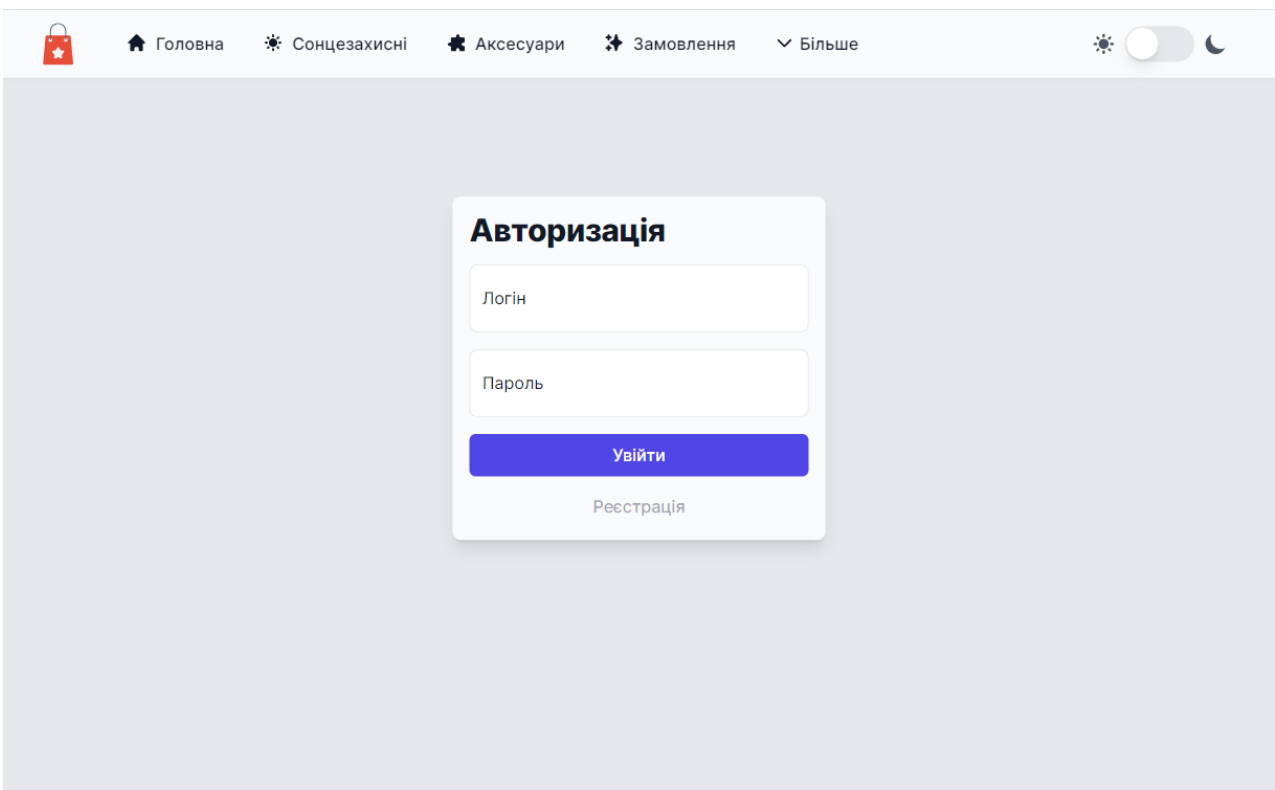
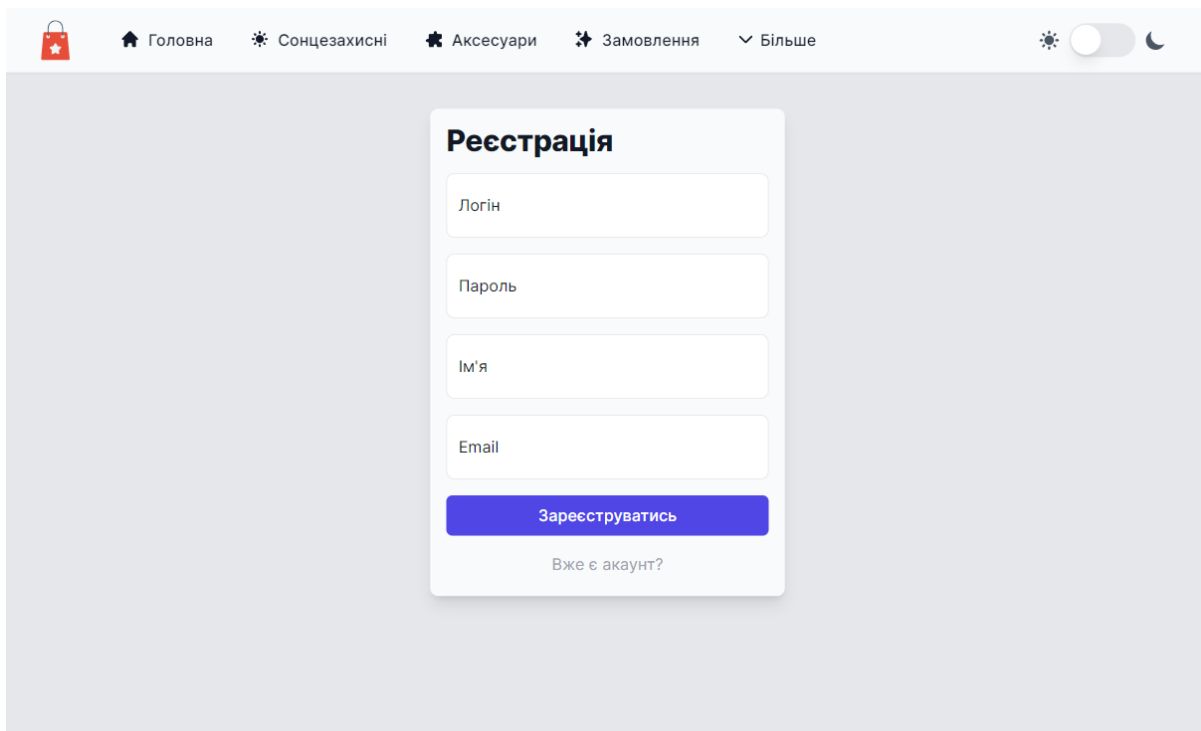


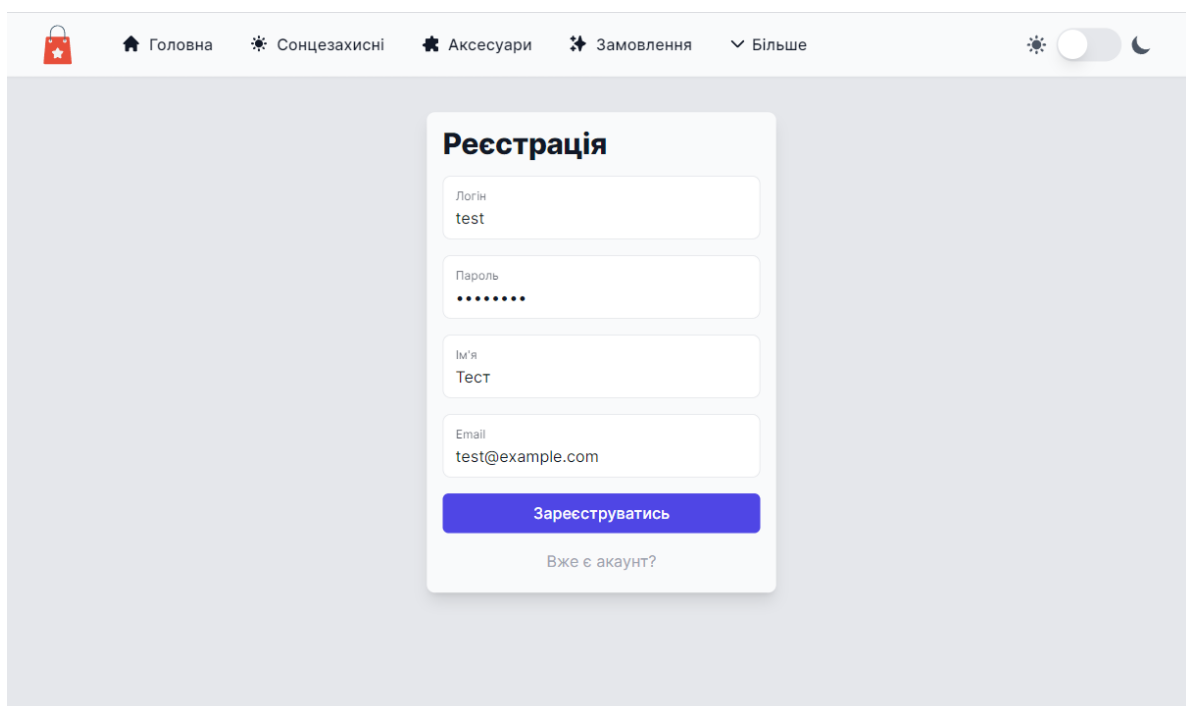
Рисунок 3.7 – Сторінка авторизації



The screenshot shows a web application header with a shopping bag icon, a home icon, and navigation links: Головна, Сонцезахисні, Аксесуари, Замовлення, and Більше. A toggle switch for dark mode is on the right. The main content area features a registration form titled "Регістрація". The form contains four input fields: "Логін", "Пароль", "Ім'я", and "Email". Below these fields is a blue button labeled "Зареєструватись" and a link "Вже є акаунт?".

Рисунок 3.8 – Сторінка реєстрації

Для перевірки створимо профіль з логіном test, іменем Тест, паролем test1234 та поштою test@example.com. Результат зображено на рисунках 3.9 та 3.10:



This screenshot shows the same registration form as Figure 3.8, but with the following data entered: "Логін" is "test", "Пароль" is masked with eight dots, "Ім'я" is "Тест", and "Email" is "test@example.com". The "Зареєструватись" button and the "Вже є акаунт?" link remain visible at the bottom of the form.

Рисунок 3.9 – Сторінка реєстрації з заповненими даними

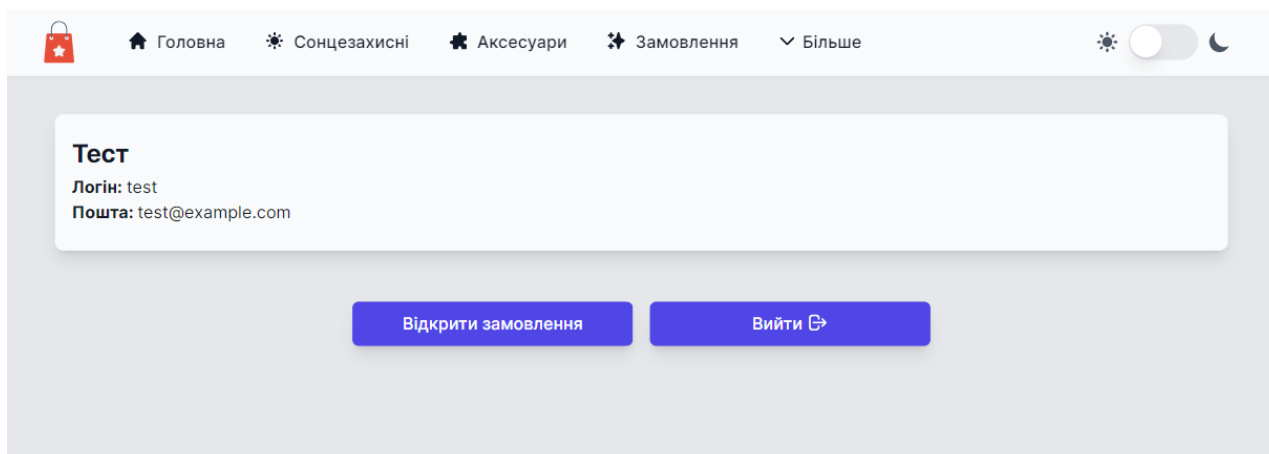


Рисунок 3.10 – Сторінка профілю після авторизації

Після заповнення даних на сторінці авторизації користувач попадає в свій щойно створений профіль, де він бачить основну свою інформацію, кнопку "Вийти" та "Відкрити замовлення". Натиснувши кнопку "Відкрити замовлення" він зможе переглянути його активні замовлення. Якщо у користувача є права редактора або адміністратора, то він зможе побачити всі замовлення.

Далі потрібно перевірити роботу інтерфейсу на різних розмірах екрану. Для тестування версії для телефонів було обрано розмір iPhone SE (1 покоління), який має розмір 375 на 667 пікселів. Для планшетів було вибрано iPad mini з розміром екрану 768 на 1024 пікселів. Для версії на комп'ютері було вибрано розмір 1280 на 720 пікселів. Вигляд сторінки сонцезахисних окулярів на цих пристроях продемонстровано на рисунках 3.11, 3.12 та 3.13 відповідно.

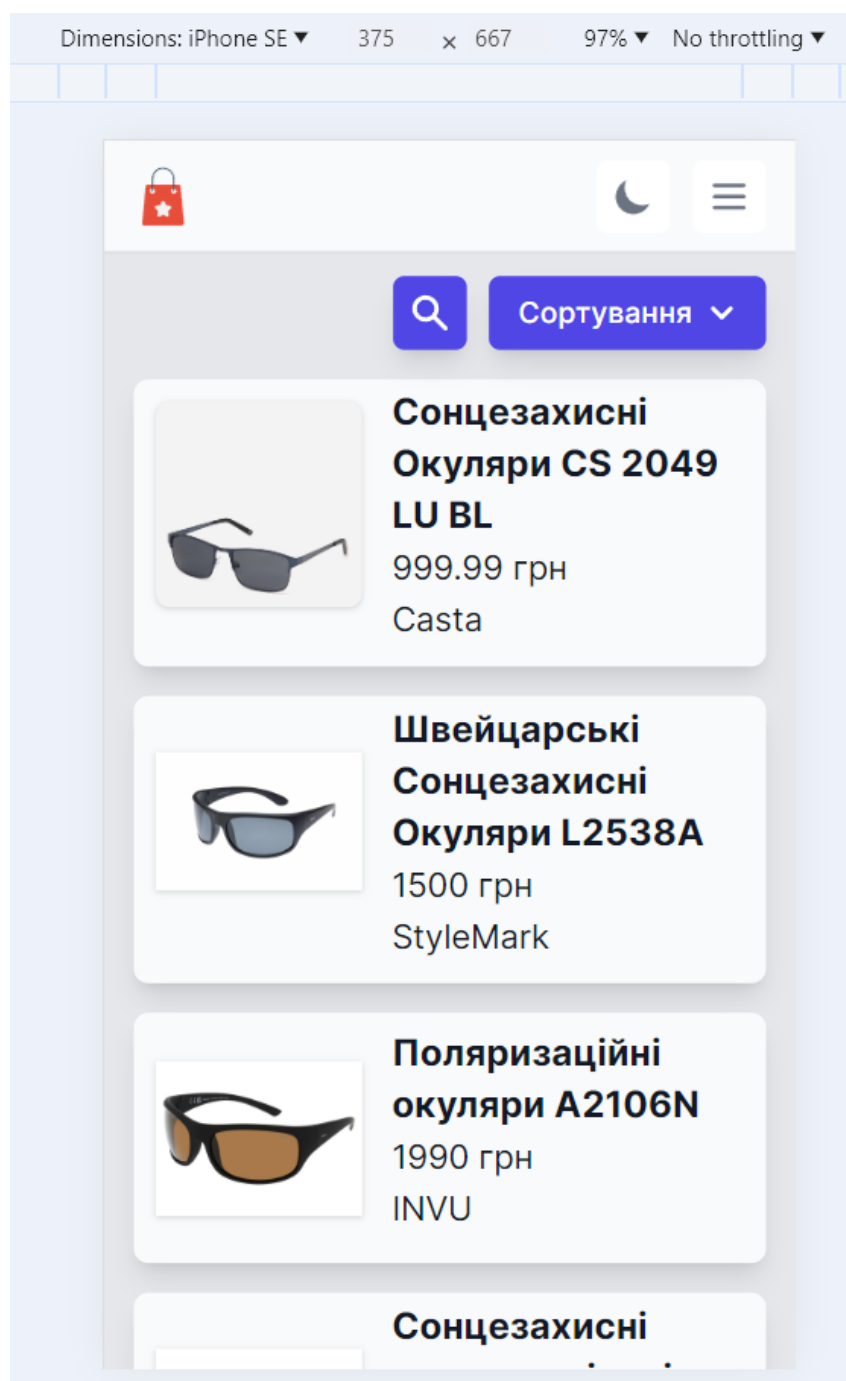


Рисунок 3.11 – Сторінка з сонцезахисними окулярами на телефоні

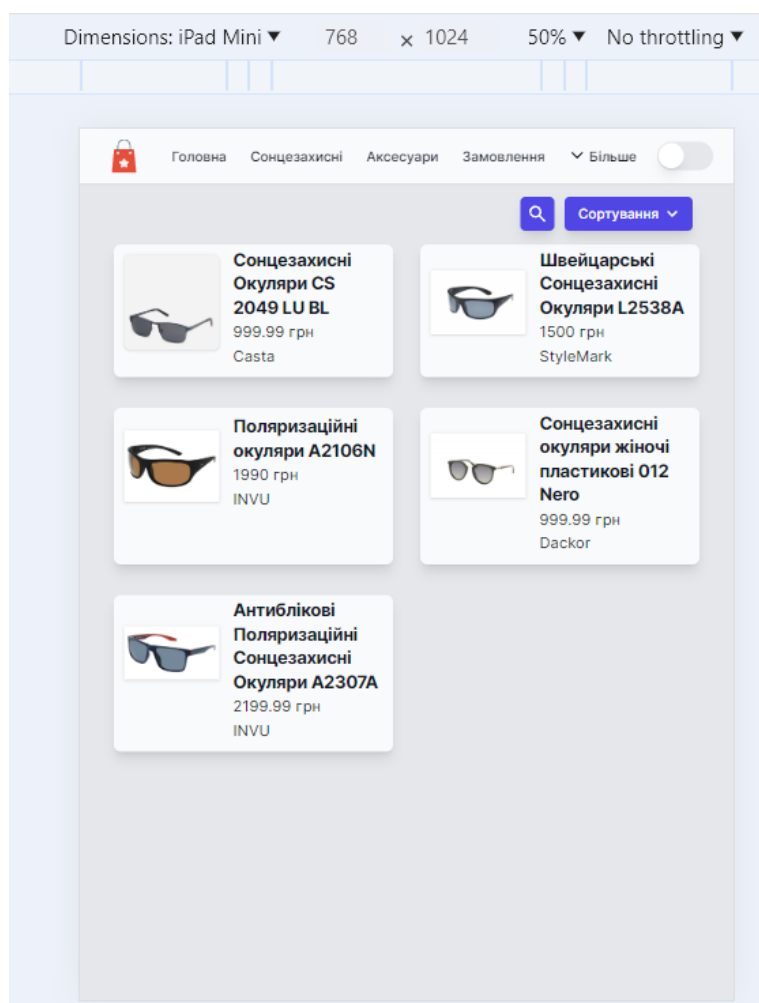


Рисунок 3.12 – Сторінка з сонцезахисними окулярами на планшетах

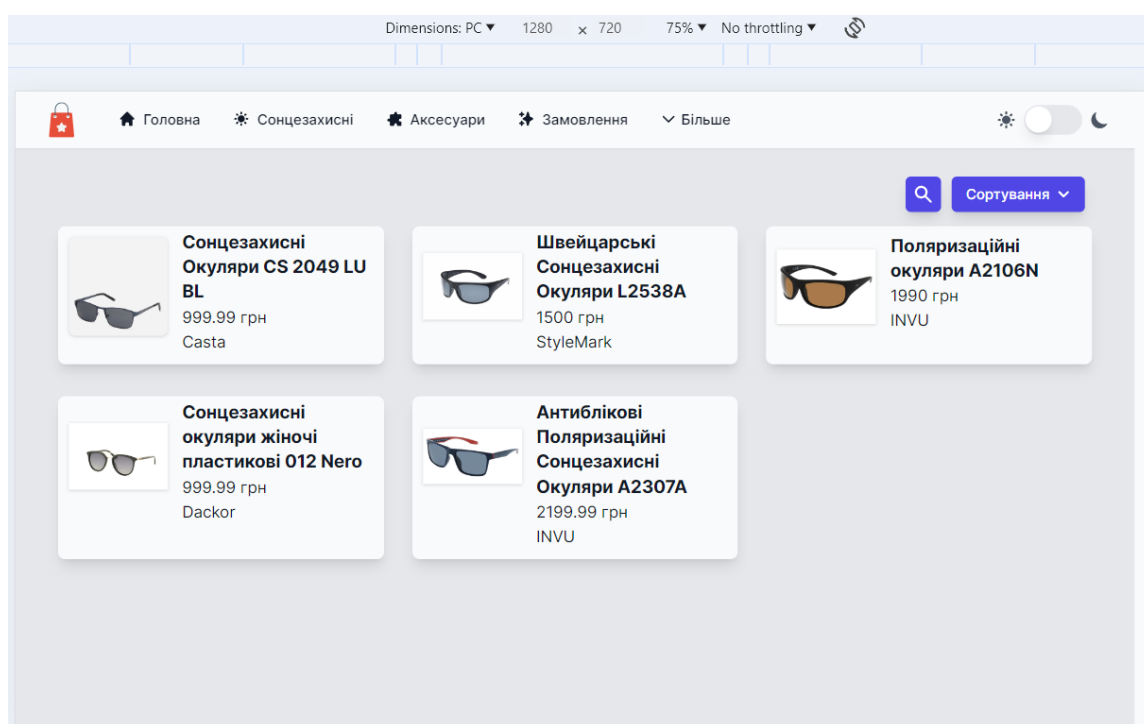


Рисунок 3.13 – Сторінка з сонцезахисними окулярами на ПК

Також перевіримо каталог товарів з сортуванням та пошуком.

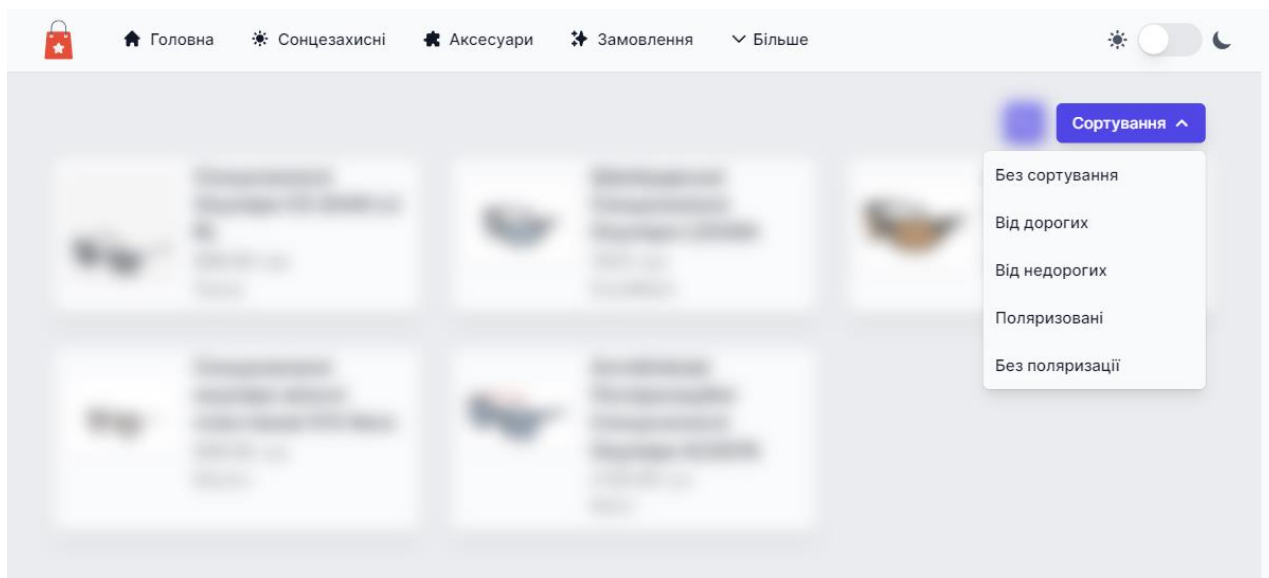


Рисунок 3.14 – Сторінка сонцезахисних окулярів з відкритим сортуванням

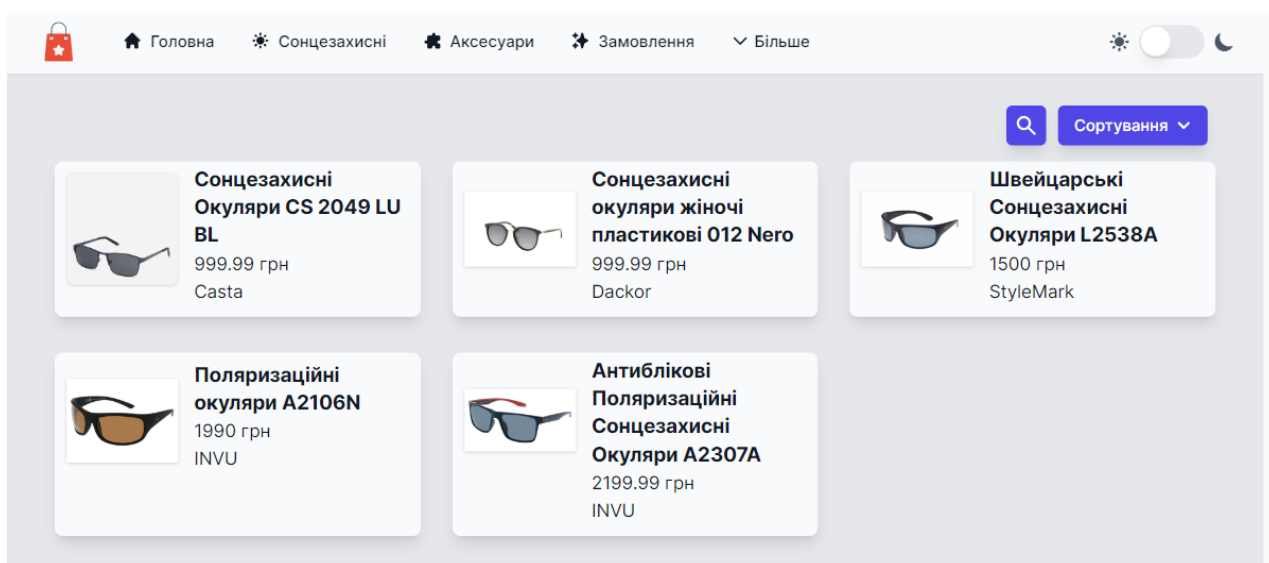


Рисунок 3.15 – Сторінка сонцезахисних окулярів відсортованих за ціною

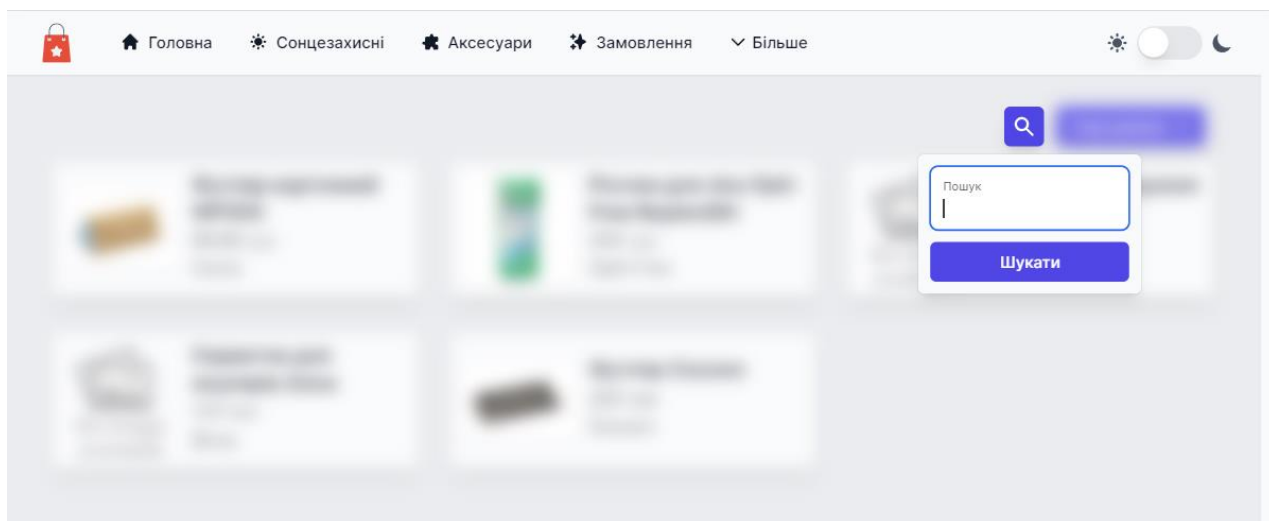


Рисунок 3.16 – Сторінка акcesуарів та лінз з відкритим пошуком

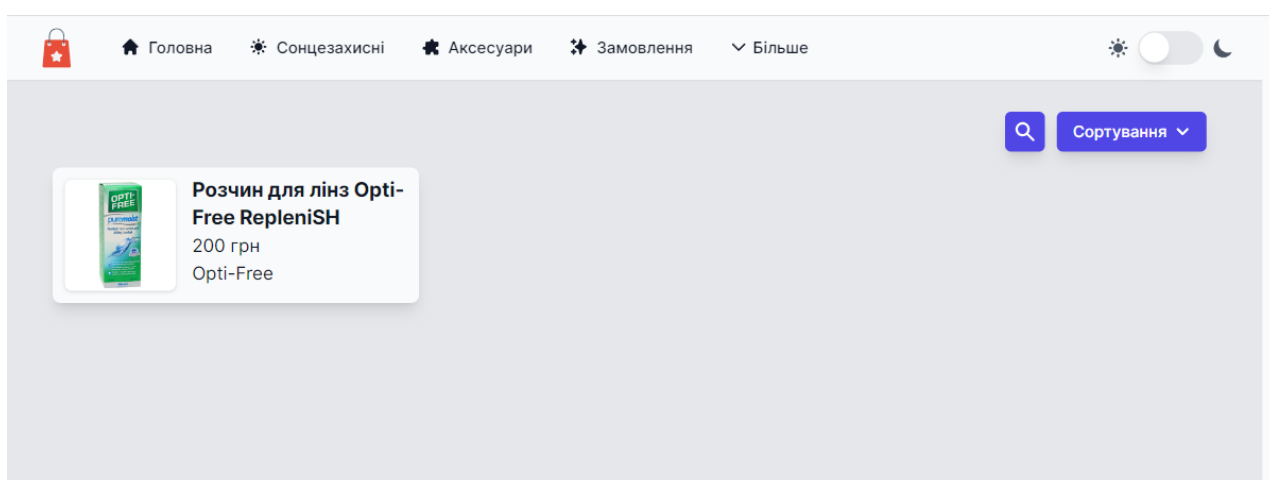


Рисунок 3.17 – Сторінка акcesуарів та лінз з пошуком "opti"

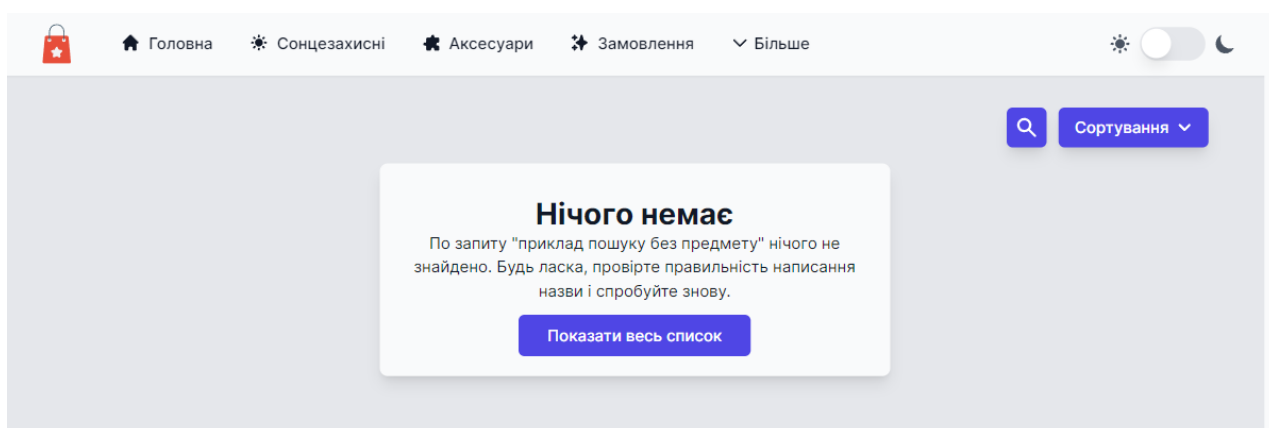


Рисунок 3.18 – Сторінка акcesуарів та лінз з вікном, яке інформує про відсутність предметів по запиту

Протестовано на працездатність пошук та сортування. Сортування працює правильно, та пошук дозволяє шукати будь-які збіги в назві та коректно виводить їх. Також виводить вікно, в якому вказується про відсутність товарів за запитом.

Також одним з важливих пунктів є наявність детальної інформації про товари, включно з фото та описами.

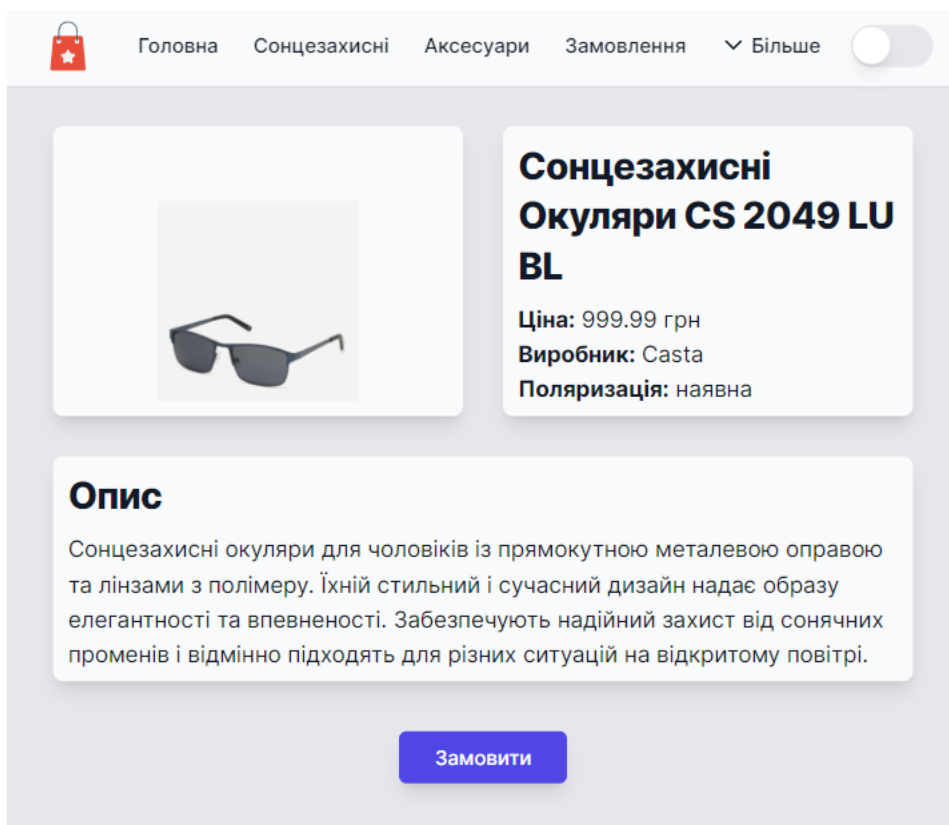


Рисунок 3.19 – Сторінка з сонцезахисними окулярами

Як видно з рисунку 3.19, інтрнет-магазин має детальну інформацію про товари, включно з фото та описом.

Важливою складовою, як було вияснено, являється наявність конструктору окулярів. Вона присутня в одному аналогу, через що він має перевагу для людей які хочуть замовити окуляри для свого рецепту, який їм виписав лікар. Тому протестуємо ще замовлення з рецептом (рис. 3.20).

Оформити замовлення

Вибіріть оправу та лінзу:

Wayfarer 1200 грн Ray-Ban	Frogskins 1400 грн Oakley	PO3019S 1800 грн Persol
---	---	---

Однофокальні 1200 грн Zeiss Німеччина	Мультифокальні 2500 грн Essilor Франція	Біф 1800 грн Hoya Япо
---	---	---

Вкажіть ваш рецепт:

Діоптрії (праве око)

Діоптрії (ліве око)

Астигматика (праве око)

Астигматика (ліве око)

DP (міжзінична відстань)

Додаткові параметри (опціонально)

Вкажіть куди доставити:

Адрес

☐ Швидка доставка

Додати

Головна

Рисунок 3.20 – Сторінка замовлення з конструктором

Можна побачити, що сторінка проста та має не навантажений інтерфейс. Спочатку користувачеві пропонується вибрати оправу та лінзи з доступних варіантів. Нижче знаходяться поля для введення рецепту, які включають діоптрії, астигматизм, міжзіничну відстань (DP) та додаткові параметри. Також є поле для вказання адреси доставки і вкінці сторінки розміщена кнопка для завершення замовлення. Після цього користувач отримає повідомлення про те, що замовлення успішно оформлене.

Отже, провівши тестування, можна сказати, що всі поставлені вимоги в розділі 1 було виконано.

3.6 Висновок до розділу 3

У цьому розділі було проведено обґрунтування вибору інструментів для технічної реалізації інтернет-магазину для окулярів, лінз та аксесуарів до них. Для фронтенду (клієнтської частини) було обрано JavaScript у поєднанні з фреймворком Vue.js та допоміжними бібліотеками, такими як Tailwind CSS, Vite, Axios та Pinia. Це рішення забезпечує створення зручного користувацького інтерфейсу, підтримку кросплатформеності, а також легкість розширення коду.

Для бекенду (серверної частини) було вибрано мову програмування PHP завдяки її простоті, інтеграції з веб-серверами та підтримці різних баз даних. Розроблено алгоритми та структуру серверної частини, що забезпечують безпечну реєстрацію та авторизацію користувачів, взаємодію з базою даних, обробку запитів для товарів і замовлень.

Крім того було вибрано середовище розробки, а саме Visual Studio Code, оскільки воно пропонує зручний інтерфейс, підтримку різних мов програмування, фреймворків, а також широкі можливості для налаштування.

Вкінці розділу було проведено тестування розробленого веб-додатку. Перевірено функціонал реєстрації та авторизації користувачів, адаптивність інтерфейсу на різних пристроях, роботу каталогу товарів з функціями пошуку та сортування, а також наявність детальної інформації про товари з фото та описами.

У результаті було створено повноцінний інтернет-магазин для продажу контактних лінз, окулярів та аксесуарів до них, який відповідає поставленим вимогам та забезпечує зручний та безпечний досвід для користувачів.

ВИСНОВКИ

Під час виконання бакалаврської дипломної роботи було проведено аналіз предметної області, вивчено особливості роботи офтальмологічної сфери та існуючі рішення інтернет-магазинів. На основі цього аналізу були визначені основні вимоги та функціональні можливості майбутньої системи.

Спроектовано структуру програмного модуля інтернет-магазину, що складається з окремих сторінок та компонентів. Також було обрано клієнт-серверну архітектуру та розроблено UML-діаграми варіантів використання для різних типів користувачів та алгоритми функціонування.

Для програмної реалізації системи було обрано сучасні технології. На клієнтській частині застосовано JavaScript з використанням фреймворку Vue.js, Tailwind CSS, Vite, Axios та Pinia. На серверній частині використано PHP для обробки запитів та взаємодією з базою даних SQLite.

Після завершення програмної реалізації було проведено тестування інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них на відповідність початковим вимогам та коректність роботи всіх функцій. Всі поставлені завдання та вимоги було виконано.

СПИСОК ДЖЕРЕЛ

1. World health organization. URL: <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment> (дата звернення: 19.02.2024).
2. AMERICAN OPTOMETRIC. URL: <https://www.aoa.org/healthy-eyes/eye-and-vision-conditions/computer-vision-syndrome> (дата звернення: 19.02.2024).
3. ПОЛОЖЕННЯ про систему офтальмологічної допомоги в Україні: Наказ Міністерства охорони здоров'я України від 14.05.2013 р. №372. URL: <https://zakon.rada.gov.ua/laws/show/z1051-13#Text> (дата звернення: 19.02.2024)
4. ЕКСІМЕР. URL: <https://eximer.ua/uk/shho-mi-likuiemo/korotkozorist/> (дата звернення: 19.02.2024).
5. Олег Олегович Щерба: Аналіз передумов розробки інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів для них. LIII Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2024). URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2024/paper/view/20796> (дата звернення: 22.05.2024).
6. Етапи створення інтернет-магазину. URL: <https://redchameleon.com.ua/ua/articles/etapy-stvorennja-internet-magazynu/> (дата звернення: 22.05.2024).
7. Люксоптика . URL: <https://luxoptica.ua/ua/> (дата звернення: 19.02.2024).
8. Оптика Люксор . URL: <https://opticaluxor.ua/> (дата звернення: 19.02.2024).
9. kupit-ochki.com.ua. URL: <https://kupit-ochki.com.ua/ua/> (дата звернення: 19.02.2024).
10. LINZA.ua. URL: <https://linza.com.ua/uk/> (дата звернення: 19.02.2024).
11. Рудий Т. В., Паранчук Я., С., Сенік В. В. Алгоритмізація та програмування. Частина 1. Структурне програмування : навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2023. 240 с.

12. Розуміння Клієнт-Серверної Архітектури на прикладах. URL: <https://dou.ua/forums/topic/44636/> (дата звернення: 11.05.2024).
13. Чому варто вивчати JavaScript? URL: <https://qagroup.com.ua/publications/why-learn-javascript/> (дата звернення: 12.05.2024).
14. Відкрита JS бібліотека React? URL: <https://web-developer.in.ua/assets/articles/react/react-js-library/react-js-library.html> (дата звернення: 20.04.2024).
15. Бібліотеки та фреймворки JavaScript: як правильно їх вибрати. URL: <https://aw.club/global/uk/blog/javascript-libraries-and-frameworks-how-to-choose> (дата звернення: 20.04.2024).
16. Вступ Vue.js. URL: <https://ua.vuejs.org/guide/introduction.html> (дата звернення: 20.04.2024).
17. Яку мову програмування можна назвати найпростішою? URL: <https://itproger.com/ua/news/kakoy-yazik-programmirovaniya-mozhno-nazvat-samim-prostim> (дата звернення: 20.04.2024).
18. Редагуйте код за допомогою веб-програми Visual Studio Code (підготовча версія). URL: <https://learn.microsoft.com/uk-ua/power-pages/configure/visual-studio-code-editor> (дата звернення: 20.04.2024).
19. WebStorm The JavaScript and TypeScript IDE. URL: <https://www.jetbrains.com/webstorm/> (дата звернення: 13.05.2024).
20. Sublime Text, ефективні шорткати та трюки. URL: <https://hackyourmom.com/kibervijna/sublime-text-efektyvni-shortkaty-ta-tryuky/> (дата звернення: 20.04.2024).

ДОДАТКИ

ДОДАТОК А

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка інтернет-магазину для продажу контактних лінз, окулярів та аксесуарів до них

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

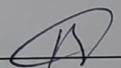
Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)

Показники звіту подібності Unichesk

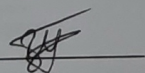
Оригінальність 95,03% Схожість 4,97%

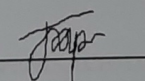
Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Озеранський В.С.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи  Щерба О.О.

Керівник роботи  Барабан С.В.

ДОДАТОК Б

ІНСТРУКЦІЯ КОРИСТВАЧА

При відкритті веб-сайту, всі користувачі потрапляють на головну сторінку (рис Б.1). На ній можна перейти до сонцезахисних окулярів, контактні лінзи та аксесуарів, окулярів під замовлення, авторизація (профіль якщо користувач авторизувався) та про роботу. Також для навігації верхній частині знаходиться меню, за допомогою якого можна зробити теж саме та змінити тему сайту на темну.

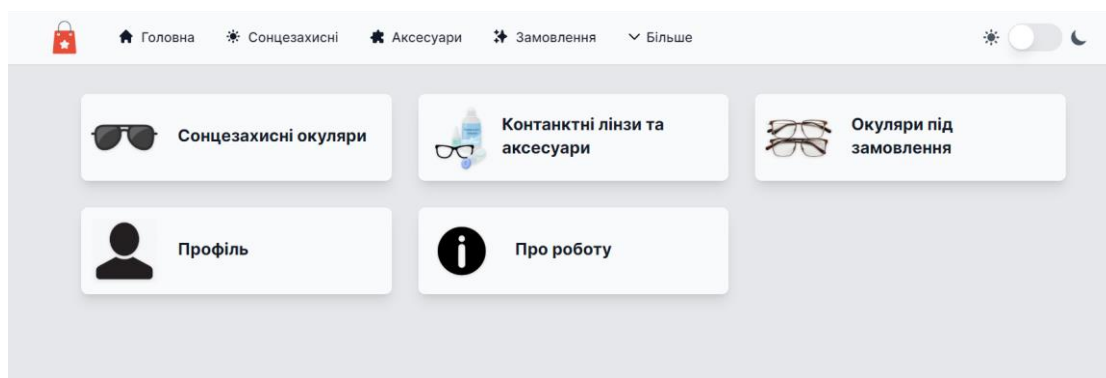


Рисунок Б.1 – Головна сторінка

Перейшовши до любого каталогу (рис. 2), ви побачите список доступних товарів з їх зображеннями, назвами та цінами.

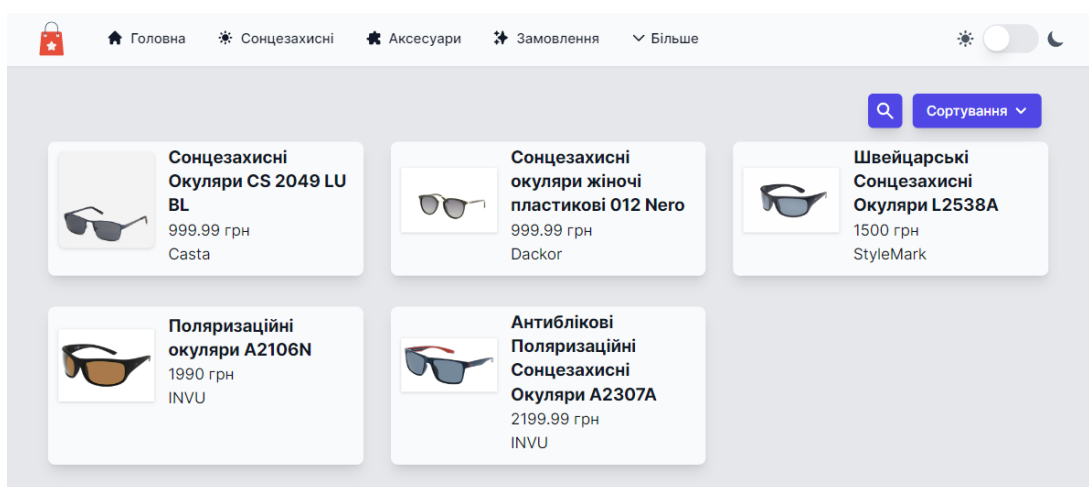


Рисунок Б.2 – Каталог з сонцезахисними окулярами

Натиснувши на сортування або на піктограму лупи користувач зможе відфільтрувати та знайти необхідний товар швидше. Після чього нажати на блок з товаром і перейти на сторінку товару (рис. Б.3).

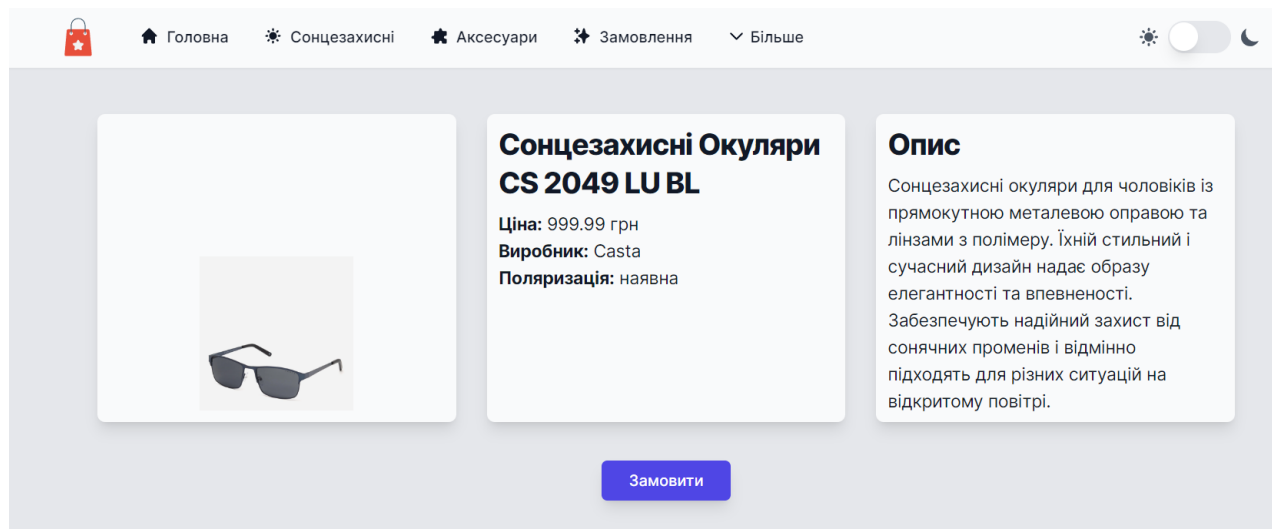


Рисунок Б.3 – Сторінка з сонцезахисними окулярами

Тут користувач зможе замовити окуляри, але якщо він не авторизований, він отримає повідомлення що для замовлення необхідно увійти, або створити профіль (рис. Б.4).

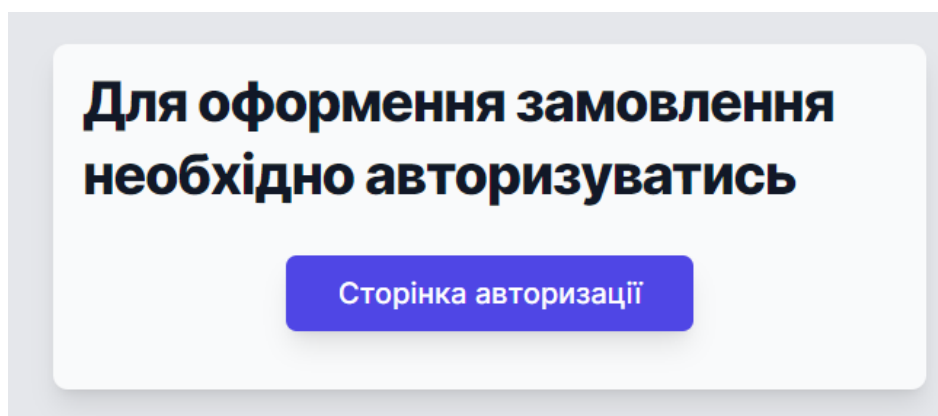
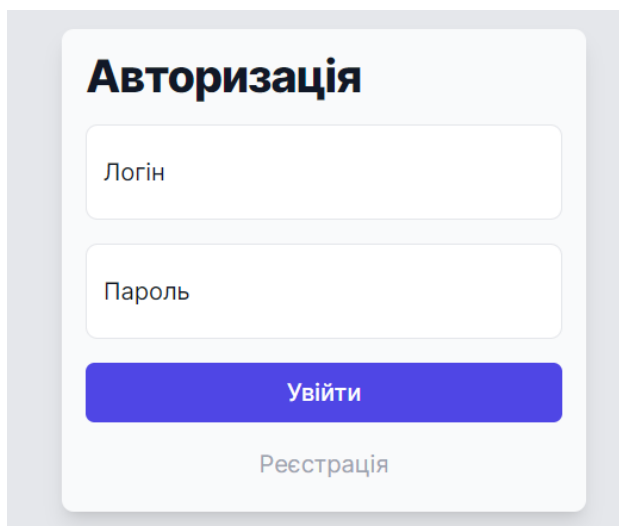


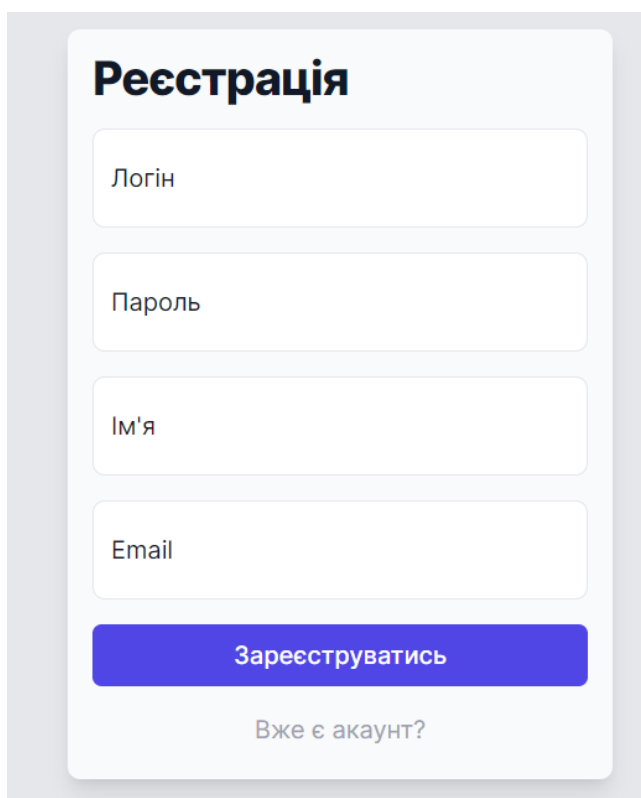
Рисунок Б.4 – Сторінка з сонцезахисними окулярами

Далі користувач попадає на сторінку авторизації (рис. Б.5), де може ввести логін і пароль, якщо в нього є профіль, або створити новий натиснувши реєстрація та попавши на сторінку реєстрації (рис. Б.6).



The image shows a login form titled "Авторизація" (Authorization). It contains two input fields: "Логін" (Login) and "Пароль" (Password). Below these fields is a blue button labeled "Увійти" (Login). At the bottom of the form, there is a link labeled "Реєстрація" (Registration).

Рисунок Б.5 – Сторінка авторизації



The image shows a registration form titled "Реєстрація" (Registration). It contains four input fields: "Логін" (Login), "Пароль" (Password), "Ім'я" (Name), and "Email". Below these fields is a blue button labeled "Зареєструватись" (Register). At the bottom of the form, there is a link labeled "Вже є акаунт?" (Already have an account?).

Рисунок Б.6 – Сторінка реєстрації

Після того як користувач увійшов, чи зареєструвався, він побаче сторінку свого профілю (рис. Б.7). На ній він зможе переглянути свої замовлення та вийти. Якщо користувач має роль редактор, або адмін він зможе з сторінки профіля добавляти товари та переглядати всі замовлення.

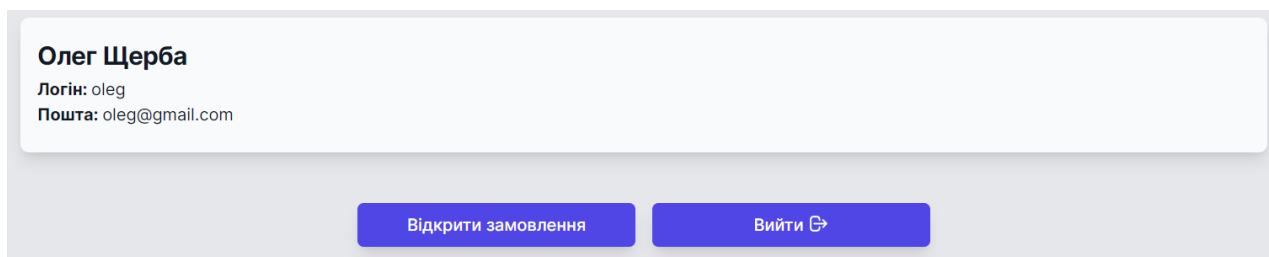


Рисунок Б.7 – Сторінка профілю користувача

Далі спробуємо замовити окуляри під замовлення, для цього перейдемо на сторінку замовлення з меню навігації зверху, або через головну. Вигляд сторінки конструктору замовлення зображено на рисунку Б.8.

Оформити замовлення

Вибірть оправу та лінзу:

Wayfarer 1200 грн Ray-Ban	Frogskins 1400 грн Oakley	PO3019S 1800 грн Persol
---	---	---

Однофокальні 1200 грн Zeiss Німеччина	Мультифокальні 2500 грн Essilor Франція	Біф 1800 грн Hoya Япо
---	---	---

Вкажіть ваш рецепт:

Діоптрії (праве око)

Діоптрії (ліве око)

Астегматика (праве око)

Астегматика (ліве око)

DP (міжнічна відстань)

Додаткові параметри (опціонально)

Вкажіть куди доставити:

Адрес

☐ Швидка доставка

Добавити

Головна

Рисунок Б.8 – Сторінка оформлення замовлення

Заповнивши необхідні данні в полях, користувач отримає повідомлення (рис. Б.9), що замовлення добавлено успішно.

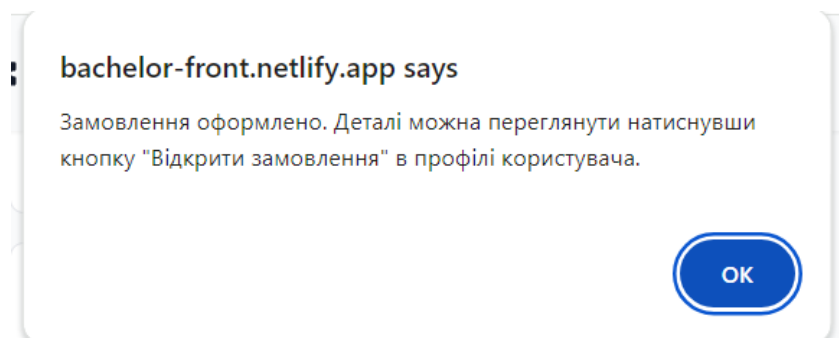


Рисунок Б.9 – Повідомлення про добавлене замовлення

Натиснувши кнопку "Відкрити замовлення" користувач зможе переглянути його замовлення (рис. Б.10).

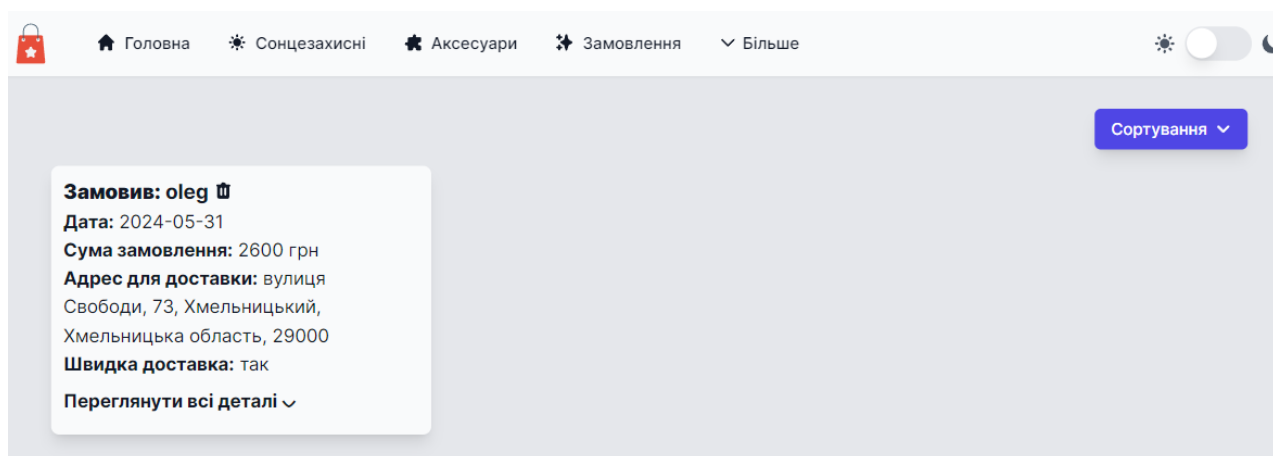


Рисунок Б.10 – Сторінка з замовленнями

Також є функціонал доступний лише редакторам, або адмінам. Для нього в них на сторінці профілю користувача доступні додаткові кнопки (рис. Б.11). А саме: Редагувати користувачів, Додати сонцезахисні, Додати аксесуар, Додати оправу, Додати лінзу, Редагувати записи.

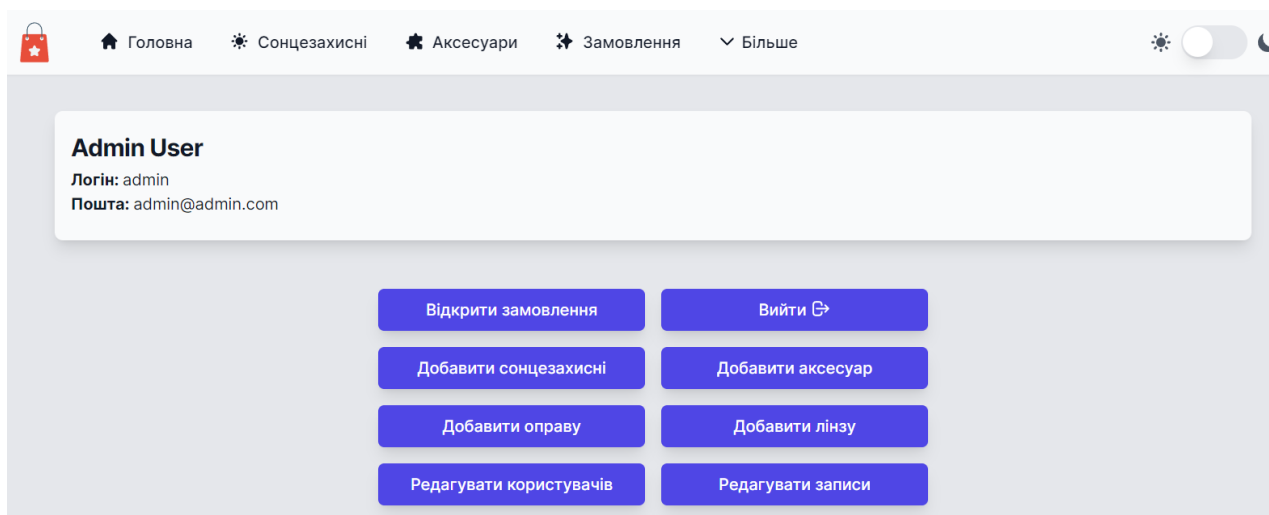


Рисунок Б.11 – Сторінка профілю користувача з роллю адмін

При натисканні кнопки "Додати сонцезахисні" з'явиться форма для додавання інформації про окуляри (рис. Б.12). Ця форма містить 4 текстових поля, чекбокс та можливість завантажити фото. Після заповнення форми, веб-додаток ретельно перевірить введені дані на коректність, включаючи текстові поля, формат зображення (підтримуються jpg, jpeg, png) та його розмір.

Рисунок Б.12 – Форма для добавляння сонцезахисних окулярів

Після успішного додавання нового запису з'явиться повідомлення, яке проінформує про успішне виконання операції (рис. Б.13).

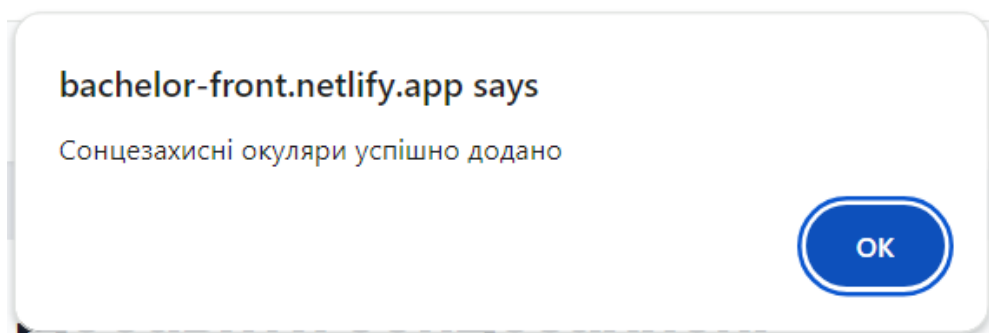


Рисунок Б.13 – Форма для добавляння сонцезисних окулярів

Аналогічно й з іншими товарами. Форма для їх додавання має схожу структуру, але з деякими нюансами, залежно від специфіки кожного продукту.

ДОДАТОК В

Лістинг програми

main.php:

```
<?php
define('DB_PATH', realpath(dirname(__FILE__) . '/../database/index.db'));
// Підключення до БД
$db = new SQLite3(DB_PATH);

// Заголовки
header('Content-Type: application/json');
header('Access-Control-Allow-Origin: *');
header('Access-Control-Allow-Methods: GET,POST');
header('Access-Control-Allow-Headers: Content-Type, Authorization, X-
Requested-With');

// Основні ролі
$roles = array(
    'user' => 0,
    'editor' => 1,
    'admin' => 2
);

function getDB() {
    global $db;
    return $db;
}

function getUserRole($token) {
```

```

$db = getDB();
$stmt = $db->prepare('SELECT RoleID FROM Users WHERE Token = :token');
$stmt->bindValue(':token', $token, SQLITE3_TEXT);
$result = $stmt->execute();
$row = $result->fetchArray(SQLITE3_ASSOC);

if ($row) {
    return $row['RoleID'];
}

return false;
}

function getUserID($token) {
    $db = getDB();
    $stmt = $db->prepare('SELECT UserID FROM Users WHERE Token = :token');
    $stmt->bindValue(':token', $token, SQLITE3_TEXT);
    $result = $stmt->execute();
    $row = $result->fetchArray(SQLITE3_ASSOC);

    if ($row) {
        return $row['UserID'];
    }

    return false;
}

```

init.php:

```

<?php
$dbPath = 'database/index.db';

if (file_exists($dbPath)) {
    echo "Firstly, delete the current database for reinitialization";
    exit;
    //unlink($dbPath);
}

// CLEAR IMAGES
$files = glob("database/images" . "/*.png");
foreach ($files as $file) {
    if (is_file($file)) {
        unlink($file);
    }
}

$db = new SQLite3($dbPath);

// CREATE TABLES
$sql = file_get_contents('../sql-raw/create.sql');
$db->exec($sql);

// ADD ADMIN USER
$db->exec("
INSERT INTO Users (LastLogin, Token, RoleID, Username, Password, Email,
FullName)
VALUES
('2024-05-07 12:00:00', '".bin2hex(random_bytes(32))."', 2, 'admin',

```

```
".password_hash('admin123', PASSWORD_DEFAULT).'", 'admin@admin.com',  
'Admin User'),  
    ('2024-05-07 12:00:00', '".bin2hex(random_bytes(32))."', 1, 'mod',  
".password_hash('mod123', PASSWORD_DEFAULT).'", 'mod@admin.com', 'Mod  
User');  
");
```

```
// ADD EXAMPLE SUNGLASSES
```

```
$sql = file_get_contents('../sql-raw/sunglasses.sql');  
$db->exec($sql);
```

```
// ADD EXAMPLE ACCESSORIES
```

```
$sql = file_get_contents('../sql-raw/accessories.sql');  
$db->exec($sql);
```

```
// ADD EXAMPLE FRAMES
```

```
$sql = file_get_contents('../sql-raw/frames.sql');  
$db->exec($sql);
```

```
// ADD EXAMPLE LENSES
```

```
$sql = file_get_contents('../sql-raw/lenses.sql');  
$db->exec($sql);
```

```
// DONE
```

```
echo 'Database initialization was successful';
```

main.js:

```
import { createApp } from 'vue'
import axios from "axios";
import '@/styles/tailwind.css'
import '@/styles/custom.css'
import App from '@/App.vue'
import { routes } from '@/routes.js'
import { createRouter, createWebHistory } from 'vue-router'
import { createPinia } from 'pinia'
```

```
const app = createApp(App)
```

```
// налаштування axios
```

```
axios.defaults.baseURL = "https://harhive.pp.ua/t/src"
```

```
axios.defaults.timeout = 9999;
```

```
const router = createRouter({
  history: createWebHistory(),
  routes,
  scrollBehavior(to, from, savedPosition) {
    if (savedPosition) {
      return savedPosition
    } else {
      return { top: 0 }
    }
  },
})
```

```
router.beforeEach((to, from, next) => {
  next();
```

```

document.body.style.overflowY = to.meta.showScroll ? 'scroll' : 'visible';
document.title = to.meta.title+" - магазин";
if(navigator.userAgent.toLowerCase().indexOf('prerender') !== -1) {
  document.querySelector('meta[property="og:title"]').setAttribute("content",
to.meta.title+" - магазин");

document.querySelector('meta[property="og:description"]').setAttribute("content",
to.meta.description);
  document.querySelector('meta[name="description"]').setAttribute("content",
to.meta.description);
  document.querySelector('meta[property="og:url"]').setAttribute("content",
window.location);
  to.meta.image &&
document.querySelector('meta[property="og:image"]').setAttribute("content",
window.location.protocol+'//'+window.location.hostname+to.meta.image);
}
})

```

```

app.use(createPinia())
app.use(router)
app.mount('#app')

```

app.vue:

```

<script setup>
import NavbarLoader from '@components/NavbarLoader.vue'
//import Error from '@components/Error.vue'

```

```
import { defineAsyncComponent } from 'vue'
```

```
const Navbar = defineAsyncComponent({
  loader: () => import('@components/Navbar.vue'),
  loadingComponent: NavbarLoader,
  delay: 0,
  //errorComponent: Error,
})
</script>
```

```
<template>
  <div class="bg-gray-200 dark:bg-gray-800 transition-colors min-h-screen">
    <header>
      <Navbar/>
    </header>
    <main class="mx-auto max-w-7xl px-4 sm:px-6 lg:px-8 pt-18 sm:pt-20 lg:pt-24">
      <router-view v-slot="{ Component }">
        <keep-alive :max="5" exclude="notAlive">
          <component :is="Component" />
        </keep-alive>
      </router-view>
    </main>
  </div>
</template>
```

ДОДАТОК Г

ГРАФІЧНА ЧАСТИНА

**"РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ ДЛЯ ПРОДАЖУ КОНТАКТНИХ
ЛІНЗ, ОКУЛЯРІВ ТА АКСЕСУАРІВ ДО НИХ"**

Виконав: студент 4 курсу, групи ЗКН-206
спеціальності 122 Комп'ютерні науки
(шифр і назва напрямку підготовки, спеціальності)
 Щерба О. О.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН
 Барабан С. В.
(прізвище та ініціали)
« 07 » 06 2024 р.

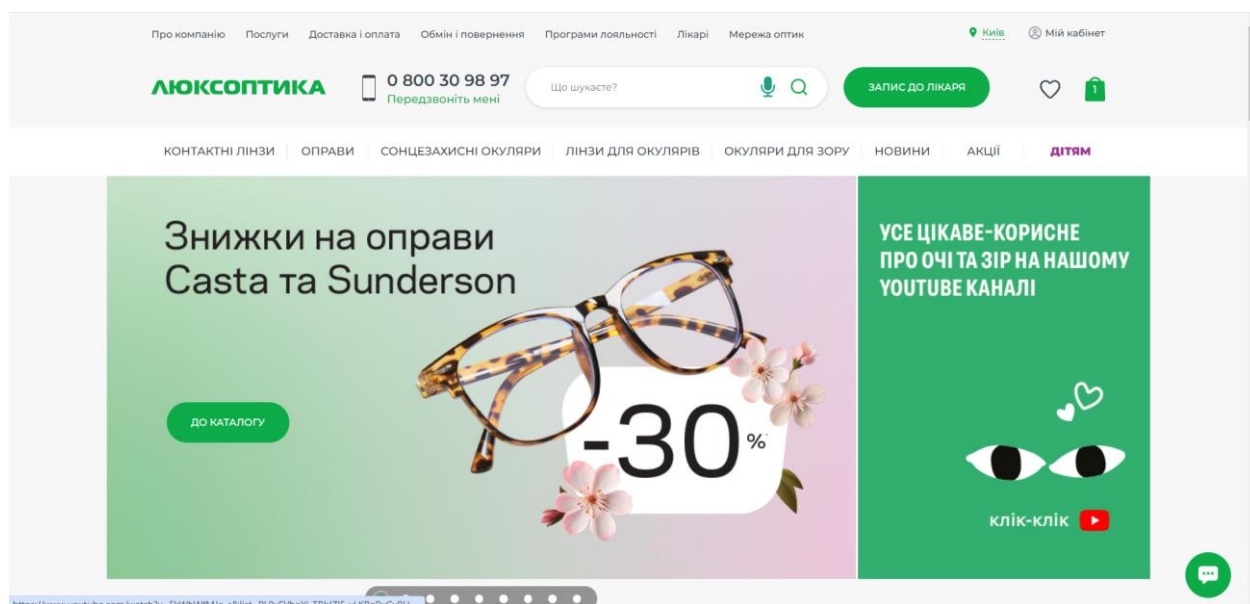


Рисунок Г.1 – Головна сторінка інтернет-магазину Люксоптика

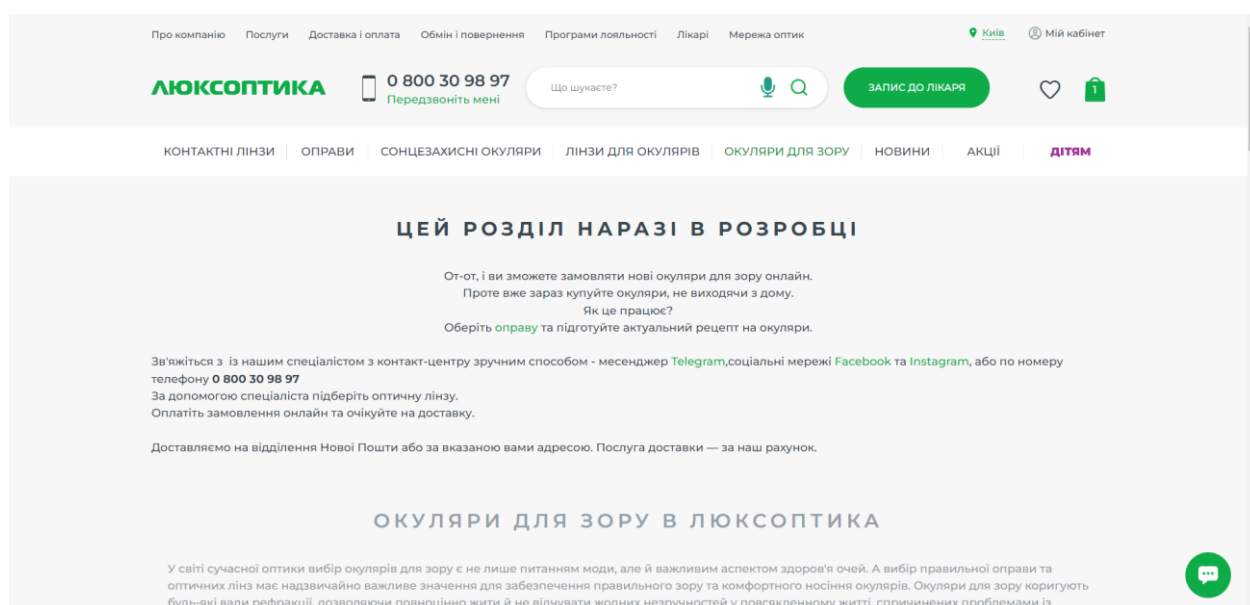


Рисунок Г.2 – Сторінка окулярів для зору в інтернет-магазині Люксоптика

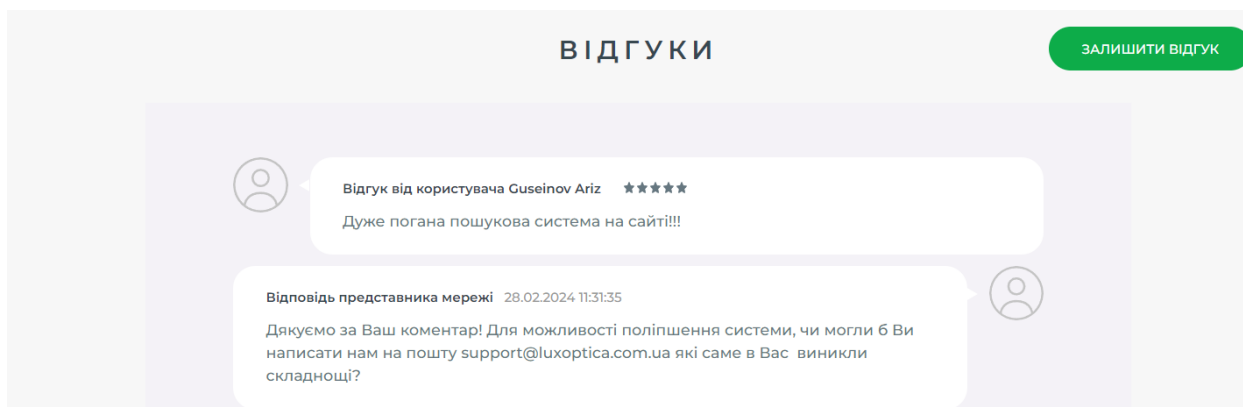


Рисунок Г.3 – Сторінка з відгуками в інтернет-магазині Люксоптика

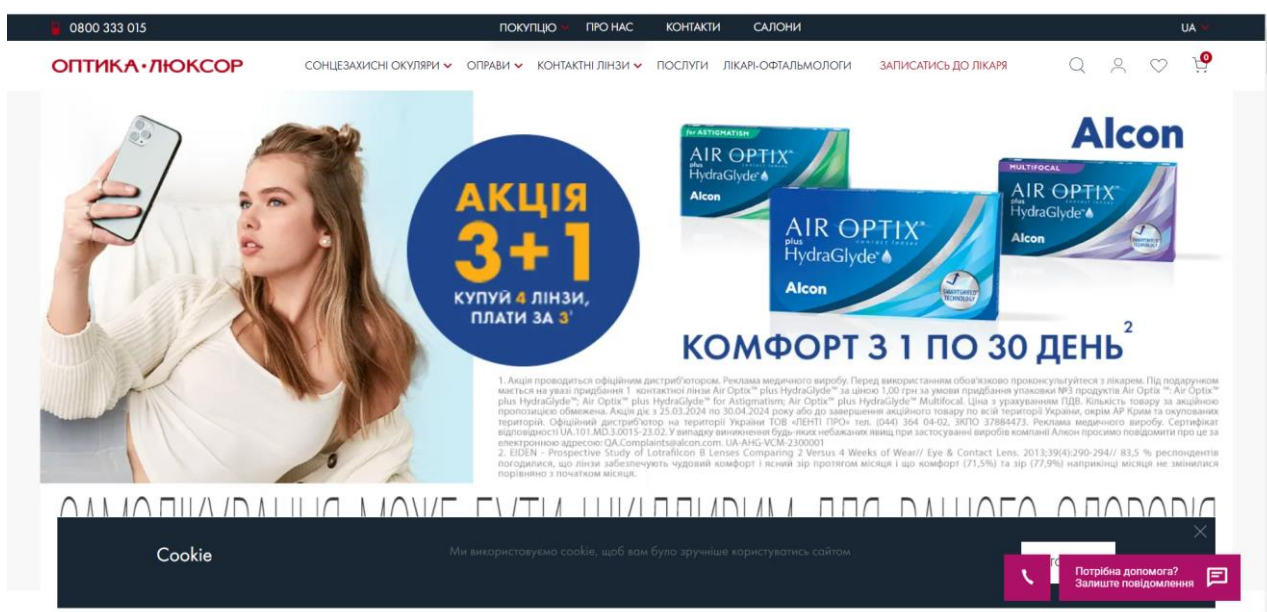


Рисунок Г.4 – Головна сторінка інтернет-магазину Оптика Люксор

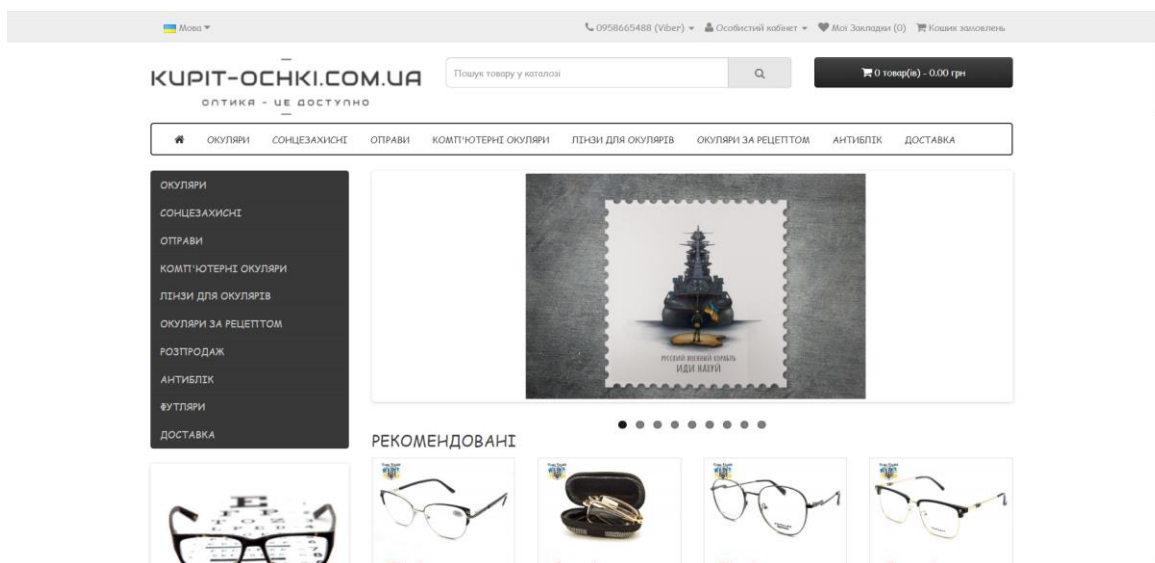


Рисунок Г.5 – Головна сторінка інтернет-магазину kupit-ochki.com.ua

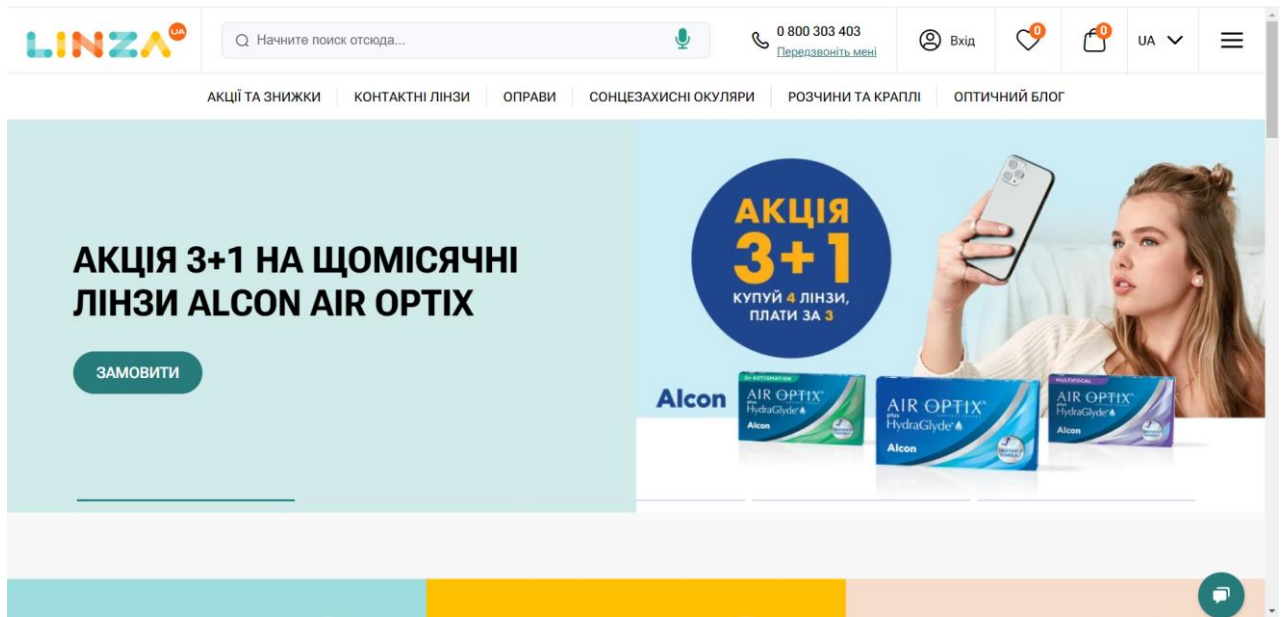


Рисунок 1.6 – Головна сторінка інтернет-магазину LINZA.ua



Рисунок Г.7 – Загальна структурна схема функціонування системи



Рисунок Г.8 – Архітектура клієнт-серверної взаємодії

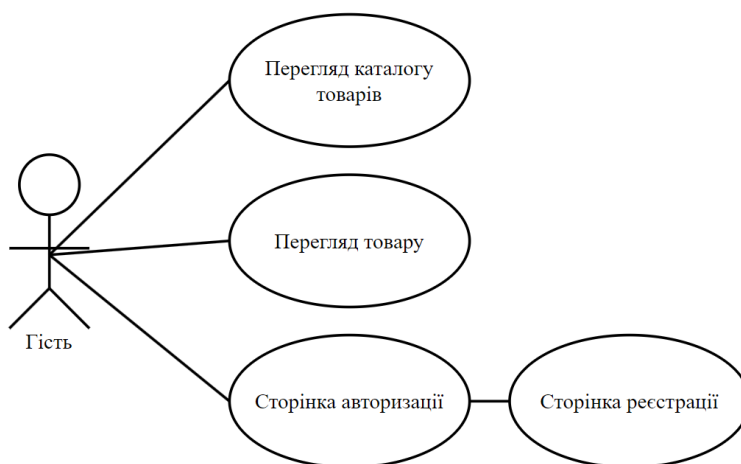


Рисунок Г.9 – Діаграма варіантів використання для класу «Гість»

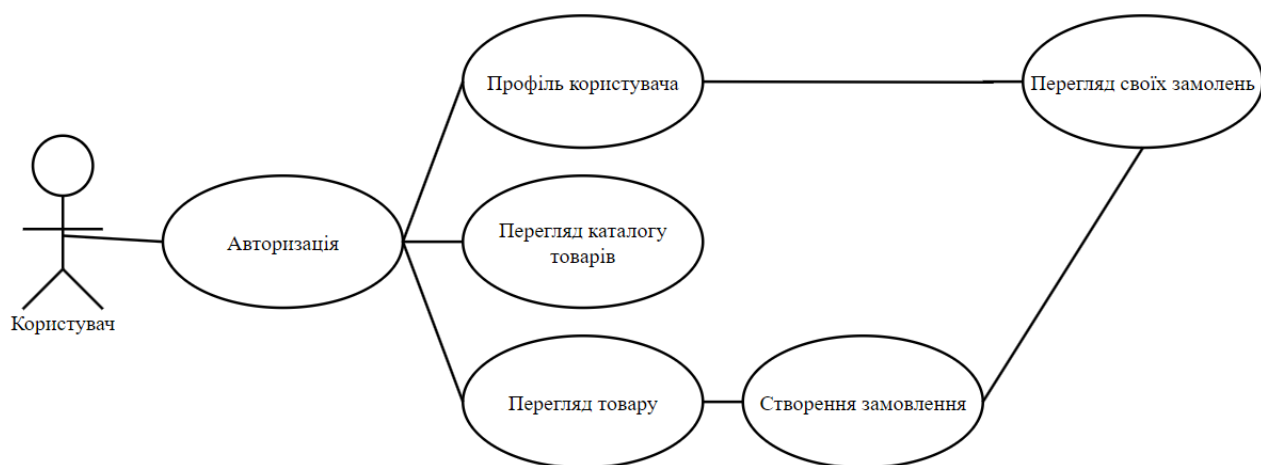


Рисунок Г.10 – Діаграма варіантів використання для класу «Користувач»

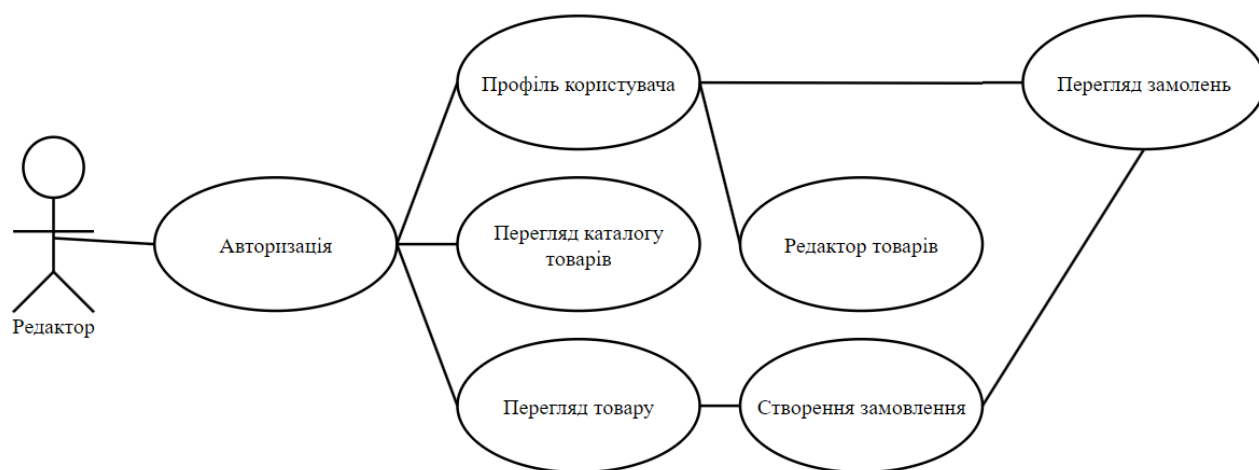


Рисунок Г.11 – Діаграма варіантів використання для класу «Редактор»

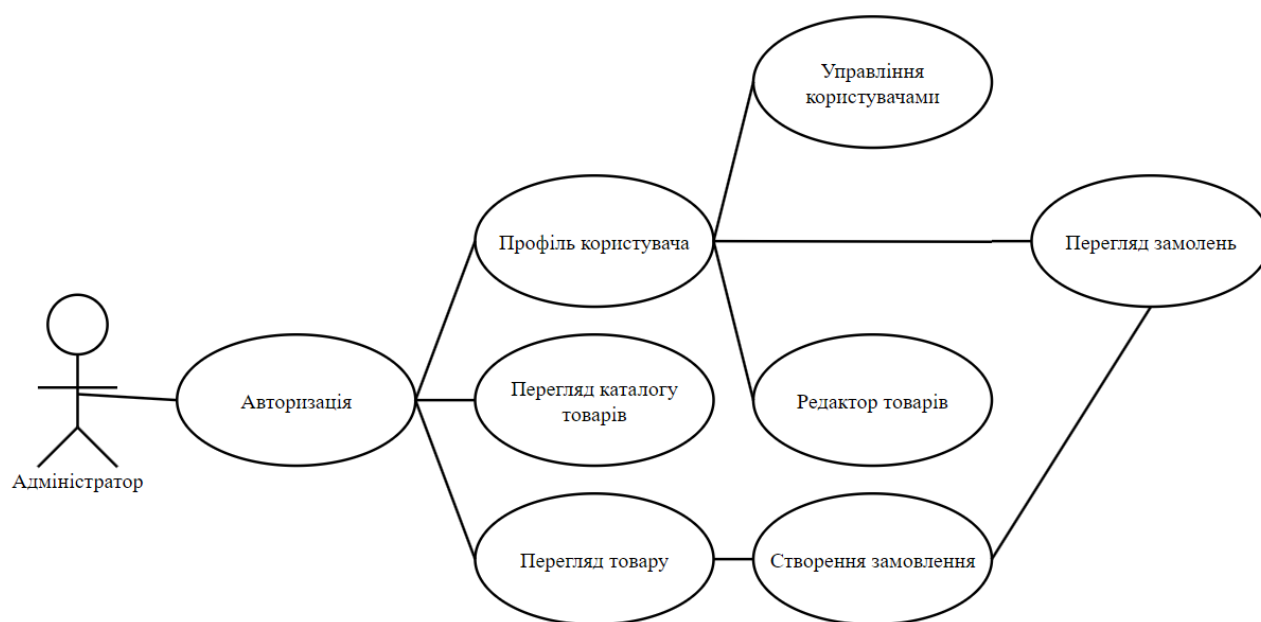


Рисунок Г.12 – Діаграма варіантів використання для класу «Адміністратор»

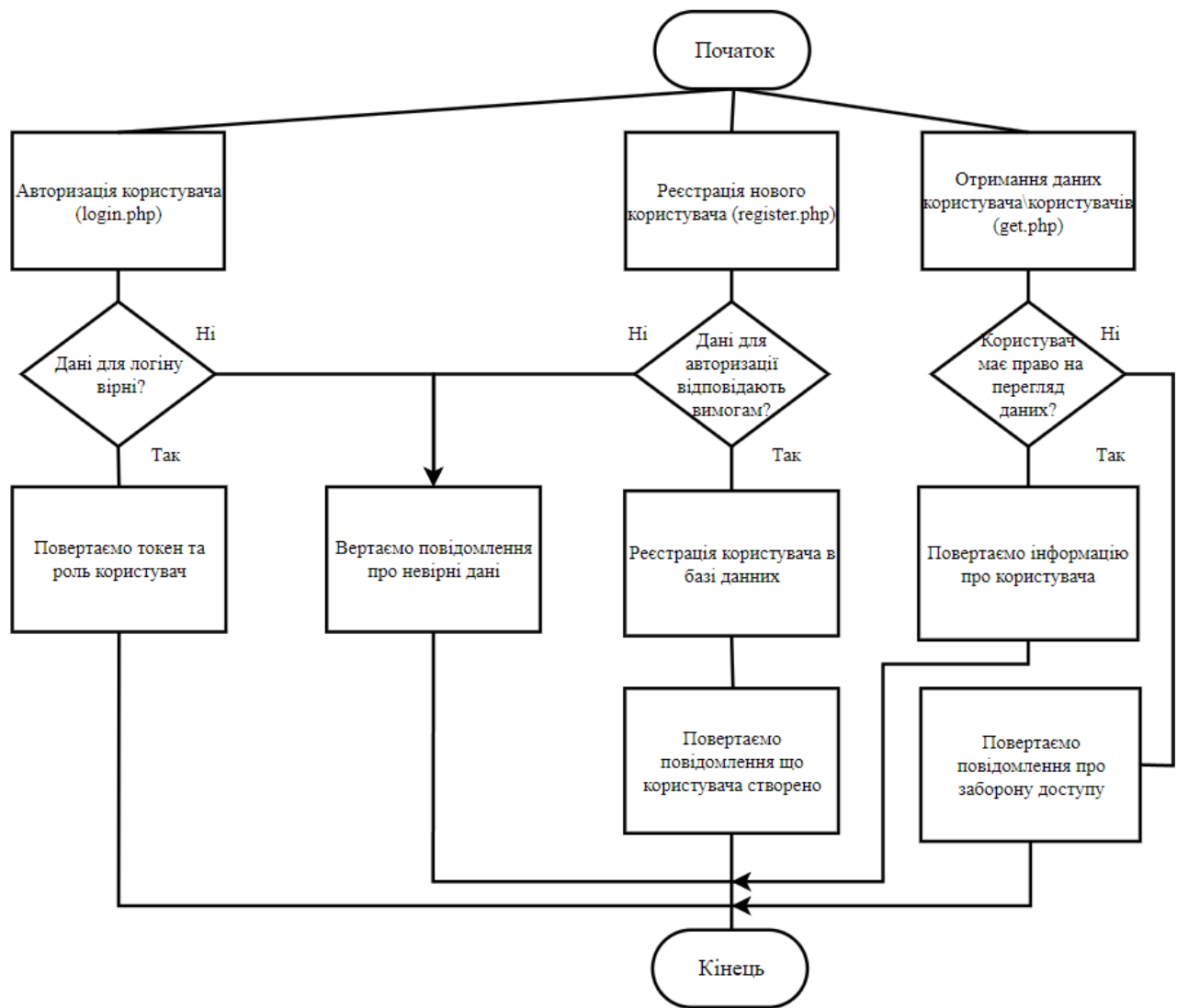


Рисунок Г.13 – Алгоритм авторизації користувача

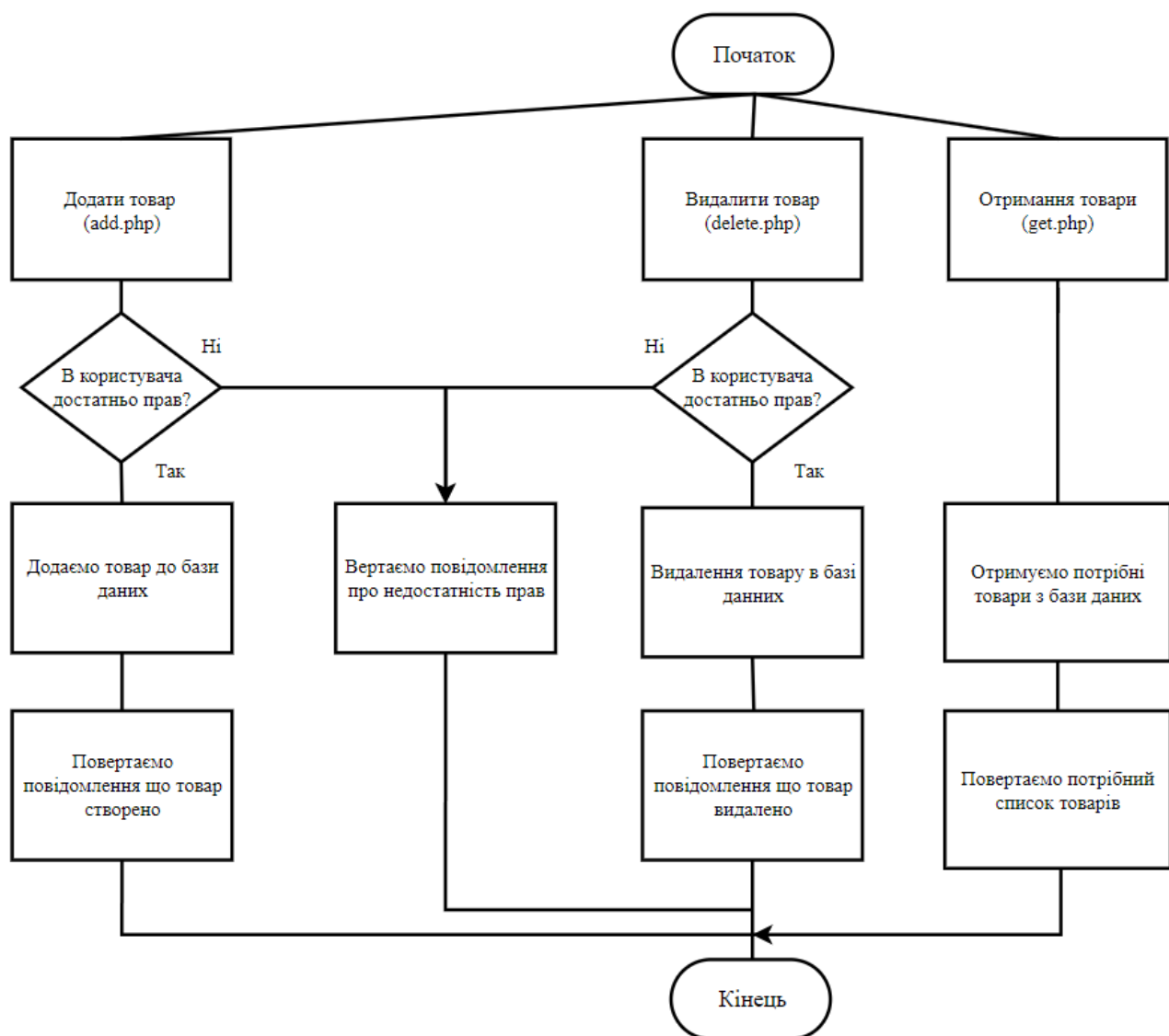


Рисунок Г.14 – Алгоритм роботи товарів

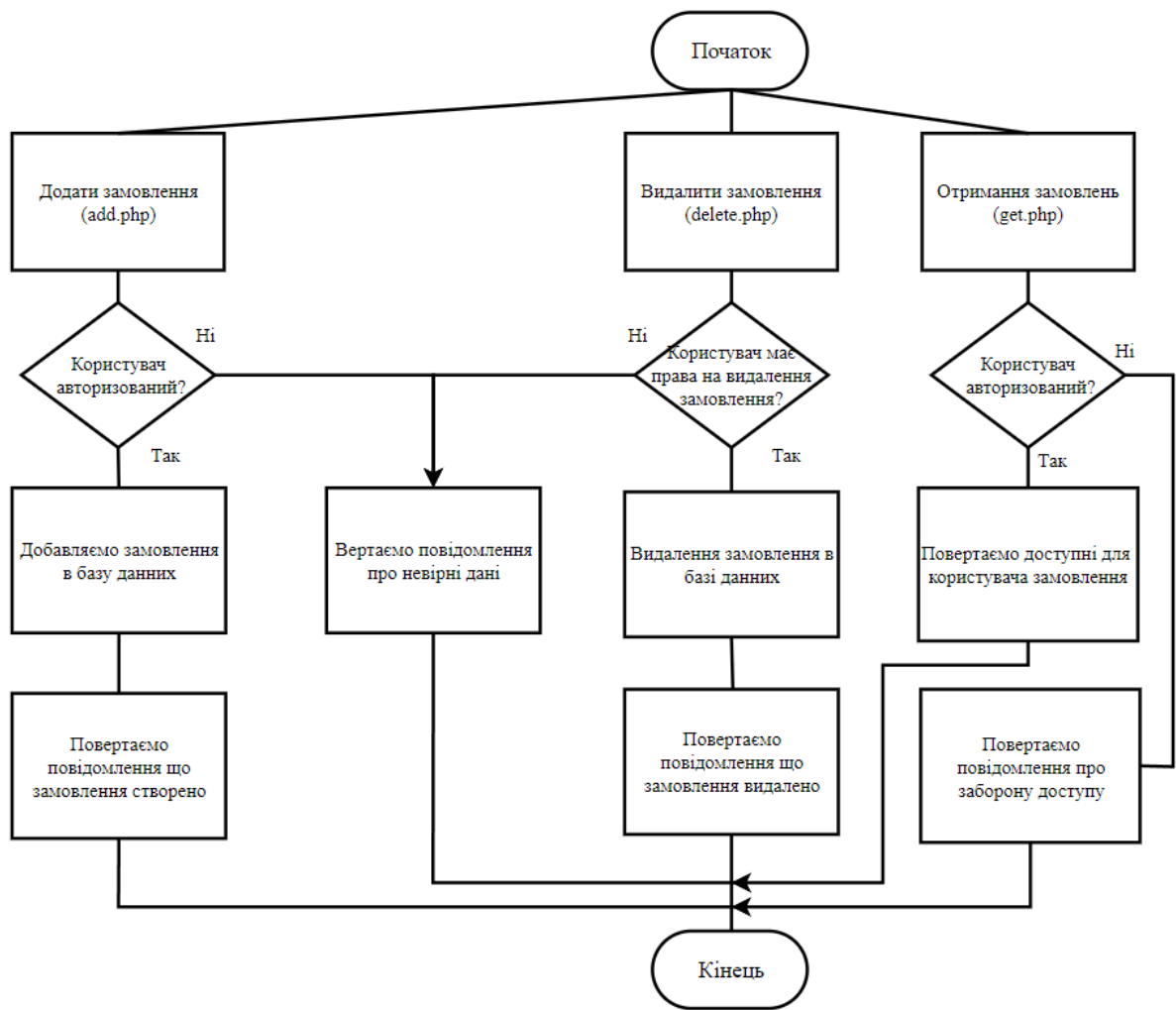


Рисунок Г.15 – Алгоритм роботи замовлень

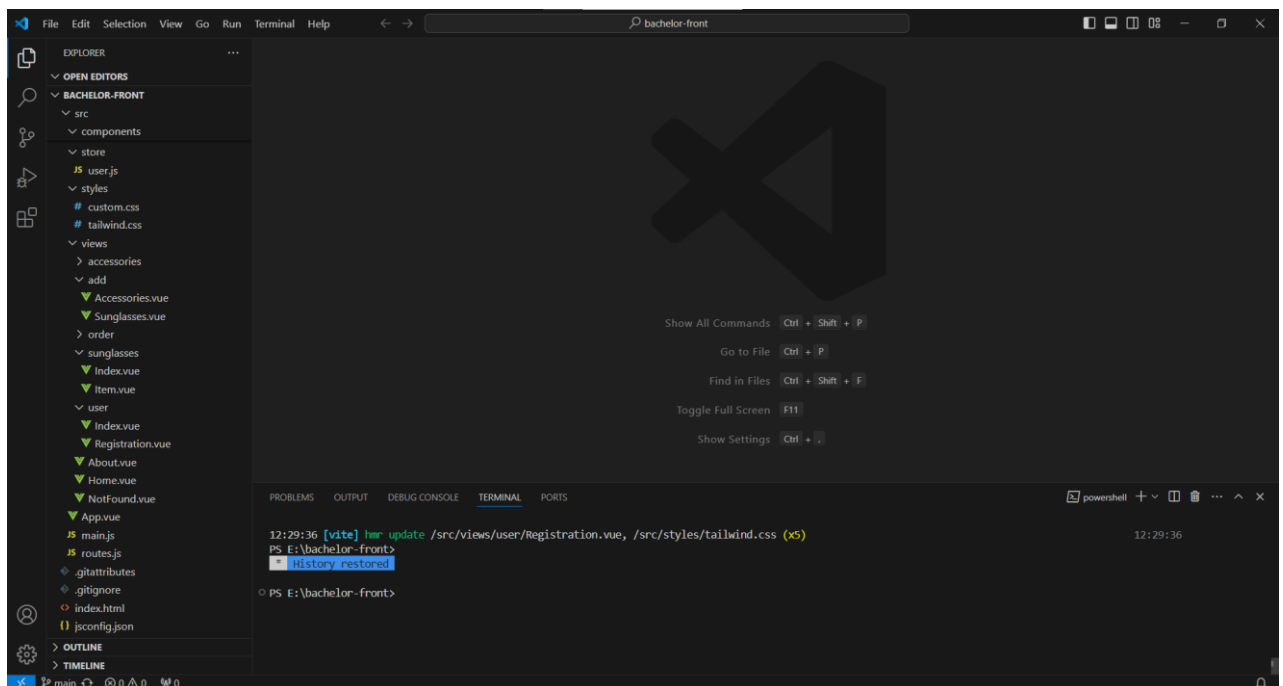


Рисунок Г.16 – Вигляд Visual Studio Code

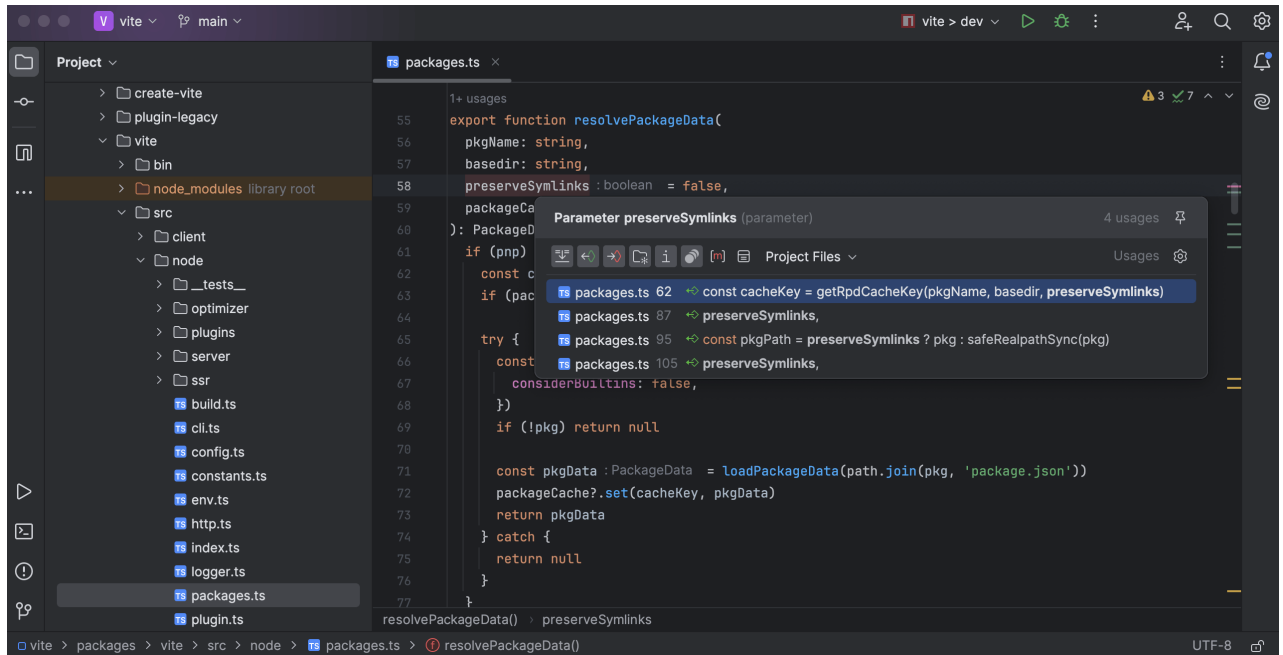


Рисунок Г.17 – Вигляд WebStorm

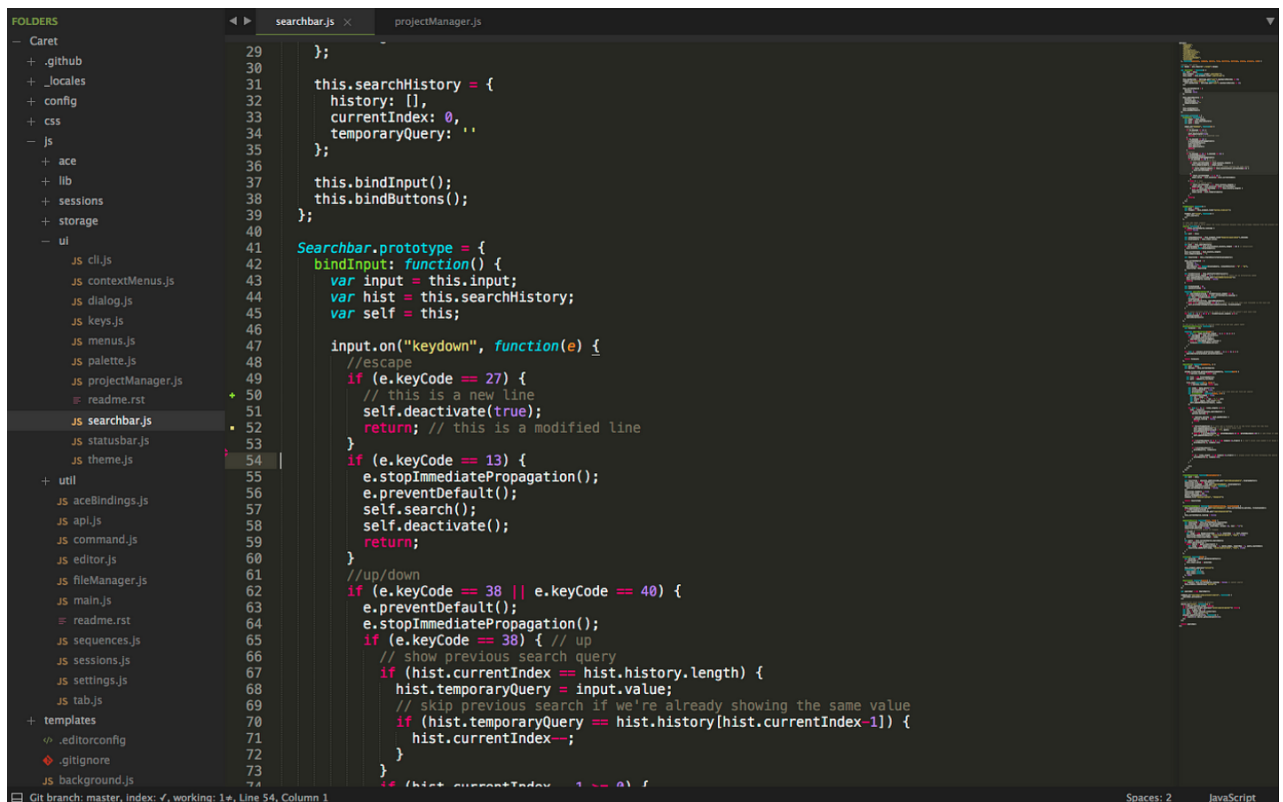


Рисунок Г.18 – Вигляд Sublime Text

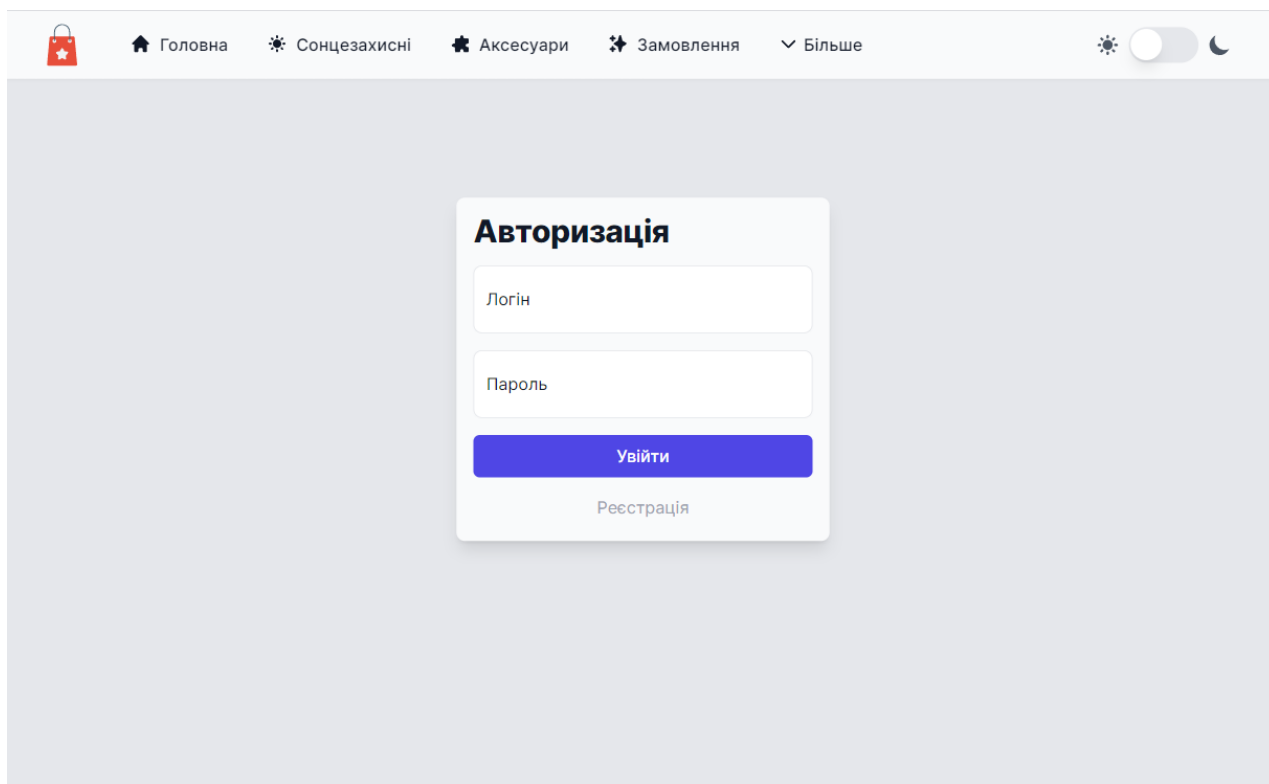


Рисунок Г.19 – Сторінка авторизації

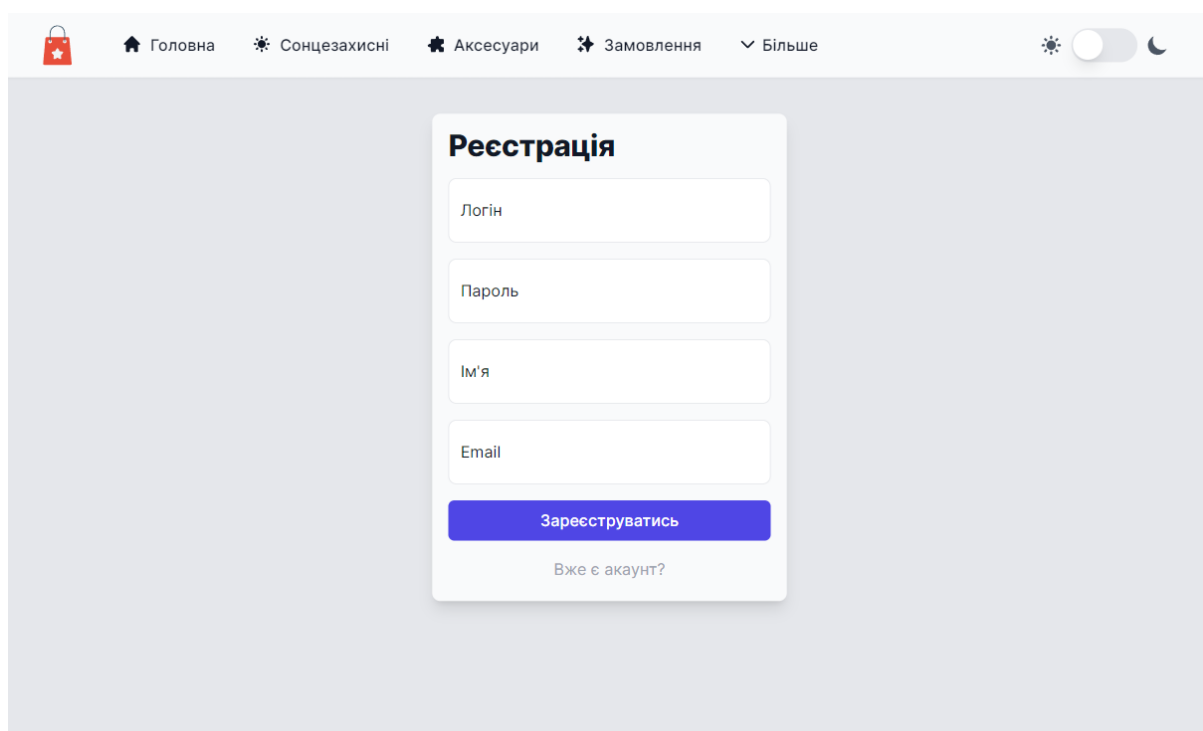
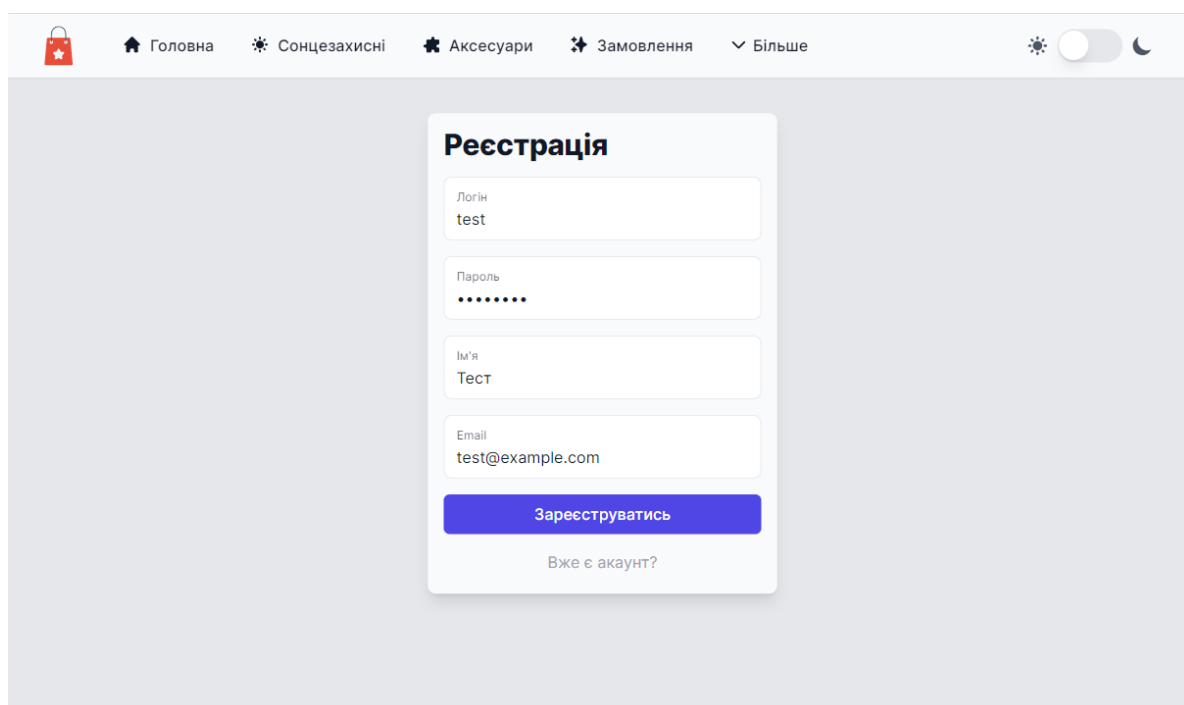


Рисунок Г.20 – Сторінка реєстрації



The image shows a web application's registration page. At the top is a navigation bar with a shopping bag icon, a home icon, and menu items: 'Головна', 'Сонцезахисні', 'Акcesуари', 'Замовлення', and 'Більше'. There is also a dark mode toggle. The main content area features a white registration card titled 'Реєстрація'. The card contains four input fields: 'Логін' (test), 'Пароль' (masked with dots), 'Ім'я' (Тест), and 'Email' (test@example.com). Below the fields is a blue 'Зареєструватись' button and a link 'Вже є акаунт?'.

Головна Сонцезахисні Акcesуари Замовлення Більше

Реєстрація

Логін
test

Пароль
.....

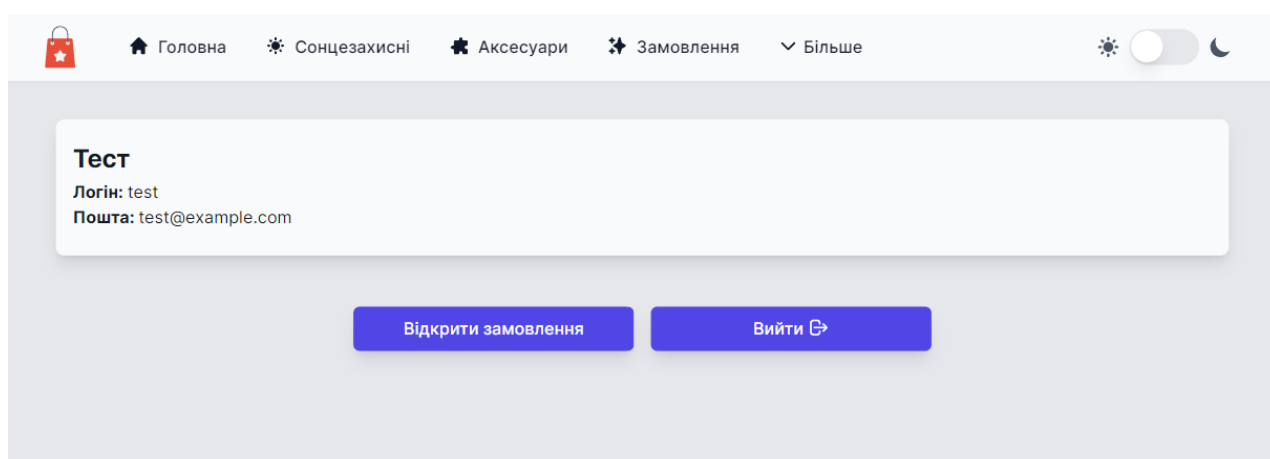
Ім'я
Тест

Email
test@example.com

Зареєструватись

[Вже є акаунт?](#)

Рисунок Г.21 – Сторінка реєстрації з заповненими даними



The image shows the user profile page after successful login. The navigation bar is identical to the previous screen. The main content area displays a white profile card for a user named 'Тест'. The card shows 'Логін: test' and 'Пошта: test@example.com'. Below the card are two blue buttons: 'Відкрити замовлення' and 'Вийти' with an external link icon.

Головна Сонцезахисні Акcesуари Замовлення Більше

Тест

Логін: test
Пошта: test@example.com

Відкрити замовлення Вийти ➔

Рисунок Г.22 – Сторінка профілю після авторизації

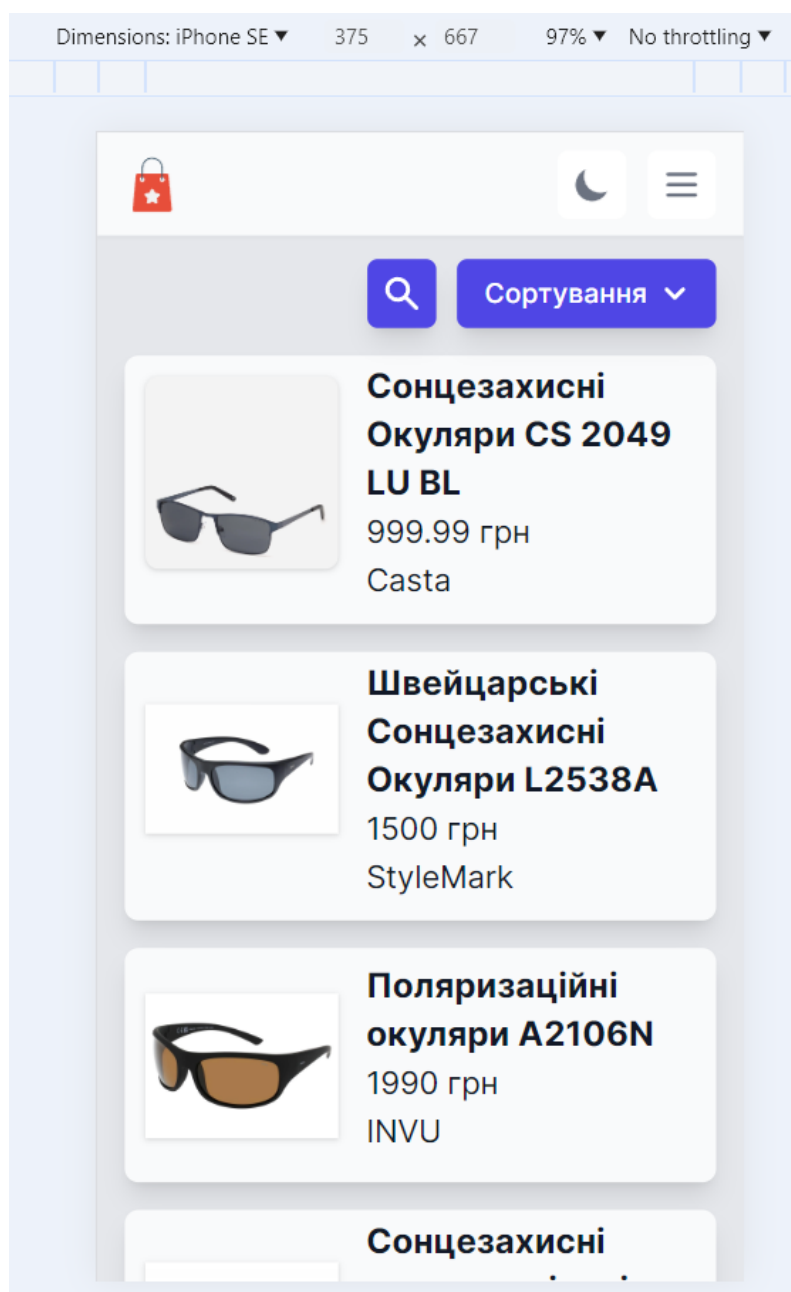


Рисунок Г.23 – Сторінка з сонцезахисними окулярами на телефоні

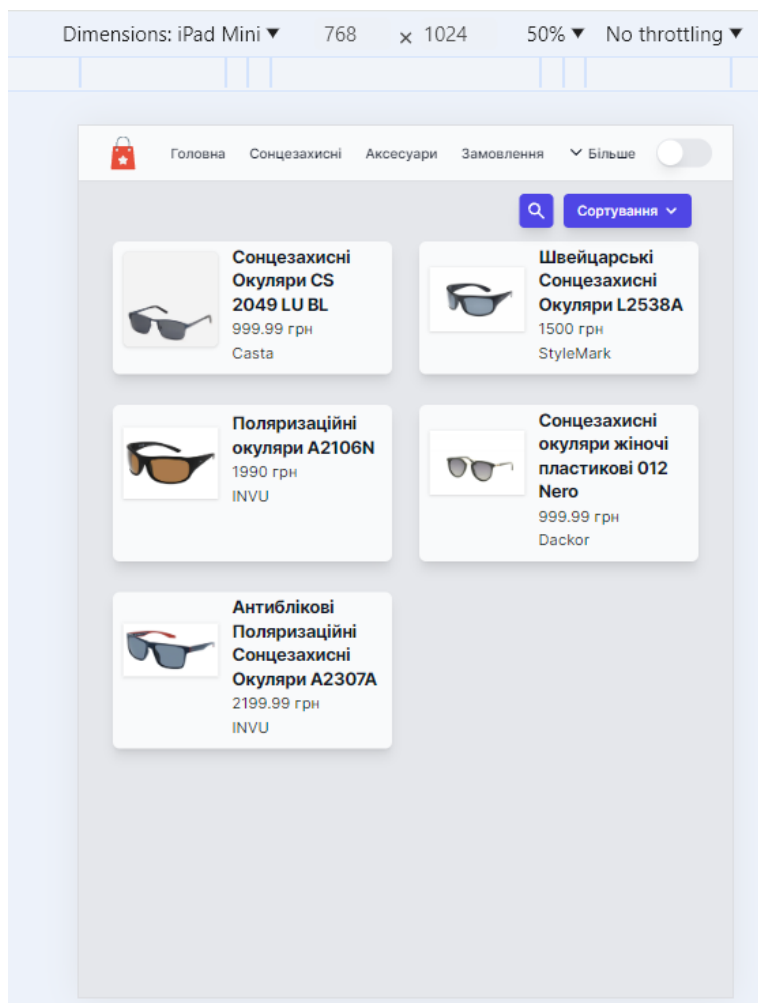


Рисунок Г.24 – Сторінка з сонцезахисними окулярами на планшетах

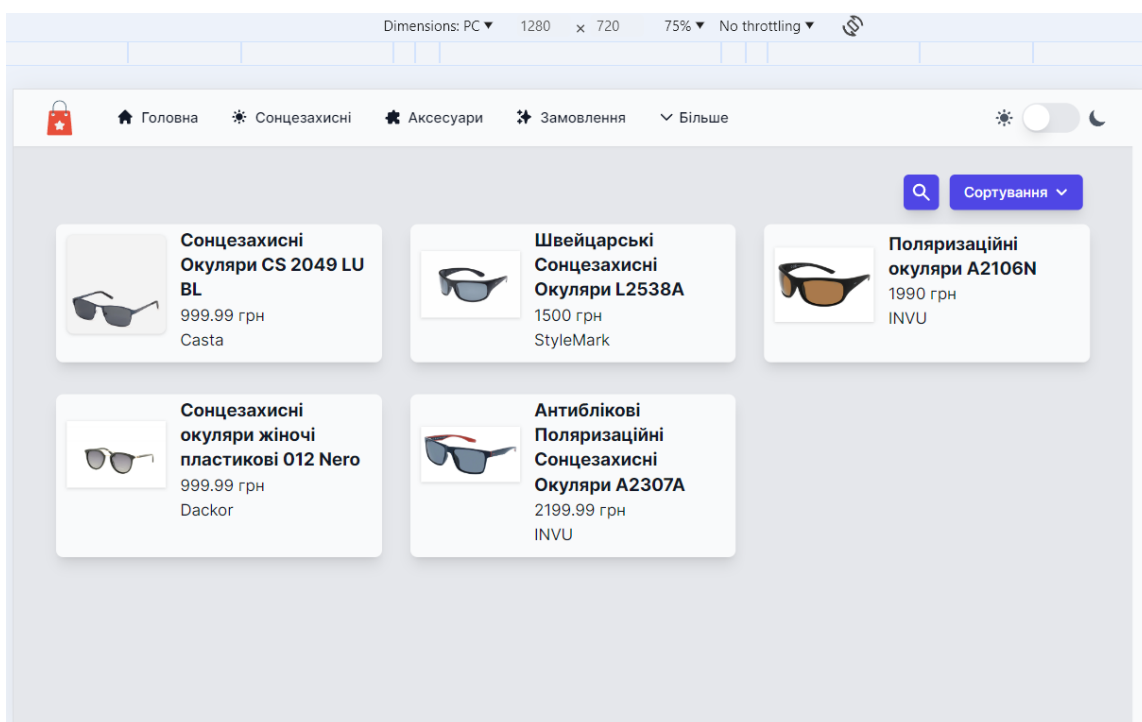


Рисунок Г.25 – Сторінка з сонцезахисними окулярами на ПК

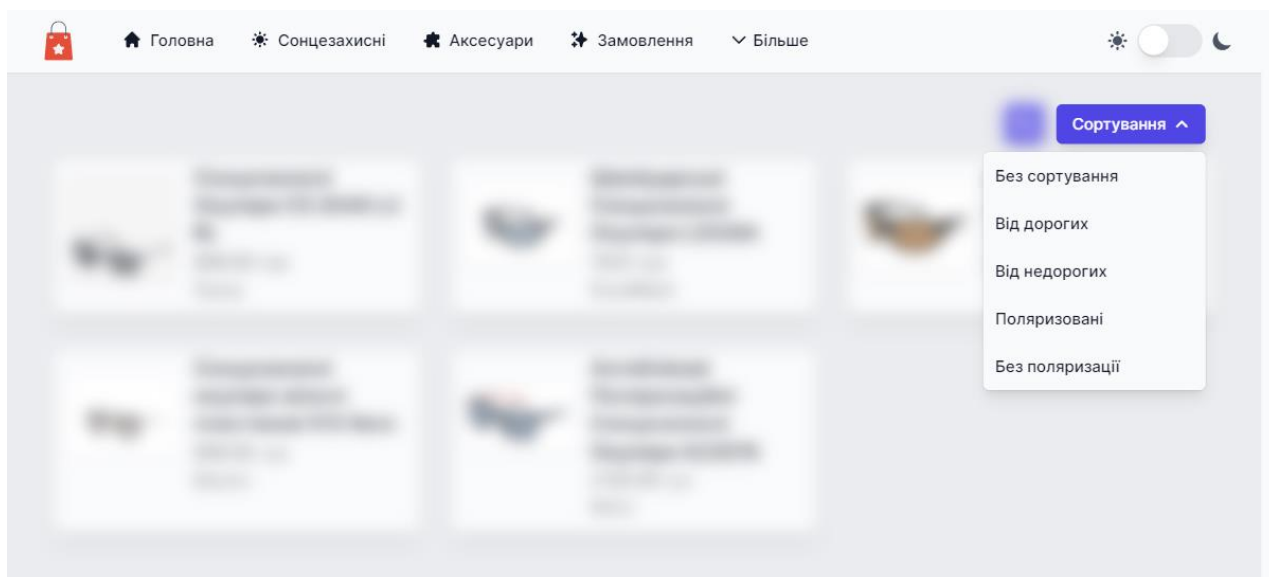


Рисунок Г.26 – Сторінка сонцезахисних окулярів з відкритим сортуванням

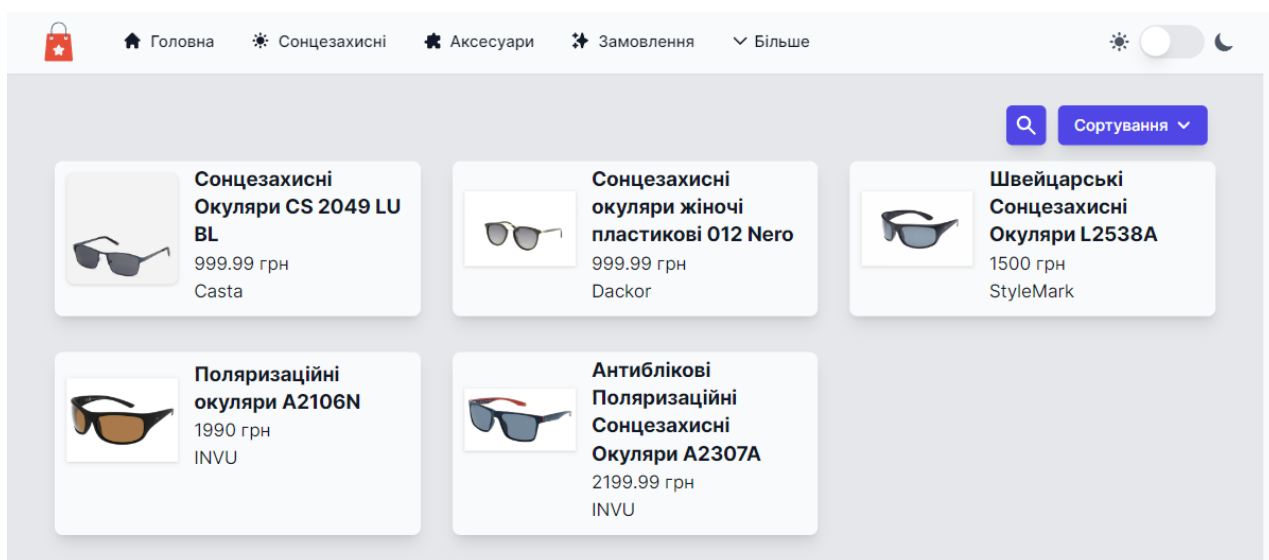


Рисунок Г.27 – Сторінка сонцезахисних окулярів відсортованих за ціною

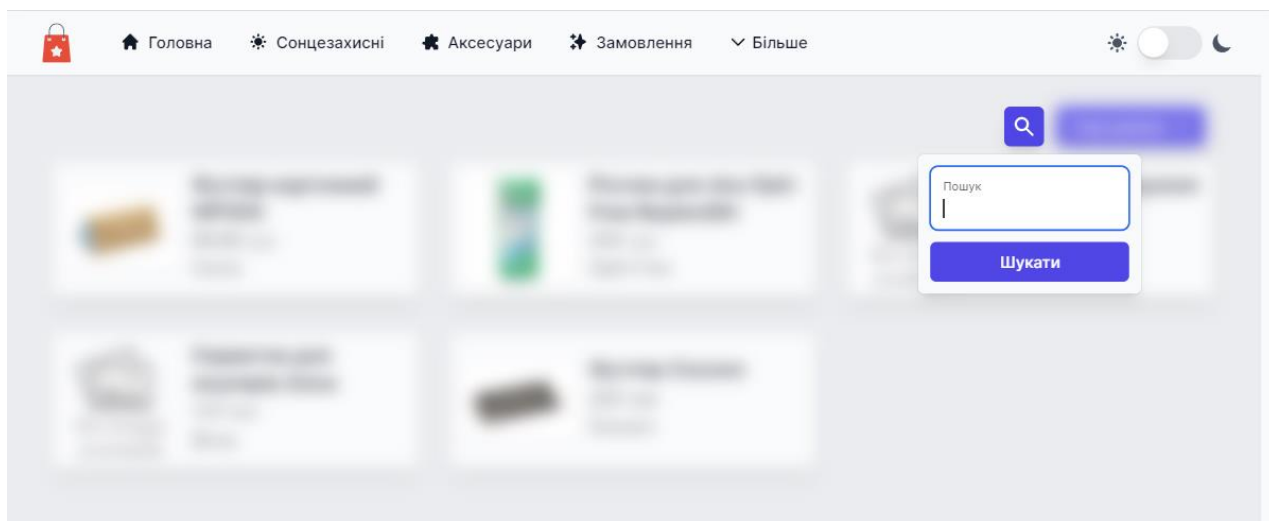


Рисунок Г.28 – Сторінка аксесуарів та лінз з відкритим пошуком

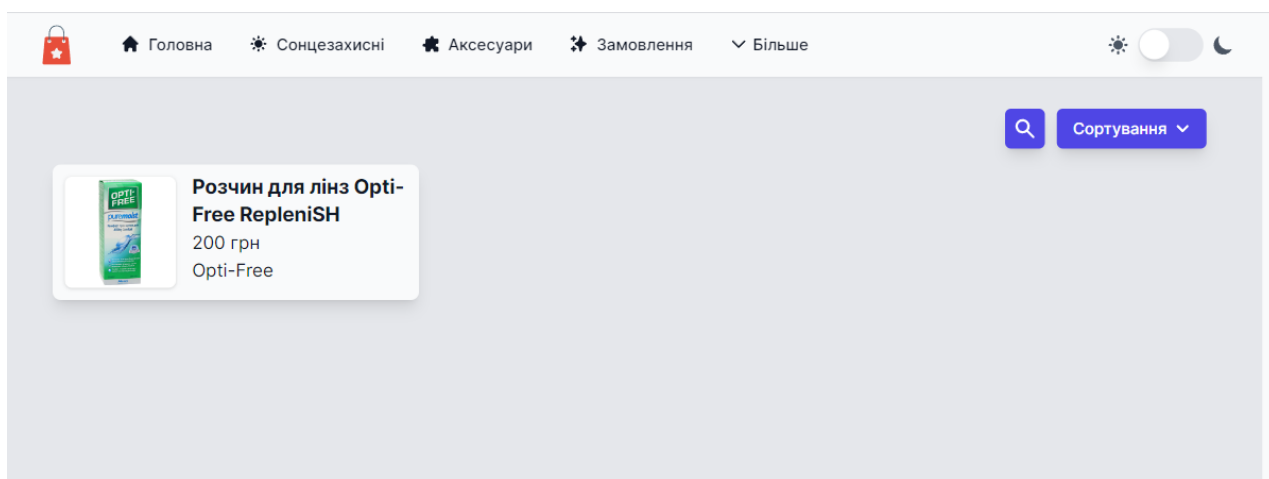


Рисунок Г.29 – Сторінка аксесуарів та лінз з пошуком "opti"

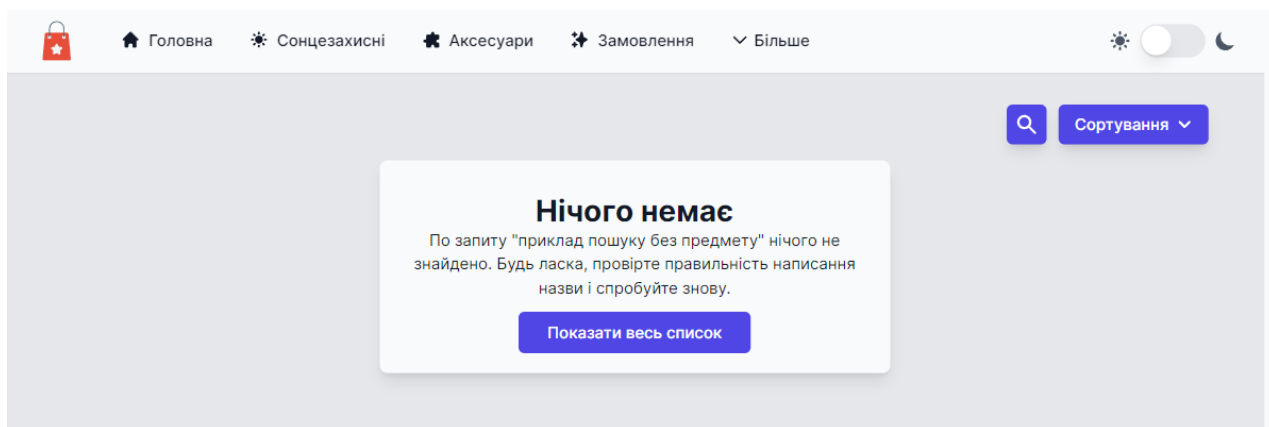


Рисунок Г.30 – Сторінка аксесуарів та лінз з вікном, яке інформує про відсутність предметів по запиті

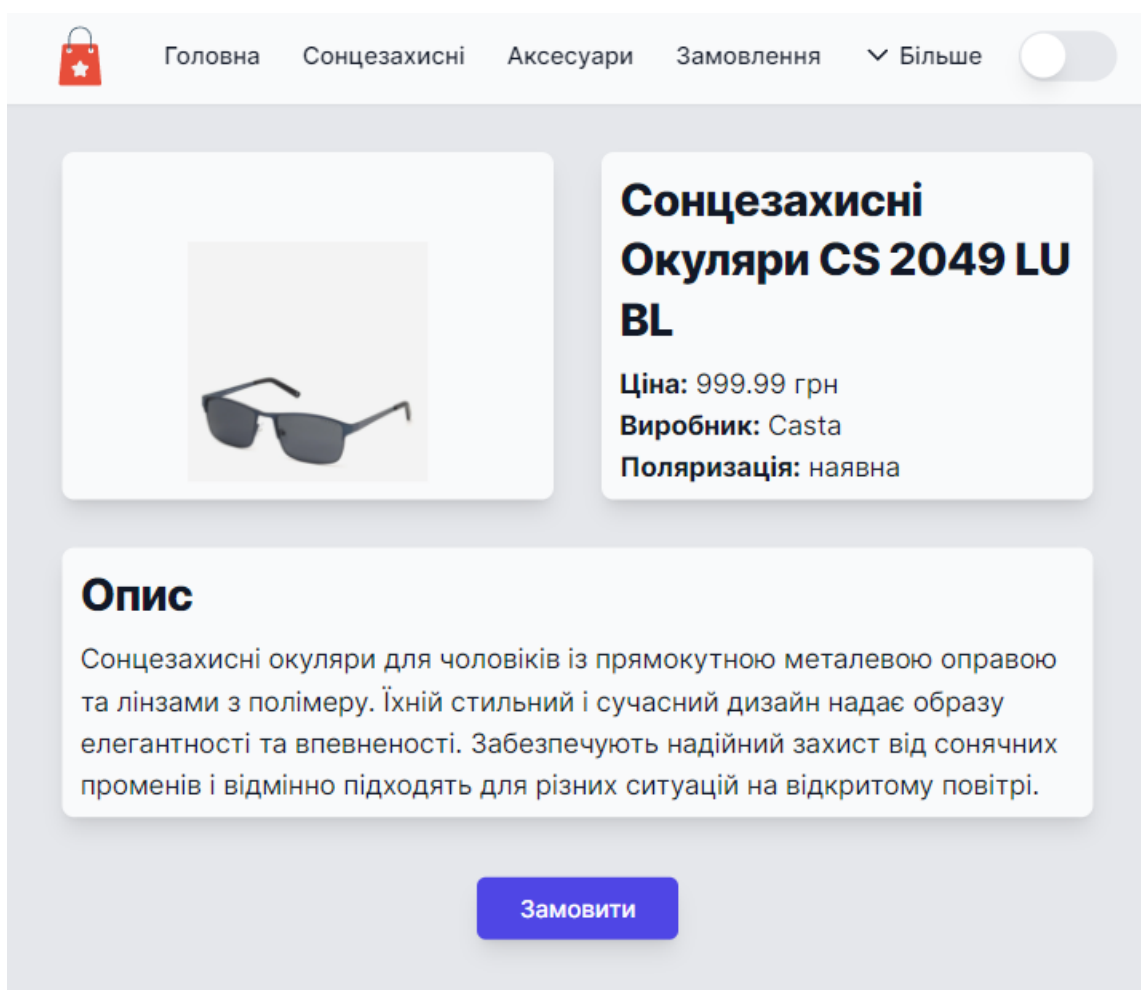


Рисунок Г.31 – Сторінка з сонцезахисними окулярами

Оформити замовлення

Вибіріть оправу та лінзу:

Wayfarer 1200 грн Ray-Ban	Frogskins 1400 грн Oakley	PO3019S 1800 грн Persol
---	---	---

Однофокальні 1200 грн Zeiss Німеччина	Мультифокальні 2500 грн Essilor Франція	Біф 1800 грн Hoya Япо
---	---	---

Вкажіть ваш рецепт:

Діоптрії (праве око)

Діоптрії (ліве око)

Астегматика (праве око)

Астегматика (ліве око)

DP (міжзнічна відстань)

Додаткові параметри (опціонально)

Вкажіть куди доставити:

Адрес

☐ Швидка доставка

Добавити

Головна

Рисунок Г.32 – Сторінка замовлення з конструктором

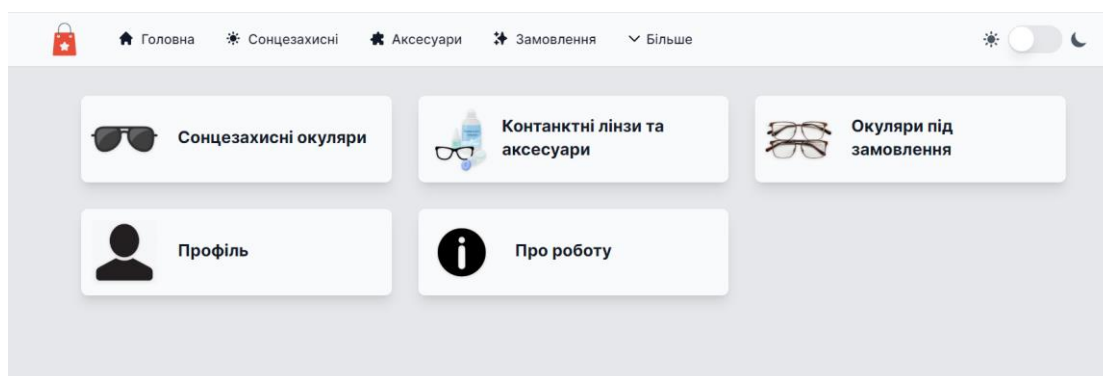


Рисунок Г.33 – Головна сторінка

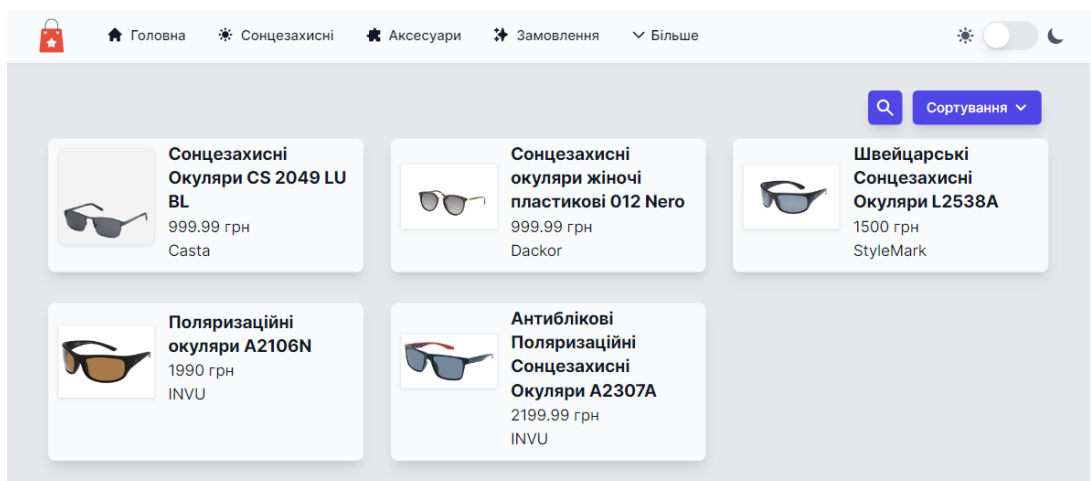


Рисунок Г.34 – Каталог з сонцезахисними окулярами

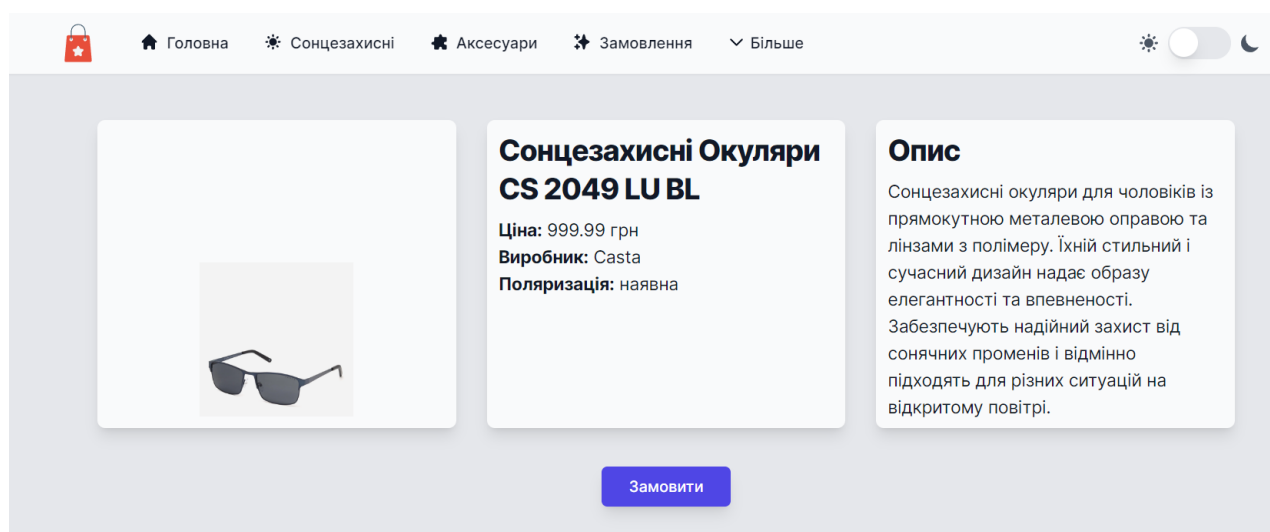


Рисунок Г.35 – Сторінка з сонцезахисними окулярами

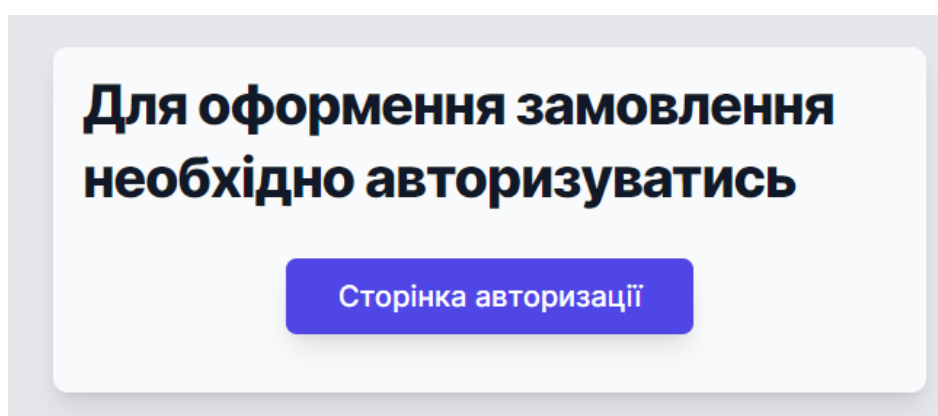
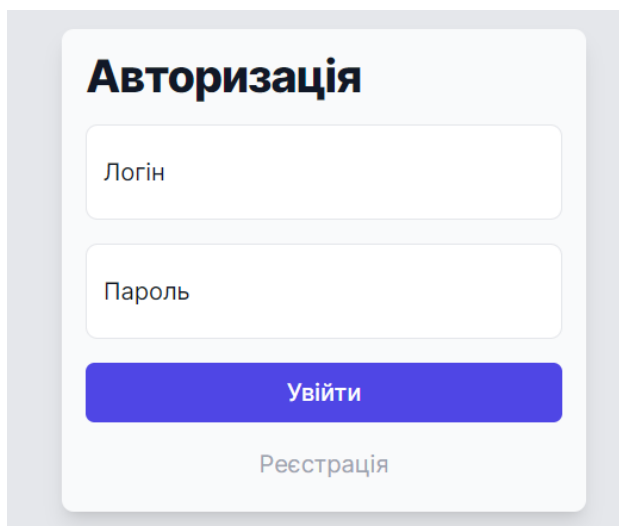
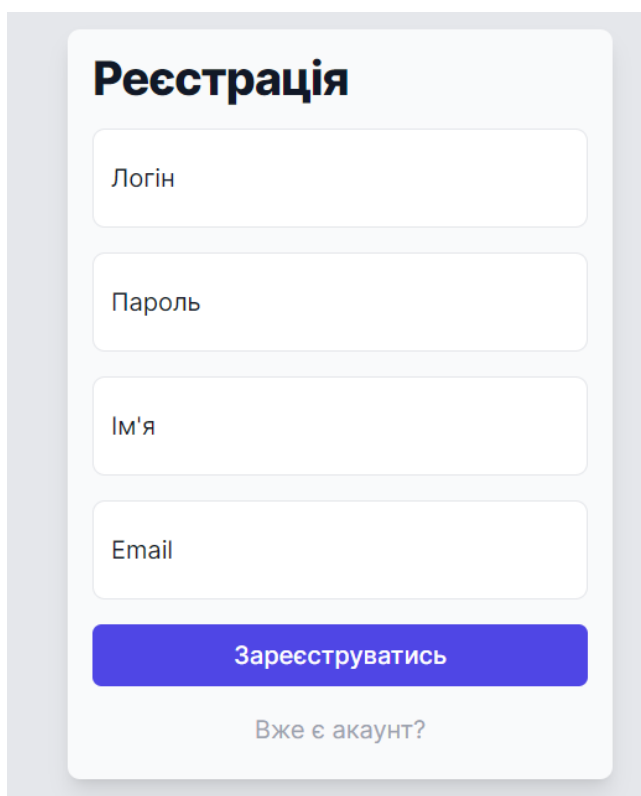


Рисунок Г.36 – Сторінка з сонцезахисними окулярами



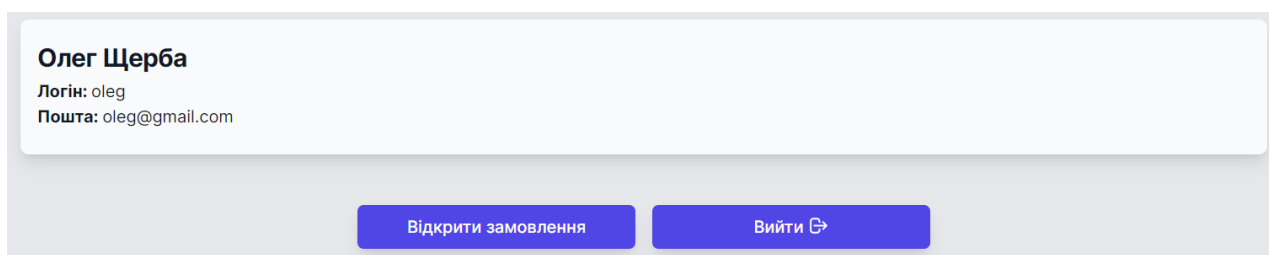
The image shows a login form titled "Авторизація" (Authorization). It contains two input fields: "Логін" (Login) and "Пароль" (Password). Below these fields is a blue button labeled "Увійти" (Login). At the bottom of the form, there is a link labeled "Реєстрація" (Registration).

Рисунок Г.37 – Сторінка авторизації



The image shows a registration form titled "Реєстрація" (Registration). It contains four input fields: "Логін" (Login), "Пароль" (Password), "Ім'я" (Name), and "Email". Below these fields is a blue button labeled "Зареєструватись" (Register). At the bottom of the form, there is a link labeled "Вже є акаунт?" (Already have an account?).

Рисунок Г.38 – Сторінка реєстрації



The image shows a user profile card for "Олег Щерба" (Oleg Shcherba). It displays the user's login "Логін: oleg" and email "Пошта: oleg@gmail.com". Below the card, there are two blue buttons: "Відкрити замовлення" (Open order) and "Вийти" (Logout) with an external link icon.

Рисунок Г.39 – Сторінка профілю користувача

Оформити замовлення

Вибіріть оправу та лінзу:

 Wayfarer 1200 грн Ray-Ban	 Frogskins 1400 грн Oakley	 PO3019S 1800 грн Persol
Однофокальні 1200 грн Zeiss Німеччина	Мультифокальні 2500 грн Essilor Франція	Біф 1800 грн Hoya Японія

Вкажіть ваш рецепт:

Діоптрії (праве око)

Діоптрії (ліве око)

Астегматика (праве око)

Астегматика (ліве око)

DP (міжзінична відстань)

Додаткові параметри (опціонально)

Вкажіть куди доставити:

Адрес

☐ Швидка доставка

Добавити

Головна

Рисунок Г.40 – Сторінка оформлення замовлення

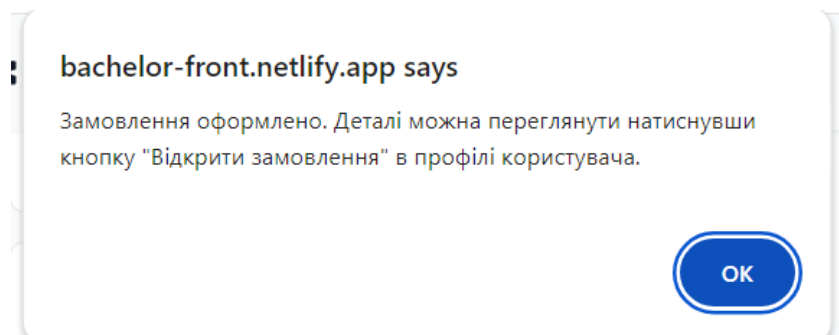


Рисунок Г.41 – Повідомлення про добавлене замовлення

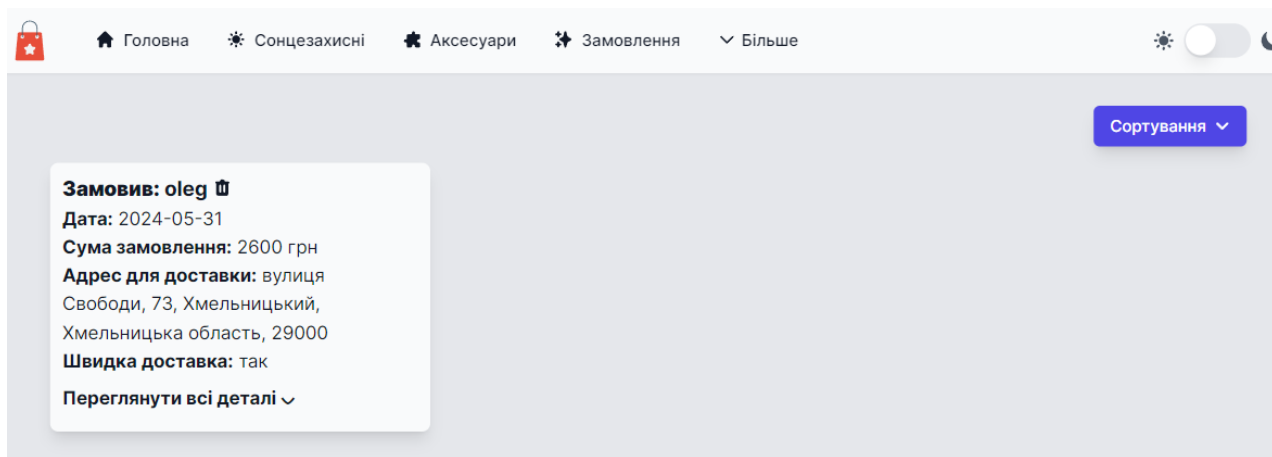


Рисунок Г.42 – Сторінка з замовленнями

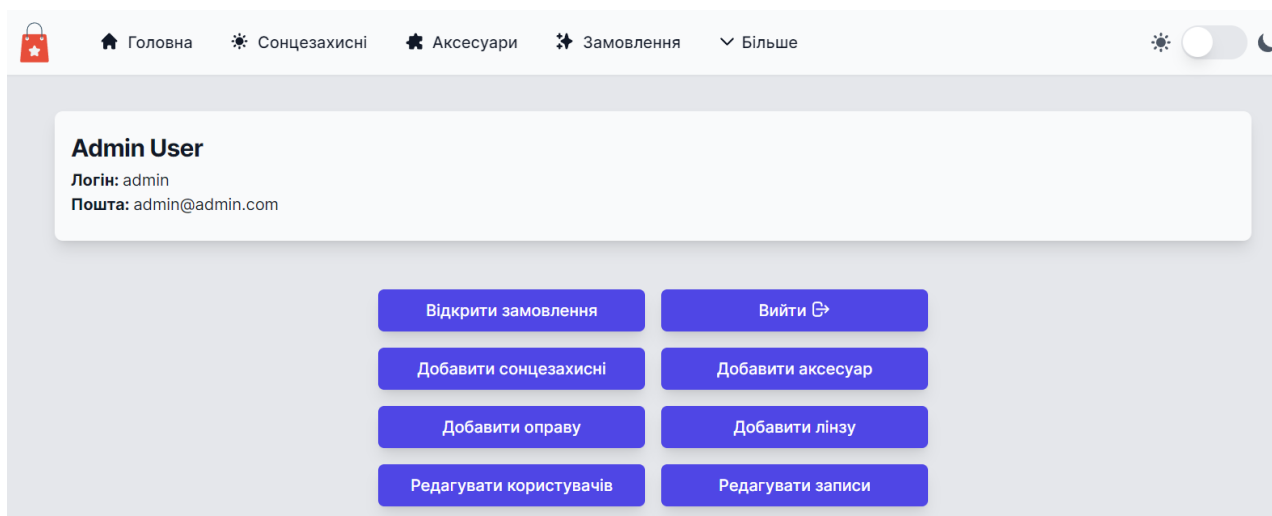
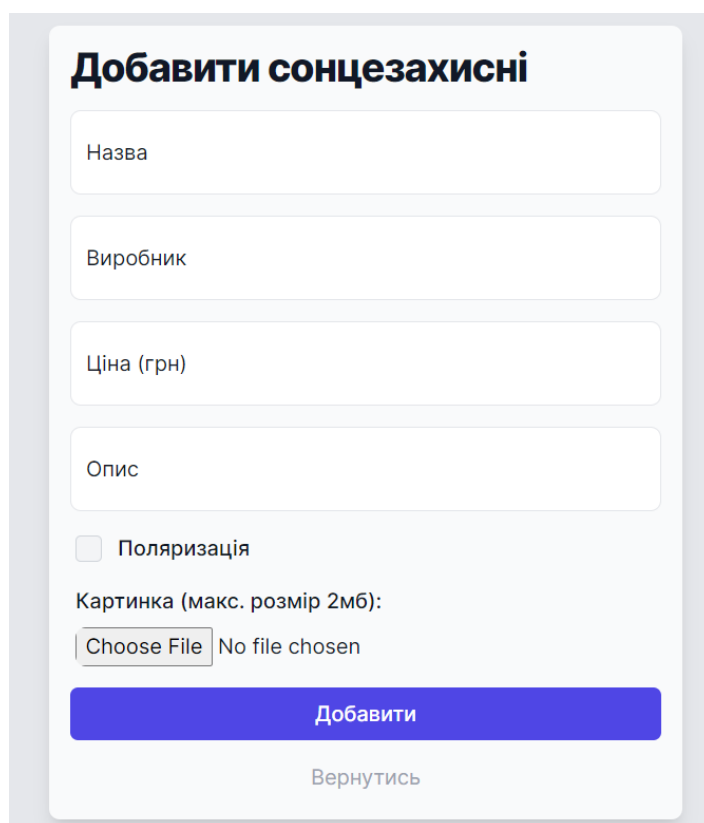


Рисунок Г.43 – Сторінка профілю користувача з роллю адмін



Добавити сонцезахисні

Назва

Виробник

Ціна (грн)

Опис

☐ Поляризація

Картинка (макс. розмір 2мб):

Choose File No file chosen

Добавити

Вернутись

Рисунок Г.44 – Форма для добавляння сонцезисних окулярів

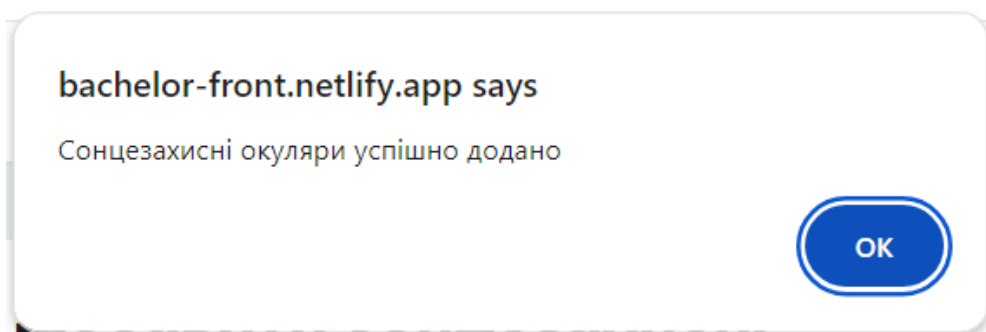


Рисунок Г.45 – Форма для добавляння сонцезисних окулярів