МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет імені Петра Могили

Факультет комп'ютерних наук

Кафедра інженерії програмного забезпечення

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії
програмного забезпечення

_____Євген ДАВИДЕНКО_____

«___» _____ 2026 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

ВЕБЗАСТОСУНОК ПРОДАЖУ ОДЯГУ

Спеціальність 121 Інженерія програмного забезпечення
Освітня програма «Інженерія програмного забезпечення»

Здобувачка _____ **Анна ЧЕБАН**

«__» _____ 20__ р.

Керівник роботи
канд. техн. наук,
доцент _____ **Євген ДАВИДЕНКО**

«__» _____ 20__ р.

Миколаїв – 2026

Завдання на виконання кваліфікаційної роботи

Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Перший (бакалаврський)
Освітній ступінь	Бакалавр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри інженерії
програмного забезпечення
____Євген ДАВИДЕНКО_____
«___» _____ 2026 р.

ЗАВДАННЯ

на кваліфікаційну бакалаврську роботу здобувачки

Чебан Анни

1. Тема кваліфікаційної роботи: Вебзастосунок продажу одягу затверджена наказом ректора ЧНУ ім. Петра Могили № 349 від «26» грудня 2025 р.
2. Строк представлення кваліфікаційної роботи «___» _____ 2026 р.
3. Очікуваний результат роботи та початкові дані якщо такі потрібні: очікуваним результатом є вебзастосунок продажу одягу.
4. Перелік питань, що підлягають розробці:

– провести аналіз предметної області;
– зробити порівняльний аналіз аналогів;
– визначити вимоги та функціонал вебзастосунку;
– виконати моделювання та проектування програмного застосунку;

– розробити клієнтську та серверну частини системи;

– провести тестування застосунку.

5. Перелік графічних матеріалів: слайди презентації.

6. Консультанти:

Консультант	**Кафедра (організація)**	**Частина роботи**

Дата видачі завдання «26» грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: **Вебзастосунок продажу одягу.**

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КБР	26.12.2025	03.02.2026	Виконано
2.	Огляд літератури за темою роботи	04.02.2026	10.02.2026	Виконано
3.	Складання календарного плану КБР	11.02.2026	12.02.2026	Виконано
4.	Аналіз предметної області	13.02.2026	16.02.2026	Виконано
5.	Розробка проєктних рішень	17.02.2026	23.02.2026	Виконано
6.	Моделювання та конструювання ПЗ	24.02.2026	18.03.2026	Виконано
7.	Кодування, тестування та апробація розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	19.03.2026	09.04.2026	Виконано
8.	Відгук керівника КБР	10.04.2026	14.04.2026	Виконано
9.	Оформлення КБР та презентації	15.04.2026	17.04.2026	Виконано
10.	Попередній захист	25.05.2026	25.05.2026	Виконано
11.	Рецензування	04.06.2026	04.06.2026	Виконано
12.	Завершення оформлення КБР та презентації	04.06.2026	04.06.2026	Виконано
13.	Захист кваліфікаційної роботи			Виконано

Здобувачка _____ **Анна ЧЕБАН**

«__» _____ 20__ р.

Керівник роботи

канд. техн. наук,

доцент _____ **Євген ДАВИДЕНКО**

«__» _____ 20__ р.

АНОТАЦІЯ

до кваліфікаційної бакалаврської роботи

«Вебзастосунок продажу одягу»

Здобувачка 408 гр.: Чебан Анна

Керівник: кандидат технічних наук, доцент Давиденко Євген

Серед інтернет-магазинів особливе місце посідають саме магазини одягу. Потреба в одязі існує від дня народження людини й супроводжує людство з моменту його появи. З розвитком культури одяг став необхідним не лише для захисту від погодних умов, таких як холод чи спека. Попит на нього давно не обмежується лише захисною функцією – прикладом є використання спеціального професійного одягу для роботи в різних галузях виробництва та сфери обслуговування.

Сьогодні для великої частини молоді, людей середнього віку й навіть літніх людей у всьому світі одяг – це спосіб самовираження. Як казала Міучча Прада: «Ваш одяг – це спосіб подати себе світу, особливо зараз, коли спілкування стає таким швидким. Мода – це моментальна мова».

У сучасному світі, що стрімко розвивається, інтернет-сервіси з продажу одягу досягли високого рівня зручності у виборі та замовленні товарів. Серед переваг існуючих вебсервісів – зручний пошук за категоріями, система сповіщень та персоналізовані рекомендації. Проте меж для розвитку не існує.

Науково-практичне значення роботи полягає в можливості використання отриманих результатів для створення та впровадження ефективного вебзастосунку інтернет-магазину одягу, який може бути застосований у реальній комерційній діяльності, а також використаний як приклад під час розробки подібних інформаційних систем.

Об’єктом кваліфікаційної роботи є процес електронної комерції інтернет-магазину одягу.

Предметом дослідження кваліфікаційної роботи є сукупність інструментів для розробки вебзастосунку інтернет-магазину одягу.

Метою кваліфікаційної роботи є розробка вебзастосунку інтернет-магазину з використанням сучасних інформаційних технологій та засобів захисту даних.

Кваліфікаційна робота складається із вступу, 4 розділів, висновків та переліку джерел посилання.

У вступі обґрунтовано актуальність теми, визначено об’єкт, предмет і мету дослідження.

У першому розділі проведено аналіз предметної області, розглянуто особливості інтернет-магазинів та існуючі аналоги.

Другий розділ присвячено проєктуванню системи, опису архітектури вебзастосунку та розробці UML-діаграм.

У третьому розділі описано реалізацію вебзастосунку, використані технології та основні функціональні можливості системи.

У висновках підсумовано результати виконаної роботи та визначено можливі напрями подальшого розвитку системи.

Кваліфікаційна робота викладена на 76 сторінках машинописного тексту, складається із вступу, 4 розділів, загальних висновків, переліку джерел посилання з 21 найменування та 1 додатку. Праця містить 14 рисунків.

Ключові слова: вебзастосунок, інтернет-магазин, електронна комерція, продаж одягу, база даних, веброзробка, користувацький інтерфейс.

ABSTRACT

to the qualifying bachelor's thesis

«Web application for clothing sales»

Student of 408 group: Cheban Anna

Supervisor: PhD in Technical Sciences, Associate Professor Davydenko Yevhen

Among online stores, clothing stores occupy a special place. The need for clothing exists from the day a person is born and has accompanied humanity since its very beginning. With the development of culture, clothing has become necessary not only for protection from weather conditions such as cold or heat. The demand for it has long gone beyond its protective function – for example, the use of special professional clothing for work in various fields of production and service industries.

Today, for a large part of young people, middle-aged individuals, and even elderly people around the world, clothing is a way of self-expression. As Miuccia Prada once said: “What you wear is how you present yourself to the world, especially today, when human contacts are so quick. Fashion is instant language.”

In the modern world, which is rapidly developing, online services for selling clothing have reached a high level of convenience in choosing and ordering products. Among the advantages of existing web services are convenient category-based search, notification systems, and personalized recommendations. However, there are no limits to further development.

The scientific and practical significance of this work lies in the possibility of using the obtained results for the creation and implementation of an effective web application for an online clothing store. Such an application can be used in real commercial activities and also serve as an example in the development of similar information systems.

The object of the qualification work is the e-commerce process of an online clothing store.

The subject of the qualification work is the set of tools used for developing a web application for an online clothing store.

The purpose of the qualification work is to develop a web application for an online store using modern information technologies and data protection tools.

The qualification work consists of an introduction, four chapters, conclusions, and a list of references.

The introduction substantiates the relevance of the topic and defines the object, subject, and purpose of the research.

The first chapter analyzes the subject area, considers the features of online stores, and reviews existing analogues.

The second chapter is devoted to system design, describing the architecture of the web application and the development of UML diagrams.

The third chapter describes the implementation of the web application, the technologies used, and the main functional capabilities of the system.

The conclusions summarize the results of the work performed and identify possible directions for the further development of the system.

The qualification work is presented on 76 pages of typed text and consists of an introduction, four chapters, general conclusions, a list of references containing 21 sources, and 1 appendice. The work includes 14 figures.

Keywords: web application, online store, e-commerce, clothing sales, database, web development, user interface.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних

ПЗ – програмне забезпечення

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

PHP – Hypertext Preprocessor

SQL – Structured Query Language

MySQL – My Structured Query Language

UML – Unified Modeling Language

StarUML – Star Unified Modeling Language

HTTP – HyperText Transfer Protocol

CRUD – Create, Read, Update, Delete

WYSIWYG – What You See Is What You Get

SEO – Search Engine Optimization

PDF – Portable Document Format

XML – eXtensible Markup Language

JSON – JavaScript Object Notation

CMS – Content Management System

ВСТУП

Розвиток інформаційних технологій та глобальної мережі Інтернет суттєво трансформував підходи до організації торговельної діяльності, зумовивши активне впровадження електронних сервісів у сфері роздрібного продажу. Вебзастосунки електронної комерції стали невід'ємною складовою сучасного бізнес-середовища, оскільки забезпечують оперативність обслуговування, автоматизацію внутрішніх процесів та розширення каналів взаємодії з клієнтами. У цьому контексті створення програмного продукту для онлайн-продажу одягу є не лише актуальним, але й методично доцільним завданням, що дозволяє поєднати теоретичні знання з практичними навичками розробки інформаційних систем.

Проєктування вебзастосунку передбачає комплексний підхід, який охоплює аналіз вимог користувачів, формування логічної структури бази даних, реалізацію серверної та клієнтської частин, а також забезпечення цілісності, надійності та безпеки обробки даних. Важливим аспектом є організація коректної взаємодії між інтерфейсом користувача та сервером, що здійснює обробку запитів, управління обліковими записами, збереження інформації про товари, замовлення та результати транзакцій. Особлива увага приділяється побудові зручного та інтуїтивно зрозумілого інтерфейсу, який сприяє ефективній навігації, швидкому пошуку продукції та спрощенню процедури оформлення замовлення.

Розроблення вебзастосунку продажу одягу також передбачає впровадження механізмів автентифікації та авторизації користувачів, розмежування прав доступу, реалізацію функціоналу кошика та системи обробки замовлень, а також створення адміністративного модуля для керування асортиментом і контролю господарських операцій. Таким чином, система має забезпечувати повний цикл взаємодії з клієнтом – від моменту реєстрації до завершення покупки – із можливістю подальшого адміністрування та аналітичного супроводу.

Практична спрямованість роботи полягає у створенні функціонального програмного продукту, який може бути використаний як основа для реального інтернет-магазину або адаптований до інших сфер електронної комерції. У процесі виконання роботи здійснюється узагальнення сучасних підходів до розробки

вебзастосунків, поглиблення знань у галузі проєктування баз даних та організації клієнт-серверної архітектури, а також формування навичок інтеграції різних програмних компонентів у межах єдиної інформаційної системи.

Отже, створення вебзастосунку продажу одягу є комплексним завданням, що поєднує технологічні, організаційні та прикладні аспекти розробки програмного забезпечення й спрямоване на формування повноцінного інструменту для здійснення електронної торгівлі в сучасному цифровому середовищі.

1 ТЕОРЕТИЧНІ ОСНОВИ

1.1 Аналіз предметної області

Предметною областю даного дослідження є електронна комерція у сфері продажу одягу через вебзастосунки [11, 15]. Сучасний розвиток інформаційних технологій та широке розповсюдження мережі Інтернет сприяли активному переходу торгівлі в онлайн-середовище. Зокрема, ринок одягу є одним із найбільш динамічних сегментів електронної комерції[16], що зумовлено високим попитом, частою зміною колекцій та значною конкуренцією.

Вебзастосунок продажу одягу являє собою інформаційну систему, що забезпечує взаємодію між продавцем і покупцем у режимі реального часу [2]. Основною метою такої системи є автоматизація процесів вибору, замовлення та оплати товарів, а також управління асортиментом продукції.

У межах предметної області можна виділити основні сутності: користувач (покупець або адміністратор), товар (одяг), категорія товару, замовлення, кошик та платіж. Користувач має можливість переглядати каталог товарів, здійснювати пошук та фільтрацію за різними параметрами (ціна, розмір, колір, бренд), додавати товари до кошика та оформлювати замовлення. Адміністратор, у свою чергу, здійснює управління товарами, категоріями, замовленнями та користувачами.

Особливістю предметної області є необхідність обробки великої кількості товарних позицій, кожна з яких може мати варіації (наприклад, різні розміри та кольори). Це потребує ефективної організації бази даних та реалізації механізмів фільтрації і пошуку.

Також важливим аспектом є забезпечення зручного користувацького інтерфейсу, оскільки від цього залежить рівень задоволеності користувачів і, відповідно, успішність вебзастосунку [20, 21]. Необхідно враховувати принципи юзабіліті, адаптивність дизайну для різних пристроїв [13, 17], а також швидкість завантаження сторінок.

Окрім цього, вебзастосунок повинен забезпечувати безпеку обробки даних користувачів, зокрема персональної інформації та платіжних даних [14]. Це

включає використання сучасних методів аутентифікації, шифрування даних та захисту від типових вебзагроз.

Таким чином, аналіз предметної області показує, що розробка вебзастосунку продажу одягу потребує комплексного підходу, який охоплює проектування бази даних, реалізацію бізнес-логіки, створення зручного інтерфейсу та забезпечення безпеки системи.

1.2 Особливості інтернет-магазинів та існуючі аналоги

Інтернет-магазини є сучасною формою організації торгівлі, що функціонує в межах електронної комерції та забезпечує продаж товарів і послуг через мережу Інтернет [11]. На відміну від традиційних торговельних точок, інтернет-магазини мають низку характерних особливостей, які визначають їхню ефективність, функціональність та популярність серед користувачів.

Однією з ключових особливостей інтернет-магазинів є їх доступність незалежно від часу та місця перебування користувача [15]. Покупець має можливість здійснювати покупки цілодобово, використовуючи будь-який пристрій із доступом до Інтернету. Це значно підвищує зручність користування та розширює аудиторію клієнтів.

Іншою важливою особливістю є широкий асортимент товарів, який може значно перевищувати можливості фізичних магазинів. Завдяки відсутності обмежень, пов’язаних із торговельною площею, інтернет-магазини можуть пропонувати велику кількість товарних позицій, включаючи різні варіанти розмірів, кольорів та моделей.

Суттєвою характеристикою є наявність зручних інструментів пошуку та фільтрації товарів [20]. Користувачі можуть швидко знаходити потрібні товари за заданими параметрами, такими як ціна, бренд, категорія, розмір або інші характеристики. Це значно спрощує процес вибору та скорочує час на пошук необхідної продукції.

Інтернет-магазини також забезпечують можливість персоналізації взаємодії з користувачем [16, 21]. На основі попередніх переглядів, покупок або вподобань

система може пропонувати релевантні товари, що підвищує ймовірність здійснення покупки та покращує користувацький досвід.

Важливою особливістю є інтеграція з платіжними системами, що дозволяє здійснювати оплату онлайн різними способами [11], такими як банківські картки, електронні гаманці або інші сервіси. Це забезпечує гнучкість та зручність для користувачів.

Окрім цього, інтернет-магазини характеризуються автоматизацією бізнес-процесів. До таких процесів належать обробка замовлень, облік товарів, формування звітності, управління доставкою та взаємодія з клієнтами. Автоматизація дозволяє зменшити кількість помилок та підвищити ефективність роботи системи.

Ще однією важливою особливістю є необхідність забезпечення безпеки даних [14]. Інтернет-магазини повинні гарантувати захист персональної інформації користувачів, а також безпечне проведення платіжних операцій. Для цього застосовуються сучасні технології шифрування, аутентифікації та захисту від кіберзагроз.

Також варто зазначити, що інтернет-магазини мають можливість збору та аналізу даних про поведінку користувачів. Це дозволяє оптимізувати роботу системи, покращувати інтерфейс та адаптувати пропозиції під потреби клієнтів.

Таким чином, інтернет-магазини поєднують у собі технологічні, економічні та користувацькі аспекти, що робить їх ефективним інструментом сучасної торгівлі та важливою складовою цифрової економіки.

У межах аналізу застосунків-аналогів було розглянуто три популярні інтернет-магазини одягу: Answear.ua, Kasta.ua та Intertop.ua [4-6]. Дослідження дозволило виявити їхні спільні та відмінні риси з точки зору архітектури, технологій реалізації, функціональних можливостей, а також переваг і недоліків.

Усі три сервіси побудовані за принципом трирівневої архітектури (3-tier web application), що передбачає розділення системи на рівень представлення, бізнес-логіки та доступу до даних [10]. Такий підхід забезпечує гнучкість, масштабованість і зручність супроводу програмного забезпечення.

Застосунок Answear.ua розроблений компанією Answear.com S.A. (Краків, Польща) з використанням мов програмування PHP, Java та C# [4]. Система надає стандартний набір функцій для інтернет-магазину, зокрема реєстрацію та авторизацію користувачів, перегляд каталогу товарів, пошук і фільтрацію, роботу з кошиком, оформлення замовлення та онлайн-оплату. Також реалізовано інтеграцію з платіжними сервісами та можливість управління профілем користувача. До переваг застосунку належить зручний інтерфейс, досягнутий завдяки оптимізації фронтенду. Водночас недоліком є відсутність єдиної сторінки з відображенням усіх категорій товарів, що може ускладнювати навігацію.

Інтернет-магазин Kasta.ua, розроблений компанією «Маркасон», заснованою Андрієм Логвіним, використовує широкий стек технологій, включаючи Clojure, Java, Rust, Python (разом із Cython), а також мови C та C++ [6]. Особливістю даного застосунку є реалізація моделі маркетплейсу, що дозволяє працювати з кількома продавцями одночасно. Функціональність системи охоплює реєстрацію та авторизацію користувачів, кошик, онлайн-оплату, а також додаткові сервіси, такі як кешбек, бонусні програми, можливість оплати частинами, списки бажань і push-сповіщення. Основною перевагою є розвинена система сповіщень (push, email, SMS), яка забезпечує своєчасне інформування користувачів. Водночас спостерігаються недоліки, пов’язані з інтеграцією кешбек-системи, зокрема некоректне нарахування або зникнення бонусів.

Застосунок Intertop.ua, що належить ТОВ «Інтертоп Україна» під керівництвом Сергія Бадрітдінова, реалізований із використанням мов C++, PHP, JavaScript та SQL [5]. Основний функціонал включає реєстрацію та авторизацію користувачів, каталог товарів із розширеними можливостями фільтрації та сортування за такими параметрами, як бренд, розмір, ціна та колір. Додатково передбачено можливість перевірки наявності товару у фізичних магазинах, що є важливою перевагою для користувачів. Сильними сторонами системи є швидка робота пошуку, що досягається завдяки використанню індексування та розподілених сервісів. До недоліків належать збої під час оформлення замовлення,

зокрема поява дублікатів товарів у кошику, а також відсутність англомовної версії сайту.

Отже, проведений аналіз дозволяє зробити висновок, що сучасні інтернет-магазини одягу мають подібну базову функціональність, проте відрізняються рівнем реалізації додаткових можливостей, стабільністю роботи та зручністю користування. Отримані результати доцільно врахувати при розробці вебзастосунку з метою уникнення виявлених недоліків і впровадження найбільш ефективних рішень.

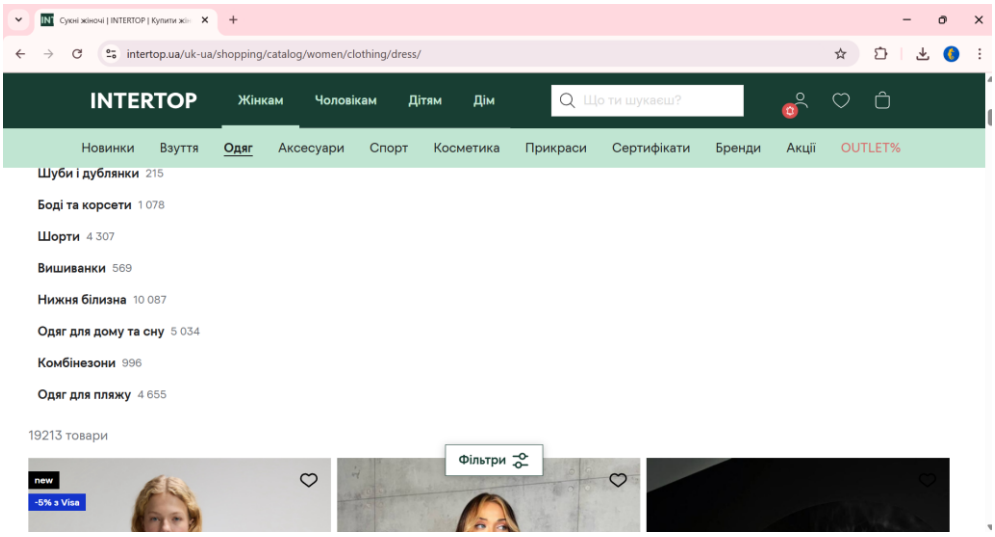

Рисунок 1.1 – Фрагмент користувацького інтерфейсу вебзастосунку інтернет-магазину Intertop

На рис.1.1 на основі наданого інтерфейсу сторінки каталогу інтернет-магазину можна виділити низку недоліків, які впливають на зручність користування та ефективність взаємодії користувача із системою.

По-перше, спостерігається перевантаженість інтерфейсу текстовими елементами у лівій частині екрана. Список категорій товарів представлений у вигляді суцільного переліку без достатньої візуальної ієрархії або групування. Це ускладнює швидке сприйняття інформації та орієнтацію користувача, особливо при великій кількості категорій.

По-друге, відсутня чітка візуальна акцентуація активного розділу або вибраної категорії. Користувач не отримує достатнього зворотного зв’язку щодо

свого поточного місцезнаходження в структурі каталогу, що може призводити до дезорієнтації під час навігації.

Третім недоліком є недостатня помітність елементів фільтрації. Кнопка «Фільтри» розміщена у центральній частині інтерфейсу, проте не має достатнього візуального виділення та може залишатися непоміченою користувачем. З огляду на важливість фільтрації в інтернет-магазинах одягу, така реалізація знижує ефективність пошуку товарів.

Крім того, верхнє меню містить значну кількість пунктів навігації (категорії, підкатегорії, додаткові розділи), що створює інформаційне перевантаження. Відсутність чіткого розмежування між основними та другорядними елементами меню ускладнює швидкий доступ до потрібних розділів.

Також варто відзначити недостатню візуальну диференціацію між окремими блоками сторінки. Наприклад, межі між списком категорій, результатами пошуку та іншими елементами інтерфейсу є слабо вираженими, що негативно впливає на сприйняття структури сторінки.

Ще одним недоліком є обмежена інформативність прев'ю товарів у нижній частині сторінки (з огляду на видиму область). Користувач не отримує достатньої кількості ключової інформації (наприклад, ціни, рейтингу або знижок) без додаткової взаємодії, що може знижувати швидкість прийняття рішення про покупку.

Таким чином, основними проблемами інтерфейсу є перевантаженість інформацією, недостатня візуальна ієрархія, слабка акцентуація важливих елементів та обмежена зручність навігації. Усунення зазначених недоліків дозволить підвищити юзабіліті вебзастосунку та покращити загальний користувацький досвід.

Сучасний ринок електронної комерції демонструє стрімке зростання: лише за 2025 рік обсяг світових онлайн-продажів перевищив 6 трильйонів доларів. Попри це, частка e-commerce у загальному обсязі роздрібної торгівлі становить лише 15%, що вказує на значний потенціал для подальшого розвитку. Вибір відповідної платформи є критичним фактором успіху для підприємця. Нижче

наведено детальний аналіз основних інструментів для створення інтернет-магазинів.

Існує спеціалізована e-commerce платформа – Shopify. Дана платформа вважається найбільш популярним вибором серед професійних продавців завдяки тому, що вона була розроблена виключно для потреб електронної торгівлі [2]. До її сильних сторін належать надзвичайна швидкість налаштування (запуск магазину можливий за 4 хвилини), цілодобова підтримка, високий рівень безпеки (включно з SSL-сертифікатом) та величезна бібліотека додатків та інтеграцій, що дозволяє уникнути витрат на розробників.

Слабкими сторонами можна вважати високу вартість обслуговування (від 29 доларів на місяць плюс оплата додаткових плагінів) та обмежені можливості кастомізації сторінок товару та оформлення замовлення. Крім того, відзначається зниження якості технічної підтримки останніми роками.

WooCommerce – найпопулярніша платформа у світі, що працює на базі WordPress [11]. До її переваг належить безкоштовне програмне забезпечення з найбільшою у світі бібліотекою плагінів, що забезпечує повну кастомізацію будь-якого елемента сайту. Вона є ідеальним рішенням для тих, хто має обмежений бюджет, оскільки витрати обмежуються лише хостингом (від 2-3 доларів на місяць). Як і всі інші платформи, дана платфорта також має слабкі сторони оскільки вона є менш зручною для користувачів без технічних навичок — глибокі зміни коду можуть вимагати залучення фахівців. Також вона вважається менш «професійною» порівняно з Shopify у питаннях вбудованої підтримки.

Існують універсальні конструктори сайтів з функціями e-commerce, наприклад, Squarespace та Wix. Ці платформи орієнтовані на створення візуально привабливого контенту, а торговельні функції були додані до них пізніше. Squarespace пропонує чудові дизайнерські шаблони, що підходять для митців. Wix є надійною платформою зі зручним інтерфейсом. Однак обидва сервіси не рекомендуються для повноцінного e-commerce бізнесу, оскільки їм бракує багатьох специфічних функцій, необхідних для масштабування торгівлі.

Big Cartel – це платформа для художників та креаторів, що дозволяє безкоштовно розміщувати до 5 товарів. Основним недоліком є обмеженість функцій, що робить перехід на інші платформи складним у міру зростання бізнесу.

Magento (Adobe Commerce) пропонує найвищий рівень кастомізації, проте є надзвичайно складною у використанні та вимагає обов'язкового залучення дизайнерів і розробників.

BigCommerce наразі орієнтується виключно на середній та великий бізнес, ігноруючи потреби дрібних підприємців.

Продажі на маркетплейсах мають кардинально іншу модель, де близько 50% усіх онлайн-транзакцій у США припадає саме на Amazon. Величезна існуюча база клієнтів дозволяє значно економити на маркетингу. Не потрібно створювати власний сайт — достатньо сторінки товару. Walmart, своєю чергою, пропонує можливість виходу в офлайн-ритейл у разі успішних продажів онлайн. Надзвичайно висока конкуренція (мільйони продавців) та значні комісійні збори (близько 15% від продажу). Головним ризиком є те, що маркетплейс може заблокувати акаунт у будь-який момент, на відміну від власного сайту на Shopify чи WooCommerce, де власник має повний контроль.

Для досягнення максимальних результатів рекомендується комбінований підхід: створення власного сайту паралельно з продажами на маркетплейсах. При виборі платформи для сайту варто орієнтуватися на бюджет та цілі: WooCommerce є оптимальним вибором для початківців з обмеженим бюджетом, тоді як Shopify забезпечує більш професійний та стабільний фундамент для довгострокового зростання бізнесу.

1.3 Психологічні аспекти взаємодії користувача з вебзастосунком

Під час розробки сучасних вебзастосунків важливу роль відіграє врахування психологічних особливостей користувачів та принципів взаємодії людини з програмним забезпеченням [20, 21]. Зручний та інтуїтивно зрозумілий інтерфейс безпосередньо впливає на ефективність роботи системи, швидкість виконання користувацьких дій та загальний рівень задоволеності користувачів. Для

вебзастосунків електронної комерції, зокрема вебзастосунку продажу одягу, правильне використання принципів UI/UX-дизайну дозволяє покращити навігацію, спростити процес оформлення замовлення та підвищити зручність взаємодії із системою [20].

Під час проєктування інтерфейсу доцільно враховувати когнітивні особливості сприйняття інформації користувачем, оскільки вони впливають на швидкість орієнтації в системі, пошук необхідних товарів та прийняття рішень під час роботи із вебзастосунком. Нижче наведено основні психологічні принципи, які враховуються при розробці інтерфейсу вебзастосунку.

Ефект першого враження полягає у тому, що початковий досвід взаємодії користувача з вебзастосунком значною мірою визначає подальше ставлення до системи[21]. Зручний дизайн, зрозуміла структура сторінок та швидке завантаження інтерфейсу формують позитивне враження про програмний продукт і підвищують ймовірність повторного використання вебзастосунку.

Суть ефекту послідовності полягає в тому, що користувачі краще запам’ятовують інформацію, яка розташована на початку та в кінці сторінки або послідовності дій. Тому при проєктуванні вебзастосунку важливо розміщувати найбільш важливі елементи інтерфейсу, такі як меню навігації, кнопки оформлення замовлення або інформацію про товар, у найбільш помітних областях сторінки.

Наступний прийом – ефект новизни, він характеризується тим, що користувачі звертають більше уваги на нову або оновлену інформацію. У вебзастосунку продажу одягу це може бути реалізовано через відображення нових колекцій, акційних пропозицій або рекомендацій товарів, що дозволяє підтримувати інтерес користувачів до системи.

Ефект повторного пред’явлення полягає у тому, що регулярна взаємодія користувача з певними елементами інтерфейсу спрощує їх сприйняття та використання. Використання однакових стилів кнопок, кольорової гами, структури сторінок та елементів навігації забезпечує цілісність інтерфейсу та зменшує когнітивне навантаження на користувача під час роботи із вебзастосунком [1, 20].

Під час розробки вебзастосунків електронної комерції важливо враховувати особливості процесу прийняття рішень користувачем. Правильна організація інтерфейсу та структури каталогу дозволяє спростити пошук товарів, зменшити когнітивне навантаження та підвищити ефективність взаємодії із системою.

Механізм уникнення втрат полягає у тому, що користувачі більш чутливо реагують на можливість втрати певної переваги, ніж на потенційне отримання вигоди. У вебзастосунку продажу одягу це може реалізовуватися через відображення інформації про обмежену кількість товару, завершення акцій або наявність товару на складі. Такі елементи інтерфейсу допомагають користувачеві швидше приймати рішення щодо оформлення замовлення.

Ефект компромісу проявляється у ситуаціях, коли користувач обирає між кількома варіантами товарів або цінових категорій. У більшості випадків користувачі схильні обирати середній варіант серед запропонованих. Тому під час проєктування каталогу товарів важливо забезпечити зрозуміле порівняння характеристик, цін та параметрів продукції.

Ефект якоріння полягає у тому, що перша отримана користувачем інформація впливає на подальше сприйняття інших даних. У вебзастосунку це може використовуватися під час відображення початкової вартості товару, знижок або акційних пропозицій. Наприклад, відображення попередньої ціни поруч із новою дозволяє користувачеві швидше оцінити вигідність покупки.

Перевантаження вибором виникає у випадках, коли користувач отримує надто велику кількість варіантів або функцій, що ускладнює процес прийняття рішення. Для уникнення цього під час розробки вебзастосунку необхідно забезпечити логічну структуру каталогу, використання фільтрів, категорій та системи пошуку, що спрощує навігацію та пошук необхідних товарів.

Ефект фреймінгу полягає у тому, що спосіб подання інформації впливає на її сприйняття користувачем. У вебзастосунку продажу одягу важливо використовувати зрозумілі формулювання, інформативні повідомлення та візуально доступне представлення даних, що покращує користувацький досвід та полегшує взаємодію із системою.

Ефект залучення користувача характеризується тим, що користувачі позитивніше сприймають програмний продукт, якщо мають можливість взаємодіяти із системою та впливати на її використання. Реалізація функцій відгуків, оцінювання товарів, списку обраного або персоналізованих рекомендацій підвищує зацікавленість користувачів у використанні вебзастосунку.

Ефект очікувань полягає у тому, що зрозумілий та передбачуваний інтерфейс сприяє більш ефективній роботі користувача із системою. Якщо вебзастосунок відповідає загальноприйнятим принципам UI/UX-дизайну, користувач швидше адаптується до роботи із системою та виконує необхідні дії без додаткових труднощів.

Упередження підтвердження проявляється у тому, що користувачі схильні звертати увагу на інформацію, яка відповідає їхнім попереднім уподобанням або інтересам. У вебзастосунку продажу одягу це враховується через реалізацію систем рекомендацій, персоналізованих добірок товарів та збереження історії переглядів, що дозволяє підвищити зручність користування системою та релевантність відображуваного контенту.

Під час розробки вебзастосунків електронної комерції важливим аспектом є забезпечення довіри користувачів до системи та мінімізація можливих ризиків під час взаємодії із вебресурсом. Надійність програмного забезпечення, зрозумілий інтерфейс та прозорість функціонування системи позитивно впливають на користувацький досвід і сприяють підвищенню ефективності використання вебзастосунку.

Ефект мінімізації ризику полягає у прагненні користувачів уникати невизначеності та потенційних помилок під час роботи із системою. Для зменшення психологічного бар’єру у вебзастосунку продажу одягу доцільно реалізувати зрозумілі умови оформлення замовлення, відображення інформації про наявність товарів, політику повернення, а також систему підтвердження дій користувача. Це підвищує рівень довіри до програмного продукту та зменшує ймовірність помилкових дій.

Ефект соціального підтвердження проявляється у тому, що користувачі схильні орієнтуватися на досвід інших людей під час прийняття рішень. У вебзастосунках електронної комерції цей принцип реалізується через систему відгуків, рейтинги товарів, кількість замовлень або рекомендації популярних товарів. Такі елементи допомагають користувачеві швидше оцінити якість продукції та підвищують довіру до системи.

Упередження «сліпої плями» характеризується тим, що користувачі не завжди усвідомлюють вплив особливостей інтерфейсу на власну поведінку під час роботи із системою. Саме тому під час проєктування вебзастосунку необхідно дотримуватись принципів прозорості, зрозумілості та етичності взаємодії з користувачем. Інтерфейс повинен допомагати користувачеві виконувати необхідні дії, а не ускладнювати процес роботи або вводити його в оману.

Етичний підхід до розробки програмного забезпечення передбачає орієнтацію на потреби користувача, забезпечення зручності використання системи та створення безпечного цифрового середовища. Основною метою вебзастосунку продажу одягу є не лише забезпечення функціональності системи, а й створення комфортного та ефективного середовища для взаємодії користувача з програмним продуктом.

Висновки до розділу 1

У результаті проведеного аналізу предметної області встановлено, що електронна комерція у сфері продажу одягу є динамічним і конкурентним середовищем, яке потребує застосування сучасних інформаційних технологій та комплексного підходу до розробки вебзастосунків. Було визначено основні сутності системи, їх взаємозв’язки та ключові функціональні можливості, необхідні для забезпечення повноцінного процесу онлайн-продажу.

Дослідження особливостей інтернет-магазинів показало, що їх ефективність визначається доступністю, широким асортиментом товарів, наявністю зручних інструментів пошуку і фільтрації, персоналізацією взаємодії з користувачем, а також високим рівнем автоматизації бізнес-процесів і безпеки даних. Водночас

важливу роль відіграє якість користувацького інтерфейсу, що безпосередньо впливає на досвід користувача та конверсію.

Аналіз існуючих аналогів (Answear.ua, Kasta.ua, Intertop.ua) дозволив виявити як спільні підходи до побудови систем (зокрема використання трирівневої архітектури), так і відмінності у функціональності, технологічних рішеннях та реалізації додаткових сервісів. Окрему увагу було приділено їхнім перевагам і недолікам, що дало змогу визначити типові проблеми, пов’язані з юзабіліті, стабільністю роботи та організацією інтерфейсу [13].

Проведений аналіз інтерфейсу також підтвердив, що навіть у популярних вебзастосунках існують недоліки, пов’язані з перевантаженням інформацією, недостатньою візуальною ієрархією та неефективною навігацією. Це підкреслює необхідність ретельного проєктування користувацького інтерфейсу при розробці власного продукту.

Крім того, було розглянуто сучасні платформи для створення інтернет-магазинів, що дозволило визначити їхні переваги, обмеження та доцільність використання залежно від потреб бізнесу. Встановлено, що вибір платформи суттєво впливає на можливості масштабування, рівень кастомізації та загальну ефективність вебзастосунку.

Таким чином, результати проведеного аналізу є підґрунтям для формування вимог до розробки вебзастосунку продажу одягу, а також дозволяють визначити напрямки вдосконалення функціональності, інтерфейсу та архітектури майбутньої системи з урахуванням виявлених недоліків і сучасних тенденцій розвитку електронної комерції.

2 ТЕХНОЛОГІЇ РОЗРОБКИ ТА ВИМОГИ ДО ВЕБЗАСТОСУНКУ

2.1 Інструментарій, моделі та методи

У процесі розробки вебзастосунку продажу одягу важливим етапом є вибір інструментальних засобів, моделей та методів, які забезпечують ефективність створення, функціонування та супроводу програмного продукту [2, 10]. Обраний стек технологій має відповідати вимогам продуктивності, масштабованості, безпеки та зручності використання.

Для реалізації клієнтської частини вебзастосунку використано мови розмітки та стилізації HTML і CSS [7-9]. HTML – це абревіатура, що розшифровується як Hypertext Markup Language (мова розмітки гіпертексту). Кожен елемент цієї назви відображає ключову функцію мови:

- гіпертекст (Hypertext) вказує на те, що HTML дозволяє створювати посилання на вебсторінках, за якими користувачі можуть переходити (клікати);
- мова розмітки (Markup Language) означає використання спеціальних позначок для структурування звичайного тексту, щоб вказати браузеру на специфіку та призначення певних фрагментів інформації.

Основою HTML є теги, які зазвичай використовуються парами: відкриваючий та закриваючий. Закриваючий тег відрізняється від відкриваючого наявністю похилої риски (слеша). Наприклад, фрагмент тексту, який потрібно виділити як важливий, обгортається тегами (відкриваючий) та (закриваючий). Браузер використовує ці теги як інструкцію. Наприклад, тег зображення <img> містить атрибут src (джерело), що вказує шлях до файлу в інтернеті, а також атрибути width (ширина) та height (висота), щоб браузер заздалегідь знав розмір об’єкта.

Головна роль HTML полягає в описі призначення тексту, а не просто в його візуальному оформленні. Це має критичне значення для кількох аспектів:

- ієрархія контенту, що реалізується завдяки використання заголовків різних рівнів (наприклад, <h3> або <h4>) допомагає структурувати інформацію;

– теги допомагають програмам зчитування екрана (screen readers) правильно інтерпретувати текст для людей з порушеннями зору [14]. Наприклад, тег підказує програмі прочитати слово з більшим акцентом, а атрибут alt у тезі зображення надає текстовий опис картинки;

– пошукова оптимізація (SEO): правильна розмітка заголовків та структури допомагає пошуковим системам, таким як Google, краще розуміти зміст сторінки.

HTML надає опис структури (наприклад, «це заголовок» або «це таблиця»), але браузер самостійно вирішує, як саме візуалізувати цей опис. Важливо пам'ятати, що різні браузери можуть відображати один і той самий HTML-код по-різному. Тому не варто покладатися виключно на HTML для точного візуального оформлення (наприклад, для надання тексту жирності чи певного розміру); його першочергове завдання – логічний опис елементів.

Багато сучасних систем управління контентом використовують візуальні редактори типу WYSIWYG (What You See Is What You Get – «що бачиш, те й отримуєш»), де користувач працює з текстом, як у звичайному документі. Однак за лаштунками такий редактор автоматично генерує HTML-код. Знання основ HTML дозволяє авторам перемикатися в режим «Текст» (Raw HTML) для точного виправлення помилок розмітки, які важко усунути через візуальний інтерфейс.

HTML є фундаментальною мовою вебтехнологій, яка перетворює звичайний текст на структурований гіпертекст [7, 8]. Вона описує роль кожного елемента на сторінці – від заголовків та таблиць до зображень і посилань, забезпечуючи коректну роботу браузерів, пошукових систем та засобів доступності

HTML забезпечує структуризацію вебсторінок, формуючи логічну організацію контенту, тоді як CSS відповідає за візуальне оформлення інтерфейсу, включаючи кольорову схему [12], розташування елементів, адаптивність та забезпечення зручності користування на різних пристроях [9, 17].

CSS (Cascading Style Sheets) – це мова, що використовується для опису зовнішнього вигляду та форматування документів, написаних мовою розмітки (зазвичай HTML). У сучасній веброзробці CSS є фундаментальним інструментом,

який дозволяє відокремити зміст сторінки від її візуального представлення. Стилі можна прописувати безпосередньо в HTML-коді, проте найбільш ефективною практикою є використання окремих файлів стилів.

Селектори в CSS призначені для пошуку та вибору елементів HTML, до яких потрібно застосувати стилі. Вибір може здійснюватися за типом елемента (наприклад, h1), за класом (з використанням крапки, наприклад, .class-name) або за ідентифікатором (ID, з використанням символу решітки #). Селектори також можна об'єднувати в ланцюжки для більш точного націлювання на вкладені елементи.

Атрибути – це безпосередньо стилі (властивості), які застосовуються до вибраних елементів.

Стилі в CSS застосовуються послідовно – зверху вниз по файлу. У разі виникнення конфліктів між правилами вступає в дію механізм специфічності. Селектори за ідентифікатором (ID) мають найвищий пріоритет, за ними йдуть класи, і лише потім – загальні селектори елементів.

Блокова модель (Box Model) у CSS кожен елемент розглядається як прямокутна область, що складається з чотирьох частин:

1) контент (Content): власне вміст елемента, розміри якого визначаються властивостями width (ширина) та height (висота). Рекомендується задавати їх у відсотках для забезпечення гнучкості макета;

2) внутрішні відступи (Padding): простір між контентом і рамкою;

3) рамка (Border): межа елемента, що має параметри розміру, типу (наприклад, solid) та кольору;

4) зовнішні відступи (Margin): простір зовні елемента, що відокремлює його від сусідніх об'єктів.

Для розташування елементів на сторінці використовується властивість display. Block та Inline визначають, чи буде елемент займати весь доступний рядок, чи лише необхідне місце. Flexbox та Grid – це потужні інструменти для створення складних сіток та вирівнювання контенту [17]. flex вважається універсальним засобом для горизонтального та вертикального центрування елементів. Також існує

властивість position (наприклад, relative або absolute), яка дозволяє розміщувати елементи в специфічних точках екрана.

CSS надає широкі можливості для візуального оформлення. Кольори задаються за назвами або шістнадцятковими кодами (hex codes). Псевдокласи наприклад, :hover дозволяє змінювати стиль елемента, коли користувач наводить на нього курсор миші. Анімації та переходи (Transitions) дають змогу створювати плавні зміни станів та складні анімації за допомогою ключових кадрів (keyframes). Тіні та ефекти, властивості box-shadow та drop-shadow додають глибини інтерфейсу [18].

Для коректного відображення сторінок на різних пристроях (мобільні телефони, планшети) використовуються медіа-запити (media queries) [13, 17]. Вони працюють за принципом умовних операторів: якщо ширина екрана відповідає певній умові, застосовується відповідний набір стилів.

На професійному рівні розробники часто використовують препроцесори (наприклад, Sass або SCSS), які розширюють можливості CSS змінними та вкладеними властивостями, а також сучасні підходи, такі як CSS-in-JS (Styled Components) у React-фреймворках.

Отже, CSS – це багатогранна мова, яка пройшла шлях від простого форматування тексту до створення складних анімованих та адаптивних інтерфейсів. Розуміння селекторів, блокової моделі та механізмів позиціювання є критично важливим для створення сучасного веб-продукту.

Серверна частина застосунку реалізована з використанням мови програмування PHP[3, 10]. Дана мова є однією з найбільш поширених у сфері веброзробки та забезпечує ефективну обробку запитів користувачів, взаємодію з базою даних, реалізацію бізнес-логіки та обробку форм. PHP дозволяє створювати динамічні вебсторінки та забезпечує інтеграцію з різними сервісами.

Мова програмування PHP, що є рекурсивною абревіатурою від «PHP: Hypertext Preprocessor», є провідним інструментом написання сценаріїв на стороні сервера, розробленим спеціально для сфери веб-розробки [10]. Створена Расмусом Лердорфом у 1994 році, вона пройшла тривалу еволюцію від простого

інструментарію для персональних сторінок до зрілої, перевіреної часом технології, яка сьогодні забезпечує функціонування майже 80 відсотків усіх вебсайтів у світі. Академічний та практичний інтерес до цієї мови зумовлений її домінуючим становищем на ринку, де вона значно випереджає конкурентів, таких як ASP.NET, Ruby чи Python, у сегменті серверних рішень.

Основне призначення PHP полягає у створенні динамічного та інтерактивного контенту для веб-ресурсів. Завдяки своїй здатності інтегруватися безпосередньо в HTML-код або використовуватися у вигляді окремих файлів, PHP дозволяє розробникам будувати складні системи шаблонів, обробляти дані, отримані від користувачів через форми, та керувати механізмами автентифікації на сайтах із членством. Технологічні можливості мови охоплюють роботу з базами даних, зокрема MySQL, шифрування конфіденційної інформації, управління сесіями та файлами куки, а також створення файлів у форматах PDF, XML та JSON.

Принцип роботи PHP базується на циклі запиту та відповіді, де сервер відіграє ключову роль. Коли браузер клієнта робить запит до веб-сервера, останній ідентифікує файли з розширенням .php і запускає процес обробки коду всередині них. Під час цього процесу сервер може взаємодіяти з базою даних для отримання або оновлення інформації, після чого він формує фінальний HTML-код і надсилає його назад у браузер користувача. Важливою перевагою такого підходу є безпека, оскільки вихідний логічний код залишається на сервері й ніколи не передається клієнту, що робить його невидимим для відвідувачів сайту.

Актуальність PHP підкріплюється її використанням як ядра для найпопулярніших систем керування вмістом (CMS), таких як WordPress, що охоплює близько 38 відсотків динамічних сайтів, а також Joomla, Drupal та Magento. Окрім цього, сучасна екосистема мови пропонує потужні фреймворки та бібліотеки, зокрема Laravel та Symfony, які значно прискорюють процес розробки та сприяють створенню безпечного й ефективного програмного забезпечення [3]. Величезна кількість великих технологічних компаній, серед яких Facebook, Slack, Wikipedia та Etsy, покладаються на PHP у своїй серверній інфраструктурі, що

свідчить про високу надійність та продуктивність цієї мови, особливо в її останніх версіях.

З погляду професійної перспективи, вивчення PHP залишається стратегічно вигідним рішенням для спеціалістів, орієнтованих на веб-розробку. Високий попит на розробників на ринку праці та конкурентоспроможний рівень заробітних плат відображають стабільність цієї технології. Попри появу інших мов, PHP зберігає свою унікальну нішу завдяки спеціалізації на веб-середовищі, простоті опанування та величезній спільноті, яка продовжує активно розвивати та підтримувати мову, забезпечуючи її відповідність сучасним вимогам безпеки та швидкодії [19].

Для збереження та обробки даних використовується система керування базами даних MySQL. Вона забезпечує надійне зберігання інформації про користувачів, товари, замовлення та інші сутності системи. Використання мови запитів SQL дозволяє здійснювати ефективні операції вибірки, додавання, оновлення та видалення даних.

У процесі проєктування системи застосовано інструменти моделювання, зокрема Software Ideas Modeler та StarUML. Дані засоби використовуються для створення UML-діаграм, що дозволяють візуалізувати структуру системи, її компоненти та взаємозв’язки між ними. Це сприяє кращому розумінню архітектури застосунку та спрощує процес його розробки.

З точки зору архітектурних рішень, вебзастосунок реалізується за трирівневою моделлю (3-tier architecture), яка включає рівень представлення (інтерфейс користувача), рівень бізнес-логіки (серверна частина) та рівень даних (база даних). Такий підхід забезпечує розділення відповідальностей, підвищує масштабованість системи та спрощує її модифікацію.

У процесі розробки застосовуються стандартні методи вебпрограмування, зокрема клієнт-серверна взаємодія, обробка HTTP-запитів, використання CRUD-операцій для роботи з даними, а також методи забезпечення безпеки, такі як валідація введених даних та захист від типових вебзагроз.

Концепція CRUD, що є акронімом від англійських термінів Create (створення), Read (читання), Update (оновлення) та Delete (видалення), становить

фундамент функціональності будь-якого інтерфейсу прикладного програмування (API) або системи управління базами даних. В академічному та практичному сенсі CRUD визначає чотири базові типи операцій, які необхідно реалізувати для повноцінного керування інформаційними сутностями в межах програмного забезпечення. Хоча традиційно цей підхід асоціюється з реляційними базами даних, сучасна парадигма розробки поширює ці принципи на будь-які обчислювальні об'єкти, включаючи файлові системи, структури в пам'яті та веб-служби.

У середовищі веб-розробки кожна з операцій CRUD має чітку кореляцію з методами протоколу HTTP, що дозволяє стандартизувати взаємодію між клієнтом і веб-сервером. Операція створення (Create) реалізується через метод POST, який призначений для відправлення нових даних на сервер з метою їх подальшого збереження в базі. Функція читання (Read) відповідає методу GET, що дозволяє отримувати доступ до наявної інформації – як до повних списків об'єктів, так і до конкретних записів за їхніми унікальними ідентифікаторами – без внесення будь-яких змін у систему. Для модифікації вже існуючих даних використовується операція оновлення (Update), яка в архітектурі веб-запитів представлена методом PUT. Завершальний етап життєвого циклу даних забезпечується операцією видалення (Delete), що виконується однойменним HTTP-методом і дозволяє безповоротно вилучати записи з серверної інфраструктури [19].

Практична необхідність впровадження моделі CRUD стає очевидною при розгляді динамічних систем, де дані постійно трансформуються. На прикладі API для керування асортиментом товарів можна простежити, як розробник використовує метод POST для додавання нової позиції з унікальною назвою та ціною, або метод GET для відображення меню користувачеві. Важливість операцій оновлення та видалення підкреслюється потребами бізнесу, такими як зміна вартості продукту або вилучення позиції, що більше не доступна для замовлення. Без реалізації повної структури CRUD вебзастосунок втрачає свою гнучкість та здатність адекватно реагувати на зміни в реальному часі.

Отже, дотримання алгоритму CRUD є критично важливим аспектом проектування веб-додатків, оскільки воно дозволяє структурувати моделі даних та забезпечує розробникам чітку схему побудови маршрутів API. Використання стандартних HTTP-методів у поєднанні з цією структурою робить взаємодію з даними передбачуваною та ефективною. Розуміння цих принципів дозволяє створювати масштабовані системи, де кожен запит до сервера має визначену роль у процесі обробки інформації, що є обов'язковою умовою для розробки сучасного високоякісного програмного забезпечення.

Крім того, при створенні інтерфейсу враховуються принципи юзабіліті та адаптивного дизайну, що дозволяє забезпечити зручність використання вебзастосунку на різних типах пристроїв, включаючи персональні комп’ютери, планшети та смартфони.

Таким чином, обраний інструментарій, моделі та методи розробки є доцільними для створення вебзастосунку продажу одягу, оскільки вони забезпечують необхідний рівень функціональності, продуктивності та зручності користування, а також відповідають сучасним вимогам до вебсистем.

2.2 Специфікація вимог

Вебзастосунок інтернет-магазину одягу призначений для здійснення онлайн-продажу товарів через мережу Інтернет [11]. Система забезпечує користувачам можливість перегляду каталогу продукції, додавання товарів до кошика, оформлення замовлення та здійснення оплати, а також надає адміністраторам інструменти для управління товарами, замовленнями та користувачами.

Розробка здійснюється з метою створення зручного та функціонального вебзастосунку електронної комерції з використанням сучасних вебтехнологій. Система орієнтована на подальше розширення функціональності та масштабування [2].

Проєкт охоплює створення вебзастосунку з клієнтською та серверною частинами, базою даних і базовою інтеграцією з платіжною системою. Розробка

мобільного застосунку не передбачена. Система не включає логістичну інфраструктуру доставки, а лише обробляє інформацію про замовлення.

Система використовується для онлайн-продажу одягу фізичним особам. Вебзастосунок дозволяє здійснювати пошук товарів, переглядати їх характеристики, оформлювати замовлення та обирати спосіб оплати.

Система передбачає такі ролі:

- гість (перегляд каталогу);
- зареєстрований користувач (оформлення замовлення, управління кошиком);
- адміністратор (управління товарами та замовленнями).

Користувачі повинні мати базові навички роботи з веббраузером.

Система складається з:

- клієнтської частини (інтерфейс користувача);
- серверної частини (обробка запитів, бізнес-логіка);
- бази даних (збереження інформації про товари, користувачів, замовлення);
- інтеграції з платіжним сервісом.

Система функціонує за наявності стабільного інтернет-з’єднання. Коректна робота гарантується у сучасних веббраузерах.

Система забезпечує створення облікового запису користувача та вхід до системи з використанням логіну та пароля. Дані користувача зберігаються у базі даних у захищеному вигляді.

Користувач має можливість переглядати список товарів із зазначенням назви, ціни, опису та залишку на складі. Передбачено фільтрацію та сортування товарів.

Користувач може додавати товари до кошика, змінювати кількість або видаляти їх. Система автоматично розраховує загальну вартість замовлення.

Під час оформлення замовлення користувач вводить дані доставки та обирає спосіб оплати (карткою або при отриманні). Після підтвердження створюється запис у базі даних та зменшується кількість товару на складі.

Адміністратор має можливість переглядати список замовлень, змінювати їх статус та управляти каталогом товарів.

Вхідна інформація:

- дані користувача (ім’я, контактний телефон, адреса доставки);
- дані про товари (назва, ціна, опис, залишок);
- параметри замовлення (кількість, спосіб оплати).

Інформація зберігається у реляційній базі даних. Передбачено цілісність даних за допомогою зовнішніх ключів та транзакцій.

Система повинна працювати на сервері з підтримкою PHP та СКБД SQL Server (або іншої реляційної СКБД). Доступ до системи здійснюється через веббраузер.

Система реалізована за клієнт-серверною архітектурою.

Технології розробки:

- Frontend: HTML, CSS3;
- Backend: PHP;
- База даних: SQL Server;
- Інтеграція платежів: Stripe.

Інтерфейс користувача має бути адаптивним та зрозумілим. Передбачено навігаційне меню, сторінку каталогу, кошик, форму оформлення замовлення та сторінку підтвердження.

Система повинна забезпечувати:

- коректну обробку не менше 100 одночасних користувачів;
- час відповіді сервера до 2 секунд;
- захист від SQL-ін’єкцій;
- збереження конфіденційності персональних даних.

Передбачена можливість розширення функціоналу: додавання системи знижок, відгуків, історії замовлень та інтеграції з службами доставки.

Висновки до розділу 2

У результаті проведеного аналізу інструментарію, моделей та методів розробки вебзастосунку інтернет-магазину одягу було визначено доцільність використання сучасного технологічного стеку, який забезпечує ефективне створення та подальший супровід програмного продукту. Поєднання HTML і CSS дозволяє реалізувати структуровану та адаптивну клієнтську частину системи, що відповідає вимогам сучасного вебдизайну та забезпечує зручність взаємодії користувача з інтерфейсом.

Використання PHP як серверної мови програмування забезпечує реалізацію динамічної логіки застосунку, обробку запитів користувачів та взаємодію з базою даних, що є критично важливим для функціонування інтернет-магазину. Застосування MySQL як системи керування базами даних дозволяє ефективно організувати зберігання та обробку інформації про товари, користувачів і замовлення з дотриманням вимог цілісності та надійності даних.

Використання архітектури клієнт-сервер та трирівневої моделі сприяє чіткому розмежуванню відповідальностей між компонентами системи, що підвищує масштабованість, підтримуваність та гнучкість розробленого застосунку. Крім того, застосування CRUD-операцій та стандартних HTTP-методів забезпечує уніфіковану та передбачувану взаємодію з даними.

Отже, обрані інструменти, технології та методи розробки є оптимальними для реалізації вебзастосунку продажу одягу, оскільки вони забезпечують необхідний рівень функціональності, продуктивності, безпеки та зручності використання, а також відповідають сучасним вимогам до веборієнтованих інформаційних систем.

3 ПРОЄКТУВАННЯ ВЕБЗАСТОСУНКУ ПРОДАЖУ ОДЯГУ

3.1 Аналіз системи, що розробляється

Для детального опису того, як користувач взаємодіє з системою та опису поведінки системи у відповідь на запити користувача розроблено сценарії використання.

Use Case 1: «Зареєструватися та увійти в систему»

Область застосування (Scope)

Інтернет-магазин.

Рівень (Level)

User-goal (орієнтований на ціль користувача).

Основний актор (Primary Actor)

Клієнт.

Зацікавлені сторони та їхні інтереси (Stakeholders and Interests):

1) клієнт зацікавлений у отриманні доступу до персонального кабінету та можливості користуватися функціоналом сайту;

2) адміністратор зацікавлений у наявності зареєстрованих користувачів для управління замовленнями та обліковими записами;

3) бізнес компанія зацікавлена у зборі контактних даних користувачів для комунікації та маркетингових цілей.

Передумови (Preconditions)

Користувач має доступ до вебсайту інтернет-магазину та володіє валідною електронною поштою.

Гарантія успішного виконання (Success Guarantee):

Обліковий запис користувача створено, електронну пошту підтверджено, користувач успішно авторизувався в системі.

Основний сценарій успіху (Main Success Scenario):

1) користувач натискає кнопку «Реєстрація»;

2) користувач вводить особисті дані: електронну пошту, пароль, ПІБ та номер телефону;

3) система перевіряє унікальність електронної пошти та зберігає дані користувача в базі даних Users;

4) система надсилає лист для підтвердження електронної пошти;

5) користувач переходить за посиланням у листі та підтверджує email;

6) користувач вводить email і пароль та входить у систему.

Розширення (Extensions):

3)а на кроці перевірки електронної пошти система повідомляє користувача, якщо така email-адреса вже існує;

6)а під час входу в систему у разі введення некоректних облікових даних система відображає повідомлення про помилку.

Спеціальні вимоги (Special Requirements):

1) паролі користувачів повинні зберігатися в зашифрованому вигляді;

2) система має бути захищена від SQL Injection та інших типів атак.

Технологічні та інформаційні варіації (Technology and Data Variations)

Для хешування паролів використовуються алгоритми bcrypt або argon2.

Аутентифікація може реалізовуватися за допомогою JWT (JSON Web Token) або session-based механізму.

Частота використання (Frequency of Occurrence)

Близько 30% усіх взаємодій користувачів із системою.

Додаткові положення (Miscellaneous)

Система повинна підтримувати можливість відновлення пароля через електронну пошту.

Use Case 2: «Переглянути та знайти товари»

Область застосування (Scope)

Інтернет-магазин.

Рівень (Level)

User-goal.

Основний актор (Primary Actor)

Користувач.

Зацікавлені сторони та їхні інтереси (Stakeholders and Interests):

1) користувач зацікавлений у пошуку та перегляді потрібного товару;

2) бізнес компанія зацікавлена у збільшенні продажів шляхом зручного доступу до каталогу.

Передумови (Preconditions)

Каталог товарів заповнений та доступний для перегляду.

Гарантія успішного виконання (Success Guarantee)

Користувач отримує релевантні результати пошуку та відкриває сторінку обраного товару.

Основний сценарій успіху (Main Success Scenario):

1) користувач відкриває каталог товарів;

2) користувач застосовує фільтри для уточнення результатів;

3) користувач вводить пошуковий запит;

4) система повертає результати з таблиці Products;

5) користувач відкриває сторінку товару та переглядає детальну інформацію.

Розширення (Extensions):

3)а. якщо за введеним запитом товари не знайдено, система відображає повідомлення «Товарів не знайдено».

Спеціальні вимоги (Special Requirements)

Пошук має виконуватися швидко (менше 2 секунд).

Інтерфейс повинен бути адаптивним для різних типів пристроїв.

Технологічні та інформаційні варіації (Technology and Data Variations)

Використання SQL LIKE або повнотекстового пошуку.

Кешування результатів пошуку для підвищення продуктивності.

Частота використання (Frequency of Occurrence)

Близько 80% усіх взаємодій користувачів із системою.

Додаткові положення (Miscellaneous):

Користувач має можливість додати товар до кошика або списку бажань (wishlist).

Use Case 3: «Оформити замовлення»

Область застосування (Scope)

Інтернет-магазин.

Рівень (Level)

User-goal.

Основний актор (Primary Actor)

Клієнт.

Зацікавлені сторони та їхні інтереси (Stakeholders and Interests):

1) клієнт зацікавлений у отриманні замовленого товару;
2) бізнес зацікавлений у здійсненні продажу;
3) служба доставки зацікавлена у коректному отриманні даних для доставки.

Передумови (Preconditions)

Користувач авторизований у системі, а кошик містить хоча б один товар.

Гарантія успішного виконання (Success Guarantee)

Замовлення створено та збережено в таблиці Orders.

Основний сценарій успіху (Main Success Scenario):

1) користувач відкриває сторінку кошика;
2) користувач обирає спосіб доставки та метод оплати;
3) система автоматично підраховує загальну суму замовлення;
4) користувач підтверджує оформлення замовлення;
5) система зберігає замовлення та надсилає електронне повідомлення користувачу.

Розширення (Extensions):

2)а. у разі помилки під час оплати система відображає повідомлення користувачу.

Спеціальні вимоги (Special Requirements)

Система повинна забезпечувати надійність платіжних операцій.

Розрахунок вартості замовлення має бути коректним і точним.

Технологічні та інформаційні варіації (Technology and Data Variations)

Інтеграція з платіжними шлюзами LiqPay або Stripe.

Частота використання (Frequency of Occurrence)

Близько 20% усіх взаємодій у системі.

Додаткові положення (Miscellaneous)

Статус замовлення змінюється автоматично відповідно до етапів обробки.

Use Case 4: «Керувати каталогом товарів»

Область застосування (Scope)

Інтернет-магазин.

Рівень (Level)

User-goal.

Основний актор (Primary Actor)

Адміністратор.

Зацікавлені сторони та їхні інтереси (Stakeholders and Interests):

1) адміністратор зацікавлений у підтримці актуальності даних каталогу;
2) бізнес зацікавлений у контролі асортименту товарів.

Передумови (Preconditions):

Адміністратор успішно авторизований у системі.

Гарантія успішного виконання (Success Guarantee)

Усі зміни збережені та коректно відображаються в каталозі товарів.

Основний сценарій успіху (Main Success Scenario):

1) адміністратор входить до адмін-панелі;
2) адміністратор відкриває розділ «Товари»;
3) адміністратор додає, редагує або видаляє товар;
4) адміністратор зберігає внесені зміни;
5) система фіксує дію в журналі AdminLogs.

Розширення (Extensions):

3)а. у разі помилки валідації даних система відображає повідомлення про помилку.

Спеціальні вимоги (Special Requirements)

Система повинна забезпечувати розмежування прав доступу для адміністраторів.

Технологічні та інформаційні варіації (Technology and Data Variations)

Реалізація CRUD-операцій.

Підтримка завантаження зображень товарів.

Частота використання (Frequency of Occurrence)

Близько 5% усіх операцій у системі.

Додаткові положення (Miscellaneous)

Журнали дій адміністратора не підлягають редагуванню.

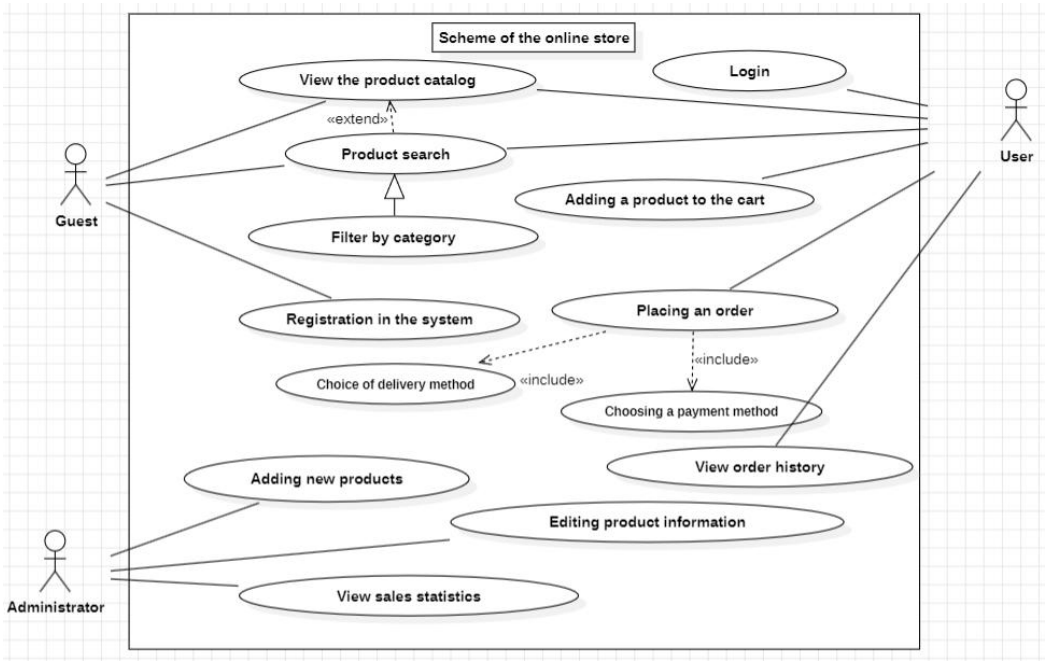

Рисунок 3.1 – Діаграма сценаріїв використання

Use Case 5: «Керувати профілем та історією замовлень»

Область застосування (Scope)

Інтернет-магазин.

Рівень (Level)

User-goal.

Основний актор (Primary Actor)

Користувач.

Зацікавлені сторони та їхні інтереси (Stakeholders and Interests):

1) користувач зацікавлений у контролі персональних даних та історії замовлень;

2) бізнес зацікавлений в актуальності інформації про клієнтів.

Передумови (Preconditions)

Користувач авторизований у системі.

Гарантія успішного виконання (Success Guarantee)

Персональні дані користувача оновлені, а історія замовлень відображається коректно.

Основний сценарій успіху (Main Success Scenario):

1) користувач входить у систему;
2) користувач відкриває розділ «Мій профіль»;
3) користувач переглядає та редагує персональні дані;
4) система оновлює інформацію в таблиці Users;
5) користувач відкриває розділ «Мої замовлення»;
6) користувач переглядає список замовлень та їх деталі.

Розширення (Extensions):

4)а. у разі введення некоректних даних система відображає повідомлення про помилку.

Спеціальні вимоги (Special Requirements)

Система повинна забезпечувати захист персональних даних користувачів.

Технологічні та інформаційні варіації (Technology and Data Variations)

Валідація даних здійснюється як на стороні клієнта (frontend), так і на стороні сервера (backend).

Частота використання (Frequency of Occurrence)

Близько 25% усіх взаємодій користувачів із системою.

Додаткові положення (Miscellaneous)

Передбачена можливість повторного оформлення замовлення.

Діаграма класів відображає структуру вебзастосунку для продажу одягу, а також основні сутності системи, їх атрибути, методи та взаємозв’язки між ними. Дана модель дозволяє формалізувати архітектуру програмного забезпечення та визначити логіку взаємодії компонентів.

Центральною сутністю системи є клас User, який описує користувача платформи. Він містить основні атрибути, зокрема ідентифікатор, ім’я, електронну пошту, пароль, роль та статус блокування. Методи класу забезпечують реєстрацію, авторизацію, вихід із системи та оновлення профілю. Користувач може взаємодіяти з кошиком і оформлювати замовлення.

Клас Cart представляє кошик користувача, який містить вибрані товари перед оформленням замовлення. Він має атрибути ідентифікатора та дати створення, а також методи додавання, видалення товарів і обчислення загальної вартості. Кошик пов’язаний із користувачем відношенням «один до одного» або «один до багатьох» залежно від реалізації.

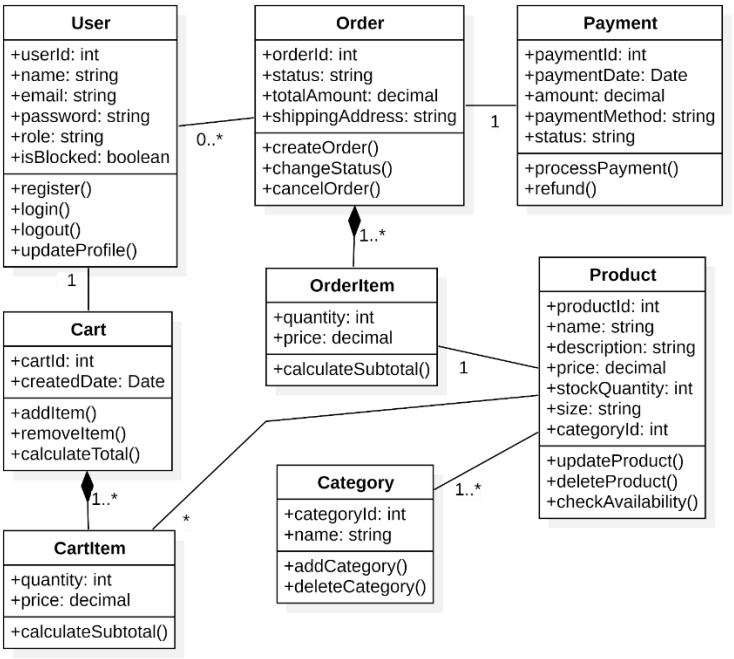

Рисунок 3.2– Діаграма класів

Клас CartItem є допоміжною сутністю, що представляє окрему позицію в кошику. Він містить інформацію про кількість товару та його ціну, а також метод для обчислення проміжної вартості. Кошик може містити множину таких елементів, що відображає зв’язок типу «один до багатьох».

Клас Order описує замовлення, яке формується на основі кошика. Він включає атрибути ідентифікатора, статусу, загальної суми та адреси доставки. Методи класу дозволяють створювати замовлення, змінювати його статус або

скасовувати. Один користувач може мати декілька замовлень, що відображає зв’язок «один до багатьох».

Клас OrderItem представляє окрему позицію в замовленні та містить інформацію про кількість і ціну товару. Він також має метод для обчислення вартості конкретної позиції. Замовлення складається з множини таких елементів, що забезпечує деталізацію структури замовлення.

Клас Product описує товар, представлений у системі. Він включає атрибути ідентифікатора, назви, опису, ціни, кількості на складі, розміру та належності до категорії. Методи класу забезпечують оновлення, видалення та перевірку доступності товару. Кожен товар належить до певної категорії.

Клас Category відповідає за класифікацію товарів. Він містить ідентифікатор та назву категорії, а також методи для додавання та видалення категорій. Зв’язок між категорією та товарами є типу «один до багатьох», оскільки одна категорія може включати декілька товарів.

Клас Payment відображає процес оплати замовлення. Він містить атрибути ідентифікатора платежу, дати, суми, способу оплати та статусу. Методи забезпечують обробку платежу та можливість повернення коштів. Кожне замовлення пов’язане з одним платежем, що реалізує зв’язок «один до одного».

Загалом, діаграма відображає логічно структуровану модель системи, побудовану на основі об’єктно-орієнтованого підходу. Вона забезпечує чітке розмежування відповідальностей між класами, що сприяє масштабованості, зручності супроводу та подальшому розширенню функціональності вебзастосунку.

ER-діаграма бази даних інформаційної системи інтернет-магазину відображає структуру сутностей, їх атрибути та взаємозв’язки між ними. Модель побудована відповідно до реляційного підходу та демонструє логічну організацію зберігання даних, необхідних для реалізації функціоналу електронної комерції, зокрема управління користувачами, товарами, замовленнями, кошиком і платежами.

Центральною сутністю системи є таблиця Users, яка містить інформацію про зареєстрованих користувачів. Вона включає первинний ключ user_id, що

однозначно ідентифікує кожного користувача, а також атрибути name, email, password_hash, role, is_blocked та created_at. Атрибут email має обмеження унікальності, що забезпечує неможливість реєстрації кількох облікових записів із однаковою електронною адресою. Поле password_hash призначене для збереження хешованого пароля з метою підвищення безпеки, тоді як role визначає рівень доступу користувача в системі. Атрибут is_blocked використовується для реалізації механізму блокування облікових записів, а created_at фіксує дату створення запису.

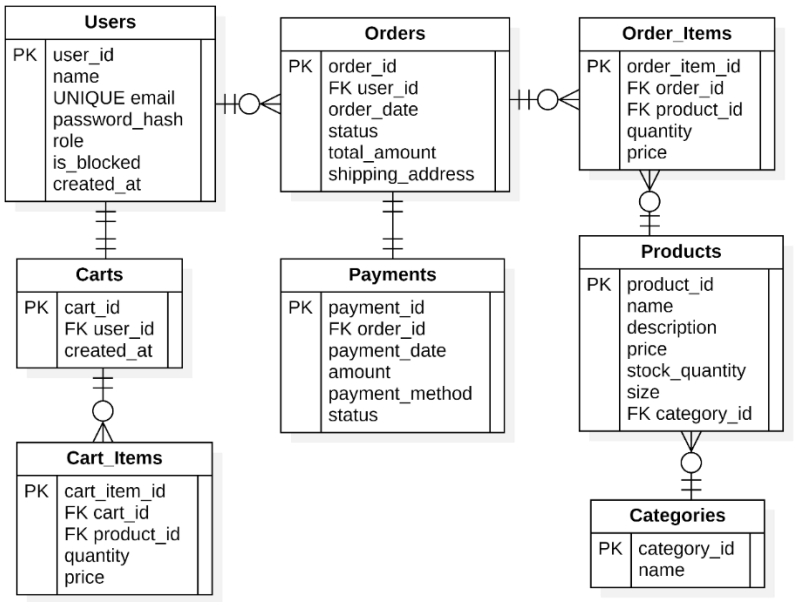

Рисунок 3.3 – ER-діаграма бази даних

Сутність Orders відображає інформацію про замовлення, створені користувачами. Первинним ключем є order_id, а зовнішній ключ user_id встановлює зв’язок із таблицею Users, що реалізує відношення типу «один до багатьох», оскільки один користувач може здійснити кілька замовлень. Таблиця також містить атрибути order_date, status, total_amount та shipping_address, які зберігають відповідно дату оформлення замовлення, його поточний стан, загальну суму та адресу доставки. Така структура дозволяє відстежувати життєвий цикл кожного замовлення від моменту створення до завершення.

Деталізація складу замовлення реалізована через сутність Order_Items, яка виступає проміжною таблицею для реалізації зв’язку «багато до багатьох» між замовленнями та товарами. Вона містить первинний ключ order_item_id, а також

зовнішні ключі order_id та product_id, що посилаються відповідно на Orders і Products. Атрибути quantity та price відображають кількість одиниць конкретного товару в межах замовлення та його ціну на момент придбання. Збереження ціни саме в цій таблиці забезпечує історичну коректність даних у випадку зміни вартості товару в майбутньому.

Сутність Products містить інформацію про товари, доступні для продажу. Первинним ключем є product_id. Інші атрибути включають name, description, price, stock_quantity та size, які характеризують основні властивості товару. Зовнішній ключ category_id пов'язує кожен товар із певною категорією, що реалізується через таблицю Categories. Таблиця Categories містить category_id як первинний ключ та name як назву категорії. Між Categories і Products встановлено зв'язок типу «один до багатьох», оскільки одна категорія може містити багато товарів, але кожен товар належить лише до однієї категорії.

Механізм тимчасового збереження обраних товарів реалізовано через сутність Carts, яка містить cart_id як первинний ключ, зовнішній ключ user_id та атрибут created_at. Зв'язок між Users і Carts також має характер «один до багатьох» або потенційно «один до одного» залежно від бізнес-логіки системи. Таблиця Cart_Items деталізує склад кошика та аналогічно до Order_Items реалізує зв'язок «багато до багатьох» між кошиком і товарами. Вона містить cart_item_id як первинний ключ, зовнішні ключі cart_id та product_id, а також атрибути quantity і price, що відображають кількість товару в кошику та його поточну ціну.

Сутність Payments призначена для зберігання інформації про фінансові операції, пов'язані із замовленнями. Первинний ключ payment_id однозначно ідентифікує кожен платіж, а зовнішній ключ order_id встановлює зв'язок із відповідним замовленням. Атрибути payment_date, amount, payment_method та status дозволяють фіксувати дату здійснення платежу, його суму, спосіб оплати та стан транзакції. Таким чином, між Orders і Payments встановлюється зв'язок типу «один до багатьох» або «один до одного» залежно від політики здійснення оплат.

У цілому представлена ER-діаграма демонструє нормалізовану структуру бази даних із чітко визначеними первинними та зовнішніми ключами, що

забезпечують цілісність даних і коректність зв’язків між сутностями. Архітектура моделі відповідає типовим вимогам до інформаційних систем електронної комерції та дозволяє ефективно реалізувати основні бізнес-процеси інтернет-магазину, включаючи управління користувачами, каталогом товарів, кошиком, замовленнями та оплатами.

3.2 Проєктування майбутнього інтерфейсу користувача

Сторінка Login page призначена для входу користувача в системі інтернет-магазину. У верхній частині сторінки розташований header із логотипом та навігацією (Каталог, Кошик, Профіль). Центральна частина містить форму входу: поле введення email, поле введення пароля та кнопку «Увійти». Поруч розміщена кнопка «Реєстрація» для переходу до створення акаунту, у випадку якщо користувач ще не зареєстрований. У нижній частині сторінки знаходиться footer із контактами та посиланням на сторінку з політикою конфіденційності. Призначенням цієї сторінки є забезпечення доступу до повного функціоналу магазину лише авторизованим користувачам.

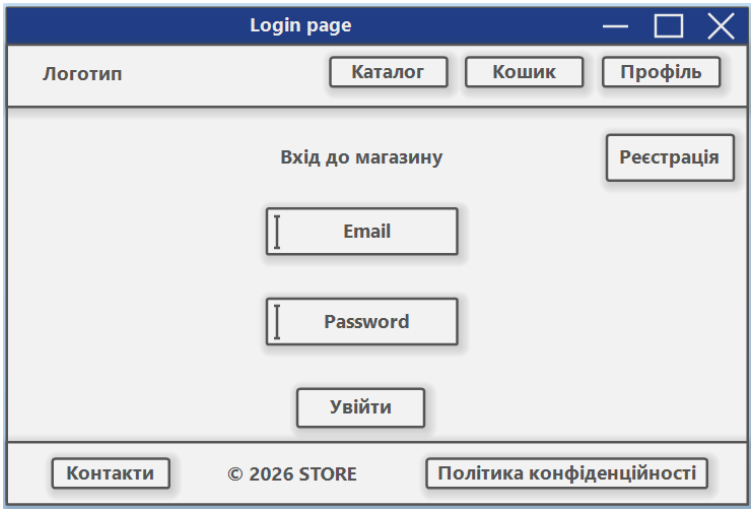

Рисунок 3.4 – Сторінка входу

Сторінка каталогу призначена для перегляду асортименту товарів інтернет-магазину та виконання пошуку необхідних позицій. У верхній частині сторінки розташований header із логотипом та навігаційним меню (Каталог, Кошик, Вийти), що забезпечує швидкий перехід між основними розділами системи.

Під header розміщена панель фільтрації та пошуку, яка містить випадаючий список категорій для обмеження відображення товарів певною групою, а також поле пошуку, що дозволяє знаходити товари за назвою. Центральна частина сторінки представлена у вигляді сітки товарів. Кожен товар відображається у вигляді картки, що містить зображення товару, його назву та ціну. Під кожною карткою розміщені кнопки «Деталі» для переходу на сторінку детального перегляду товару та «В кошик» для швидкого додавання товару до кошика.

У нижній частині сторінки знаходиться footer із контактною інформацією та посиланням на сторінку політики конфіденційності. Призначенням цієї сторінки є забезпечення зручного перегляду каталогу товарів, навігації між категоріями та формування кошика покупок.

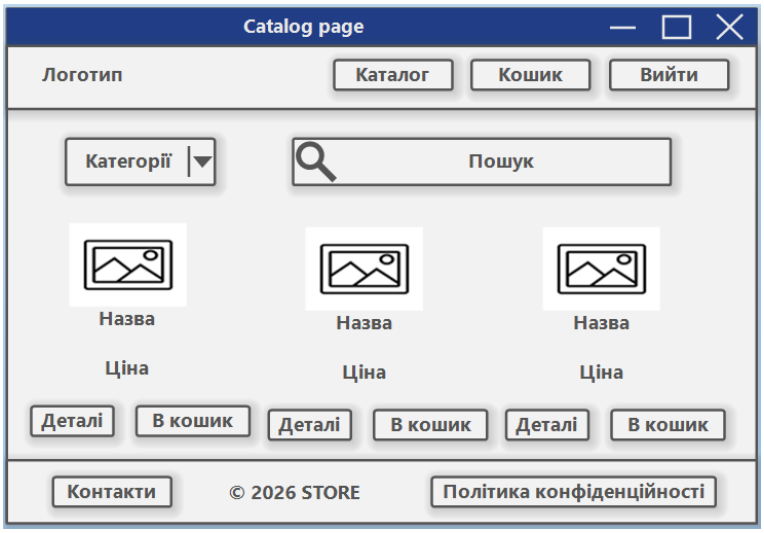

Рисунок 3.5 – Каталог товарів

Сторінка деталей товару призначена для детального перегляду інформації про обраний товар перед його додаванням до кошика. У верхній частині сторінки розташований header із логотипом і навігаційним меню (Каталог, Кошик, Вийти), що дозволяє користувачу повернутися до каталогу або перейти до кошика.

Під header розміщена кнопка «Назад до каталогу», яка забезпечує швидке повернення до списку товарів. Центральна частина сторінки поділена на два блоки: ліворуч відображається зображення товару, а праворуч — інформаційний блок із назвою товару, його ціною та текстовим описом характеристик. Під інформаційним

блоком розміщена кнопка «Додати в кошик», яка дозволяє користувачу додати товар до списку покупок.

У нижній частині сторінки розташований footer із контактною інформацією та посиланням на політику конфіденційності. Призначенням сторінки є надання користувачу повної інформації про товар для прийняття рішення щодо його придбання.

Сторінка кошику призначена для перегляду товарів, доданих користувачем до кошика, а також керування їх кількістю перед оформленням замовлення. У верхній частині сторінки знаходиться header із логотипом та навігаційним меню (Каталог, Кошик, Вийти), що забезпечує швидку навігацію між розділами магазину.

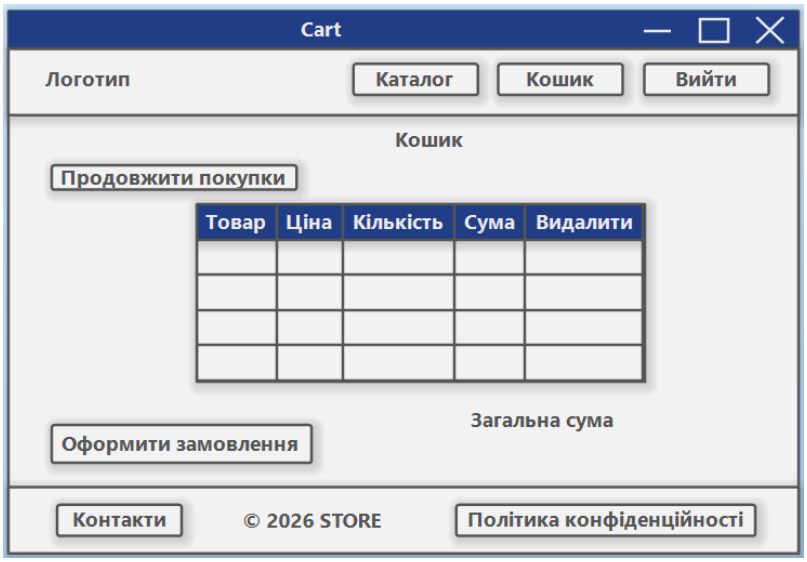

Рисунок 3.6 – Сторінка кошику

Під header розташована кнопка «Продовжити покупки», яка дозволяє користувачу повернутися до каталогу товарів. Центральна частина сторінки представлена у вигляді таблиці кошика, що містить інформацію про додані товари: назву товару, ціну, кількість, суму по кожній позиції та кнопку видалення товару з кошика. Користувач має можливість змінювати кількість товарів, що автоматично впливає на загальну суму замовлення.

Під таблицею розміщений блок із відображенням загальної вартості всіх товарів у кошику, а також кнопка «Оформити замовлення», яка переводить

користувача до наступного етапу покупки. У нижній частині сторінки знаходиться footer із контактною інформацією та посиланням на політику конфіденційності. Призначенням сторінки є надання користувачу можливості перевірити склад замовлення та виконати перехід до його оформлення.

Сторінка оформлення замовлення призначена для введення користувачем даних доставки та вибору способу оплати перед підтвердженням замовлення. У верхній частині сторінки розташований header із логотипом та навігаційним меню (Каталог, Кошик, Вийти), що забезпечує доступ до основних розділів системи.

Під header знаходиться кнопка «До кошику», яка дозволяє повернутися до попереднього етапу редагування замовлення. Центральна частина сторінки містить форму оформлення замовлення, що складається з полів введення ПІБ отримувача, номера телефону, міста та адреси доставки. Нижче розміщений блок вибору способу оплати, який пропонує користувачу два варіанти: оплата карткою або оплата при отриманні.

У нижній частині форми розташована кнопка «Підтвердити», яка завершує процес введення даних та переводить користувача до етапу оплати або підтвердження замовлення. Footer сторінки містить контактну інформацію та посилання на політику конфіденційності. Призначенням цієї сторінки є збір необхідних даних для доставки товару та завершення процесу оформлення покупки.

Висновки до розділу 3

У межах даного розділу було здійснено ґрунтовний аналіз інструментарію, моделей та методів, що застосовуються при розробці вебзастосунку для продажу одягу. Особливу увагу приділено обґрунтуванню вибору технологічного стеку, який включає PHP, MySQL, HTML та CSS. Встановлено, що використання мови PHP є доцільним для реалізації серверної логіки завдяки її широкому застосуванню у веброзробці, наявності значної кількості бібліотек та підтримки інтеграції з базами даних. Система керування базами даних MySQL забезпечує надійне зберігання та ефективну обробку структурованих даних, що є критично важливим

для функціонування інтернет-магазину. У свою чергу, HTML і CSS виступають основою для створення зрозумілого, структурованого та візуально привабливого користувацького інтерфейсу, а також забезпечують адаптивність вебзастосунку до різних типів пристроїв.

У ході дослідження було розроблено та проаналізовано сценарії використання системи, які дозволили формалізувати взаємодію між користувачами та функціональними компонентами вебзастосунку. Це дало змогу чітко визначити ролі користувачів, послідовність виконання основних операцій, а також ключові бізнес-процеси, що відбуваються в системі. Використання сценаріїв сприяло кращому розумінню вимог до функціональності та дозволило уникнути неоднозначностей на етапі проєктування.

Побудова діаграми класів забезпечила структуроване представлення програмної моделі системи. У межах цієї моделі було визначено основні класи, їх атрибути, методи та взаємозв’язки, що відображають логіку функціонування вебзастосунку. Це дозволило реалізувати принципи об’єктно-орієнтованого програмування, зокрема інкапсуляцію та розподіл відповідальностей між компонентами системи, що позитивно впливає на її масштабованість та супроводжуваність.

Додатково було розроблено ER-діаграму бази даних, яка відображає логічну структуру зберігання інформації та зв’язки між сутностями. Це забезпечило цілісність даних, узгодженість структури таблиць і можливість ефективного виконання операцій обробки інформації. Завдяки цьому було закладено основу для створення надійної та продуктивної бази даних, що відповідає вимогам предметної області.

Важливим етапом проєктування стало створення mockups (прототипів інтерфейсу), які дозволили візуалізувати майбутній вигляд вебзастосунку ще до початку його реалізації. Це дало змогу оцінити зручність навігації, логіку розташування елементів інтерфейсу, а також відповідність принципам юзабіліті. Завдяки цьому було зменшено ризик виникнення помилок у дизайні та підвищено якість користувацького досвіду.

Таким чином, у результаті виконаної роботи сформовано цілісне уявлення про архітектуру, функціональні можливості та структуру вебзастосунку. Проведений аналіз інструментарію та моделей дозволяє зробити висновок про доцільність обраних рішень і створює надійну основу для подальшої реалізації системи. Отримані результати сприяють підвищенню ефективності процесу розробки, зменшенню кількості помилок на наступних етапах та забезпеченню високої якості кінцевого програмного продукту.

4 РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ПРОДАЖУ ОДЯГУ

4.1 Створення таблиць бази даних

Організація даних у сучасному цифровому світі базується на використанні спеціалізованого програмного забезпечення – систем керування базами даних, які забезпечують структурування та збереження інформації на фізичних носіях серверів. Найбільш затребуваними є реляційні бази даних, що використовують мову SQL для взаємодії з інформаційними масивами, оскільки вони дозволяють перетворити хаотичні записи на логічно впорядковані структури, доступні для швидкого пошуку та аналізу. Практичне розгортання таких систем часто здійснюється за допомогою контейнеризації через Docker або шляхом прямої інсталяції сервера, наприклад PostgreSQL, а взаємодія з базою відбувається через графічні інтерфейси на кшталт PG Admin або універсальні інструменти типу DBeaver. Для встановлення зв'язку між клієнтським додатком і сервером необхідно чітко конфігурувати параметри підключення, включаючи адресу хоста (LocalHost для локальних систем), програмний порт (стандартно 5432 для Postgres), а також аутентифікаційні дані користувача.

Фундаментальним елементом архітектури SQL є таблиця, яка виступає логічним простором для зберігання однотипних за змістом об'єктів, де кожен запис представлений рядком, а його атрибути – колонками. Процес проектування бази даних починається з використання оператора CREATE TABLE, який визначає схему майбутнього об'єкта, де для кожної колонки встановлюється суворий тип даних, наприклад BIGINT для цілих чисел або VARCHAR для текстових рядків із лімітованою довжиною. Важливу роль відіграють обмеження цілісності, такі як PRIMARY KEY, що гарантує унікальність кожного запису, та NOT NULL, яке забороняє відсутність даних у критично важливих полях. Окрім первинного створення структури, розробник має можливість динамічно змінювати її за допомогою команди ALTER TABLE, що дозволяє додавати нові колонки або правила (constraints) без видалення наявної інформації, що є критично важливим для систем, які вже перебувають в експлуатації.

Маніпуляція даними всередині таблиць охоплює операції створення, оновлення та видалення записів. Додавання нової інформації реалізується через команду INSERT INTO, де необхідно чітко дотримуватися відповідності між переліком колонок та порядком значень у блоці VALUES. Модифікація існуючих даних виконується оператором UPDATE, а видалення – командою DELETE, причому в обох випадках критично важливим є використання конструкції WHERE. Це ключове слово дозволяє задавати умови фільтрації, запобігаючи випадковій зміні або знищенню всього масиву даних у таблиці. Гнучкість запитів забезпечується логічними операторами AND та OR, які дають змогу формувати складні критерії вибору об'єктів за декількома ознаками одночасно.

Найпотужнішим інструментом SQL є механізм вибірки даних SELECT, який дозволяє не лише отримувати окремі колонки або всі записи за допомогою символу зірочки, але й об'єднувати інформацію з різних таблиць. Реляційний характер баз даних проявляється через встановлення зв'язків типу «відношення» за допомогою зовнішніх ключів – FOREIGN KEY. Це дозволяє пов'язувати записи, наприклад, ідентифікувати користувача, якому належить конкретна фінансова операція, забезпечуючи посилальну цілісність, при якій неможливо створити запис із посиланням на неіснуючий об'єкт в іншій таблиці. Для об'єднання таких таблиць у межах одного запиту застосовується оператор JOIN. Зокрема, INNER JOIN повертає лише ті рядки, для яких знайдено відповідність в обох таблицях, тоді як зовнішні об'єднання (LEFT, RIGHT або FULL OUTER JOIN) дозволяють включати у результат навіть ті записи, що не мають пари, заповнюючи відсутні значення маркером NULL.

Для проведення аналітичних розрахунків у SQL передбачено агрегатні функції, такі як SUM для підрахунку сум, MAX та MIN для пошуку екстремальних значень, а також AVG для обчислення середнього арифметичного. Коли виникає потреба отримати статистику в розрізі певних категорій, наприклад сумарні витрати окремого користувача, використовується конструкція GROUP BY, яка групує записи за вказаною колонкою перед застосуванням агрегатної функції. Якщо ж необхідно відфільтрувати результати вже після проведення групування,

стандартний оператор WHERE не може бути застосований, тому розробники використовують спеціалізоване слово HAVING. Такий комплексний підхід до побудови запитів дозволяє створювати складні звіти та забезпечувати функціонування масштабних інформаційних систем із високим рівнем автоматизації обробки даних.

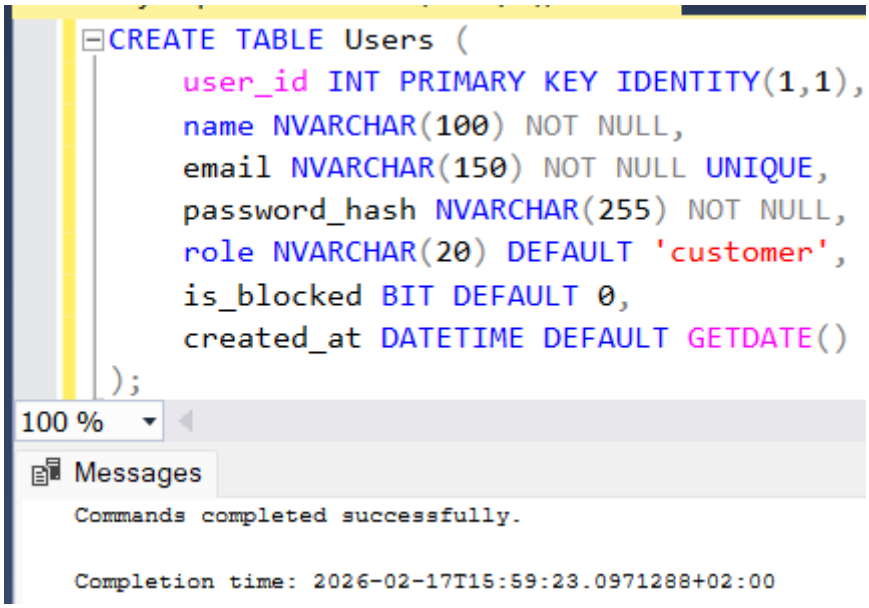

Рисунок 4.1 – Створення таблиці користувачів

Створення структури бази даних розпочато з проєктування таблиці користувачів, яка виконує ключову роль у системі, оскільки забезпечує зберігання облікових записів і пов’язаних із ними атрибутів. На першому етапі було визначено поле ідентифікатора користувача, яке реалізовано як ціле число з автоматичним збільшенням значення для кожного нового запису. Такий підхід гарантує унікальність кожного користувача та спрощує організацію зв’язків із іншими таблицями. Далі було створено поле для збереження імені користувача, яке є обов’язковим для заповнення, що дозволяє однозначно ідентифікувати особу в межах системи. Наступним кроком стало додавання поля електронної пошти, для якого встановлено обмеження унікальності, що виключає можливість реєстрації кількох облікових записів із однаковою адресою. Для забезпечення безпеки передбачено окреме поле для збереження хешованого пароля, що виключає зберігання паролів у відкритому вигляді. Також реалізовано поле ролі користувача,

яке автоматично набуває значення за замовчуванням, що дозволяє розмежовувати права доступу без додаткових дій під час створення запису. Додатково введено логічне поле блокування, яке визначає, чи має користувач доступ до системи, причому за замовчуванням доступ дозволено. Завершальним елементом цієї таблиці стало поле дати створення запису, яке автоматично фіксує момент реєстрації користувача, що важливо для аудиту та відстеження активності.

Наступним етапом було створення таблиці категорій, яка використовується для логічного групування товарів. Спочатку визначено унікальний ідентифікатор категорії, який, аналогічно до таблиці користувачів, автоматично збільшується для кожного нового запису. Це забезпечує зручність у роботі зі зв'язками між таблицями. Після цього додано поле для назви категорії, яке є обов'язковим, оскільки кожна категорія повинна мати зрозуміле текстове представлення. Така структура дозволяє централізовано керувати переліком категорій і використовувати їх для класифікації товарів.

Завершальним етапом стало створення таблиці товарів, яка є основною для зберігання інформації про продукцію. У першу чергу визначено поле унікального ідентифікатора товару з автоматичною генерацією значень, що забезпечує унікальність кожного запису. Далі було передбачено поле назви товару, яке є обов'язковим для заповнення та використовується для відображення продукції користувачам. Окрім цього, створено поле опису, яке дозволяє зберігати розширену інформацію про товар без обмеження довжини, що є важливим для детального представлення характеристик. Для зберігання вартості товару використано числове поле з фіксованою точністю, що дозволяє коректно обробляти грошові значення. Також передбачено поле кількості товару на складі, яке є обов'язковим і використовується для контролю наявності продукції. Додатково введено поле розміру, яке може бути використане для специфічних характеристик товару, наприклад, одягу чи взуття. Наступним кроком реалізовано поле, яке відповідає за належність товару до певної категорії. Це поле пов'язане із таблицею категорій через зовнішній ключ, що забезпечує цілісність даних і гарантує, що кожен товар може бути віднесений лише до існуючої категорії. Такий зв'язок

дозволяє ефективно організувати структуру бази даних, забезпечити узгодженість інформації та спростити виконання запитів для отримання впорядкованих даних.

Проєктування наступного етапу структури бази даних було спрямоване на реалізацію механізмів роботи з кошиком користувача та оформленням замовлень. Насамперед було створено таблицю, призначену для зберігання інформації про кошики. У процесі її формування визначено поле унікального ідентифікатора кошика, яке автоматично генерується для кожного нового запису, що забезпечує однозначну ідентифікацію кожного екземпляра кошика в системі. Далі було передбачено поле для збереження ідентифікатора користувача, якому належить кошик. Це поле не є автономним, а пов’язане з відповідним полем у таблиці користувачів, що дозволяє встановити логічний зв’язок між обліковим записом і створеним кошиком. Таким чином забезпечується можливість зберігати індивідуальні кошики для кожного користувача. Додатково було введено поле дати створення кошика, значення якого автоматично встановлюється на момент додавання запису, що дає змогу відстежувати час ініціалізації кошика та аналізувати поведінку користувачів.

Наступним кроком стало створення таблиці елементів кошика, яка деталізує вміст кожного кошика. На початку визначено унікальний ідентифікатор кожного запису, що забезпечує незалежність зберігання кожної позиції товару. Після цього було додано поле, яке посилається на конкретний кошик, дозволяючи об’єднати всі позиції в межах одного кошика. Також передбачено поле для ідентифікації товару, що встановлює зв’язок із таблицею товарів і забезпечує доступ до інформації про відповідний продукт. Обов’язковим елементом є поле кількості товару, яке визначає, скільки одиниць певного продукту додано до кошика. Для коректного обліку вартості було реалізовано поле ціни, яке фіксує вартість товару на момент додавання до кошика, що дозволяє уникнути невідповідностей у разі подальшої зміни ціни в основній таблиці товарів. Всі зв’язки в цій таблиці реалізовані через зовнішні ключі, що гарантує цілісність даних і виключає можливість існування записів без відповідних кошиків або товарів.

Завершальним етапом стало створення таблиці замовлень, яка використовується для фіксації фактів придбання товарів користувачами. У її структурі визначено унікальний ідентифікатор замовлення з автоматичною генерацією значень, що дозволяє однозначно ідентифікувати кожне замовлення. Далі було додано поле для збереження ідентифікатора користувача, яке пов’язане з таблицею користувачів, забезпечуючи встановлення належності замовлення конкретному клієнту. Особливу увагу приділено полю статусу замовлення, яке є обов’язковим і використовується для відображення поточного стану обробки, наприклад, створено, обробляється або виконано. Також реалізовано поле загальної суми замовлення, що дозволяє зберігати агреговану вартість усіх товарів, включених до нього. Для забезпечення можливості доставки передбачено поле адреси, в якому зберігається інформація про місце отримання товару. Як і в попередніх таблицях, додано поле дати створення запису, яке автоматично фіксує момент оформлення замовлення. Завдяки встановленню зовнішнього ключа забезпечується узгодженість даних між таблицями та підтримується цілісність всієї інформаційної системи.

4.2 Підключення БД до серверу

Процес підключення бази даних до серверної частини застосунку було реалізовано з використанням універсального механізму доступу до баз даних, який забезпечує стандартизовану взаємодію з різними системами керування базами даних. На початковому етапі було визначено основні параметри підключення, зокрема ім’я сервера, на якому розміщено базу даних, а також назву самої бази даних, до якої необхідно встановити з’єднання. Додатково передбачено змінні для облікових даних користувача, які можуть використовуватися для автентифікації при підключенні, хоча в даному випадку використовується спрощений варіант доступу.

Наступним кроком стало створення об’єкта підключення, який виконує роль посередника між серверною логікою та базою даних. Для цього сформовано спеціальний рядок підключення, що містить інформацію про тип драйвера, адресу

сервера та назву бази даних. У цьому рядку також передбачено параметр, який дозволяє встановлювати з'єднання навіть у випадку використання самопідписаного сертифіката, що є актуальним для локального середовища розробки. Після формування рядка підключення ініціалізується відповідний об'єкт, який і забезпечує подальшу взаємодію з базою даних, включаючи виконання запитів та отримання результатів.

Особливу увагу приділено обробці можливих помилок під час встановлення з'єднання. Для цього використано механізм перехоплення виняткових ситуацій, який дозволяє контролювати процес підключення та реагувати на збої. У разі виникнення помилки формується повідомлення, що містить інформацію про причину невдалого підключення, після чого виконання програми примусово припиняється. Такий підхід дозволяє уникнути подальшої роботи системи в некоректному стані.

db.php

config > db.php

```php
<?php
$serverName = "DESKTOP-60I2LBM\SQLEXPRESS";
$database = "Store";
$username = "";
$password = "";

try {
    $pdo = new PDO(
        "sqlsrv:Server=$serverName;Database=$database;TrustServerCertificate=true",
        $username,
        $password
    );

    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

} catch (PDOException $e) {
    die("Помилка підключення до БД: " . $e->getMessage());
}
```

Рисунок 4.2 – Підключення серверної частини

Додатково після успішного встановлення з'єднання було налаштовано режим обробки помилок для об'єкта доступу до бази даних. Зокрема, обрано режим, при якому всі помилки під час виконання запитів будуть генерувати винятки. Це забезпечує більш надійний контроль за виконанням операцій із базою даних, оскільки дозволяє централізовано обробляти помилки та спрощує процес налагодження і тестування системи.

У результаті виконаних дій було створено надійний механізм підключення до бази даних, який забезпечує стабільну взаємодію серверної частини застосунку з інформаційним сховищем, контроль помилок і можливість подальшого розширення функціональності.

4.3 Реалізація функціоналу реєстрації та входу

Реалізація механізму реєстрації користувача була виконана у вигляді окремого серверного сценарію, який забезпечує як обробку введених даних, так і взаємодію з базою даних. На початковому етапі відбувається підключення файлу конфігурації, що містить параметри доступу до бази даних і створює об’єкт для виконання запитів. Це дозволяє забезпечити централізоване керування підключенням і повторне використання налаштувань у різних частинах застосунку.

Після цього ініціалізується змінна для збереження повідомлення про помилку, яка використовується для інформування користувача у випадку некоректного введення даних або виникнення інших проблем. Основна логіка обробки виконується лише за умови, що форма була відправлена методом передавання даних, призначеним для надсилання інформації на сервер. Такий підхід дозволяє відокремити початкове відображення сторінки від обробки введених користувачем даних.

Далі здійснюється отримання значень, введених у форму. Текстові дані очищуються від зайвих пробілів на початку та в кінці, що запобігає помилкам, пов’язаним із некоректним форматом введення. Пароль і підтвердження пароля отримуються без змін для подальшої перевірки їх відповідності. Після отримання даних виконується поетапна валідація. Спочатку перевіряється, чи всі обов’язкові поля заповнені, що гарантує наявність мінімально необхідної інформації. Далі здійснюється перевірка правильності формату електронної пошти, що дозволяє відсіяти некоректні значення. Наступним кроком є перевірка довжини пароля з метою забезпечення базового рівня безпеки. Після цього перевіряється співпадіння пароля та його підтвердження, що виключає помилки користувача під час введення.

У разі успішного проходження первинної перевірки виконується додаткова перевірка наявності користувача з таким самим електронним адресом у базі даних. Для цього формується запит, який визначає кількість записів із заданим значенням електронної пошти. Якщо виявляється, що такий користувач уже існує, формується відповідне повідомлення про помилку, що запобігає дублюванню облікових записів.

Якщо ж перевірка не виявляє конфліктів, виконується етап безпосереднього створення нового користувача. Перед збереженням пароля застосовується механізм хешування, який перетворює його у захищений формат. Це забезпечує неможливість відновлення початкового значення пароля навіть у разі несанкціонованого доступу до бази даних. Після цього формується запит на додавання нового запису до таблиці користувачів із передачею відповідних параметрів. У результаті виконання цього запиту в базі даних з'являється новий користувач із зазначеними атрибутами.

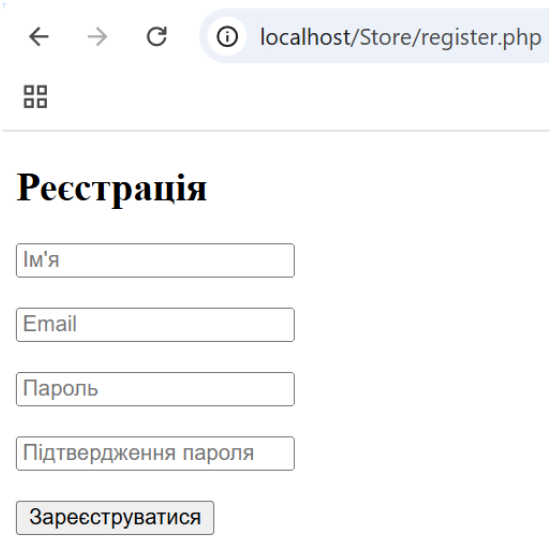

Рисунок 4.3 – Результат реалізації функціоналу реєстрації

Після успішного завершення операції виконується перенаправлення користувача на сторінку входу до системи із передачею ознаки успішної реєстрації. Це дозволяє відобразити відповідне повідомлення або змінити поведінку інтерфейсу. У разі виникнення будь-якої помилки на попередніх етапах користувач отримує повідомлення, яке виводиться безпосередньо на сторінці реєстрації.

Завершальним компонентом розробки цієї частини проєкту є розмітка сторінки, яка формує інтерфейс для введення даних. Вона містить поля для введення імені, електронної пошти, пароля та підтвердження пароля, а також елемент керування для відправлення форми. У разі наявності помилки відповідне повідомлення відображається у візуально виділеному вигляді, що забезпечує зручність взаємодії користувача із системою. Таким чином, реалізований механізм поєднує перевірку даних, забезпечення безпеки та зручний інтерфейс для реєстрації нових користувачів.

Реалізація механізму автентифікації користувача була виконана у вигляді окремого серверного сценарію, який забезпечує як відображення форми входу, так і обробку введених облікових даних. На початковому етапі передбачено можливість відображення інформаційного повідомлення про успішну реєстрацію користувача. Таке повідомлення формується на основі спеціального параметра, що передається разом із запитом, і слугує для інформування користувача про необхідність виконання входу після створення облікового запису. Це дозволяє логічно пов’язати процеси реєстрації та авторизації в єдиний користувацький сценарій.

Далі формується базовий інтерфейс сторінки входу, який містить поля для введення електронної пошти та пароля, а також елемент керування для відправлення форми. На цьому етапі сторінка виконує виключно функцію збору даних від користувача без їх обробки.

У процесі вдосконалення функціональності було реалізовано повноцінну обробку автентифікації із використанням сесійного механізму. На початку виконання сценарію ініціалізується сесія, що дозволяє зберігати дані користувача між різними запитами та забезпечує підтримку стану авторизації. Після цього підключається конфігураційний файл, який встановлює з’єднання з базою даних, що є необхідним для перевірки облікових даних.

Як і у випадку з реєстрацією, вводиться змінна для збереження повідомлення про помилку, яке буде відображено користувачу у разі некоректного введення даних або невдалої спроби входу. Основна логіка виконується лише після

відправлення форми, що дозволяє відокремити етап відображення інтерфейсу від етапу обробки даних.

Після отримання даних із форми виконується їх попередня обробка, зокрема видалення зайвих пробілів у полі електронної пошти. Далі формується запит до бази даних, спрямований на пошук користувача з указаною електронною адресою. У результаті виконання цього запиту отримується запис користувача, якщо такий існує.

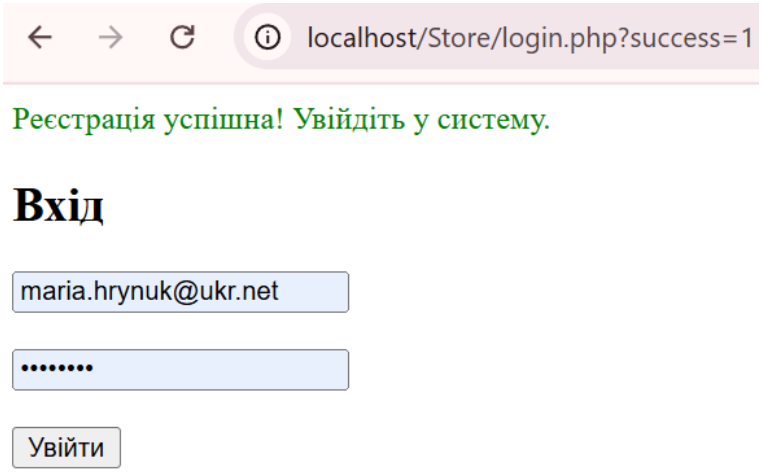

Рисунок 4.4 – Результат реалізації функціоналу автентифікації

Наступним етапом є перевірка автентичності введеного пароля. Для цього використовується спеціальний механізм порівняння, який співставляє введене значення з раніше збереженим у базі даних хешованим паролем. Якщо перевірка проходить успішно, це означає, що користувач ввів коректні облікові дані. У такому випадку в сесії зберігаються основні ідентифікаційні параметри користувача, зокрема його унікальний ідентифікатор та ім’я. Це дозволяє системі надалі розпізнавати користувача та надавати йому доступ до захищених розділів застосунку.

Після успішної автентифікації виконується перенаправлення користувача на сторінку каталогу товарів, що фактично означає завершення процесу входу та перехід до основної функціональності системи. У випадку, якщо користувач із вказаною електронною адресою не знайдений або введений пароль не відповідає

збереженому значенню, формується повідомлення про помилку, яке інформує про невірність облікових даних.

Завершальним елементом є оновлена структура сторінки, яка поєднує в собі як інтерфейс для введення даних, так і механізм відображення повідомлень про помилки. У разі виникнення помилки відповідний текст виводиться у візуально виділеному вигляді, що підвищує зручність взаємодії користувача із системою. Таким чином, реалізований підхід забезпечує надійний процес автентифікації, збереження стану користувача та інтеграцію з іншими компонентами вебзастосунку.

4.4 Створення сторінки каталогу

Реалізовано функціональність сторінки каталогу інтернет-магазину одягу, яка забезпечує авторизованим користувачам можливість перегляду асортименту товарів, виконання пошуку, фільтрації за категоріями, додавання товарів до кошика та управління списком обраного. Логіка роботи сторінки побудована відповідно до клієнт-серверної архітектури, де серверна частина відповідає за обробку запитів, взаємодію з базою даних і формування динамічного вмісту сторінки, а клієнтська частина забезпечує відображення інформації та взаємодію користувача з інтерфейсом.

На початковому етапі відбувається ініціалізація сесії користувача, що дозволяє зберігати інформацію про його авторизацію та персональні дані протягом усього часу роботи із системою. Після цього здійснюється підключення до конфігураційного файлу, який містить параметри з’єднання з базою даних. Такий підхід забезпечує централізоване управління доступом до СКБД та підвищує безпечність системи.

Далі реалізовано механізм захисту сторінки від несанкціонованого доступу. Якщо користувач не авторизований, система автоматично перенаправляє його на сторінку входу. Це забезпечує контроль доступу до функціоналу каталогу, зокрема можливості додавання товарів до кошика та списку обраного.

Після перевірки авторизації система отримує параметри пошуку та обраної категорії з HTTP-запиту. Якщо користувач ввів текст для пошуку або обрав конкретну категорію товарів, ці параметри передаються на сервер і використовуються для формування відповідного запиту до бази даних. Формування запиту здійснюється динамічно: початково обираються всі записи з таблиці товарів, після чого, за наявності введених критеріїв, додаються умови фільтрації. Таким чином забезпечується можливість одночасного застосування кількох параметрів — текстового пошуку та категорії. Для запобігання SQL-ін’єкціям використовується механізм підготовлених запитів із параметрами, що гарантує безпечну обробку введених користувачем даних.

Отримані з бази даних результати сортуються у зворотному порядку додавання, що дозволяє відображати найновіші товари першими. Після виконання запиту формується масив товарів, який передається до частини відображення.

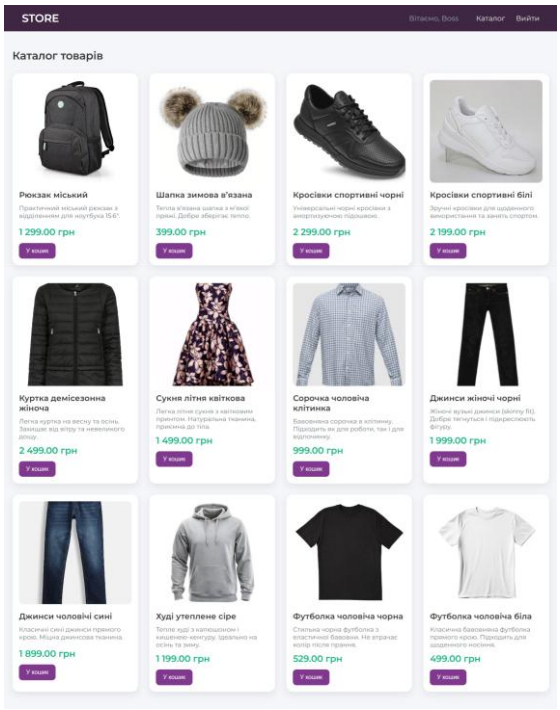

Рисунок 4.5 – Каталог товарів

Паралельно здійснюється окремий запит до таблиці обраних товарів для визначення, які саме позиції вже додані користувачем до списку бажаного. Це дозволяє інтерфейсу коректно відобразити стан кожного товару та показати відповідний індикатор.

У частині клієнтського інтерфейсу формується навігаційна панель із логотипом магазину, формою пошуку та меню користувача. Значення поля пошуку та вибраної категорії зберігаються після відправлення форми, що покращує зручність користування системою. Основна частина сторінки містить перелік товарів, кожен з яких відображається у вигляді окремої картки. Лістинг програмного коду сторінки каталогу наведено в додатку.

Картка товару містить зображення, назву, короткий опис, інформацію про наявність, ціну та кнопки взаємодії. Назва, опис і зображення виводяться з урахуванням екранування спеціальних символів, що забезпечує захист від XSS-атак. Система додатково аналізує кількість товару на складі: якщо залишок обмежений, користувачеві відображається відповідне повідомлення, а у разі відсутності товару кнопка додавання до кошика блокується. Такий механізм запобігає створенню замовлень на недоступні позиції.

Для кожного товару реалізовано можливість переходу до детальної сторінки з розширеною інформацією. Також передбачено інтерактивний елемент додавання до списку обраного, який працює через окремий обробник і дозволяє користувачеві швидко керувати вподобаними позиціями без переходу на інші сторінки.

Додавання товару до кошика здійснюється через передачу ідентифікатора товару на серверну частину, де відповідний обробник фіксує дію користувача. Уся взаємодія між клієнтською та серверною частинами організована через HTTP-запити, що забезпечує узгоджену роботу системи.

Загалом сторінка каталогу реалізує повний цикл взаємодії користувача з асортиментом продукції: від пошуку й перегляду до формування наміру придбати товар. Архітектурно модуль інтегрований із системою авторизації, базою даних та підсистемою керування кошиком і списком обраного, що забезпечує цілісність, безпечність і логічну завершеність функціонування інтернет-магазину.

4.5 Реалізація кошику

Створено повний механізм функціонування кошика інтернет-магазину, що охоплює як відображення обраних товарів користувачем, так і процес додавання

нових позицій до кошика із збереженням даних у базі даних. Архітектурно рішення побудоване відповідно до принципів клієнт-серверної взаємодії, де серверна частина відповідає за обробку запитів, перевірку коректності даних і взаємодію з таблицями бази даних, а клієнтська частина забезпечує зручне відображення інформації та можливість редагування вмісту кошика.

На початковому етапі відбувається ініціалізація сесії користувача, що дозволяє ідентифікувати його в межах системи та отримати доступ до персоналізованих даних. Після цього здійснюється підключення до конфігураційного файлу з параметрами з'єднання з базою даних. Далі реалізується механізм перевірки авторизації: якщо користувач не має активної сесії, він автоматично перенаправляється на сторінку входу. Таким чином забезпечується обмеження доступу до функціоналу кошика виключно для зареєстрованих і авторизованих осіб.

Після підтвердження авторизації система отримує ідентифікатор користувача із сесії та формує запит до бази даних для отримання переліку товарів, доданих до кошика. Використовується з'єднання таблиці кошика з таблицею товарів, що дозволяє отримати не лише технічні ідентифікатори записів, а й назву товару, його актуальну ціну та кількість, обрану користувачем. Такий підхід забезпечує цілісність даних і гарантує, що відображається саме актуальна інформація з бази даних. Запит виконується з використанням підготовлених параметризованих конструкцій, що підвищує рівень безпеки та запобігає можливості SQL-ін'єкцій [14].

Отримані результати формують масив позицій кошика, який використовується для побудови інтерфейсу сторінки. У клієнтській частині відображається навігаційна панель із логотипом магазину та посиланнями на основні розділи системи. Додатково виводиться ім'я користувача, отримане із сесії, що персоналізує взаємодію з системою.

Основний блок сторінки містить таблицю з переліком товарів у кошику. Якщо кошик порожній, система виводить відповідне повідомлення, що інформує користувача про відсутність обраних позицій. У випадку наявності товарів

формується таблична структура, у якій для кожної позиції відображається назва, ціна, кількість, підсумкова вартість та елемент керування для видалення. Під час виведення текстових даних застосовується екранування спеціальних символів, що забезпечує захист від міжсайтових скриптових атак.

Для кожної позиції реалізовано можливість зміни кількості товару. Це здійснюється через окрему форму, яка передає ідентифікатор запису кошика та нове значення кількості до відповідного обробника. Таким чином користувач може оперативно коригувати замовлення без необхідності повторного додавання товарів. Підсумкова вартість кожної позиції обчислюється шляхом множення ціни на кількість, після чого здійснюється накопичення загальної суми замовлення. Загальна сума відображається окремим блоком під таблицею, що забезпечує наочність фінансового результату перед оформленням замовлення.

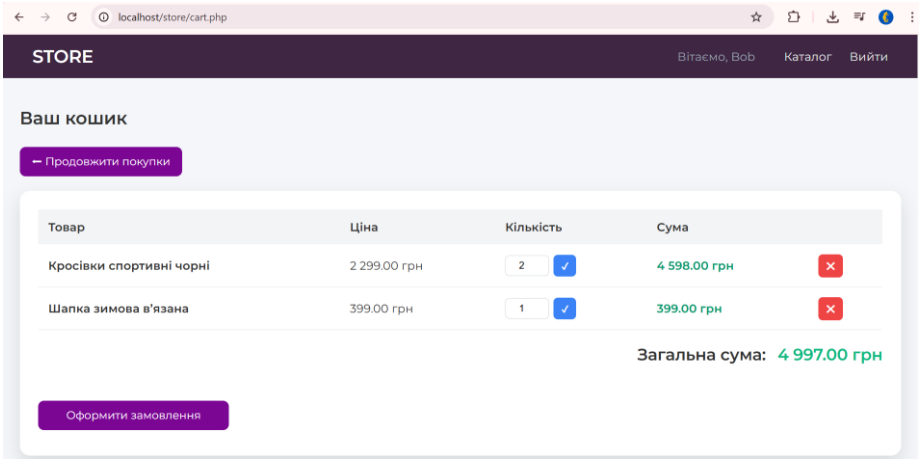

Рисунок 4.6 – Сторінка кошику

Передбачено також можливість видалення товару з кошика шляхом передачі ідентифікатора відповідного запису до окремого обробника. Завершальним елементом сторінки є кнопка переходу до оформлення замовлення, яка ініціює наступний етап процесу купівлі.

Окремий серверний модуль відповідає за додавання товарів до кошика. Після ініціалізації сесії та перевірки авторизації система обробляє запит лише у випадку його надходження методом POST, що відповідає принципам безпечної передачі даних. Отримується ідентифікатор товару та кількість, при цьому кількість

автоматично обмежується мінімальним значенням, що запобігає введенню некоректних або від’ємних даних.

Далі виконується перевірка існування товару в базі даних. Це дозволяє переконатися в коректності переданого ідентифікатора та уникнути додавання неіснуючих позицій. Після підтвердження валідності товару система перевіряє, чи вже існує відповідний запис у кошику користувача. Якщо товар уже був доданий раніше, здійснюється оновлення кількості шляхом її збільшення. Якщо ж позиція відсутня, створюється новий запис у таблиці кошика з фіксацією користувача, товару та кількості.

Після завершення операції користувач автоматично перенаправляється на сторінку кошика, що дозволяє одразу переглянути результат виконаної дії. Така організація логіки забезпечує узгодженість між базою даних і відображенням інформації, підтримує актуальність даних та реалізує повноцінний механізм управління кошиком у межах вебзастосунку інтернет-магазину одягу.

4.6 Панель адміністратора

Реалізовано адміністративну частину вебзастосунку інтернет-магазину одягу, яка забезпечує управління товарами та перегляд замовлень. Даний функціональний модуль призначений виключно для користувачів із розширеними правами доступу та побудований відповідно до принципів розмежування ролей, безпеки доступу й централізованого адміністрування даних.

На початковому етапі в кожному адміністративному модулі ініціалізується сесія користувача, після чого здійснюється перевірка наявності ідентифікатора користувача та його ролі. Якщо користувач не авторизований або не має ролі адміністратора, виконання сценарію припиняється з повідомленням про заборону доступу. Таким чином реалізовано механізм контролю доступу, який запобігає несанкціонованому перегляду або модифікації даних. Додатково використовується окремий модуль автентифікації, що централізує перевірку прав адміністратора для всіх сторінок адміністративної панелі.

Головна сторінка адміністративної панелі виконує функцію навігаційного центру. Вона містить посилання на управління товарами та перегляд замовлень, а також можливість повернення до основного каталогу магазину. Такий підхід дозволяє адміністратору швидко перемикатися між різними функціональними підсистемами.

Модуль перегляду замовлень реалізує отримання даних із таблиці замовлень бази даних із сортуванням у зворотному хронологічному порядку. Це дозволяє першочергово відображати найновіші замовлення. Отримані записи виводяться у табличній формі із зазначенням ідентифікатора замовлення, ідентифікатора користувача, загальної суми та статусу оплати. Така структура надає адміністратору можливість контролювати фінансовий стан операцій, аналізувати поточні замовлення та відслідковувати їхній статус.

Модуль управління товарами передбачає відображення повного переліку продукції, що зберігається у базі даних. Дані впорядковуються за спаданням ідентифікатора, що забезпечує відображення останніх доданих товарів на початку списку. Інтерфейс організовано у вигляді адміністративної панелі з боковою навігацією та основною областю контенту. Для кожного товару відображається його ідентифікатор, зображення, назва, ціна та набір дій для редагування або видалення. Зображення підтягується з відповідного каталогу, а у випадку його відсутності використовується стандартне резервне зображення. Для текстових полів застосовується обробка спеціальних символів, що забезпечує захист від XSS-атак.

Окремий модуль відповідає за додавання нового товару. Після перевірки прав адміністратора та підключення до бази даних система очікує надходження даних методом POST. Реалізовано механізм завантаження зображення товару із перевіркою допустимих форматів файлів. У разі відповідності формату зображення зберігається у файловій системі із унікалізованою назвою, що формується на основі поточного часу. Це дозволяє уникнути конфліктів імен файлів. Після обробки зображення здійснюється вставка нового запису до таблиці товарів із зазначенням назви, опису, ціни, категорії, кількості на складі та шляху до зображення. Після

успішного додавання відбувається перенаправлення до списку товарів, що забезпечує логічну завершеність операції.

Модуль редагування товару реалізує двоетапну логіку роботи. На першому етапі за переданим ідентифікатором здійснюється отримання поточних даних товару з бази. Якщо запис відсутній, система повідомляє про помилку. На другому етапі, у разі надходження оновлених даних методом POST, виконується перевірка можливого завантаження нового зображення. Якщо адміністратор обрав новий файл, відбувається перевірка формату та збереження файлу аналогічно процедурі додавання. Якщо нове зображення не вибрано, використовується попередній файл. Далі здійснюється оновлення відповідного запису в таблиці товарів із заміною змінених значень. Після завершення операції виконується повернення до списку товарів.

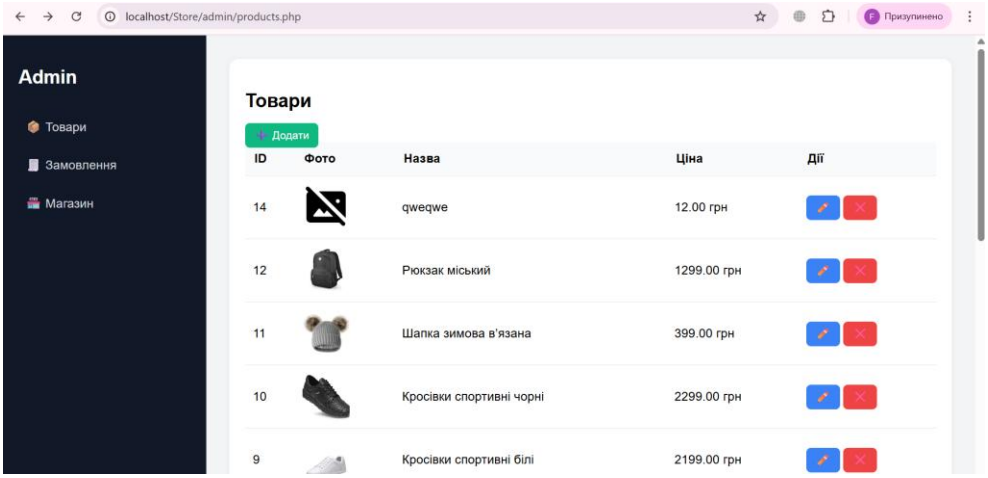

Рисунок 4.7 – Адміністративна частина вебзастосунку

Загалом адміністративна підсистема забезпечує повний цикл управління товарним асортиментом і моніторингом замовлень. Реалізовано механізми автентифікації, перевірки прав доступу, безпечної роботи з базою даних через підготовлені запити, контроль завантаження файлів та централізовану навігацію. Така організація дозволяє підтримувати актуальність даних магазину, оперативно реагувати на зміни в асортименті та контролювати процес оформлення замовлень у межах вебзастосунку.

4.7 Використання стилів

Таблиця стилів CSS визначає візуальну концепцію та інтерфейсну поведінку вебзастосунку інтернет-магазину одягу. Даний стилістичний модуль формує єдину дизайн-систему платформи, забезпечує узгодженість елементів інтерфейсу, адаптивність відображення та підвищує зручність взаємодії користувача з системою.

На початковому рівні встановлюється глобальне скидання відступів і параметрів блочної моделі для всіх елементів сторінки, що забезпечує передбачувану поведінку верстки у різних браузерах. Як основний шрифт застосовується Montserrat, імпортований із зовнішнього джерела, що формує сучасний мінімалістичний стиль інтерфейсу. Для всього документа визначено світлий фон та нейтральну кольорову палітру тексту, що сприяє зручності сприйняття інформації.

Верхня панель навігації реалізована як гнучкий контейнер із вирівнюванням елементів по горизонталі. Вона включає логотип, навігаційні посилання та форму пошуку. Колірна гама навігаційної панелі побудована на темних відтінках бордового, що створює контраст із загальним світлим фоном сторінки. Форма пошуку передбачає текстове поле, селектор категорій та кнопку відправлення запиту. Для інтерактивних елементів застосовано ефекти фокусування та наведення, що забезпечують візуальний зворотний зв’язок під час взаємодії користувача з інтерфейсом.

Основна область контенту обмежена фіксованою шириною та центрована відносно сторінки, що забезпечує структуроване розміщення інформації. Блок відображення товарів реалізовано за допомогою сіткової моделі, яка автоматично розподіляє картки товарів залежно від доступного простору. Передбачено як фіксовану ширину картки, так і адаптивний режим із використанням auto-fill, що дозволяє коректно масштабувати інтерфейс на різних розмірах екранів.

Картка товару має світлий фон, заокруглені кути та тінь, що створює ефект піднятого елемента. При наведенні застосовується анімація зміщення та посилення тіні, що додає динамічності інтерфейсу. Зображення товару масштабується із

збереженням пропорцій, а текстові елементи структуровані за принципом ієрархії: назва, короткий опис і ціна. Колір ціни виділено зеленим відтінком для акцентування фінансової інформації.

Система кнопок уніфікована через загальний клас, який визначає колір, радіус заокруглення, розміри та анімацію зміни стану. Забезпечено однакове відображення як для елементів типу button, так і для гіперпосилань, стилізованих під кнопки. Додатково передбачено спеціалізовані кнопки для перегляду деталей товару, додавання до кошика та роботи зі списком обраного. Кнопка wishlist реалізована у вигляді іконки з позиціонуванням у верхньому куті картки, що дозволяє швидко додавати товар до обраного без переходу на іншу сторінку.

Форми авторизації та реєстрації оформлені у вигляді окремого блоку з тінню та центруванням на сторінці. Вхідні поля мають стандартизовані відступи, радіус заокруглення та зміну кольору рамки при фокусі. Повідомлення про успішні або помилкові операції стилізовані окремими класами із різними кольоровими схемами, що дозволяє користувачу швидко інтерпретувати результат виконаної дії.

Окремий розділ стилів присвячено сторінці окремого товару, де застосовано двоколонкову сітку з великим зображенням та детальною інформацією. Використано збільшені розміри шрифту для назви та ціни, а також оптимізовано міжрядковий інтервал для довгого текстового опису. Це забезпечує читабельність та зручність ознайомлення з характеристиками продукції.

Модуль кошика включає стилізацію таблиці товарів, підсумкової вартості та елементів керування кількістю. Таблиця має чітку структурну організацію з виділенням заголовків і меж рядків. Передбачено інтерактивні кнопки для оновлення кількості товару та його видалення. Загальна сума замовлення візуально акцентується збільшеним шрифтом та кольором. У разі порожнього кошика відображається окремий інформаційний блок із центруванням тексту.

Головна сторінка містить секцію hero із повноекранним слайдером фонових зображень, реалізованим через CSS-анімацію. Застосовано ефект плавної зміни прозорості з циклічним відображенням кількох зображень. Поверх фону накладено затемнюючий шар для підвищення контрастності тексту. Контент у межах hero-

блоку центрується як по вертикалі, так і по горизонталі, що формує виразний візуальний вступ до сайту.

Додатково реалізовано секції з описом переваг магазину, які оформлені у вигляді гнучких блоків із можливістю переносу на новий рядок при зменшенні ширини екрану. Нижній колонтитул має темний фон і слугує завершальним елементом сторінки.

Передбачено базову адаптивність інтерфейсу через медіазапити. При зменшенні ширини екрана навігаційна панель переходить у вертикальний режим відображення, а текстові елементи масштабуються для комфортного читання. Таким чином забезпечується коректне функціонування вебзастосунку як на десктопних пристроях, так і на мобільних платформах.

Загалом представлений CSS-модуль формує цілісну, структуровану та сучасну візуальну архітектуру інтернет-магазину одягу, забезпечуючи єдність стилю, інтуїтивність інтерфейсу та адаптивність відображення.

Висновки до розділу 4

У межах виконаного розділу було комплексно реалізовано архітектурні, функціональні та візуальні складові вебзастосунку інтернет-магазину одягу. Послідовність виконання робіт охопила створення структури бази даних, налаштування взаємодії з серверною частиною, реалізацію механізмів автентифікації користувачів, розроблення основних користувацьких сторінок та формування адміністративного середовища керування ресурсами системи.

На етапі проєктування та створення таблиць бази даних було закладено логічну модель збереження інформації, що забезпечує цілісність, зв’язність і масштабованість системи. Структура бази даних дозволяє ефективно зберігати відомості про користувачів, товари, замовлення, вміст кошика та список обраного, а також підтримує коректні зв’язки між сутностями. Це створює надійну основу для подальшої реалізації бізнес-логіки застосунку.

Підключення до сервера бази даних забезпечило стабільний механізм взаємодії між серверною частиною вебзастосунку та сховищем даних.

Налаштування з’єднання та використання підготовлених запитів сприяли підвищенню безпеки обробки даних і зменшенню ризиків несанкціонованого втручання.

Реалізація функціоналу реєстрації та входу дозволила впровадити систему ідентифікації користувачів, що є ключовим елементом персоналізованої взаємодії з платформою. Використання сесійного механізму забезпечило контроль доступу до захищених сторінок та розмежування прав користувачів відповідно до їхньої ролі. Таким чином було створено основу для безпечної роботи як звичайних клієнтів, так і адміністраторів системи.

Сторінка каталогу реалізувала механізм відображення асортименту товарів із можливістю пошуку та фільтрації за категоріями. Це підвищує зручність навігації та оптимізує процес вибору продукції. Інтеграція функцій додавання товарів до кошика та списку обраного забезпечила інтерактивність та покращила користувацький досвід.

Модуль кошика реалізує логіку накопичення обраних позицій, автоматичного обчислення вартості та редагування кількості товарів. Завдяки цьому користувач отримує повний контроль над своїм замовленням перед його підтвердженням. Взаємодія кошика з базою даних гарантує актуальність інформації та коректність фінансових розрахунків.

Адміністративна панель забезпечує централізоване керування ресурсами інтернет-магазину. Через відповідний інтерфейс адміністратор має можливість додавати, редагувати та видаляти товари, а також переглядати замовлення користувачів. Реалізоване розмежування доступу гарантує, що управлінські функції доступні виключно авторизованим особам із відповідними правами.

Завершальним етапом стала стилізація інтерфейсу, що забезпечила цілісність дизайну, візуальну привабливість та адаптивність відображення на різних пристроях. Використання сучасних принципів верстки та єдиної кольорової концепції сприяло формуванню професійного вигляду вебзастосунку та покращенню зручності його використання.

Отже, у межах розділу було створено функціонально завершену інформаційну систему інтернет-магазину одягу, яка поєднує структуровану базу даних, безпечну серверну взаємодію, повноцінний користувацький функціонал і сучасне візуальне оформлення. Реалізовані компоненти забезпечують стабільну роботу платформи та створюють основу для її подальшого розширення й удосконалення.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було досягнуто поставленої мети, що полягала у розробці повнофункціонального вебзастосунку інтернет-магазину одягу із використанням сучасних технологій вебпрограмування та засобів організації баз даних. Створена система забезпечує повний цикл взаємодії користувача з онлайн-магазином – від реєстрації та авторизації до формування замовлення й завершення процедури покупки.

У ході роботи було послідовно виконано всі поставлені завдання. Проведено аналіз предметної області та визначено основні вимоги до функціоналу системи. Спроєктовано структуру бази даних із урахуванням логічних зв’язків між сутностями, що забезпечило цілісність і узгодженість збереження інформації. Реалізовано механізми підключення до сервера бази даних та організовано безпечну взаємодію між клієнтською та серверною частинами застосунку. Розроблено модулі реєстрації та авторизації користувачів із розмежуванням прав доступу, що дозволило створити окремі рівні функціональності для звичайних користувачів і адміністратора.

У межах реалізації користувацької частини було створено сторінку каталогу з можливістю пошуку та фільтрації товарів, сторінку кошика з функцією редагування кількості обраних позицій та обчислення загальної вартості замовлення, а також механізм оформлення замовлення із фіксацією відповідних даних у базі. Адміністративна панель забезпечує керування асортиментом товарів і перегляд інформації про замовлення, що дозволяє ефективно контролювати діяльність магазину. Окрему увагу приділено стилізації інтерфейсу, що сприяє формуванню цілісного візуального образу застосунку та підвищує зручність його використання.

Отримані результати підтверджують, що поставлені у роботі завдання виконані повністю, а розроблений вебзастосунок відповідає визначеним вимогам та може бути використаний як основа для подальшого розвитку або впровадження у практичну діяльність. Перспективами подальшого вдосконалення системи можуть бути розширення функціональних можливостей, впровадження

аналітичних модулів, оптимізація продуктивності та підвищення рівня захисту інформації.

Таким чином, виконана робота має як навчальне, так і практичне значення, демонструє успішне застосування теоретичних знань у сфері веброзробки та підтверджує досягнення поставленої мети.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Вебдизайн: що це таке і як вивчити з нуля. URL: https://goit.global/ua/articles/vebdyzayn-shcho-tse-take-i-iak-vyvchyty-z-nulia/ (дата звернення: 02.02.2026).
2. Етапи розробки інтернет-магазину. URL: https://webstudio2u.net/ua/website-development.html (дата звернення: 07.01.2024).
3. Зандстра М. PHP. Об’єкти, шаблони та методики програмування. Київ, 2022. 866 с.
4. Інтернет-магазин одягу Answear. URL: https://answear.ua/c/vona (дата звернення: 13.03.2026).
5. Інтернет-магазин одягу Intertop. URL: https://intertop.ua/uk-ua/ (дата звернення: 16.03.2026).
6. Інтернет-магазин одягу Kasta. URL: https://kasta.ua/ (дата звернення: 05.03.2026).
7. Лабберс П. HTML5 для професіоналів: потужні інструменти для розробки сучасних веб-додатків. Київ, 2011. 267 с.
8. Лоусон Б., Шарп Р. Вивчаємо HTML5. Київ, 2020. 272 с.
9. Макфарланд Д. Велика книга CSS3. 3-тє вид. Київ, 2014. 608 с.
10. Ніксон Р. PHP, MySQL, JavaScript і HTML5. 4-те вид. Київ, 2018. 768 с.
11. Тардаскіна Т. М., Стрельчук Є. М., Терешко Ю. В. Електронна комерція : навчальний посібник. Одеса: ОНАЗ ім. О. С. Попова, 2011. 244 с.
12. Трофименко О. Г. Веб-дизайн та HTML-програмування. Одеса: Фенікс, 2017. 215 с.
13. Що таке адаптивний дизайн сайту та як його зробити. URL: https://hostiq.ua/blog/ukr/adaptive-design/ (дата звернення: 06.02.2026).
14. Accessibility for a Content-Driven Web App Frontend. Google for Developers. URL: https://developers.google.com/solutions/content-driven/frontend/accessibility (accessed: 11.02.2026).

15. Berezovska L., Kyrychenko A. Development of Electronic Commerce in Ukraine and the EU. Economy and Society. 2022. № 42. DOI: https://doi.org/10.32782/2524-0072/2022-42-15.

16. Bilovodska O., Poretskova M. Barriers to Online Purchase: Case Study Consumer Behaviour in Fashion Industry E-Commerce. Economic Journal of Lesya Ukrainka Volyn National University. 2023. № 2. P. 102–112. DOI: https://doi.org/10.29038/2786-4618-2023-02-102-112.

17. Frain B. Responsive Web Design with HTML5 and CSS3. Birmingham: Packt Publishing, 2015. 409 p.

18. Get Started with Bootstrap. Bootstrap Documentation. URL: https://getbootstrap.com/docs/5.3/getting-started/introduction/ (accessed: 13.02.2026).

19. PHP Manual. URL: http://php.net/manual/en/index.php (accessed: 15.01.2024).

20. Unger R., Chandler C. A Project Guide to UX Design: For User Experience Designers in the Field or in the Making. 2nd ed. Berkeley: New Riders, 2012. 337 p.

21. Walter A. Designing for Emotion. 2nd ed. New York: A Book Apart, 2020. 112 p.

ДОДАТОК

Лістинг коду файлу catalog.php

```
<?php
session_start();
require_once __DIR__ . "/config/db.php";

if (!isset($_SESSION['user_id'])) {
    header("Location: login.php");
    exit;
}

$search = $_GET['search'] ?? '';
$category = $_GET['category'] ?? '';

// SQL
$sql = "SELECT * FROM Products WHERE 1=1";
$params = [];

if (!empty($search)) {
    $sql .= " AND name LIKE ?";
    $params[] = "%" . $search . "%";
}

if (!empty($category)) {
    $sql .= " AND category = ?";
    $params[] = $category;
}

$sql .= " ORDER BY id DESC";

$stmt = $pdo->prepare($sql);
$stmt->execute($params);
$products = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Каталог товарів</title>
    <link rel="stylesheet" href="assets/css/style.css">
</head>
<body>
```

```
<div class="navbar">
    <div class="logo">KOGUT</div>
    <form method="GET" action="catalog.php" class="search-form">

        <input type="text" name="search"
            placeholder="Пошук товарів..."
            value="<?= htmlspecialchars($search) ?>">

        <select name="category">
            <option value="">Всі категорії</option>

            <option value="Футболки" <?= $category == 'Футболки' ? 'selected' : '' ?>>
                Футболки
            </option>

            <option value="Худі" <?= $category == 'Худі' ? 'selected' : '' ?>>
                Худі
            </option>

            <option value="Джинси" <?= $category == 'Джинси' ? 'selected' : '' ?>>
                Джинси
            </option>

            <option value="Сорочки" <?= $category == 'Сорочки' ? 'selected' : '' ?>>
                Сорочки
            </option>

            <option value="Сукні" <?= $category == 'Сукні' ? 'selected' : '' ?>>
                Сукні
            </option>

            <option value="Куртки" <?= $category == 'Куртки' ? 'selected' : '' ?>>
                Куртки
            </option>

            <option value="Кросівки" <?= $category == 'Кросівки' ? 'selected' : '' ?>>
            Кросівки
            </option>

            <option value="Шапки" <?= $category == 'Шапки' ? 'selected' : '' ?>>
                Шапки
            </option>

            <option value="Рюкзаки" <?= $category == 'Рюкзаки' ? 'selected' : '' ?>>
```

```
            Рюкзаки
        </option>
    </select>

    <button type="submit">Пошук</button>

</form>
<div class="nav-links">
    <span style="color:#9ca3af; margin-right:15px;">
        Вітаємо, <?= htmlspecialchars($_SESSION['name']) ?>
    </span>
    <a href="catalog.php">Каталог</a>
    <a href="cart.php">Кошик</a>
    <a href="wishlist.php">💔 Обране</a>
    <a href="logout.php">Вийти</a>
</div>
</div>

<div class="container">


<h2 style="margin-bottom:25px;">Каталог товарів</h2>

<?php
$userId = $_SESSION['user_id'];

$stmt = $pdo->prepare("SELECT product_id FROM Wishlist WHERE user_id=?");
$stmt->execute([$userId]);

$wishlistItems = $stmt->fetchAll(PDO::FETCH_COLUMN);
?>

<div class="products">
<?php foreach ($products as $product): ?>

    <div class="product-card">

        <img src="images/<?= htmlspecialchars($product['image']) ?>" alt="Товар">

        <div class="product-title">
            <?= htmlspecialchars($product['name']) ?>
        </div>

        <form action="wishlist_toggle.php" method="POST" class="wishlist-form">
            <input type="hidden" name="product_id" value="<?= $product['id'] ?>">
```

```
            <button class="wishlist-btn">
              <?php if (in_array($product['id'], $wishlistItems)): ?>
                ❤️
              <?php else: ?>
                🤍
              <?php endif; ?>
            </button>
          </form>

          <div class="product-description">
            <?= htmlspecialchars($product['description']) ?>
          </div>

          <?php if ($product['stock'] > 0 && $product['stock'] < 5): ?>
            <p>В наявності: <?= $product['stock'] ?> шт.</p>
          <?php endif; ?>
          <?php if ($product['stock'] <= 0): ?>
            <p style="color:red;">Немає в наявності</p>
          <?php endif; ?>

          <div class="price">
            <?= number_format($product['price'], 2, '.', ' ') ?> USD
          </div>

          <a class="btn details-btn" href="product.php?id=<?= $product['id'] ?>">
            Детальніше
          </a>
          <form action="add_to_cart.php" method="POST">
            <input type="hidden" name="product_id" value="<?= $product['id'] ?>">
            <?php if ($product['stock'] > 0): ?>
            <button type="submit" class="btn add-cart">🛒 Додати в кошик</button>
          <?php else: ?>
            <button class="btn" disabled>❌ Немає</button>
          <?php endif; ?>
          </form>
        </div>
    <?php endforeach; ?>
</div>
</div>

</body>
</html>
```