

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему:

Розробка веб-застосунку інтернет-магазину засобами мови
програмування JavaScript, бібліотеки React та платформи
Firebase

Виконала: студентка IV курсу, групи СПс-41

спеціальності 121 – Інженерія програмного забезпечення

(шифр і назва спеціальності)

Вовнянка О.Р.
(підпис) (прізвище та ініціали)

Керівник Яворська Є.Б.
(підпис) (прізвище та ініціали)

Нормоконтроль Стоянов Ю.М.
(підпис) (прізвище та ініціали)

Завідувач кафедри Петрик М.Р.
(підпис) (прізвище та ініціали)

Рецензент
(підпис) (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра програмної інженерії
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

проф. Петрик М.Р.

(підпис)

(прізвище та ініціали)

« 6 » квітня 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавр
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

студенту Вовнянці Оксані Романівній

1. Тема роботи Розробка веб-застосунку інтернет-магазину засобами мови програмування
JavaScript, бібліотеки React та платформи Firebase

Керівник роботи Яворська Євгенія Богданівна канд. техн. наук, зав. кафедри
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від « 06 » квітня 2026 року №

2. Термін подання студентом роботи 22.06.2026

3. Вихідні дані до роботи наукові літературні джерела

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

1 Аналіз предметної області.

2. Проектування та реалізація

3. Реалізація та тестування веб-застосунку.

4. Безпека життєдіяльності, основи охорони праці.

Висновки

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Мариненко С.Ю.		
Нормоконтроль	Стоянов Ю.М.		

7. Дата видачі завдання 6 квітня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Розробка технічного завдання</i>	6.04 – 12.04	
2.	<i>Робота над першим розділом «Аналіз предметної області»</i>	13.04 – 26.04	
3.	<i>Робота над другим розділом «Проектування веб-застосунку»</i>	27.04 – 03.05	
4.	<i>Робота над третім розділом «Реалізація та тестування веб-застосунку»</i>	04.05 – 17.05	
	<i>Робота над четвертим розділом «Безпека життєдіяльності, основи охорони праці»</i>	18.05 – 24.05	
5.	<i>Оформлення пояснювальної записки і графічного матеріалу</i>	25.05 – 7.06	
6.	<i>Перевірка на академічний плагіат, перевірка керівником та консультантами</i>	8.06 – 14.06	
7.	<i>Попередній захист кваліфікаційної роботи бакалавра</i>	15.06 – 21.06	
8.	<i>Захист кваліфікаційної роботи бакалавра</i>		
9.			
10.			
11.			
12.			

Студент

(підпис)

Вовнянка О.Р

(прізвище та ініціали)

Керівник роботи

(підпис)

Яворська С.Б

(прізвище та ініціали)

АНОТАЦІЯ

Вовнянка О. Р. Розробка веб-застосунку інтернет-магазину засобами мови програмування JavaScript, бібліотеки React та платформи Firebase : робота на здобуття кваліфікаційного ступеня бакалавра : спец. 121 – інженерія програмного забезпечення / наук. кер. Є. Б. Яворська. Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2026. 54 с. // С. - 54, табл. - 7, рис. - 18, бібліогр. – 19, слайд. – 14, додат. - 4 ,

Ключові слова: вебзастосунок, інтернет-магазин, електронна комерція, React, TypeScript, Firebase Authentication, Cloud Firestore, кошик покупок, авторизація користувачів.

Метою роботи є розробка веб-застосунку інтернет-магазину для перегляду товарів, реєстрації та авторизації користувачів, формування кошика покупок і оформлення замовлень.

У першому розділі проаналізовано існуючі інтернет-магазини, обґрунтовано вибір технологій та визначено вимоги до системи.

У другому розділі виконано проєктування веб-застосунку, описано архітектуру програмного забезпечення, структуру бази даних та UML-діаграми.

У третьому розділі реалізовано веб-застосунок із використанням React, TypeScript та Firebase.

У четвертому розділі розглянуто питання охорони праці та безпеки життєдіяльності під час розробки програмного забезпечення.

Об'єктом дослідження є процеси автоматизації продажу товарів через веб-застосунки електронної комерції.

Предметом дослідження є методи та засоби розробки веб-застосунків інтернет-магазинів із використанням технологій React, Firebase Authentication та Cloud Firestore.

Методи дослідження включають аналіз предметної області, проєктування баз даних, веброзробку та тестування програмного забезпечення.

ABSTRACT

Vovnianska O. R. Development of an Online Store Web Application Using the JavaScript Programming Language, React Library, and Firebase Platform : Bachelor's Qualification Thesis : Specialty 121 – Software Engineering / Supervisor Ye. B. Yavorska. Ternopil : Ivan Puluj Ternopil National Technical University, 2026. 54 p. . // Pages - 54, tables - 7, figures - 18, bibliographic references - 19, presentation slides – 14, appendices -4.

Keywords: web application, online store, e-commerce, React, TypeScript, Firebase Authentication, Cloud Firestore, shopping cart, user authentication.

The purpose of the work is to develop an online clothing store web application that provides product catalog browsing, user registration and authentication, shopping cart management, and order placement.

The first chapter analyzes existing online stores, justifies the choice of technologies, and defines the system requirements.

The second chapter presents the design of the web application, including the software architecture, database structure, and UML diagrams.

The third chapter describes the implementation of the web application using React, TypeScript, and Firebase. A user authentication system, product catalog, shopping cart, and order processing functionality were developed, and system testing was carried out.

The fourth chapter examines occupational health and safety issues related to software development.

The object of the research is the processes of automating product sales through e-commerce web applications.

The subject of the research is the methods and tools for developing online store web applications using React, Firebase Authentication, and Cloud Firestore technologies.

The research methods include domain analysis, database design, web development, and software testing.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

API – інтерфейс прикладного програмування.

CSS – каскадні таблиці стилів.

DB – база даних.

HTML – мова розмітки гіпертекстових документів.

HTTP – протокол передачі гіпертексту.

JS – мова програмування JavaScript.

JSON – формат обміну даними JavaScript Object Notation.

NoSQL – нереляційна база даних.

React – бібліотека для створення користувацьких інтерфейсів.

TS – мова програмування TypeScript.

UI – інтерфейс користувача.

UX – користувацький досвід.

Firebase – платформа для розробки веб- та мобільних застосунків.

Firestore – хмарна NoSQL база даних платформи Firebase.

SPA – односторінковий вебзастосунок.

ЗМІСТ

ВСТУП.....	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1 Словник предметної області	10
1.2 Опис предметної області	11
1.3 Аналіз існуючих аналогів інтернет-магазинів	12
1.4 Формування вимог до програмного забезпечення	14
1.4.1 Функціональні вимоги.....	14
1.4.2 Нефункціональні вимоги.....	15
1.5. Проектування відношень між акторами та прецедентами.....	16
1.6 Висновки до першого розділу.....	19
2 ПРОЄКТУВАННЯ ВЕБ-ЗАСТОСУНКУ.....	20
2.1 Методології проектування	20
2.2 Мова програмування.....	21
2.3 Середовище розробки	22
2.4 Платформа Firebase	22
2.5 Проектування архітектури веб-застосунку	23
2.6 Проектування структури бази даних.....	24
2.7 Діаграма діяльності процесу оформлення замовлення	26
2.8 Висновки до другого розділу	28
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ.....	30
3.1 Реалізація структури веб-застосунку	30
3.2 Реалізація системи авторизації та реєстрації	32
3.3 Реалізація каталогу товарів	34
3.4 Реалізація сторінки товару та кошика.....	36
3.5 Реалізація оформлення замовлення.....	37
3.6 Тестування веб-застосунку	40
3.6.1 Тестування реєстрації користувачів.....	40
3.6.2 Тестування авторизації.....	40

3.6.3 Тестування роботи каталогу товарів.....	41
3.6.4 Тестування кошика покупок	41
3.6.5 Тестування оформлення замовлення	42
3.7 Висновки до третього розділу.....	42
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	44
4.1 Ергономічні проблеми безпеки життєдіяльності.....	44
4.2 Вимоги ергономіки до організації робочого місця оператора ПК.....	45
4.3. Висновки до четвертого розділу.....	46
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50
ДОДАТКИ.....	55
ДОДАТОК А.....	56
ДОДАТОК Б	57
ДОДАТОК В.....	58
ДОДАТОК Г	59

ВСТУП

Актуальність теми дослідження. Сучасний розвиток електронної комерції сприяє активному впровадженню вебтехнологій у сферу продажу товарів та послуг. Інтернет-магазини стали одним із найбільш популярних способів здійснення покупок завдяки доступності, зручності та можливості швидкого отримання необхідної інформації про продукцію. У зв'язку зі зростанням кількості користувачів та обсягів даних виникає потреба у створенні сучасних вебзастосунків, які забезпечують швидку роботу, зручний інтерфейс, надійне зберігання даних та безпечну взаємодію з користувачами. Одним із ефективних підходів до реалізації таких систем є використання бібліотеки React та хмарної платформи Firebase.

Мета дослідження – розробка вебзастосунку інтернет-магазину одягу з використанням мови програмування JavaScript, бібліотеки React та платформи Firebase для забезпечення зручного перегляду товарів, авторизації користувачів та оформлення замовлень.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести аналіз предметної області та існуючих аналогів інтернет-магазинів одягу.
2. Визначити функціональні та нефункціональні вимоги до програмного продукту.
3. Спроекувати архітектуру вебзастосунку та структуру бази даних.
4. Розробити користувацький інтерфейс інтернет-магазину.
5. Реалізувати систему реєстрації та авторизації користувачів засобами Firebase Authentication.
6. Реалізувати механізм зберігання даних та обробки замовлень за допомогою Cloud Firestore.
7. Провести тестування вебзастосунку та оцінити його працездатність.

Об'єкт дослідження – процеси організації електронної торгівлі засобами вебтехнологій.

Предмет дослідження – методи та засоби розробки вебзастосунків інтернет-магазинів із використанням React та Firebase.

Методи дослідження – аналіз предметної області, об'єктно-орієнтоване проєктування, UML-моделювання, сучасні технології веброзробки, тестування програмного забезпечення.

Практичне значення результатів полягає у створенні вебзастосунку інтернет-магазину одягу, який забезпечує перегляд каталогу товарів, реєстрацію та авторизацію користувачів, управління кошиком покупок і оформлення замовлень. Розроблений програмний продукт може бути використаний як основа для створення повноцінних систем електронної комерції та подальшого розширення їх функціональних можливостей.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Ця робота ґрунтується на сучасних технологіях веброзробки та принципах створення електронних комерційних систем. Використання мови програмування JavaScript, бібліотеки React та платформи Firebase дозволяє реалізувати сучасний вебзастосунок із високою продуктивністю, зручним користувацьким інтерфейсом та можливістю ефективного керування даними. Застосування зазначених технологій сприяє створенню масштабованого програмного рішення, яке відповідає сучасним вимогам ринку електронної комерції та забезпечує якісну взаємодію між продавцем і покупцем [1].

Розробка веб-застосунку інтернет-магазину поєднує питання проєктування програмного забезпечення, створення сучасного користувацького інтерфейсу та організації ефективного зберігання й обробки даних. Особливу увагу приділено використанню сучасних вебтехнологій, які забезпечують високу швидкодію системи, зручність користування та можливість подальшого масштабування програмного продукту.

1.1 Словник предметної області

Інтернет-магазин – місце в інтернеті, де відбувається прямий продаж товарів споживачеві, враховуючи доставку. При цьому розміщення споживацької інформації, замовлення товару і угода відбуваються там само, всередині мережі.

Головна сторінка – початкова сторінка інтернет-магазину, яка, зазвичай, надає відомості про тематику інтернет-магазину та матеріали, які можна побачити на подальших сторінках.

Сторінка контактів – це сторінка, на якій знаходяться мобільні номери компанії, для зв'язку з центром підтримки при виникненні непередбаченої ситуації.

Сторінка «Про нас» – це сторінка, на якій знаходиться основна інформація про компанію.

Сторінка товару – це сторінка, на якій знаходиться вся інформація про товар.

Ціна – фундаментальна економічна категорія, яка означає кількість грошей, за яку продавець згоден продати, а покупець готовий купити одиницю товару (певної цінності, в т. ч. нематеріальної, наприклад, знання).

Корзина – це місце, куди потрапляють вибрані покупцем товари для купівлі.

Категорія – це сукупність товарів певного класу. Таблиця “Категорій” має вигляд дерева і кожен товар у системі прив'язується до певної категорії.

Товар – продукт природи і людської праці або тільки людської праці у матеріальній і нематеріальній субстанції та у формі послуг, який завдяки своїм властивостям здатен задовольняти наявні чи передбачувані суспільні потреби.

Користувач – людина, що користується послугами обчислювальної техніки для отримання інформації чи розв'язування різноманітних задач.

Адміністратор – це особа, яка здійснює роботу з якісного і ефективного обслуговування відвідувачів, консультує їх із питань, які стосуються товарів і послуг, які надає підприємство, організація.

Акаунт – сукупність наданої інформації про користувача, засобів та прав користувача відносно багатокористувацької системи.

Авторизація – керування рівнями та засобами доступу до певного захищеного ресурсу, як у фізичному розумінні, так і в галузі цифрових технологій та ресурсів системи залежно від ідентифікатора і пароля користувача або надання певних повноважень на виконання деяких дій у системі обробки даних.

Реєстрація – процес створення нового акаунту користувача.

1.2 Опис предметної області

Сфера інформаційних технологій та електронної комерції динамічно розвивається, охоплюючи дедалі більше галузей роздрібної торгівлі, зокрема індустрію одягу. Онлайн-продаж одягу став звичним елементом повсякденного життя споживачів, оскільки поєднує зручність вибору, широкий асортимент та доступність товарів незалежно від місця перебування покупця.

Предметна область онлайн-магазину включає різні категорії продукції, такі як верхній одяг, повсякденний та святковий одяг, взуття й аксесуари. Важливу роль відіграють параметри товарів, зокрема розмір, матеріал, колір, сезонність та виробник. Для ефективної роботи інтернет-магазину необхідно забезпечити зручну навігацію, детальний опис товарів, якісні зображення та можливість фільтрації продукції за ключовими характеристиками [2,3].

З огляду на різноманітність потреб споживачів, онлайн-магазин одягу повинен орієнтуватися на різні цільові групи, враховуючи індивідуальні вподобання, стиль життя та фінансові можливості користувачів. Важливими аспектами також є забезпечення безпеки персональних даних, надійність платіжних операцій та зручність процесу оформлення замовлення.

Даний розділ присвячений аналізу основних особливостей предметної області онлайн-торгівлі одягом, визначенню ключових факторів, що впливають на вибір і придбання товарів, а також формуванню вимог до інтернет-магазину з метою забезпечення максимальної задоволеності потреб споживачів у даному сегменті ринку.

1.3 Аналіз існуючих аналогів інтернет-магазинів

Перед початком розробки веб-застосунку інтернет-магазину доцільно провести аналіз існуючих аналогів, що дозволяє визначити найбільш затребувані функціональні можливості сучасних систем електронної комерції, виявити їх переваги та недоліки, а також сформулювати вимоги до майбутнього програмного продукту.

Одним із найбільш популярних інтернет-магазинів одягу в Україні є Kasta. Даний вебресурс пропонує широкий асортимент товарів, включаючи жіночий, чоловічий та дитячий одяг, взуття й аксесуари. Серед основних переваг Kasta можна виділити зручну систему пошуку та фільтрації товарів, наявність персональних рекомендацій, можливість додавання товарів до списку обраного та адаптивний інтерфейс для мобільних пристроїв. Крім того, сервіс регулярно

пропонує акції та спеціальні пропозиції для користувачів. Разом із тим значна кількість рекламних елементів та велика кількість інформації на окремих сторінках можуть ускладнювати навігацію для нових користувачів [6].

Іншим відомим представником ринку є Answer, який спеціалізується на продажі брендового одягу, взуття та аксесуарів [7]. Вебсайт має сучасний дизайн, зручну структуру каталогу та якісну систему фільтрації продукції. Користувачі можуть здійснювати пошук товарів за брендом, кольором, сезоном, розміром та іншими характеристиками. Перевагою сервісу є також висока швидкість роботи та інтуїтивно зрозумілий інтерфейс. Недоліком можна вважати значну кількість категорій і підкатегорій, що іноді ускладнює пошук необхідного товару.

Для аналізу також було розглянуто офіційний інтернет-магазин Reserved [8]. Даний вебресурс відрізняється мінімалістичним дизайном та високою швидкістю завантаження сторінок. Особлива увага приділяється візуальному представленню товарів, що позитивно впливає на сприйняття продукції користувачами. Для кожного товару доступний детальний опис, фотографії високої якості та інформація про доступні розміри. Разом із тим функціональні можливості сервісу є менш різноманітними порівняно з великими маркетплейсами.

Проведений аналіз показав, що сучасний інтернет-магазин одягу повинен забезпечувати зручний пошук товарів, ефективну систему фільтрації, можливість реєстрації та авторизації користувачів, формування кошика покупок, оформлення замовлень та адаптивне відображення інтерфейсу на різних типах пристроїв. Також важливими характеристиками є висока швидкість роботи веб-застосунку, надійне зберігання даних та зручність адміністрування каталогу товарів [9].

На основі проведеного аналізу було визначено функціональні можливості, які доцільно реалізувати у власному веб-застосунку інтернет-магазину одягу. До них належать система реєстрації та авторизації користувачів, каталог товарів із можливістю пошуку та фільтрації, сторінка детального перегляду товару, кошик покупця, оформлення замовлень та адміністративна панель для управління товарами й замовленнями.

Таблиця 1.3 – Порівняльна характеристика існуючих аналогів

№	Функціональність	Kasta	Answer	Reserved	Розроблюваний вебзастосунок
1	Реєстрація користувачів	+	+	+	+
2	Авторизація	+	+	+	+
3	Каталог товарів	+	+	+	+
4	Пошук товарів	+	+	+	+
5	Фільтрація товарів	+	+	+	+
6	Кошик покупця	+	+	+	+
7	Оформлення замовлення	+	+	+	+
8	Адаптивний дизайн	+	+	+	+
9	Адміністративна панель	-	-	-	+

1.4 Формування вимог до програмного забезпечення

Формування вимог до програмного забезпечення (ПЗ) є ключовим етапом в розробці інтернет-магазину для продажу одягу. Для забезпечення оптимальної функціональності та задоволення потреб користувачів, необхідно визначити та деталізувати вимоги до різних аспектів системи [11,12].

1.4.1 Функціональні вимоги

Для авторизованих користувачів система надає широкий набір функціональних можливостей. Адміністратор має можливість створювати нові профілі користувачів, переглядати список усіх зареєстрованих користувачів, а також редагувати персональну інформацію, зокрема ім'я, номер телефону, фотографію профілю, адресу електронної пошти та пароль.

Користувачі можуть виконувати вхід до власного облікового запису, здійснювати зручне керування категоріями товарів та переглядати каталог продукції без необхідності перезавантаження сторінки після вибору категорії. Для кожного товару передбачено компактне прев'ю, яке містить основну інформацію про продукт, зокрема фотографію, назву моделі та вартість.

Система забезпечує зручний перегляд характеристик товарів. Сторінки

різних категорій можуть відрізнятися кольоровим оформленням для покращення навігації. Користувач має можливість переглядати детальну інформацію про товар, додавати його до кошика та переглядати перелік обраних товарів у кошику. Також передбачено можливість отримання консультації щодо продукції. Для адміністраторів додатково реалізовано функції перегляду аналітичної інформації.

Неавторизовані користувачі мають доступ до функцій реєстрації нового облікового запису та автентифікації в системі.

У процесі розробки програмного забезпечення функціональні можливості системи можуть бути розширені та доповнені. Перспективними напрямками розвитку є реалізація функції додавання товарів до списку обраного, перегляд обраних товарів, налаштування параметрів облікового запису, оформлення замовлень, формування та надсилання відгуків щодо виявлених проблем, а також додавання нових товарів до каталогу адміністраторами системи.

Результатом роботи програмного забезпечення є коректне відображення товарів у каталозі та забезпечення можливості їх додавання до кошика для подальшого оформлення замовлення.

1.4.2 Нефункціональні вимоги

Надійність

Однією з основних вимог до системи є зручність використання. Застосунок повинен мати інтуїтивно зрозумілий інтерфейс, який забезпечує комфортну роботу користувачів незалежно від рівня їх підготовки. Інтерфейс має бути адаптивним та коректно відображатися на різних мобільних пристроях.

Система повинна забезпечувати високий рівень доступності протягом усього робочого дня, а для адміністраторів – цілодобово. Якщо користувач тривалий час не виконує жодних дій у системі, має автоматично запускатися механізм повторної аутентифікації із запитом логіна та пароля. Час, необхідний для технічного обслуговування системи, не повинен перевищувати 3 % від загального часу її роботи.

Ефективність

Програмний засіб повинен забезпечувати швидке виконання операцій. Час відгуку для стандартних завдань не повинен перевищувати 0,5 секунди, а для більш складних операцій – 1,5 секунди.

Придатність до експлуатації

Система повинна підтримувати можливість масштабування та бути готовою до роботи з великою кількістю товарів різних категорій без втрати продуктивності.

Обмеження проєктування

Програмний засіб повинен стабільно, коректно та швидко функціонувати в середовищі виконання. Розробка системи здійснюється за клієнт-серверною архітектурою, яка складається з рівня представлення даних (Front-end) та рівня бізнес-логіки й обробки даних (Back-end).

Вимоги до захисту та доступу

Доступ до веб-застосунку здійснюється через веб-браузер із використанням мережі Інтернет.

Вимоги до інтерфейсу системи

Інтерфейс веб-застосунку повинен коректно відображатися на персональних комп'ютерах, планшетах та мобільних пристроях. Дизайн застосунку має бути мінімалістичним, зрозумілим і функціональним. Сторінка товару повинна містити повний та зрозумілий опис продукції. Для кожного товару необхідно відображати якісне зображення, яке дає змогу оцінити його зовнішній вигляд. Керування категоріями товарів має бути простим, зручним та ефективним [11].

Вимоги до програмного забезпечення

Веб-застосунок повинен підтримувати роботу в сучасних веб-браузерах Google Chrome, Mozilla Firefox, Microsoft Edge та Safari.

1.5. Проєктування відношень між акторами та прецедентами

В описаній системі, згідно з можливими точками розширення, виділено два актори:

Користувач – авторизувавшись, переглядає товари і шукає йому потрібні і додає в обране якщо йому сподобався товар, чи в корзину, якщо хоче його купити і оформляє замовлення. Якщо в нього виникають питання чи проблеми, то він може написати відгук, або подзвонити в службу підтримки магазину.

Адміністратор – додає нові товари, слідкує за роботою бази даних, аналізує та відповідає на повідомлення користувачів. Приймає міри щодо вирішення помилок та покращення роботи системи.

База даних – додає та надає доступ до відгуків користувачам, в залежності від їх статусу (звичайний користувач – тільки надсилати, адміністратор – надсилати та читати надіслані відгуки), а також зберігає та надає реєстраційні дані користувачів (дані надаються додатку задля автоматичної авторизації після реєстрації).

Користувачів у системі є два типи: адміністратор та користувач. Різниця у користуванні полягає у тому, що у адміністратора є доступ до бази даних з відгуками користувачів. Схема взаємодії основних компонентів системи показана на рисунку 1.1.

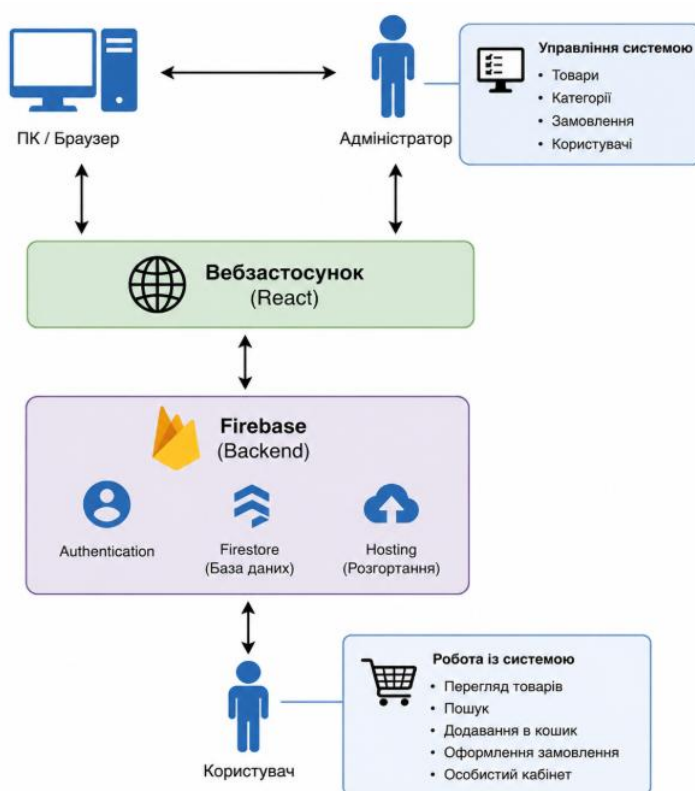


Рисунок 1.1 – Схема взаємодії основних компонентів системи

Таблиця 1.1 – Варіанти використання системи

Код	Основний актор	Найменування	Формулювання
K1	Користувач	Реєстрація нового користувача в додатку	Для того щоб отримати повний доступ для користування функціями веб-додатку потрібно зареєструватися
K2	Користувач	Авторизація	Ввести номер телефону та пароль для входу.
K3	Користувач	Вибір категорії	Авторизуватися та вибрати потрібну категорію.
K4	Користувач	Перегляд інформації про товар	Натиснути на товар для перегляду його детальної інформації.
K5	Користувач	Додання товару в корзину	Перейти на сторінку товару та натиснути «Додати в корзину».
K6	Користувач	Перегляд товарів в корзині	Натиснути «Корзина» для перегляду доданих товарів.
K7	Користувач	Отримання консультації	Знайти контакти підтримки та звернутися через електронну пошту або телефон.
A1	Адміністратор	Вхід адміністратора	Увійти через спеціальну форму для адміністратора.
A2	Адміністратор	Перевірка відгуків	Переглянути електронну пошту підтримки та дати відповіді на відгуки.
A3	Адміністратор	Створення та зміна профілів користувачів	У Firebase створити чи змінити профіль користувача
A4	Адміністратор	Перегляд списку існуючих профілів	У Firebase переглянути всі профілі користувачів.
A5	Адміністратор	Редагування персональної інформації	У Firebase змінити інформацію вибраного користувача
A6	Адміністратор	Перегляд аналітики	У Firebase переглянути аналітичні дані про використання веб-застосунку.

Для кращого візуального сприйняття зв'язків між акторами та варіантами використання можна створити діаграму варіантів використання. На діаграмі варіантів використання, на якій зображено актори та варіанти використання, можна побачити той функціонал, який необхідно реалізувати в системі рисунок 1.2.

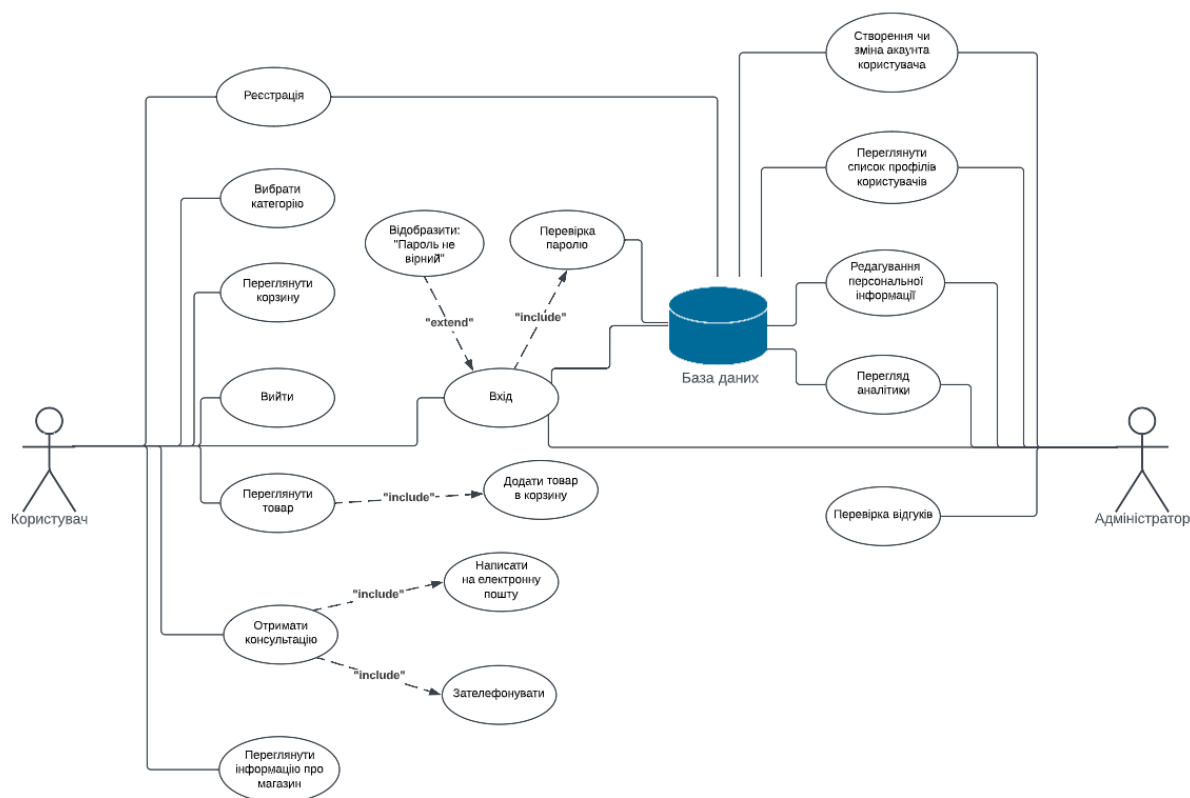


Рисунок 1.2 – Діаграма варіантів використання (Додаток Б)

1.6 Висновки до першого розділу

У першому розділі було проведено аналіз предметної області розробки веб-застосунку інтернет-магазину. Розглянуто основні поняття та особливості функціонування систем електронної комерції, визначено ключові терміни предметної області та проаналізовано процес взаємодії користувачів із системою.

У ході дослідження було виконано аналіз існуючих аналогів інтернет-магазинів одягу, що дозволило визначити їх основні переваги та недоліки, а також сформуванати перелік функціональних можливостей, які доцільно реалізувати у розроблюваному веб-застосунку. На основі проведеного аналізу сформовано функціональні та нефункціональні вимоги до системи.

Також було визначено основних акторів системи та побудовано діаграму варіантів використання, яка відображає взаємодію користувачів із вебзастосунком. Отримані результати стали основою для подальшого проєктування архітектури системи, вибору технологій розробки та реалізації програмного продукту.

2 ПРОЄКТУВАННЯ ВЕБ-ЗАСТОСУНКУ

В даному розділі розглянуто методологію проєктування, обґрунтовано вибір мови програмування JavaScript, бібліотеки React та платформи Firebase, а також описано середовище розробки, структуру бази даних і особливості проєктування користувацького інтерфейсу веб-застосунку інтернет-магазину.

2.1 Методології проєктування

Була обрана методологія аналізу вимог і гнучкий процес розробки ПЗ ICONIX, що ґрунтується на варіантах використання. ICONIX надає більше вимог та проектної документації і прагне уникнути паралічу аналізу. Процес ICONIX використовує лише чотири діаграми на основі UML у чотири-етапному процесі, який перетворює текст прецеденту на робочий код.

Принциповою відмінністю ICONIX є використання аналізу надійності, методу подолання розриву між аналізом та проєктуванням. Аналіз стійкості зменшує неоднозначність в описах варіантів використання, гарантуючи, що вони написані в контексті предметної області, що супроводжується моделями. Цей процес значно спрощує проєктування, тестування та оцінку варіантів використання.

Процес ICONIX поділено на чотири етапи. На кожному етапі робота з попередньої віхи переглядається та оновлюється.

Огляд вимог. Перед початком процесу ICONIX необхідно провести певний аналіз вимог. На основі цього аналізу можна визначити варіанти використання, створити модель предметної області та створити кілька прототипів графічних інтерфейсів.

Попередній огляд проєкту. Після визначення варіантів використання можна написати текст, що описує взаємодію користувача та системи. Виконується аналіз надійності, щоб знайти потенційні помилки у тексті варіанта використання, та відповідним чином оновлюється модель предметної області. Текст прецеденту

важливий для визначення того, як користувачі будуть взаємодіяти з системою. Вони також надають розробнику перевірити правильність результатів аналізу вимог.

Детальний аналіз проекту. На цьому етапі процесу ICONIX модель предметної області та текст прецеденту з етапу 2 використовуються для проектування системи, що будується. Діаграма класів створюється з моделі предметної області, а текст варіантів використання використовується створення діаграм послідовності.

Розгортання. Модульні тести написані для перевірки того, що система відповідатиме тексту варіанта використання та діаграм послідовності. Нарешті, код пишеться з використанням діаграм класів та послідовностей в якості настанови [7].

Ця методологія була обрана тому, що дозволить швидко та ефективно виконати задачі поставлені перед розробником проекту, йому не доведеться витратити більшу частину свого часу на формування діаграм, він зможе якісно описати усі вимоги та спроектувати систему згідно них, після чого протестувати її. Ця методологія не є найлегшою, проте знаходиться в золотій середині між швидкістю та ефективністю. Після того як проект пройде через усі чотири необхідні етапи, він буде завершеним та готовим до експлуатації.

2.2 Мова програмування

Для розробки веб-застосунку інтернет-магазину одягу було обрано мову програмування JavaScript, бібліотеку React та технологію CSS. Даний стек технологій забезпечує створення сучасного, швидкого та зручного вебресурсу.

JavaScript використовується як основна мова програмування клієнтської частини веб-застосунку. Вона дозволяє створювати динамічні сторінки, забезпечувати взаємодію з користувачем та виконувати обробку даних без перезавантаження сторінки.

React використовується для розробки користувацького інтерфейсу. Бібліотека забезпечує компонентний підхід до створення вебзастосунків, що

спрощує розробку, підтримку та подальше масштабування проєкту [13].

CSS застосовується для оформлення зовнішнього вигляду вебсайту. За допомогою CSS реалізується дизайн сторінок, налаштовуються кольори, шрифти, розташування елементів та адаптивність інтерфейсу для різних пристроїв.

Обрані технології дозволяють створити функціональний, сучасний та масштабований вебзастосунок, що відповідає вимогам до інтернет-магазинів.

2.3 Середовище розробки

Для розробки веб-застосунку використовується середовище Visual Studio Code (VS Code) – сучасний та безкоштовний редактор коду від Microsoft. Він забезпечує підтримку мов JavaScript, HTML та CSS, має зручний інтерфейс і широкий набір розширень для веброзробки.

Однією з переваг VS Code є інтеграція з системою контролю версій Git, а також підтримка розширення Live Server, яке дозволяє переглядати зміни в проєкті в режимі реального часу. Використання VS Code сприяє підвищенню ефективності розробки та спрощує підтримку програмного продукту [14].

2.4 Платформа Firebase

Для реалізації серверної частини веб-застосунку використовується платформа Firebase, розроблена компанією Google. Вона забезпечує зручні засоби для зберігання даних, авторизації користувачів та розгортання вебзастосунків.

У проєкті використовуються такі сервіси Firebase:

Cloud Firestore – для зберігання інформації про товари, користувачів та замовлення;

Firebase Authentication – для реєстрації та авторизації користувачів;

Firebase Hosting – для розгортання та розміщення веб-застосунку.

Використання Firebase дозволяє спростити процес розробки, забезпечити надійне зберігання даних та підвищити продуктивність системи [15].

2.5 Проєктування архітектури веб-застосунку

Під час розробки веб-застосунку інтернет-магазину важливим етапом є проєктування архітектури системи. Архітектура визначає структуру програмного забезпечення, взаємодію між окремими компонентами та спосіб обробки даних. Правильно спроектована архітектура забезпечує стабільну роботу системи, спрощує її підтримку та дозволяє легко розширювати функціональні можливості в майбутньому.

Для реалізації веб-застосунку було обрано клієнт-серверну архітектуру, у якій користувач взаємодіє із системою через веббраузер. Клієнтська частина розроблена за допомогою бібліотеки React і відповідає за відображення інтерфейсу користувача, обробку подій та взаємодію з сервісами Firebase [16].

Загальна архітектура розроблюваного веб-застосунку представлена на рисунку 2.1.



Рисунок 2.1 – Загальна архітектура системи

Користувач працює із вебзастосунком через браузер, де відображаються сторінки каталогу товарів, кошика, оформлення замовлень та особистого кабінету. Усі дії користувача обробляються React-компонентами, які забезпечують динамічне оновлення інформації без перезавантаження сторінки.

Для реалізації серверної частини використовується платформа Firebase, яка надає набір готових хмарних сервісів. Сервіс Firebase Authentication забезпечує реєстрацію та авторизацію користувачів, що дозволяє контролювати доступ до функціональних можливостей системи. Cloud Firestore використовується для зберігання інформації про товари, категорії, користувачів, кошик та замовлення. Firebase Storage призначений для зберігання графічних матеріалів, зокрема фотографій товарів. Розгортання веб-застосунку здійснюється за допомогою Firebase Hosting, який забезпечує доступ до системи через мережу Інтернет.

2.6 Проєктування структури бази даних

Одним із важливих етапів проєктування веб-застосунку інтернет-магазину одягу є розробка структури бази даних. Від правильності її побудови залежить ефективність зберігання інформації, швидкість обробки запитів та можливість подальшого розширення функціональності системи [17].

Для реалізації сховища даних було обрано сервіс Cloud Firestore платформи Firebase. Дана база даних має документно-орієнтовану структуру та забезпечує швидкий доступ до інформації, автоматичну синхронізацію даних і високий рівень масштабованості.

Структура бази даних розроблюваного веб-застосунку представлена на рисунку 2.2.

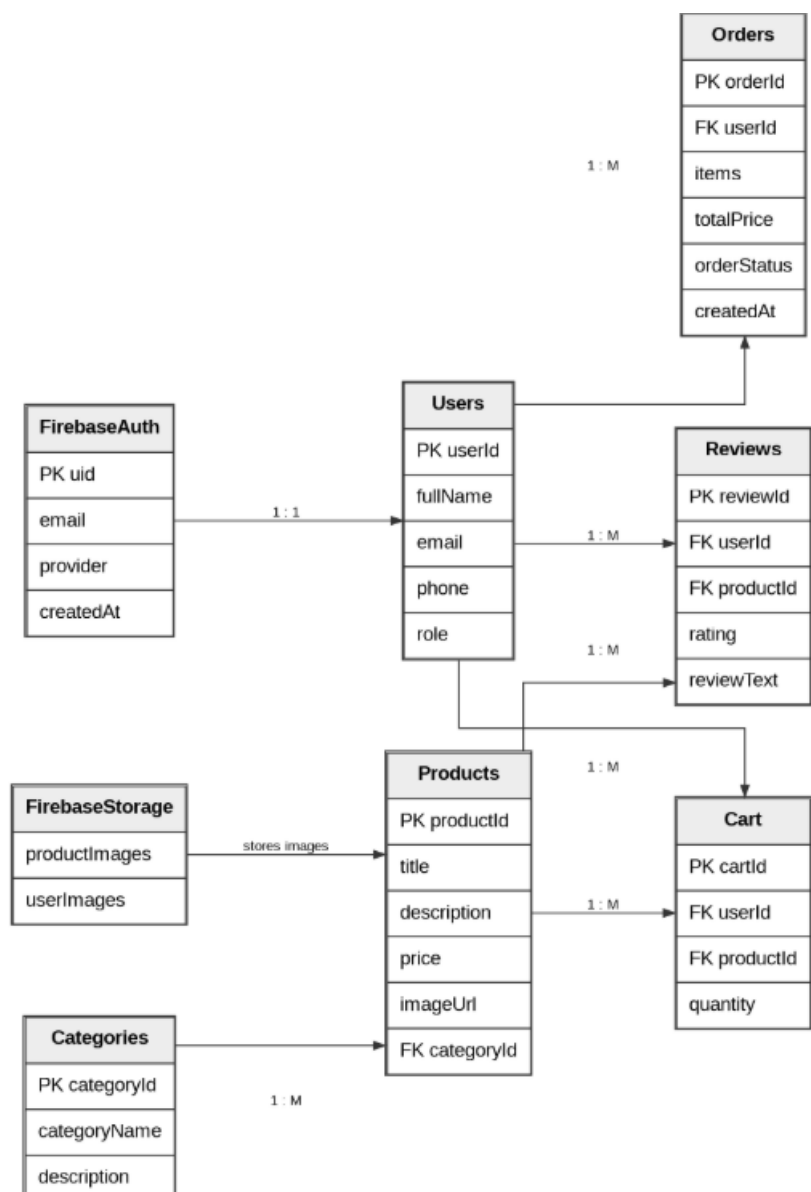


Рисунок 2.2 – Схема структури бази даних інтернет-магазину

База даних складається з кількох основних сутностей, які забезпечують роботу системи та зберігання необхідної інформації.

Сутність Users містить дані про зареєстрованих користувачів системи. До основних атрибутів належать ідентифікатор користувача, ім'я, адреса електронної пошти, номер телефону та роль користувача. Роль визначає рівень доступу до функціональних можливостей системи та дозволяє розмежувати звичайних користувачів і адміністраторів.

Сутність Products призначена для зберігання інформації про товари. Для кожного товару зберігаються назва, опис, ціна, зображення, категорія та інші

характеристики. Саме ця сутність формує каталог продукції, який відображається користувачам веб-застосунку.

Сутність Categories використовується для групування товарів за категоріями. Використання окремої колекції категорій спрощує пошук продукції та забезпечує зручну навігацію по каталогу.

Сутність Cart містить інформацію про товари, які користувач додав до кошика. Для кожного запису зберігається посилання на користувача, товар та його кількість. Це дозволяє формувати індивідуальний кошик для кожного користувача.

Сутність Orders використовується для зберігання оформлених замовлень. У ній міститься інформація про користувача, перелік товарів, дату створення замовлення, його поточний статус та загальну вартість покупки.

Для реалізації можливості оцінювання товарів може використовуватися сутність Reviews, яка містить текст відгуку, оцінку товару та посилання на користувача, що залишив відгук.

Запропонована структура бази даних забезпечує логічний зв'язок між основними сутностями системи та дозволяє ефективно організувати процес зберігання інформації. Використання Cloud Firestore також спрощує взаємодію між клієнтською частиною веб-застосунку та серверними сервісами Firebase.

Таким чином, спроектована структура бази даних забезпечує надійне зберігання інформації, швидкий доступ до даних та підтримує основні бізнес-процеси інтернет-магазину одягу, пов'язані з керуванням товарами, користувачами та замовленнями.

2.7 Діаграма діяльності процесу оформлення замовлення

Одним із ключових бізнес-процесів веб-застосунку інтернет-магазину є процес оформлення замовлення. Для наочного відображення послідовності дій користувача та реакції системи було побудовано діаграму діяльності. Використання даної діаграми дозволяє детально проаналізувати логіку функціонування системи та визначити основні етапи виконання процесу покупки

товару.

Діаграма діяльності відображає повний цикл оформлення замовлення від моменту вибору товару до підтвердження успішного створення замовлення. На відміну від діаграми варіантів використання, яка демонструє взаємодію користувача із системою на загальному рівні, діаграма діяльності детально показує послідовність виконання окремих операцій та переходів між ними [16].

Процес починається з перегляду каталогу товарів, де користувач може ознайомитися з доступною продукцією, виконати пошук необхідного товару або скористатися системою фільтрації. Після вибору потрібної позиції користувач переходить на сторінку товару для перегляду детальної інформації, характеристик та вартості.

Наступним етапом є додавання товару до кошика. Після виконання цієї дії система оновлює інформацію про вміст кошика та надає користувачеві можливість продовжити покупки або перейти безпосередньо до оформлення замовлення.

Під час оформлення замовлення користувач заповнює необхідні контактні дані, вказує адресу доставки та іншу інформацію, необхідну для обробки замовлення. Після введення даних система виконує перевірку правильності та повноти заповнення полів. У разі виявлення помилок користувачеві пропонується виправити введену інформацію.

Після успішної перевірки даних система формує замовлення та передає його до бази даних для збереження. Одночасно виконується оновлення інформації про стан замовлення та очищення кошика користувача. Після завершення цих операцій користувач отримує повідомлення про успішне оформлення покупки.

Діаграма діяльності процесу оформлення замовлення представлена на рисунку 2.3.

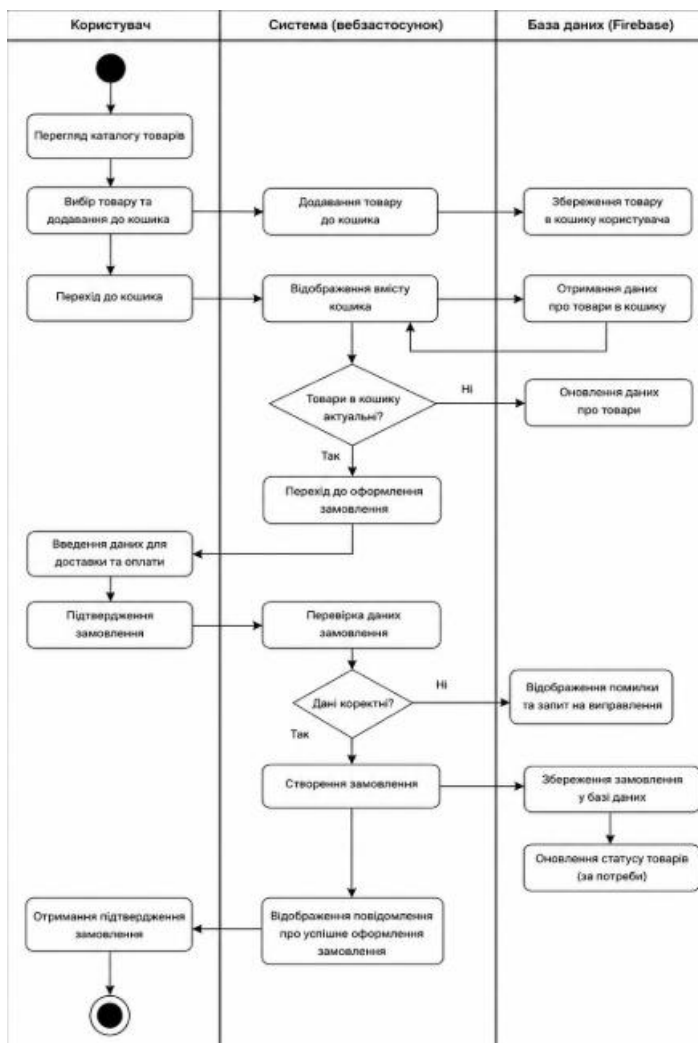


Рисунок 2.3 – Діаграма діяльності процесу оформлення замовлення

Побудована діаграма дозволяє краще зрозуміти основний сценарій взаємодії користувача із системою під час здійснення покупки. Вона також є основою для подальшої реалізації відповідного функціоналу у веб-застосунку та перевірки правильності його роботи під час тестування.

2.8 Висновки до другого розділу

У другому розділі було розглянуто основні етапи проєктування веб-застосунку інтернет-магазинута обґрунтовано вибір технологій для його реалізації. Для розробки програмного продукту було обрано мову програмування JavaScript, бібліотеку React, платформу Firebase та середовище розробки Visual Studio Code, які забезпечують створення сучасного, функціонального та масштабованого веб-

застосунку.

У ході проєктування було визначено архітектуру системи та принципи взаємодії її основних компонентів. Також розроблено структуру бази даних, яка забезпечує зберігання інформації про користувачів, товари, кошик та замовлення. Побудована схема бази даних дозволяє ефективно організувати зберігання та обробку даних у системі.

Крім того, було виконано моделювання процесу оформлення замовлення за допомогою діаграми діяльності, що дозволило наочно відобразити послідовність дій користувача та взаємодію із системою під час здійснення покупки.

Отримані результати проєктування створюють основу для подальшої реалізації веб-застосунку та його тестування, що буде розглянуто у наступному розділі дипломної роботи.

З РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ

Після завершення етапу проєктування було виконано реалізацію веб-застосунку інтернет-магазину одягу засобами JavaScript, React та Firebase. На даному етапі створено основні програмні модулі системи, реалізовано взаємодію з базою даних та забезпечено виконання основних функцій, необхідних для роботи інтернет-магазину.

3.1 Реалізація структури веб-застосунку

У процесі розробки було створено вебзастосунок інтернет-магазину одягу із використанням бібліотеки React та мови програмування TypeScript. Для побудови архітектури застосунку використано компонентний підхід, що дозволяє розділити програму на незалежні модулі та забезпечити повторне використання елементів інтерфейсу [18].

Структура проєкту веб-застосунку представлена на рисунку 3.1, організована за принципом розподілу функціональних частин на окремі каталоги. Основна логіка застосунку знаходиться у папці `src`, де розміщені сторінки, компоненти, контексти, файли конфігурації та допоміжні модулі. Для навігації між сторінками використовується маршрутизація, що забезпечує швидкий перехід між розділами без перезавантаження сторінки.

Особливістю структури є використання окремих контекстів для керування авторизацією користувача та кошиком покупок, що дозволяє централізовано керувати даними в різних частинах застосунку.

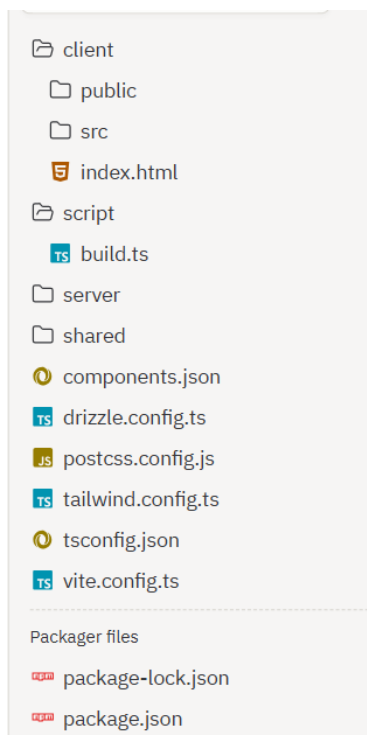


Рисунок 3.1 – Структура проєкту веб-застосунку

Детальна структура клієнтської частини наведена на рисунку 3.2. У папці `src` розміщено сторінки веб-застосунку, React-компоненти, контексти керування станом та модулі взаємодії з Firebase. Така організація забезпечує логічний поділ програмного коду та спрощує подальший розвиток системи.

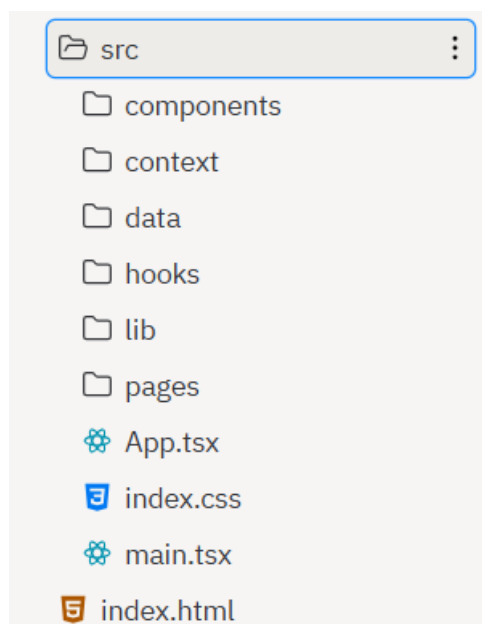


Рисунок 3.2 – Структура клієнтської частини веб-застосунку.

3.2 Реалізація системи авторизації та реєстрації

Для забезпечення доступу користувачів до функціональних можливостей веб-застосунку було реалізовано систему реєстрації та авторизації. Реалізація даного механізму виконана за допомогою сервісу Firebase Authentication, який забезпечує безпечне створення та керування обліковими записами користувачів.

Під час реєстрації користувач вводить ім'я, адресу електронної пошти та пароль. Після надсилання форми система створює новий обліковий запис у Firebase Authentication та автоматично генерує унікальний ідентифікатор користувача. Окрім створення облікового запису, додаткові дані користувача зберігаються у хмарній базі даних Cloud Firestore для подальшого використання у веб-застосунку.

Форма реєстрації користувача представлена на рисунку 3.3. Вона забезпечує введення основних персональних даних та створення нового облікового запису в системі [17,18].

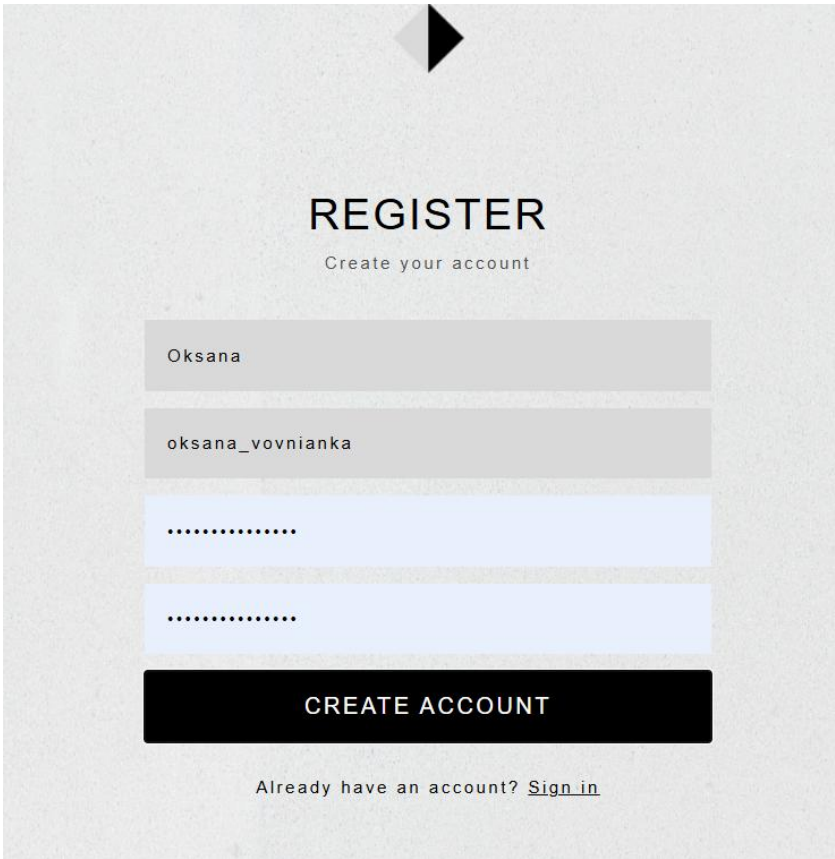
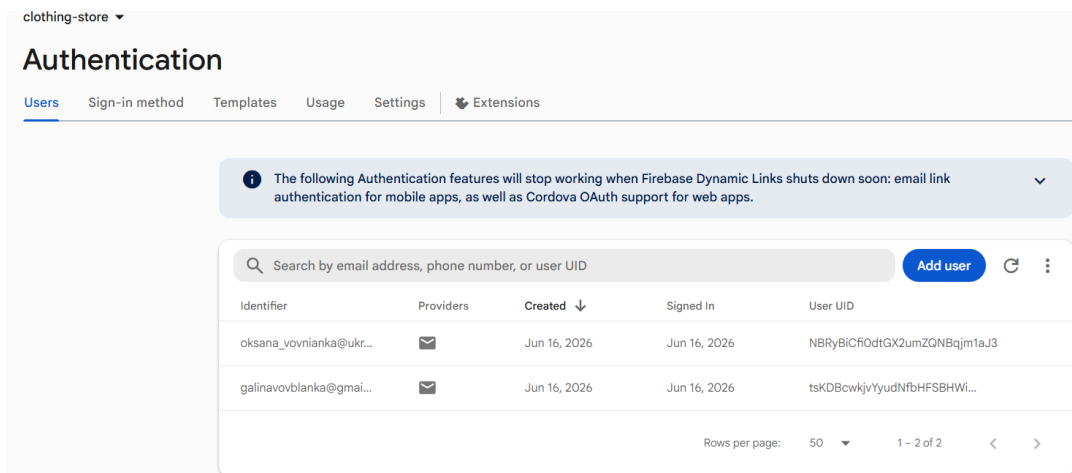
The image shows a registration form titled "REGISTER" with the subtitle "Create your account". It features four input fields: a text field for the name "Oksana", a text field for the email "oksana_vovnianka", and two password fields represented by dots. Below the fields is a large black button labeled "CREATE ACCOUNT". At the bottom, there is a link that says "Already have an account? Sign in".

Рисунок 3.3 – Форма реєстрації користувача

Після успішної реєстрації дані користувача автоматично додаються до сервісу Firebase Authentication. Приклад створеного облікового запису наведено на рисунку 3.4.



The screenshot shows the 'Authentication' page in the Firebase console for a project named 'clothing-store'. It features a navigation bar with 'Users', 'Sign-in method', 'Templates', 'Usage', 'Settings', and 'Extensions'. A warning banner at the top states: 'The following Authentication features will stop working when Firebase Dynamic Links shuts down soon: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.' Below this is a search bar and an 'Add user' button. A table lists two users:

Identifier	Providers	Created ↓	Signed In	User UID
oksana_vovnianka@ukr...	✉	Jun 16, 2026	Jun 16, 2026	NBRyBiCf0dtGX2umZQN8qjm1aJ3
galinavovblanka@gmai...	✉	Jun 16, 2026	Jun 16, 2026	tsKDBcwkvjvYyudNfbHFSBHWf...

At the bottom right, it shows 'Rows per page: 50' and '1 – 2 of 2'.

Рисунок 3.4 – Дані користувача у Firebase Authentication

Для авторизації користувачів реалізовано окрему форму входу до системи. Під час входу здійснюється перевірка введених облікових даних та встановлення активної сесії користувача. Після успішної авторизації користувач отримує доступ до функцій веб-застосунку, пов'язаних із оформленням замовлень та керуванням кошиком покупок.

Форма авторизації представлена на рисунку 3.5.

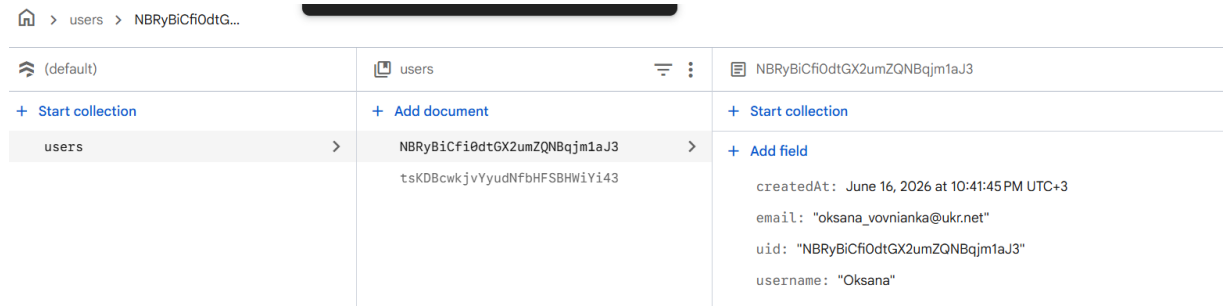


The screenshot shows a 'SIGN IN' form with the following elements:

- A title 'SIGN IN' and a subtitle 'Welcome back'.
- An input field containing the email 'oksana_vovnianka@ukr.net'.
- A password input field represented by a blue rectangle with dots.
- A black button labeled 'SIGN IN'.
- A link at the bottom: 'Don't have an account? [Register](#)'.

Рисунок 3.5 – Форма авторизації користувача

Для збереження додаткової інформації про користувачів використовується Cloud Firestore. Після створення облікового запису до колекції users записуються унікальний ідентифікатор користувача, ім'я, адреса електронної пошти та дата реєстрації. Приклад запису користувача у Firestore наведено на рисунку 3.6.



users NBRyBiCfi0dtGX2umZQNBqjm1aJ3	users NBRyBiCfi0dtGX2umZQNBqjm1aJ3	NBRyBiCfi0dtGX2umZQNBqjm1aJ3 + Start collection + Add field
users	NBRyBiCfi0dtGX2umZQNBqjm1aJ3 tsKDBcwkjvYyudNFbHFSBHWiY143	createdAt: June 16, 2026 at 10:41:45 PM UTC+3 email: "oksana_vovnianka@ukr.net" uid: "NBRyBiCfi0dtGX2umZQNBqjm1aJ3" username: "Oksana"

Рисунок 3.6 – Дані користувача у Cloud Firestore

Таким чином, у веб-застосунку реалізовано повноцінну систему реєстрації та авторизації користувачів на базі Firebase Authentication. Використання Cloud Firestore дозволяє зберігати додаткову інформацію про користувачів та забезпечує її подальше використання під час роботи із системою.

3.3 Реалізація каталогу товарів

Каталог товарів є основним функціональним модулем веб-застосунку та призначений для ознайомлення користувачів із доступним асортиментом продукції. У каталозі відображаються товари із зазначенням їх назви, категорії, ціни та графічного зображення.

Для реалізації каталогу використано компонентний підхід бібліотеки React. Кожен товар представлений у вигляді окремої картки, що містить основну інформацію про продукцію. Такий спосіб відображення дозволяє забезпечити зручну навігацію та швидкий доступ до сторінки детального перегляду товару.

Загальний вигляд каталогу товарів наведено на рисунку 3.7. Як видно з рисунка 3.7, користувач має можливість переглядати доступні товари та здійснювати перехід до детальної інформації про обрану позицію.

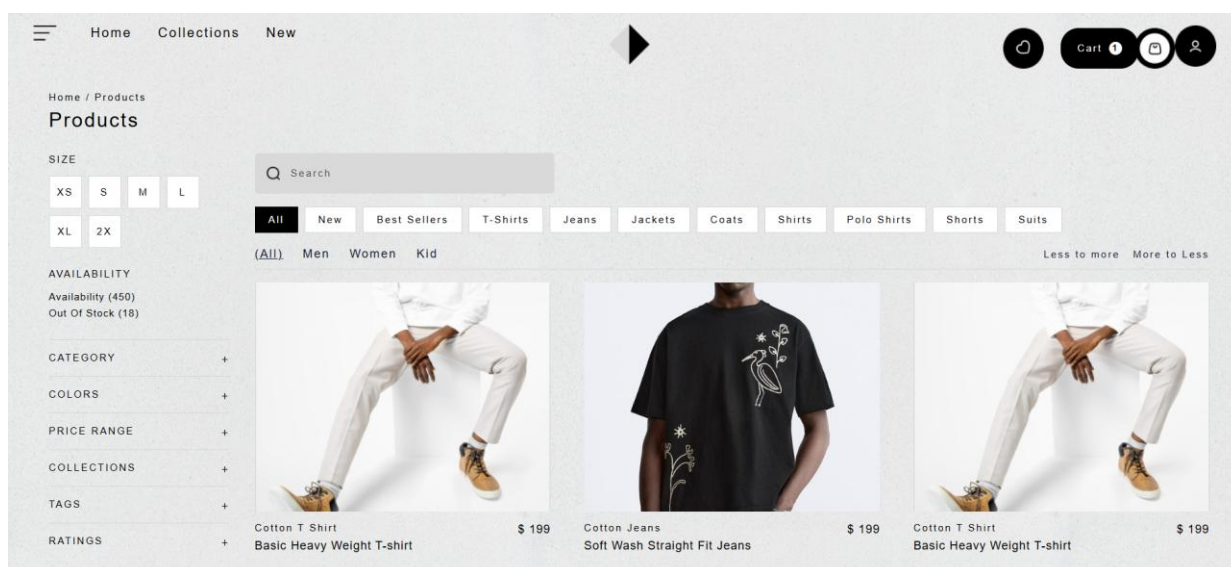


Рисунок 3.7 – Сторінка каталогу товарів

Для підвищення зручності користування у системі реалізовано механізм пошуку товарів та їх фільтрації за категоріями. Користувач може швидко знайти необхідний товар без перегляду повного каталогу продукції.

Пошук виконується за назвою товару, а фільтрація дозволяє відображати лише товари певної категорії. Це значно спрощує взаємодію користувача із системою та скорочує час пошуку необхідної продукції.

Приклад відображення товарів у вигляді карток наведено на рисунку 3.8.

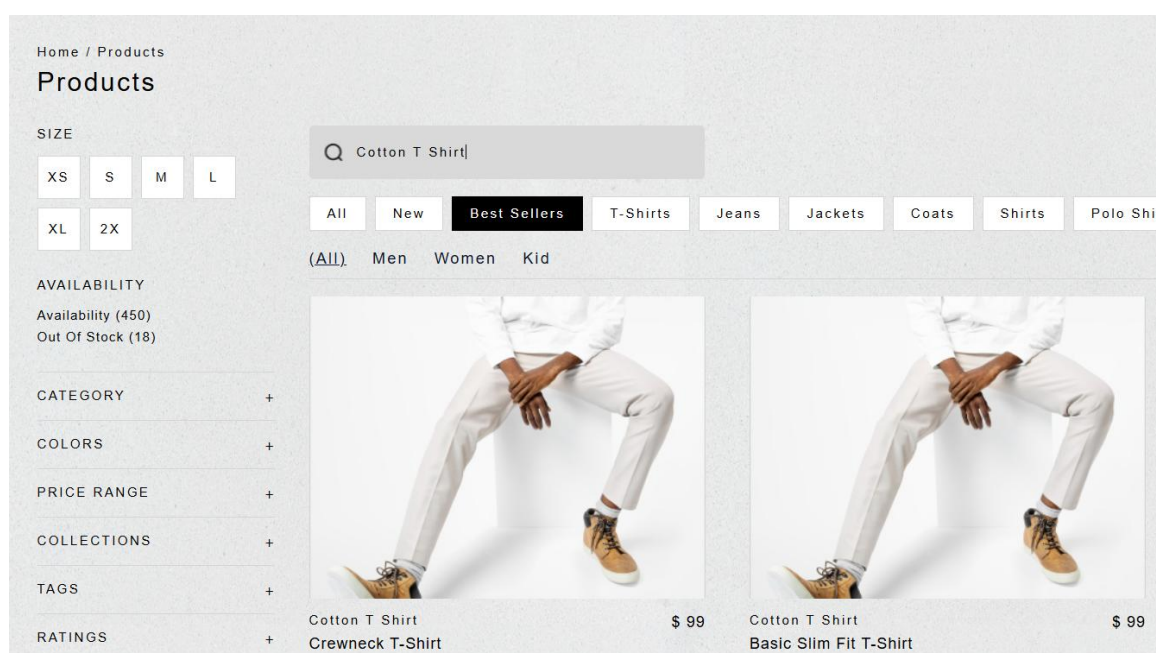


Рисунок 3.8 – Картки товарів у каталозі

Після вибору товару користувач може перейти до сторінки його детального перегляду для ознайомлення з додатковими характеристиками та оформлення покупки.

3.4 Реалізація сторінки товару та кошика

Для забезпечення можливості детального ознайомлення з продукцією у веб-застосунку реалізовано окрему сторінку перегляду товару. На даній сторінці відображається розширена інформація про продукцію, включаючи назву, опис, ціну, фотографії та доступні характеристики [19,20].

Сторінка товару формується автоматично після вибору відповідної позиції у каталозі. Це дозволяє користувачу отримати повну інформацію про товар перед здійсненням покупки.

Загальний вигляд сторінки товару представлено на рисунку 3.9.

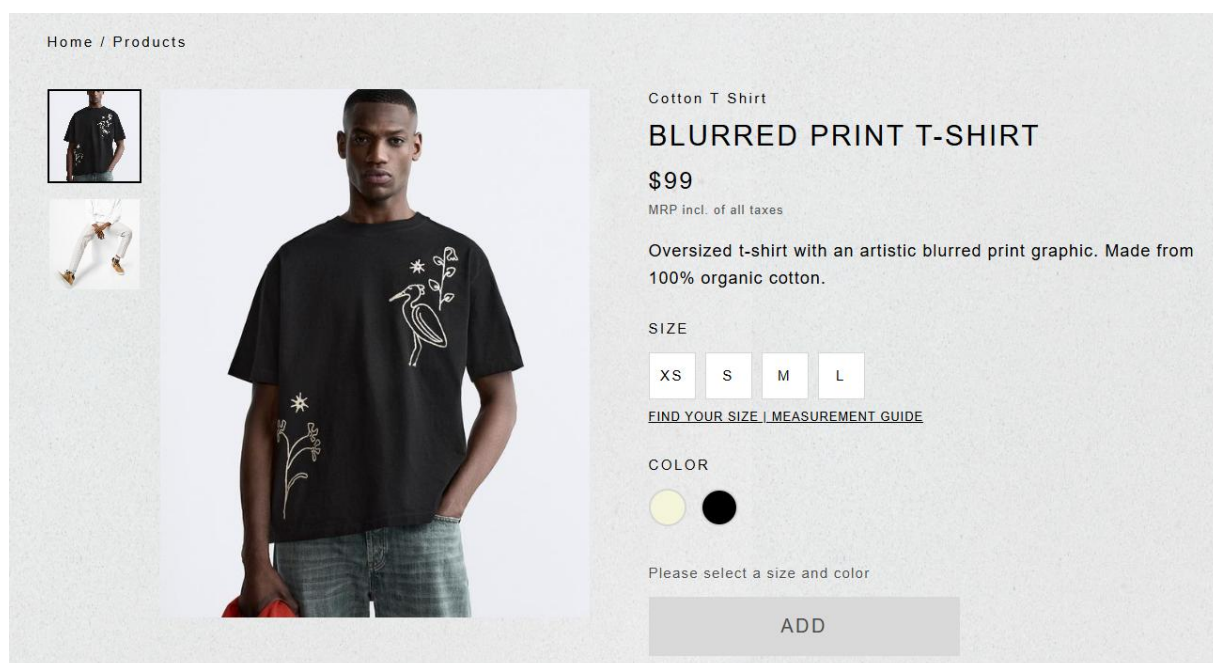


Рисунок 3.9 – Сторінка детального перегляду товару

Для здійснення покупки користувач може додати обраний товар до кошика. Після натискання кнопки додавання інформація про товар автоматично

зберігається у контексті кошика та стає доступною для подальшого оформлення замовлення.

Кошик покупок призначений для тимчасового зберігання обраних товарів та керування ними перед оформленням замовлення. У кошику користувач може переглядати перелік товарів, змінювати їх кількість та видаляти непотрібні позиції.

Інтерфейс кошика покупок наведено на рисунку 3.10.

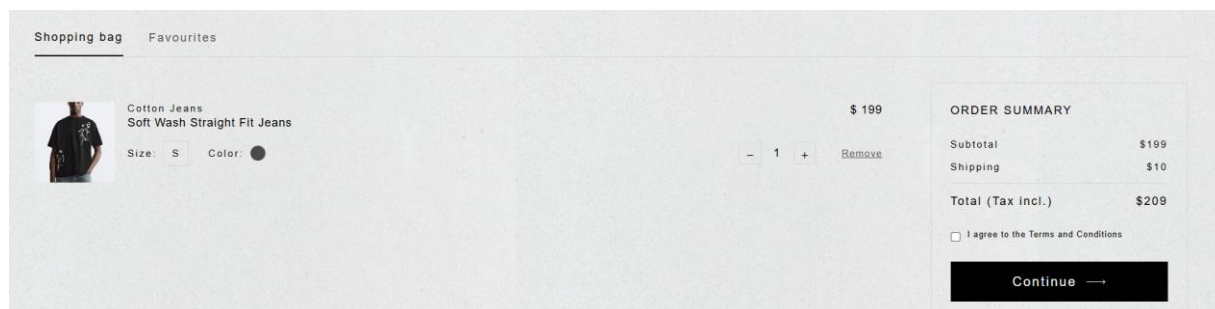


Рисунок 3.10 – Сторінка кошика покупок

У процесі роботи кошика автоматично виконується розрахунок вартості замовлення залежно від кількості обраних товарів. Це дозволяє користувачу контролювати загальну суму покупки до моменту переходу до оформлення замовлення.

3.5 Реалізація оформлення замовлення

Однією з основних функцій веб-застосунку є можливість оформлення замовлення. Для цього було реалізовано окрему сторінку Checkout, яка забезпечує введення контактних даних покупця, адреси доставки та перегляд інформації про обрані товари.

Після переходу до сторінки оформлення замовлення користувачу відображається форма для введення необхідних персональних даних. До переліку обов'язкових полів належать ім'я, прізвище, номер телефону, адреса доставки, місто, поштовий індекс, країна та регіон проживання. Крім цього, система автоматично підставляє адресу електронної пошти авторизованого користувача.

Загальний вигляд форми оформлення замовлення наведено на рисунку 3.11.

Home / Products

CHECKOUT
INFORMATION
Contact info

oksana_vovnianka@ukr.net +38097820226

Shipping address

Oksana Vovnianka

Ternopil Ternopil

46000 Ukraine

Ternopil

YOUR ORDER

 1 Cotton Jeans
Soft Wash Straight Fit
Jeans
S / #4A4A4A \$199

Subtotal \$199

Shipping Calculated at next step

Total \$209

SHIPPING

Standard Shipping \$10.00

Рисунок 3.11 – Форма оформлення замовлення

Під час оформлення замовлення користувач також має можливість переглядати інформацію про додані до кошика товари. Для кожної позиції відображається назва товару, його кількість, обрані характеристики та вартість. Додатково система автоматично виконує розрахунок проміжної суми замовлення, вартості доставки та загальної суми до сплати.

Приклад відображення інформації про замовлення наведено на рисунку 3.12.

YOUR ORDER

 1 Cotton Jeans
Soft Wash Straight Fit
Jeans
S / #4A4A4A \$199

Subtotal \$199

Shipping Calculated at next step

Total \$209

Рисунок 3.12 – Інформація про товари під час оформлення замовлення

Після натискання кнопки підтвердження замовлення система формує об'єкт замовлення, який містить інформацію про користувача, перелік товарів, адресу доставки, вартість доставки та загальну суму покупки. Отримані дані передаються до бази даних Cloud Firestore для подальшого зберігання та обробки.

Для ідентифікації покупця використовується унікальний ідентифікатор користувача, отриманий під час авторизації через Firebase Authentication. Це дозволяє пов'язувати створені замовлення із конкретними обліковими записами користувачів.

Після успішного створення замовлення система очищає кошик покупок та відображає повідомлення про успішне завершення операції. Приклад повідомлення наведено на рисунку 3.13.

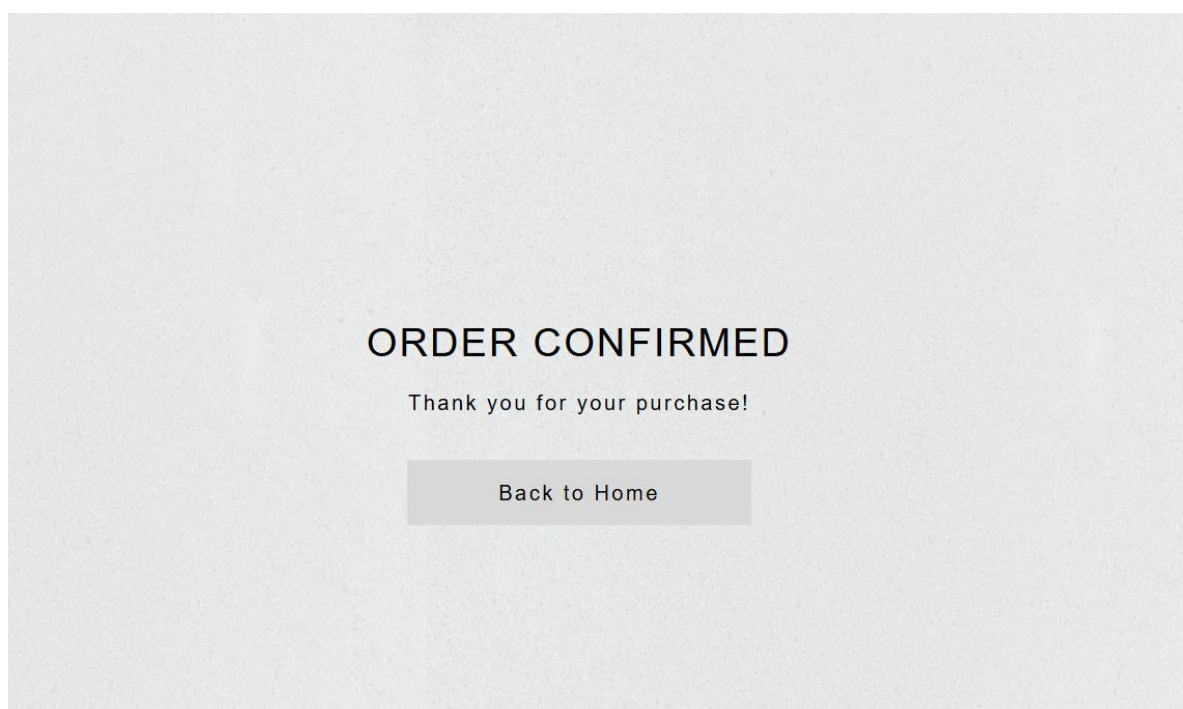


Рисунок 3.13 – Повідомлення про успішне оформлення замовлення

У результаті реалізації даного модуля було забезпечено повний цикл оформлення покупки – від введення контактних даних користувача до збереження інформації про замовлення у хмарній базі даних.

3.6 Тестування веб-застосунку

Для перевірки працездатності розробленого веб-застосунку було проведено функціональне тестування основних модулів системи. Під час тестування перевірялися процеси реєстрації та авторизації користувачів, робота каталогу товарів, функціонування кошика покупок та оформлення замовлень.

Метою тестування було підтвердження коректної роботи програмного продукту та відповідності реалізованих функцій поставленим вимогам. Результати тестування наведено у таблицях 3.1–3.5.

3.6.1 Тестування реєстрації користувачів

Для перевірки роботи модуля реєстрації було виконано тестування створення нового облікового запису користувача. Перевірялася можливість введення реєстраційних даних та створення нового користувача у Firebase Authentication.

Таблиця 3.1 – Результати тестування реєстрації користувачів

№	Тестова дія	Очікуваний результат	Результат
1	Введення коректних реєстраційних даних	Створення нового облікового запису	Успішно
2	Введення існуючої адреси електронної пошти	Відображення повідомлення про помилку	Успішно
3	Перевірка запису даних у Firestore	Створення документа користувача	Успішно

Проведене тестування підтвердило коректну роботу механізму реєстрації користувачів.

3.6.2 Тестування авторизації

Для перевірки роботи модуля авторизації було протестовано вхід користувача до системи з використанням коректних та некоректних облікових даних.

Таблиця 3.2 – Результати тестування авторизації

№	Тестова дія	Очікуваний результат	Результат
1	Вхід з коректними даними	Успішна авторизація користувача	Успішно
2	Вхід з неправильним паролем	Повідомлення про помилку	Успішно
3	Перевірка збереження сесії користувача	Користувач залишається авторизованим	Успішно

Під час тестування помилок у роботі механізму авторизації виявлено не було.

3.6.3 Тестування роботи каталогу товарів

Для перевірки каталогу товарів було протестовано відображення продукції та можливість переходу до сторінки окремого товару.

Таблиця 3.3 – Результати тестування каталогу товарів

№	Тестова дія	Очікуваний результат	Результат
1	Відкриття каталогу товарів	Відображення списку товарів	Успішно
2	Вибір товару	Перехід на сторінку товару	Успішно
3	Відображення інформації про товар	Коректне відображення даних	Успішно

3.6.4 Тестування кошика покупок

Для перевірки функціонування кошика було виконано тестування додавання, редагування та видалення товарів.

Таблиця 3.4 – Результати тестування кошика покупок

№	Тестова дія	Очікуваний результат	Результат
1	Додавання товару до кошика	Товар додається до кошика	Успішно
2	Зміна кількості товару	Перерахунок загальної суми	Успішно
3	Видалення товару	Товар видаляється з кошика	Успішно

Проведене тестування підтвердило коректну роботу механізму керування кошиком покупок.

3.6.5 Тестування оформлення замовлення

Для перевірки процесу оформлення замовлення було протестовано введення даних користувача, формування замовлення та підтвердження покупки.

Таблиця 3.5 – Результати тестування оформлення замовлення

№	Тестова дія	Очікуваний результат	Результат
1	зас	Коректне прийняття даних	Успішно
2	Підтвердження замовлення	Створення нового замовлення	Успішно
3	Відображення повідомлення про успішне оформлення	Виведення повідомлення ORDER CONFIRMED	Успішно
4	Очищення кошика після оформлення	Видалення товарів із кошика	Успішно

3.7 Висновки до третього розділу

У третьому розділі було здійснено проектування, розробку та тестування веб-застосунку інтернет-магазину одягу. Під час реалізації програмного продукту використано сучасні вебтехнології React, TypeScript та Firebase, що забезпечили створення функціонального та зручного у використанні програмного рішення.

У процесі розробки було створено структуру веб-застосунку, реалізовано систему реєстрації та авторизації користувачів на основі Firebase Authentication, а також налаштовано зберігання даних у хмарній базі даних Cloud Firestore. Розроблено каталог товарів, сторінку детального перегляду продукції, кошик покупок та механізм оформлення замовлень.

Особливу увагу приділено реалізації взаємодії користувача із системою. Користувачі мають можливість переглядати асортимент товарів, здійснювати пошук необхідної продукції, додавати товари до кошика, редагувати його вміст та оформляти замовлення через спеціалізовану форму.

Проведене тестування підтвердило коректну роботу всіх основних функціональних модулів системи та відповідність програмного продукту поставленим вимогам.

4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

У даному розділі розглядаються питання безпеки життєдіяльності та охорони праці під час професійної діяльності розробника програмного забезпечення. Особливу увагу приділено ергономічним проблемам безпеки життєдіяльності, які виникають під час тривалої роботи за персональним комп'ютером, а також вимогам до організації робочого місця оператора ПК.

4.1 Ергономічні проблеми безпеки життєдіяльності

Ергономіка є важливою складовою безпеки життєдіяльності людини та спрямована на створення комфортних і безпечних умов праці. Особливо актуальними ергономічні питання є для працівників ІТ-сфери та розробників програмного забезпечення, діяльність яких пов'язана з тривалим використанням персонального комп'ютера. Під час розробки веб-застосунків працівник значну частину робочого часу проводить за монітором, виконуючи інтелектуальну роботу, що потребує високої концентрації уваги [20] .

Недотримання ергономічних вимог може призводити до швидкої втомлюваності, погіршення зору, виникнення захворювань опорно-рухового апарату, головного болю та зниження працездатності. Найпоширенішими проблемами є неправильне розташування монітора, недостатнє або надмірне освітлення, незручне розміщення клавіатури та маніпулятора типу «миша», а також невідповідність висоти робочого столу та крісла антропометричним характеристикам працівника.

Одним із важливих чинників є правильна організація робочого простору. Монітор повинен розташовуватися на відстані 50–70 см від очей користувача, а його верхня межа має бути на рівні очей або трохи нижче. Це дозволяє зменшити навантаження на шийний відділ хребта та органи зору. Клавіатура повинна бути розташована таким чином, щоб руки працівника перебували у природному положенні без надмірного напруження.

Значну роль у забезпеченні ергономічності відіграє освітлення робочого місця. Недостатнє освітлення призводить до перенапруження зорового аналізатора, тоді як надмірна яскравість викликає дискомфорт та швидке стомлення. Оптимальним є поєднання природного та штучного освітлення із рівномірним розподілом світлового потоку по робочій зоні.

Важливим елементом ергономіки є дотримання режиму праці та відпочинку. Під час тривалої роботи за комп'ютером рекомендується регулярно робити короткі перерви для зниження навантаження на очі та м'язову систему. Виконання простих фізичних вправ сприяє покращенню кровообігу та зменшенню ризику виникнення професійних захворювань.

Таким чином, врахування ергономічних вимог під час організації праці програміста сприяє збереженню здоров'я працівника, підвищенню його продуктивності та створенню безпечних умов виконання професійної діяльності.

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Організація робочого місця оператора персонального комп'ютера повинна відповідати ергономічним вимогам, що забезпечують комфортні та безпечні умови праці. Для фахівців у галузі програмної інженерії правильне облаштування робочого місця має особливе значення, оскільки більша частина робочого часу пов'язана з використанням комп'ютерної техніки [21].

Робоче місце повинно бути обладнане столом і кріслом, які дозволяють підтримувати правильне положення тіла під час роботи. Висота столу має забезпечувати зручне розміщення клавіатури та монітора, а конструкція крісла повинна передбачати регулювання висоти сидіння та кута нахилу спинки. Це дозволяє зменшити навантаження на хребет і запобігти виникненню професійних захворювань.

Особливу увагу необхідно приділяти розташуванню монітора. Рекомендована відстань від очей користувача до екрана становить 50–70 см. Екран повинен розміщуватися таким чином, щоб погляд працівника був спрямований

трохи вниз, що сприяє зниженню навантаження на м'язи шиї та очей. Для запобігання появі відблисків монітор необхідно встановлювати перпендикулярно до джерел природного освітлення.

Клавіатура та комп'ютерна миша повинні розташовуватися в межах зони досяжності рук без необхідності виконання зайвих рухів. Під час роботи передпліччя мають бути розташовані паралельно до поверхні підлоги, а кисті рук не повинні перебувати у напруженому стані.

Одним із важливих факторів є забезпечення належного мікроклімату приміщення. Температура повітря повинна перебувати в межах комфортних значень, а рівень вологості та швидкість руху повітря мають відповідати санітарним нормам. Також необхідно забезпечити ефективну вентиляцію приміщення для підтримання сприятливих умов праці.

Для зменшення негативного впливу роботи за комп'ютером рекомендується дотримуватися встановлених режимів праці та відпочинку. Періодичні перерви дозволяють знизити втому, відновити працездатність та запобігти перенавантаженню організму [20,21].

Отже, дотримання ергономічних вимог до організації робочого місця оператора ПК є важливою умовою забезпечення безпечної та ефективної праці. Правильно організоване робоче місце сприяє підвищенню продуктивності працівника, зменшенню рівня втоми та збереженню його здоров'я.

4.3. Висновки до четвертого розділу

У четвертому розділі було розглянуто питання безпеки життєдіяльності та охорони праці під час роботи з персональним комп'ютером. Проаналізовано основні ергономічні проблеми, які можуть виникати під час тривалої роботи за комп'ютером, зокрема навантаження на органи зору, опорно-руховий апарат та нервову систему працівника.

Також було досліджено вимоги до організації робочого місця оператора ПК, що включають правильне розташування обладнання, забезпечення належного

освітлення, створення сприятливого мікроклімату та дотримання режимів праці й відпочинку. Встановлено, що дотримання ергономічних вимог сприяє зниженню рівня втоми, підвищенню продуктивності праці та збереженню здоров'я працівників.

ВИСНОВКИ

У бакалаврській кваліфікаційній роботі було розглянуто теоретичні та практичні аспекти розробки веб-застосунку інтернет-магазину засобами мови програмування JavaScript, бібліотеки React та платформи Firebase. Актуальність теми зумовлена стрімким розвитком електронної комерції та зростанням попиту на сучасні програмні рішення, які забезпечують зручний доступ до товарів і послуг через мережу Інтернет. Використання веб-технологій дозволяє автоматизувати процеси продажу товарів, покращити взаємодію між продавцем і покупцем та підвищити ефективність ведення бізнесу.

У процесі виконання роботи було проведено аналіз предметної області та досліджено особливості функціонування сучасних інтернет-магазинів. Також було розглянуто існуючі аналоги програмних продуктів, визначено їх переваги та недоліки, що дозволило сформулювати функціональні та нефункціональні вимоги до майбутньої системи. На основі проведеного аналізу було обґрунтовано вибір сучасних технологій розробки, зокрема мови програмування JavaScript, бібліотеки React, мови TypeScript та платформи Firebase.

У роботі виконано проектування веб-застосунку, розроблено його архітектуру та структуру програмних модулів. Для моделювання системи використано UML-діаграми, які відображають взаємодію користувачів із програмним продуктом та основні бізнес-процеси. Також було спроектовано структуру бази даних для зберігання інформації про користувачів, товари та замовлення.

Практична частина роботи присвячена реалізації веб-застосунку інтернет-магазину. У ході розробки створено сучасний користувацький інтерфейс із використанням бібліотеки React та мови TypeScript. Реалізовано систему реєстрації та авторизації користувачів за допомогою Firebase Authentication, що забезпечує безпечний доступ до функціональних можливостей системи. Для зберігання та обробки даних використано хмарну базу даних Cloud Firestore, яка дозволяє ефективно працювати з інформацією в режимі реального часу.

У розробленому веб-застосунку реалізовано основні функції інтернет-магазину, зокрема перегляд каталогу товарів, пошук та вибір продукції, перегляд детальної інформації про товар, додавання товарів до кошика покупок, редагування вмісту кошика та оформлення замовлень. Передбачено можливість збереження даних користувачів та замовлень у базі даних для подальшої обробки.

Для перевірки працездатності програмного продукту проведено комплексне функціональне тестування. У ході тестування перевірено роботу механізмів реєстрації та авторизації користувачів, функціонування каталогу товарів, кошика покупок та процесу оформлення замовлень. Результати тестування підтвердили коректність роботи всіх основних модулів системи та відповідність програмного продукту поставленим вимогам.

Окрему увагу в роботі приділено питанням безпеки життєдіяльності та охорони праці. Було розглянуто ергономічні проблеми, що виникають під час тривалої роботи за персональним комп'ютером, а також вимоги до організації робочого місця оператора ПК. Проведений аналіз показав важливість дотримання ергономічних вимог для збереження здоров'я працівників, підвищення їх працездатності та забезпечення комфортних умов праці.

Отже, у результаті виконання бакалаврської кваліфікаційної роботи було досягнуто поставленої мети та виконано всі визначені завдання. Розроблено повноцінний веб-застосунок інтернет-магазину, який забезпечує ефективну взаємодію користувачів із системою та реалізує основні процеси електронної комерції. Отримані результати можуть бути використані для подальшого вдосконалення програмного продукту, розширення його функціональних можливостей та впровадження нових сервісів для підвищення зручності роботи користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методичні вказівки до виконання кваліфікаційної роботи бакалавра для здобувачів спеціальності 121 – Інженерія програмного забезпечення, всіх форм навчання / укладачі Михалик Д.М., Цуприк Г.Б., Бревус В.М. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2024. 45 с.
2. Фріман Е., Робсон Е. Вивчаємо JavaScript. – Харків : Видавництво «Ранок», 2022. – 640 с.
3. Banks A., Porcello E. Learning React. – Sebastopol : O'Reilly Media, 2023. – 350 р.
4. Kleppmann M. Designing Data-Intensive Applications. – Sebastopol : O'Reilly Media, 2017. – 616 р.
5. Duckett J. HTML and CSS: Design and Build Websites. – Indianapolis : Wiley, 2021. – 490 р.
6. Kasta – офіційний сайт [Електронний ресурс]. – Режим доступу: <https://kasta.ua/> (дата звернення: 18.06.2026).
7. Answer – офіційний сайт [Електронний ресурс]. – Режим доступу: <https://answer.ua/> (дата звернення: 18.06.2026).
8. Reserved – офіційний сайт [Електронний ресурс]. – Режим доступу: <https://www.reserved.com/ua/uk> (дата звернення: 18.06.2026).
9. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. – Indianapolis : Wiley, 2021. – 640 р.
10. Фленаган Д. JavaScript. Детальний посібник. – Київ : Видавництво Старого Лева, 2023. – 784 с.
11. Петрик М., Мудрик І., Петрик О., Ю. Стоянов. Сучасні технології ООП-проектування та автоматичного генерування програмного коду: навчальний посібник, Тернопіль: ТНТУ імені Івана Пулюя, 2018. 48 с.

12. Види функціонального та нефункціонального тестування. URL: <https://dan-it.com.ua/uk/blog/vidy-funkcionalnogo-i-nefunkcionalnogo-testirovaniya/> (дата звернення: 17.05.2026).
13. React Documentation [Електронний ресурс]. – Режим доступу: <https://react.dev> (дата звернення: 17.05.2026).
14. TypeScript Documentation [Електронний ресурс]. – Режим доступу: <https://www.typescriptlang.org/docs> (дата звернення: 17.05.2026).
15. Firebase Documentation [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs> (дата звернення: 17.05.2026).
16. Cloud Firestore Documentation [Електронний ресурс]. – Режим доступу: <https://firebase.google.com/docs/firestore> (дата звернення: 17.05.2026).
17. Vite Documentation [Електронний ресурс]. – Режим доступу: <https://vitejs.dev/guide> (дата звернення: 17.05.2026).
18. Mozilla Developer Network (MDN). JavaScript Guide [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 17.05.2026).
19. Mozilla Developer Network (MDN). HTML Reference [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 17.05.2026).
20. Mozilla Developer Network (MDN). CSS Reference [Електронний ресурс]. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 17.05.2026).
21. Основи охорони праці. / Під ред. Ткачука К.Н., Халімовського Н.О. – К.: Основа, 2006. 448 с.
22. Зеркалов Д.В. Безпека життєдіяльності та основи охорони праці. Навчальний посібник. К.: «Основа». 2016. 267 с.
23. Про охорону праці : Закон України від 14.10.1992 р. № 2694-XII. — Відомості Верховної Ради України, 1992. — № 49. — С. 668.
24. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями : НПАОП 0.00-7.15-18 : затв. наказом

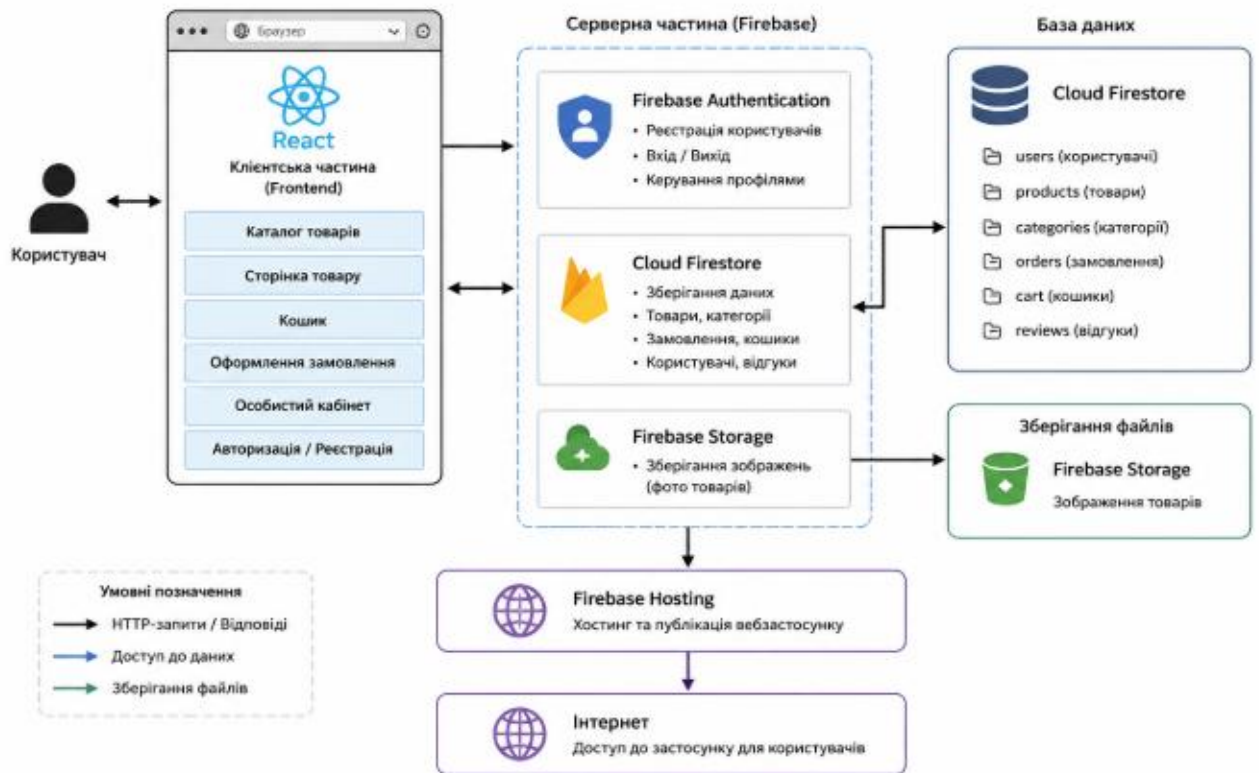
Мінсоцполітики України від 14.02.2018 р. № 207. — Офіційний вісник України, 2018. — № 40. — С. 76.

ДОДАТКИ

ДОДАТОК А

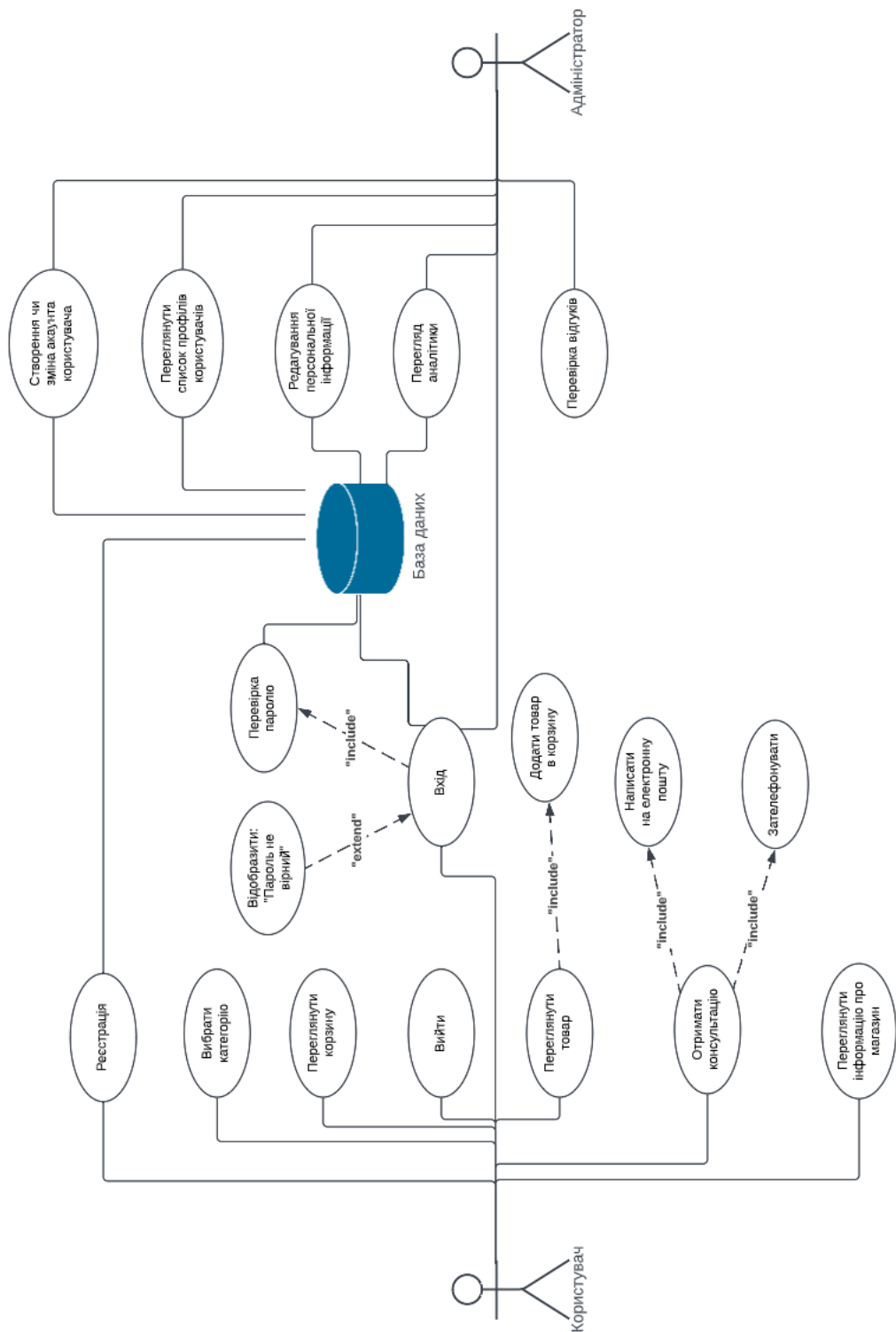
Загальна архітектура системи

Архітектура вебзастосунку інтернет-магазину одягу



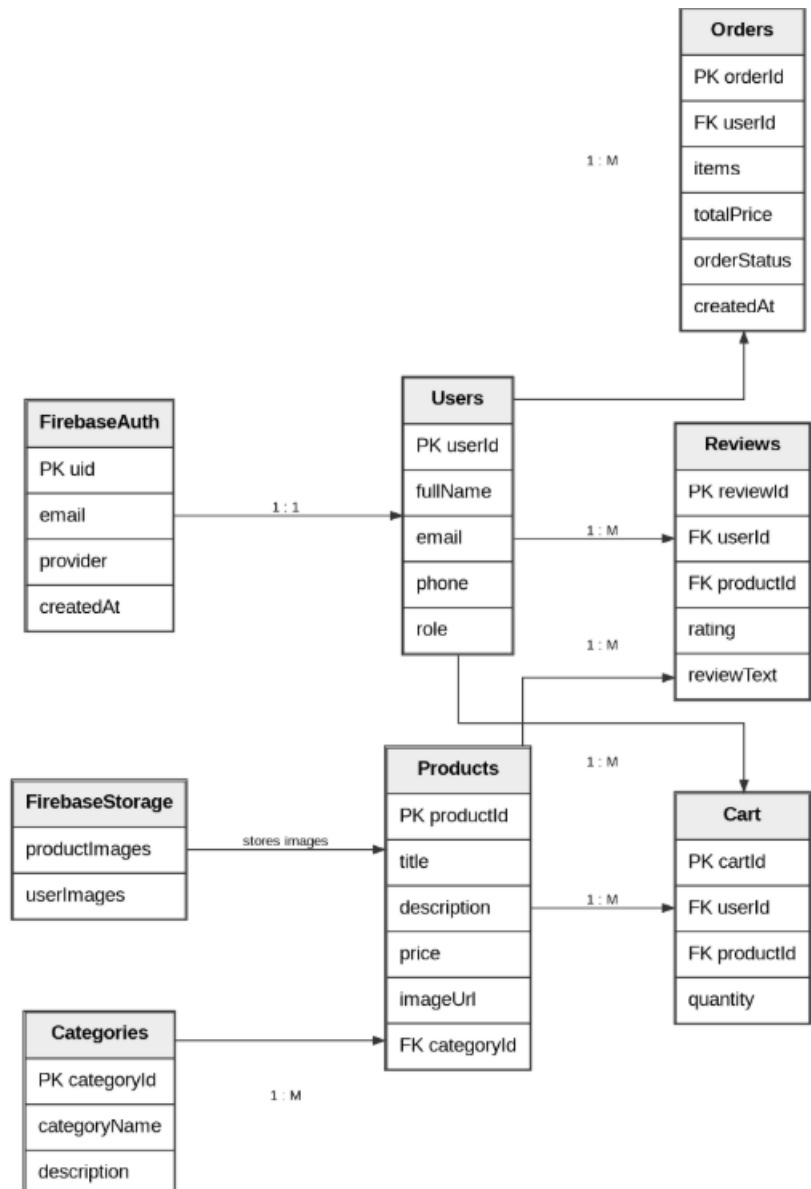
ДОДАТОК Б

Діаграма варіантів використання



ДОДАТОК В

Схема структури бази даних інтернет-магазину



ДОДАТОК Г

Діаграма діяльності процесу оформлення замовлення

