

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ ЖІНОЧОГО ОДЯГУ «YOUR
FAVORITE» З ВИКОРИСТАННЯ REACT, MONGODB ТА NEXT.JS**

**DEVELOPMENT OF THE "YOUR FAVORITE" WOMEN'S CLOTHING
ONLINE STORE USING REACT, MONGODB AND NEXT.JS**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ПЗс-21
Шемейко Лілія Василівна

Керівник:
Андрушак Ігор Євгенович

Кваліфікаційну роботу
допущено до захисту

« 10 » 06 2025р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

«20» 12 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Шемейко Лілії Василівні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: Розробка інтернет-магазину жіночого одягу «Your favorite» з використання React, MongoDB та Next.js

Керівник роботи: д.т.н., професор, Андрущак І. Є.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

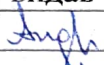
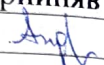
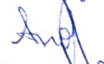
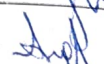
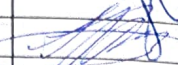
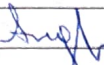
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: специфікація функціональності інтернет-магазину, вимоги до безпеки та адаптивності інтерфейсу, документація до фреймворків та бібліотек, довідкові матеріали з розробки REST API, вимоги до структури бази даних, сучасні підходи до розміщення веб-проектів, методичні вказівки до виконання кваліфікаційної роботи бакалавра.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): обґрунтувати актуальність теми та необхідність створення веб-сайту, визначити функціональні й нефункціональні вимоги до системи, проаналізувати методи реалізації інтернет-магазинів, розробка веб-сайту, протестувати працездатність і коректність функціонування, спроектувати та реалізувати базу даних, розгорнути веб-сайт на обраному хостингу, реалізувати заходи для забезпечення безпеки.

5. Перелік графічного матеріалу: 21 рисунки, 4 таблиць, 2 додатки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Андрущак І. Є.		
Специфікація вимог до розробленої системи	Андрущак І. Є.		
Розробка об'єкта проектування	Андрущак І. Є.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	2,53 %		
Академічна доброчесність	Андрущак І. Є.		

7. Дата видачі завдання «20» грудня 2024 р.

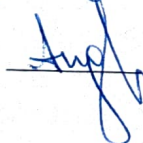
№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	

Здобувач вищої освіти



Лілія ШЕМЕЙКО

Керівник кваліфікаційної роботи



Ігор АНДРУЩАК

АНОТАЦІЯ

Шемейко Л. В. Розробка інтернет-магазину жіночого одягу «Your favorite» з використання React, MongoDB та Next.js. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається з вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі проаналізовано сучасний стан розвитку електронної комерції, зокрема в сегменті онлайн-продажу жіночого одягу. Розглянуто приклади популярних українських інтернет-магазинів, виявлено їхні переваги та недоліки, що обґрунтовує актуальність створення нового рішення.

У другому розділі визначено вимоги до веб-застосунку, здійснено огляд підходів до розробки сайтів, розглянуто переваги використання React, Next.js та MongoDB. Розроблено логічну структуру сайту, спроектовано архітектуру бази даних, а також визначено механізми безпечного зберігання даних та адміністрування.

У третьому розділі описано практичну реалізацію веб-сайту «Your Favorite»: реалізовано адаптивний інтерфейс, інтерактивний каталог товарів, кошик, форму оформлення замовлення, адміністративну панель та інтеграцію з API служби доставки. Проведено тестування працездатності сайту на різних пристроях та у різних режимах. Розглянуто питання захисту даних і продуктивності.

Ключові слова: інтернет-магазин, електронна комерція, React, Next.js, MongoDB, Tailwind CSS, веб-розробка, інтеграція API, адаптивний дизайн, база даних, адміністративна панель, тестування, Vercel.

ABSTRACT

Shemeiko L. Development of the "Your Favorite" Women's Clothing Online Store Using React, MongoDB and Next.js. Manuscript. Qualification work of the bachelor's degree program "Software Engineering", specialty "Software Engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's thesis consists of an introduction, three chapters, conclusions and suggestions, a list of references and appendices.

The first chapter analyzes the current state of e-commerce development, particularly in the segment of online women's clothing sales. Examples of popular Ukrainian online stores are reviewed, their advantages and disadvantages are identified, which substantiates the relevance of creating a new solution.

The second chapter defines the requirements for the web application, provides an overview of web development approaches, and examines the advantages of using React, Next.js, and MongoDB. The logical structure of the website is developed, the database architecture is designed, and mechanisms for secure data storage and content administration are described.

The third chapter presents the practical implementation of the "Your Favorite" website: an adaptive interface, an interactive product catalog, a shopping cart, an order checkout form, an admin panel, and integration with the delivery service API have been implemented. The performance of the website was tested on different devices and in various modes. Issues of data protection and performance optimization are also considered.

Keywords: online store, e-commerce, React, Next.js, MongoDB, Tailwind CSS, web development, API integration, adaptive design, database, admin panel, testing, Vercel.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ІНТЕРНЕТ-ТОРГІВЛІ У СФЕРІ ЖІНОЧОГО ОДЯГУ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	12
Висновки до розділу 1	12
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЕНОЇ СИСТЕМИ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	17
Висновки до розділу 2	26
РОЗДІЛ 3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ «YOUR FAVORITE» ДЛЯ ПРОДАЖУ ЖІНОЧОГО ОДЯГУ	27
3.1 Практична реалізація об'єкта проектування	27
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	42
3.3 Розробка бази даних	45
3.4 Розміщення сайту на вибраній платформі хостингу	47
3.5 Безпека розроблюваного веб-сайту	49
Висновки до розділу 3	50
ВИСНОВКИ	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54
ДОДАТКИ	56

ВСТУП

Актуальність теми. У сучасному цифровому суспільстві електронна комерція стала ключовим інструментом ведення бізнесу. Особливо активний розвиток спостерігається у сфері онлайн-продажу одягу, що зумовлено зручністю, доступністю та широким вибором для кінцевого користувача. Проте, попри зростання кількості інтернет-магазинів, існує значна частка сайтів, які не забезпечують належного користувацького досвіду через недосконалий інтерфейс, низький рівень персоналізації, перевантажену навігацію та технічні обмеження. Актуальним є створення нового веб-сайту, який би поєднував сучасні технології, естетичний дизайн і зручність користування.

Таким чином, метою даної кваліфікаційної роботи є розробка функціонального, адаптивного та безпечного інтернет-магазину жіночого одягу «Your Favorite», що забезпечує ефективну взаємодію з користувачем, реалізує основні функції онлайн-торгівлі й відповідає сучасним технічним та візуальним вимогам.

Об'єктом роботи є процес розробки веб-сайту для електронної комерції.

Предметом роботи є методи, технології та інструменти створення інтернет-магазинів з використанням сучасного фронтенд- та бекенд-стеку (React, Next.js, Tailwind CSS, MongoDB).

Для досягнення поставленої мети необхідно реалізувати наступні завдання:

- обґрунтувати актуальність обраної теми та пояснити необхідність створення веб-сайту;
- визначити функціональні та нефункціональні вимоги до майбутнього веб-ресурсу;
- здійснити огляд способів і методів реалізації інтернет-магазину;
- реалізувати веб-сайт відповідно до сформульованих вимог;
- виконати тестування працездатності та коректності роботи сайту;

- спроектувати та реалізувати структуру бази даних для зберігання інформації про товари, категорії та замовлення;
- здійснити розгортання веб-сайту на обраній платформі хостингу;
- реалізувати заходи для підвищення рівня безпеки веб-застосунку.

Апробація результатів роботи:

1) Andrushchak I., Shemeiko L. Functional and technical features of modern web development. Proceedings of the XIX International Scientific and Practical Conference. Oslo, Norway. 2025. Pp. 50-55. URL: <https://isg-konf.com/current-problems-of-the-development-of-science-trends-and-innovations/> (додаток А);

2) Andrushchak I., Shemeiko L. Options for technical security in the operation of modern web sites. Proceedings of the XX International Scientific and Practical Conference. Seville, Spain. 2025. Pp. 59-64. URL: <https://isg-konf.com/innovative-scientific-research-latest-theories-modern-methods-and-practices/> (додаток Б).

РОЗДІЛ 1

АНАЛІЗ СУЧАСНОГО СТАНУ ІНТЕРНЕТ-ТОРГІВЛІ У СФЕРІ ЖІНОЧОГО ОДЯГУ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Цифровізація суспільства значно трансформувала всі сфери життєдіяльності, включаючи торгівлю. Однією з найпомітніших тенденцій останнього десятиліття є стрімкий розвиток електронної комерції – моделі ведення бізнесу, що дозволяє здійснювати купівлю-продаж товарів і послуг через мережу Інтернет. Цей підхід відкрив нові можливості для як великих компаній, так і малого бізнесу, дозволяючи значно знизити витрати на обслуговування фізичних торгових точок та вийти на глобальний ринок [1].

Веб-сайти стали головною платформою для ведення онлайн-продажів. У порівнянні з класичними торговими об'єктами, вони мають безліч переваг: доступність 24/7, автоматизація процесу замовлення, швидкий пошук товарів, інтеграція з платіжними сервісами, зручна доставка. Окреме значення мають інтернет-магазини, орієнтовані на певну аудиторію, наприклад, на жінок, які шукають стильний, сучасний, доступний одяг.

За останні роки сегмент жіночого онлайн-одягу суттєво виріс як в Україні, так і в усьому світі. Онлайн-шопінг став не лише популярною альтернативою фізичним магазинам, а й більш привабливим варіантом завдяки зручності, економії часу та доступу до широкого асортименту. Компанії, що займаються роздрібною торгівлею одягом, активно впроваджують нові цифрові рішення, інтегруючи веб-аналітику, алгоритми рекомендацій, адаптивний дизайн, відгуки та інші елементи, які покращують користувацький досвід [2].

За останні роки з'явилося багато прикладів успішних реалізацій інтернет-магазинів у цій ніші. Серед найбільш відомих в Україні:

– MustHave – український бренд з мінімалістичним, інтуїтивно зрозумілим інтерфейсом, адаптованим під мобільні пристрої. Проте

функціональність сайту досить базова, а персоналізація товарів практично відсутня;

- One by One – сайт характеризується чистим і лаконічним інтерфейсом, що добре відповідає естетиці бренду. Основний акцент зроблено на базовий жіночий одяг. Водночас функціональність ресурсу обмежена, адже бракує розширених фільтрів, персональних рекомендацій, а особистий кабінет містить лише мінімальний набір функцій;

- Answer.ua – мультибрендова платформа з великим асортиментом товарів. До недоліків належать перевантажений інтерфейс і складність пошуку через недосконалу систему фільтрів;

- Kasta.ua – один із найбільших онлайн-маркетплейсів в Україні. Позитивною стороною є широкий вибір брендів і доставка, а негативною – агресивна реклама та надмірне комерційне навантаження, що погіршує користувацький досвід.

Ці сервіси показують успішність онлайн-торгівлі, однак мають недоліки, що залишають простір для створення нових платформ з покращеним функціоналом, зручністю і швидкодією. Повну порівняльну характеристику наведено в таблиці 1.1.

Таблиця 1.1 – Порівняльна таблиця інтернет-магазинів жіночого одягу

Назва сайту	Переваги	Недоліки
MustHave	Зручний мінімалістичний інтерфейс, адаптація до мобільних пристроїв, швидка навігація	Базовий функціонал, відсутність персоналізації, обмежений кабінет користувача
One by One	Український бренд з базовим асортиментом, чистий та естетичний дизайн, фокус на мінімалізм	Обмежений функціонал, відсутність розширених фільтрів і особистих рекомендацій
Answer.ua	Великий вибір товарів, система фільтрів, акційні пропозиції	Перевантажений інтерфейс, складна навігація, неточна фільтрація
Kasta.ua	Широка партнерська мережа, гнучкі умови доставки, велика кількість брендів	Агресивна реклама, перевантаження банерами, хаотична навігація

Загальний аналіз сучасного ринку електронної комерції у сфері продажу одягу вказує на наявність як позитивних тенденцій, так і ряду проблемних аспектів. З одного боку, онлайн-магазини активно розвиваються, використовуючи цифрові інструменти для автоматизації продажів, покращення клієнтського сервісу та розширення асортименту. Покупці отримують можливість зручно обирати товари, порівнювати ціни, здійснювати оплату та отримувати замовлення без необхідності виходити з дому.

З іншого боку, в більшості існуючих систем зберігаються технічні й концептуальні недоліки. До них можна віднести низький рівень персоналізації, обмежені функціональні можливості, недосконалу систему фільтрації товарів, складну або перевантажену структуру навігації, а також недостатню оптимізацію для мобільних пристроїв. У багатьох випадках користувацький інтерфейс не відповідає сучасним стандартам UI/UX, що ускладнює взаємодію з ресурсом і може негативно впливати на конверсію продажів.

Ще однією проблемою є відсутність цілісного брендингу та емоційного зв'язку з користувачем. Часто сайти технічно функціональні, але не створюють привабливого візуального враження або атмосфери довіри. Це особливо актуально у сфері моди, де естетика і стиль відіграють ключову роль у сприйнятті товару.

Також спостерігається тенденція до використання застарілих технологій, що ускладнює масштабування платформи, впровадження нових функцій або інтеграцію з іншими цифровими сервісами. Через це компаніям доводиться або повністю переробляти сайт, або залишатися в межах обмеженого функціоналу, що стримує розвиток бізнесу.

З огляду на вищезазначене, актуальною є розробка нової системи, яка не лише відповідала б базовим вимогам користувачів, а й враховувала би сучасні технологічні тенденції. Така система має поєднувати ефективність, зручність, адаптивність та естетику, забезпечуючи позитивний досвід для кінцевого споживача і створюючи нову якість у сегменті онлайн-продажу одягу [3].

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Метою кваліфікаційної роботи є розробка функціонального, зручного та безпечного веб-застосунку інтернет-магазину жіночого одягу «Your Favorite» із застосуванням сучасних веб-технологій. Проєкт має забезпечити повноцінну комерційну діяльність шляхом реалізації системи перегляду товарів, інтерактивного кошика, оформлення замовлення та адміністративного керування контентом.

Для досягнення поставленої мети необхідно реалізувати наступні завдання:

- обґрунтувати актуальність обраної теми та пояснити необхідність створення веб-сайту;
- визначити функціональні та нефункціональні вимоги до майбутнього веб-ресурсу;
- здійснити огляд способів і методів реалізації інтернет-магазину;
- реалізувати веб-сайт відповідно до сформульованих вимог;
- виконати тестування працездатності та коректності роботи сайту;
- спроектувати та реалізувати структуру бази даних для зберігання інформації про товари, категорії та замовлення;
- здійснити розгортання веб-сайту на обраній платформі хостингу;
- реалізувати заходи для підвищення рівня безпеки веб-застосунку.

Висновки до розділу 1

У першому розділі було проведено докладний аналіз сучасного стану електронної комерції, зокрема в сегменті онлайн-продажу жіночого одягу. Було визначено основні тенденції розвитку галузі, переваги використання веб-сайтів у торгівлі, а також виявлено типові проблеми, з якими стикаються користувачі під час взаємодії з популярними інтернет-магазинами. Аналіз конкурентних платформ показав, що значна частина існуючих рішень має обмежений

функціонал, слабку персоналізацію, недосконалі фільтри та перевантажений інтерфейс, що знижує загальну ефективність користувацького досвіду.

На основі виявлених проблем було сформульовано мету кваліфікаційної роботи – створення сучасного веб-застосунку «Your Favorite», який поєднує функціональність, зручність, адаптивний дизайн і високий рівень безпеки. У підрозділі 1.2 було визначено комплекс завдань, необхідних для реалізації проєкту.

Таким чином, результати розділу стали теоретичною основою для переходу до етапу практичної реалізації проєкту, що буде розглянуто у наступних розділах кваліфікаційної роботи.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЕНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

На початковому етапі розробки веб-застосунку «Your favorite» було проведено аналіз поточних тенденцій в електронній комерції та сформульовано мету створення сучасного інтернет-магазину жіночого одягу. Для досягнення поставлених завдань визначено ключові функціональні компоненти системи, способи взаємодії користувачів з нею, а також типи даних, які будуть оброблятися у процесі її функціонування.

Правильно сформульовані вимоги до веб-застосунку дозволяють зменшити кількість помилок під час розробки, забезпечити узгодженість усіх модулів системи та підвищити ефективність проектування архітектури. Основні функціональні вимоги до системи включають:

- реалізація повноцінного каталогу товарів;
- створення картки товару з детальним описом і візуалізацією;
- можливість додавання товарів у кошик;
- сторінка оформлення замовлення з формою доставки;
- адміністративна панель для управління контентом сайту та обробки замовлень;
- підключення до бази даних для зберігання інформації про товари, користувачів та замовлення;
- підтримка серверного рендерингу та SEO-оптимізації.

Крім функціональних можливостей, були сформульовані також нефункціональні вимоги до веб-застосунку:

- забезпечення високої швидкодії;
- простий та інтуїтивно зрозумілий інтерфейс користувача;
- стійкість до навантажень та можливість масштабування;
- кросбраузерність;

- адаптивний дизайн, що забезпечує коректне відображення на різних пристроях;
- захист даних;
- резервне копіювання даних.

Для моделювання логіки системи використано засоби UML. Це дозволило візуалізувати структуру майбутнього застосунку, логіку переходів між його складовими частинами та дії користувачів. На рисунку 2.1 зображено загальну схему взаємодії користувача з інтерфейсом сайту.

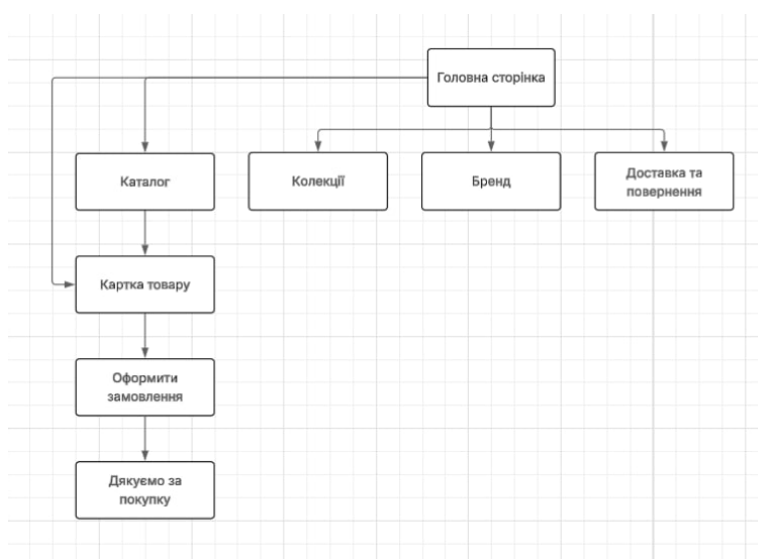


Рисунок 2.1 – Схема взаємодії користувача з сайтом

Для структурування функціоналу адміністративної частини системи створено UML-діаграму (рис. 2.2), яка відображає ключові дії адміністратора: керування товарами, замовленнями та перегляд статистики.

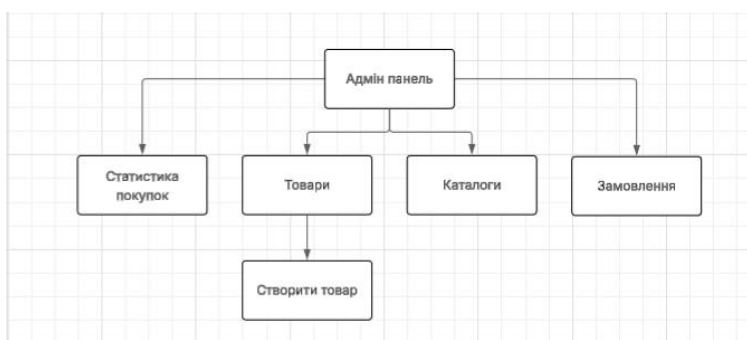


Рисунок 2.2 – Схема взаємодії адміністратора з адмін-панеллю

У процесі проєктування особлива увага приділялася інтерфейсу користувача. Враховуючи сучасні підходи до UX/UI-дизайну, було реалізовано модульну структуру сторінок. Кожен компонент інтерфейсу виконує визначену функцію, що спрощує підтримку, масштабування та знижує навантаження на систему.

Умовно структура кожної сторінки поділена на три ключові частини, що забезпечують логічну організацію контенту та зручність користувача (рис. 2.3):

- Header – верхня навігаційна панель, що містить логотип, меню та іконку кошика;
- Body – динамічний вміст сторінки, який змінюється залежно від поточного маршруту (каталог, товар, оформлення тощо);
- Footer – нижній блок з контактною інформацією, посиланнями на соціальні мережі та авторськими правами.



Рисунок 2.3 – Схема сторінки

Для наочного представлення основних процесів, які відбуваються в системі під час взаємодії з користувачем або адміністратором, було сформовано

таблицю сценаріїв. У ній (табл. 2.1) наведено типові події, що ініціюються користувачем, та відповідні реакції системи, які дозволяють реалізувати логіку роботи веб-застосунку на практиці.

Таблиця 2.1 – Сценарії взаємодії користувача і системи

Подія	Реакція системи
Користувач натискає «Купити»	Товар додається до кошика
Користувач відкриває товар	Відображається детальна інформація про товар
Користувач оформлює замовлення	Відкривається форма доставки
Адміністратор редагує товар	Відображається форма з полями товару
Зміна кількості у кошику	Оновлюється сума замовлення
Користувач переходить до кошика	Відображається вміст кошика
Користувач натискає «Оформити замовлення»	Запускається процес підтвердження замовлення
Адміністратор видаляє товар	Товар видаляється з бази даних
Користувач натискає на логотип	Користувача перенаправляє на головну сторінку

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Під час створення сучасних веб-застосунків, особливо таких складних і багатофункціональних, як інтернет-магазини, розробнику надається широкий вибір інструментів і підходів. У кожного з них є свої сильні й слабкі сторони, і який саме варіант обрати – залежить від багатьох факторів: наскільки великий проєкт, скільки є часу, який бюджет, чи працює над сайтом одна людина чи команда, а також які вимоги ставляться до дизайну, продуктивності та можливості масштабування в майбутньому.

Одні підходи дозволяють швидко створити сайт із мінімальними знаннями програмування, інші – вимагають глибокого занурення у код, але дають повний контроль над кожною деталлю. Саме тому на початку роботи важливо було проаналізувати доступні варіанти і зрозуміти, який з них найкраще підходить для реалізації інтернет-магазину з урахуванням поставлених цілей [4].

Серед основних підходів до створення сайтів можна виділити наступні:

- конструктори сайтів – це найпростіший спосіб швидко зібрати веб-сторінку без жодних знань програмування. Все працює за принципом «вибрав шаблон – налаштував – опублікував». Ідеально підходять для невеликих сайтів або коли потрібно швидко запустити базову версію проєкту. Проте функціональність таких платформ часто обмежена, особливо якщо потрібно реалізувати щось індивідуальне;

- системи керування вмістом (CMS) – дають змогу зручно працювати з текстами, зображеннями, товарами тощо через зручну адмінку. Вони не вимагають глибоких знань у програмуванні, що робить їх зручними для створення блогів, інформаційних сайтів або простих онлайн-магазинів. Але з унікальними функціями можуть виникати труднощі, і тоді без кодування не обійтись;

- класичні серверні фреймворки – це потужні інструменти для розробки складних сайтів, де є потреба в гнучкому контролі над даними, логікою, обліковими записами, обробкою запитів тощо. Підходять для реалізації корпоративних систем, внутрішніх сервісів чи великих проєктів з нестандартною архітектурою. Але варто врахувати, що розробка з таким підходом потребує досвіду і більше часу;

- модерні веб-фреймворки – створені для побудови сучасних інтерфейсів, де сторінка поводить ся як повноцінний додаток: усе працює швидко, зручно, без постійного перезавантаження. Їх використовують для інтернет-магазинів, особистих кабінетів, інтерактивних каталогів тощо. Такі технології чудово поєднуються з серверною логікою, API та базами даних;

- розробка з нуля – це повна свобода дій: сам створюєш і верстку, і функціональність, і серверну частину. Такий підхід дає змогу максимально точно реалізувати потрібну логіку, але вимагає хороших знань, терпіння і багато часу. Найчастіше використовується, коли інші варіанти не підходять або коли розробнику потрібно повністю контролювати систему.

У ході виконання цієї роботи було розглянуто два відомі онлайн-

конструктори сайтів – Weebly та Wix, які надають користувачам зручні інструменти для створення власного вебсайту без потреби в спеціальних технічних знаннях чи досвіді програмування [5].

Weebly – американський онлайн-конструктор сайтів, що займає близько 16 відсотків світового ринку у своїй ніші. Йому довірили створення свого веб-ресурсу понад 50 мільйонів користувачів, більшість з яких – з англomовних країн. Ідея, яку просувають засновники платформи, доволі проста: створити сайт має змогу кожен, незалежно від технічних знань, дизайнерських навичок чи досвіду в ІТ. Все, що потрібно – доступ до інтернету та бажання бути онлайн. Weebly позиціонує себе як інструмент для максимально швидкого запуску сайту: не дарма його асоціюють із гаслом «Сайт за годину». Завдяки функції drag-and-drop («перетягни і постав»), [6] більшість дій виконується інтуїтивно – платформа буквально бере на себе все складне, залишаючи користувачеві лише вибір контенту та стилю.

Переваги Weebly:

- дуже простий drag-and-drop інтерфейс – ідеальний для новачків;
- інтеграція з Square – зручно для ведення e-commerce і фізичних продажів;
- автоматична мобільна адаптація сторінок;
- вбудовані SEO-інструменти та аналітика;
- прості тарифні плани.

Недоліки Weebly:

- обмежена кількість шаблонів і менше варіантів дизайну;
- менше додатків у магазині розширень, ніж у Wix;
- обмежена гнучкість дизайну – важко створити нестандартний макет;
- розвиток платформи дещо уповільнений порівняно з конкурентами.

Wix – один із найвідоміших онлайн-конструкторів сайтів у світі, особливо популярний серед англomовної аудиторії, вже зібрав понад 100 мільйонів користувачів з усіх куточків планети. Його обирають за універсальність (можна створити як просту візитку, так і повноцінний інтернет-магазин чи особистий

блог), адаптивність під мобільні та десктопні пристрої, зрозумілий інтерфейс, а також за те, що навіть безкоштовна версія платформи часто буває цілком достатньою для старту [7].

Переваги Wix:

- сотні шаблонів на будь-яку тему;
- Wix ADI (штучний інтелект) – автоматичне створення сайту;
- потужний Wix Editor для повного контролю дизайну;
- App Market з 500+ додатками – розширення функціональності;
- інтеграції з маркетингом, SEO, аналітикою, онлайн-бронюванням;
- Wix Velo – можливість писати власний код і створювати складні

застосунки.

Недоліки Wix:

- шаблон не можна змінити після створення сайту;
- може здатися надто складним для новачків через велику кількість функцій;
- безкоштовна версія показує рекламу Wix;
- експорт сайту неможливий – працює тільки в межах екосистеми Wix.

Онлайн-конструктори Weebly та Wix хоча й забезпечують швидкий старт без потреби в програмуванні, не підходять для реалізації даного проєкту. Обмеження у гнучкості налаштувань, залежність від екосистеми платформи та обмежений контроль над функціональністю не дозволяють реалізувати складну логіку та індивідуальний дизайн, необхідні для повноцінного інтернет-магазину.

Серед популярних систем управління контентом (CMS) було розглянуто WordPress і Joomla (табл. 2.2) – два потужні інструменти для створення та адміністрування веб-сайтів. WordPress вирізняється простотою у використанні, великою кількістю безкоштовних тем і плагінів, а також активною спільнотою, що робить його ідеальним варіантом для новачків, блогів та невеликих інтернет-магазинів [8]. Натомість Joomla пропонує більшу гнучкість у налаштуванні, розширену систему керування правами доступу та вбудовану підтримку

мультимовності, що робить її зручнішою для створення складніших корпоративних або інформаційних порталів.

Таблиця 2.2 – Порівняння можливостей CMS WordPress та Joomla за основними критеріями

Критерій	WordPress	Joomla
Простота використання	Дуже простий, інтуїтивно зрозумілий інтерфейс для новачків	Більш складний інтерфейс, потребує часу на вивчення
Кількість шаблонів і плагінів	Дуже велика кількість, багато безкоштовних	Менше доступних шаблонів і розширень, деякі платні
Спільнота та підтримка	Велика активна спільнота, багато документації, форумів	Менша спільнота, менше ресурсів для підтримки
SEO-оптимізація	Потужна завдяки плагінам (Yoast SEO, Rank Math)	Можливості є, але потребують додаткового налаштування
Мультимовність	Через плагіни (WPML, Polylang)	Вбудована підтримка мультимовності без додаткових розширень
Безпека	Часто стає мішенню через популярність	Менш популярна, тому менше атак, трохи краща безпека
Управління правами доступу	Обмежена система доступу	Розвинена ACL-система, гнучке налаштування ролей
Підходить для	Блоги, особисті сайти, невеликі магазини	Корпоративні сайти, портали, проєкти з багатьма рівнями доступу
Інтеграція e-commerce	Чудова через WooCommerce	Є рішення (VirtueMart тощо), але менш зручні
Масштабованість та складність сайтів	Підходить для простих проєктів	Краще працює зі складними структурами, багаторівневими меню
Оновлення	Часті оновлення тем і плагінів, потребує регулярного моніторингу	Менше залежить від зовнішніх компонентів, стабільніше

WordPress і Joomla мають свої переваги та обмеження: WordPress ідеально підходить для швидкого старту, особистих сайтів і малого бізнесу, тоді як Joomla краще справляється зі складнішими структурами та корпоративними проєктами. Обидві системи можуть бути ефективними, залежно від цілей та масштабу сайту.

Попри зручність і функціональність CMS WordPress та Joomla, жодна з них

не є оптимальним варіантом для реалізації даного проєкту. Через обмежену гнучкість у побудові складної бізнес-логіки, кастомізації інтерфейсу та інтеграції сучасних технологій, було вирішено створити веб-застосунок з нуля із використанням фреймворків і власного коду.

Для того щоб визначити найбільш доцільний інструмент для реалізації веб-платформи, було проведено порівняння Next.js, Laravel та Django [9] за низкою практичних критеріїв.

Next.js – це фреймворк на базі React, який не замінює цю бібліотеку, а доповнює її. Він надає розширені можливості для розробки веб-застосунків, зокрема серверний рендеринг, генерацію статичних сторінок, вбудовану маршрутизацію та підтримку API-роутів. Таким чином, розробка з використанням Next.js фактично означає, що React застосовується як основа для створення інтерфейсу користувача, а фреймворк Next.js додає необхідний функціонал для побудови повноцінного веб-застосунку.

Переваги Next.js :

- підтримка SSR та SSG – ідеально для SEO та динамічних сторінок;
- дуже швидкий рендеринг на стороні користувача (відчуття мобільного застосунку);
- компонентний підхід – логіка + UI в одному місці (React);
- повна свобода у побудові фронтенду (і API-роути теж підтримуються);
- швидке створення MVP – чудово підходить для стартапів і сучасних інтерфейсів;
- ідеально інтегрується з Vercel – автодеплой, SSR/ISR, CDN [10].

Недоліки:

- немає вбудованої адмінки – потрібно створювати самостійно або підключати готові UI-бібліотеки;
 - складніша організація даних без чіткої структури (на відміну від Django);
 - авторизація реалізується вручну або через NextAuth – не завжди просто.
- Laravel – це сучасний PHP-фреймворк, що реалізує архітектурну модель

MVC (Model-View-Controller) та призначений для створення веб-застосунків будь-якої складності. Він не є мовою програмування, а виступає надбудовою над PHP, яка суттєво спрощує та структурує процес розробки. Laravel забезпечує зручні засоби для організації маршрутизації, взаємодії з базами даних через ORM Eloquent, обробки запитів, автентифікації користувачів, валідації форм, керування сесіями тощо. Завдяки цьому фреймворк дозволяє зосередитись безпосередньо на бізнес-логіці застосунку, мінімізуючи рутину та дублювання коду. Laravel активно використовується як у малих стартапах, так і в масштабних комерційних проєктах, адже поєднує зручність у використанні, гнучкість та потужну технологічну екосистему [11].

Переваги Laravel (PHP):

- все «з коробки» – роутинг, міграції, ORM, шаблони, авторизація;
- зручна адмінка (через Laravel Nova / Voyager);
- потужна екосистема – Laravel Breeze, Jetstream, Livewire, Sanctum;
- чітка структура MVC – зручно для організації коду;
- дуже швидка розробка CRUD-проєктів;
- низький поріг входу – чудово для студентів або початківців.

Недоліки:

- фронтенд менш гнучкий – Blade менш реактивний ніж React або Vue;
- менш придатний для SPA / JAMstack-підходу;
- не підходить для складних обчислень, CV, AI.

Django – це високорівневий веб-фреймворк з відкритим кодом, створений на Python. Його мета – спростити та пришвидшити розробку надійних вебзастосунків, дотримуючись принципу «DRY» і архітектури MTV (Model-Template-View). Django надає готові інструменти для роботи з базами даних, маршрутизації, валідації форм, авторизації та адміністрування. Однією з ключових переваг є вбудована адмінпанель, що дозволяє керувати контентом без додаткового коду. Завдяки підтримці бібліотек Python, Django підходить як для класичних CRUD-застосунків, так і для проєктів з аналітикою, ML або обробкою зображень.

Переваги Django (Python):

- швидкий старт: адмінка, ORM, безпека – все вбудовано;
- працює з ML / AI / OpenCV – Python дає велику перевагу в науці про дані;

- надійна система авторизації та валідації форм;
- добре підходить для захищених, масштабованих API;
- ідеальний для CRUD, REST API, адмінок, аналітики.

Недоліки:

- менше свободи на фронтенді – шаблони не такі динамічні як React;
- складніше розгортання, ніж у Laravel або Next.js (особливо без Docker);
- має більше обов’язкових структурних вимог, що може обмежувати новачків.

Для узагальнення переваг та недоліків кожного з розглянутих інструментів і визначення найбільш доцільного для розробки, було сформовано порівняльну таблицю (табл. 2.3).

Таблиця 2.3 – Порівняння фреймворків за критеріями

Критерій	Next.js	Laravel	Django
SSR/SEO оптимізація	+	–	–
Швидкість фронтенду (UX/UI)	+	–	±
Наявність вбудованої адмінки	–	+	+
Простота реалізації REST API	+	+	+
Підтримка компонентного підходу	+	–	–
Підходить для SPA / JAMstack	+	–	–
Простота старту (налаштування, документація)	+	+	±
Швидкість розробки MVP	+	+	+
Підтримка авторизації/реєстрації «з коробки»	±	+	+
Спільнота та екосистема	+	+	+

З урахуванням зазначених переваг, для реалізації веб-застосунку було

обрано стек сучасних технологій, які відповідають як функціональним, так і нефункціональним вимогам проєкту. У якості основи фронтенду використано бібліотеку ReactJS у зв'язці з фреймворком Next.js, що дозволяє реалізувати серверний рендеринг, покращити SEO та забезпечити швидке завантаження сторінок. Додатково застосовується TypeScript – надбудова над JavaScript, яка дозволяє впроваджувати статичну типізацію для підвищення надійності та зрозумілості коду [12].

HTML та CSS використовуються для структурування і базової стилізації інтерфейсу, а Tailwind CSS – як утилітарний фреймворк для створення адаптивного дизайну з мінімальними зусиллями. Це забезпечує швидке макетування і спрощує підтримку візуальної частини застосунку [13].

У якості бази даних використано MongoDB – документно-орієнтовану NoSQL-систему, що ідеально поєднується з JavaScript-стеком та дозволяє гнучко зберігати інформацію про товари, користувачів і замовлення. MongoDB забезпечує масштабованість і високу швидкодію, що є критично важливим для динамічних e-commerce платформ [14].

Серверну частину реалізовано за допомогою вбудованих API-роутів у Next.js. Це дозволяє створювати повноцінний бекенд без потреби у використанні окремого серверного фреймворку. API-роути відповідають за обробку запитів користувача, взаємодію з базою даних, автентифікацію, створення та збереження замовлень, що дає змогу зберігати єдину кодову базу для фронтенду та бекенду. Такий підхід суттєво спрощує розробку, покращує підтримку та підвищує узгодженість усіх компонентів системи.

Такий вибір технологій дозволив створити масштабований, гнучкий та ефективний інтернет-магазин, що відповідає сучасним вимогам до продуктивності, адаптивності та зручності користування.

Для реалізації проєкту було обрано професійне середовище розробки WebStorm [15]. Це середовище ідеально підходить для роботи з JavaScript, TypeScript і React, має вбудовані засоби автодоповнення, відлагодження, інтеграцію з системами контролю версій, а також підтримку сучасних

фреймворків. WebStorm забезпечує високу продуктивність, зручну навігацію по коду, гнучке налаштування середовища і стабільну роботу з великими проєктами, що робить його доцільним вибором для розробки повнофункціонального веб-застосунку.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи було сформульовано вимоги до сайту «Your Favorite» – сучасної e-commerce платформи для продажу жіночого одягу. Проведено детальний аналіз функціональних та нефункціональних вимог, що стали основою для структурування архітектури системи та визначення її ключових модулів: від вітрини товарів і кошика до адміністративної панелі та механізмів обробки замовлень.

З метою забезпечення гнучкості, масштабованості та високої продуктивності застосунку було досліджено кілька методів реалізації веб-ресурсів: від використання конструкторів сайтів і CMS до повноцінної розробки на основі фреймворків. У результаті порівняльного аналізу прийнято рішення реалізовувати веб-застосунок із нуля з використанням сучасного технологічного стеку. ReactJS, Next.js, TypeScript, Tailwind CSS для фронтенду. MongoDB – як бази даних. API-роути Next.js – для реалізації бекенду.

Крім того, для моделювання логіки взаємодії з користувачами й адміністраторами було використано діаграми UML, що дало змогу наочно відобразити функціональні процеси та переходи між компонентами системи. Як середовище розробки обрано WebStorm, що забезпечило високу ефективність роботи з JavaScript-стеком та зручність у налаштуванні, відлагодженні та управлінні кодовою базою.

Таким чином, у розділі було обґрунтовано всі технічні, логічні та архітектурні рішення, що дозволили створити надійний, сучасний і гнучкий веб-застосунок, який повністю відповідає як поточним вимогам електронної комерції, так і перспективам масштабування проєкту в майбутньому.

РОЗДІЛ 3

РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ «YOUR FAVORITE» ДЛЯ ПРОДАЖУ ЖІНОЧОГО ОДЯГУ

3.1 Практична реалізація об'єкта проєктування

Розробка інтернет-магазину «Your Favorite» здійснювалася у професійному середовищі WebStorm, яке є зручним та функціональним інструментом для створення сучасних веб-інтерфейсів. Перегляд результатів у процесі розробки відбувався через локальний сервер, що дало змогу миттєво тестувати внесені зміни у браузері без потреби у повному перезавантаженні.

Першим етапом стало створення навігаційного меню – одного з ключових компонентів інтерфейсу користувача. Цей компонент реалізовано з використанням JSX-синтаксису у React (ліст. 3.1).

Лістинг 3.1 – Код меню навігації

```
<ul className="hidden absolute top-1/2 left-1/2 transform -translate-y-
1/2 -translate-x-1/2 lg:flex lg:mx-auto lg:items-center lg:w-auto lg:gap-
24 xl:gap-32">
  {mainNav.map((item) => (
    <li key={item.title} className="text-sm whitespace-nowrap">
      <Link
        className={
          item.href === pathname
            ? 'text-black': 'text-gray-400 hover:text-black'}
        href={item.href}
        title={item.title}
        prefetch={false}
        onClick={handleLinkClick}> {item.title}
      </Link>
    </li>
  ))}
</ul>
```

кінець лістингу 3.1

Пункти меню генеруються методом «map()» по масиву «mainNav», а навігацію забезпечує компонент «Link». Стилiзація виконана з використанням Tailwind CSS для адаптивності інтерфейсу (рис. 3.1).

Рисунок 3.1 – Вигляд навігаційної панелі

У шапці реалізований логотип сайту «Your favorite», виконаний у стилізованій формі вішалки з серцем, що символізує турботу про стиль та комфорт (рис. 3.2). Колірна гама у спокійному світло-коричневому відтінку підкреслює елегантність бренду. Такий логотип використовується у верхній частині сайту як візуальний ідентифікатор компанії та є важливою частиною інтерфейсу користувача. Також цей логотип є посиланням на сторінку «Головна».



Рисунок 3.2 – Логотип вебсайту

Важливою частиною шапки сайту є компонент «CardCount», що відображає кількість товарів у кошику. Це дозволяє користувачу оперативно оцінити кількість товарів у кошику (ліст. 3.2).

Лістинг 3.2 – Елемент відображення корзини в header

```
import { cn } from '@lib/utils';
export type CardCountProps = React.ComponentPropsWithoutRef<'button'> &
{count: number;};
export function CardCount({className,count,onClick,...props}:
CardCountProps) {
  return (
    <button
      className={cn('flex items-center gap-1', className)}
      title="Відкрити кошик"
      onClick={onClick}
      {...props}>
      <svg>...</svg>
      <span>{count}</span>
    </button>);
}
```

кінець лістингу 3.2

Важливим елементом функціональності сайту є відображення бічної панелі з вмістом кошика (ліст. 3.3).

Лістинг 3.3 – Код корзини з товарами

```
{isOpened && !isDisabled && (
  <div
    className="fixed top-0 right-0 bottom-0 left-0 bg-white shadow-
md overflow-y-auto flex flex-col xl:left-auto xl:bottom-auto xl:!min-h-
[400px] xl:w-[660px] xl:max-h-screen"// @ts-ignore
    style={{ minHeight: '100vh', minHeight: '100dvh' }}>
      <div className="w-full h-[58px] px-4 border-b border-gray-400
xl:border-white flex items-center justify-between flex-shrink-0 xl:px-8
xl:pr-16">
        <CardCount disabled count={products.length} />
        <button title="Закрити кошик" onClick={handleToggle}>
          <Cross1Icon />
        </button>
      </div>
      <div className="flex flex-col flex-grow justify-between gap-4
px-4 py-5 xl:px-8 xl:pb-8">
        <ProductList
          products={products}
          onRemoveProduct={handleRemoveProduct}
          setProducts={setProducts}/>
        <div>
          <p className="mb-2">Сума: {totalAmount} €</p>
          <Link
            className="flex w-full"
            href={appLinks.checkout.href}
            title={appLinks.checkout.title}
            onClick={handleLinkClick}
            prefetch={false}>
            <Button className="w-full" variant="outline">
              {appLinks.checkout.title}
            </Button>
          </Link>
        </div>
      </div>
    </div>
  )}
```

кінець лістингу 3.3

У цьому компоненті виводиться список товарів, сума замовлення, а також кнопка для переходу до оформлення. Усі зміни у кошику відображаються у реальному часі без перезавантаження сторінки (рис. 3.3).

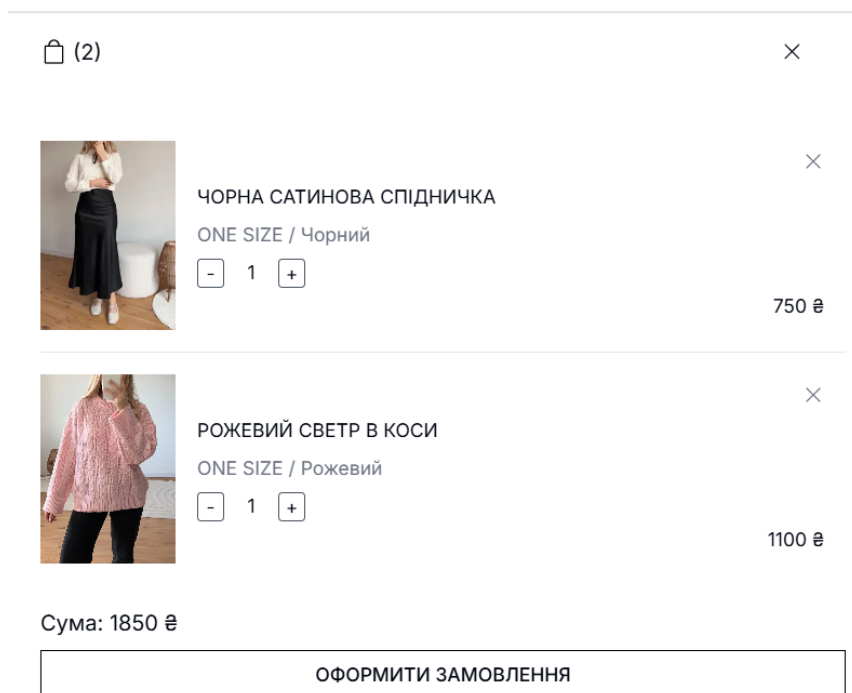


Рисунок 3.3 – Відображення кошика з товарами

Дані про товари та категорії зчитуються з бази даних MongoDB за допомогою асинхронних запитів. У лістингу 3.4 наведено приклад виводу перших двох категорій товарів на головній сторінці.

Лістинг 3.4 – Вивід перших двох категорій на головній сторінці

```
const categories = await Category.find().limit(2)
const data = await Promise.all(categories.map(async (value) => {
const products = await Product.find({category: {$elemMatch: { name:
value.name }}}).limit(3);
    return JSON.parse(JSON.stringify(products.map(value => value)));
}));
{data.map((value, index) => (
    <section key={index}>
        <Divider />
        <Products data={value} category={categories[index].name}/>
    </section>
))}
```

кінець лістингу 3.4

Для кожної категорії здійснюється запит на отримання до трьох відповідних товарів, що дозволяє гнучко оновлювати контент відповідно до змін у базі даних (рис. 3.4).

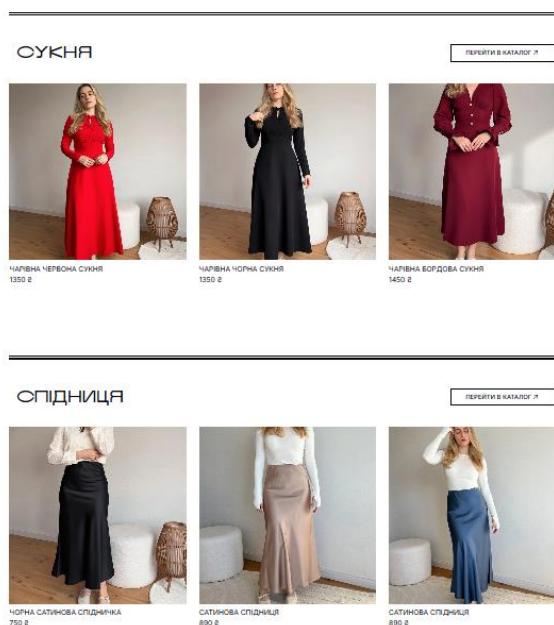


Рисунок 3.4 – Відображення перших двох категорій товарів на головній сторінці сайту

Каталог товарів генерується динамічно, що зображено у лістингу 3.5. У запиті враховано лише ті товари, що доступні до продажу, а список категорій формується як зі статичних, так і динамічних значень. Дані передаються до компонента «Catalog», де реалізовано клієнтську фільтрацію та адаптивний рендеринг залежно від пристрою.

Лістинг 3.5 – Збір каталогів з бази даних

```
export default async function Page() {
  await mongooseConnect();
  noStore()
  const data = await Category.find();
  const products = JSON.parse(JSON.stringify(await Product.find({ count:
{$ne: 0} }))))
  let categoriesList = [
    {name: "Вся колекція", value: "all"}, {name: "Sale", value: "sale"}];
  data.map((value) => categoriesList.push({name: value.name, value:
value.name}));
  return (
    <Suspense>
      <Catalog categoriesList={categoriesList} data={products}/>
    </Suspense>);
}
```

кінець лістингу 3.5

Клієнтський компонент «Catalog» відповідає за відображення товарів відповідно до обраної категорії. За допомогою React-хуків «useState», «useEffect» та «useSearchParams» здійснюється фільтрація товарів згідно з параметрами з URL. Визначення типу пристрою (десктоп або мобільний) відбувається через хук «useMediaQuery», що дозволяє динамічно обирати відповідний компонент для виведення товарів: «ProductsDesktop» або «ProductsMobile» (ліст. 3.6).

Лістинг 3.6 – Відображення каталогів на клієнті

```
const [filter, setFilter] = useState('all');
const isDesktop = useMediaQuery('(min-width: 768px)');
const Component = isDesktop ? ProductsDesktop : ProductsMobile;
const params = useSearchParams();
const filterParam = params.get('filter');
useEffect(() => {
  if (filterParam) {setFilter(filterParam);}}, [filterParam]);
return (
  <div className="pt-24 min-h-screen">
    <Container className="flex gap-y-[10px] gap-x-[20px] sm:gap-x-[40px] flex-wrap uppercase px-[20px]">
      {categoriesList.map(value => (
        <div key={value.value}
          className={cn(
            filter == value.value ?
              'text-black border-b-[1px] border-black font-[500]' :
              'text-[#696969]', 'cursor-pointer',
            value.value === "sale" ?
              'text-red-500 border-red-500 font-[500]' : '',))}
          onClick={() => setFilter(value.value)}>
            {value.name}
          </div>
        )}}
    </Container>
    {typeof isDesktop !== 'undefined' && <Component filter={filter}
data={data}/>}
  </div>
);
};
```

кінець лістингу 3.6

Відображення категорій реалізовано у вигляді інтерактивного списку, де активна категорія підсвічується, а категорія «Sale» має червоне оформлення. У

результаті користувач отримує гнучкий, адаптивний і зручний інтерфейс для перегляду та фільтрації товарів (рис. 3.5).

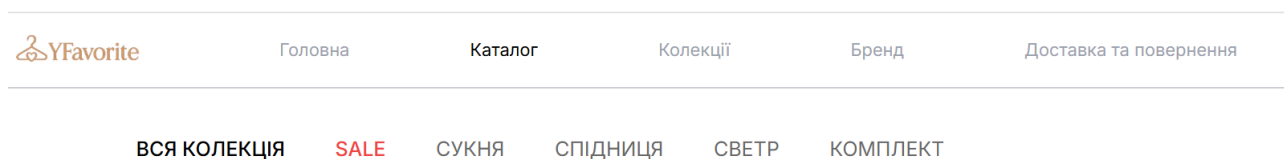


Рисунок 3.5 – Відображення каталогу товарів, отриманого з бази даних

Для відображення товарів відповідно до обраної категорії використовується клієнтський компонент, що реалізує фільтрацію масиву товарів за параметрами з бази даних. Категорії формуються у вигляді інтерактивного списку, де активна категорія підсвічується, а категорія «Sale» має окреме червоне оформлення. Виведення результатів здійснюється у вигляді адаптивної сітки з товарами, де кожен елемент містить зображення, назву, ціну та позначку «COMING SOON» для недоступних позицій. Компонент побудований з використанням JSX і Tailwind CSS з урахуванням доступності та підтримкою адаптивного рендерингу (рис. 3.6).

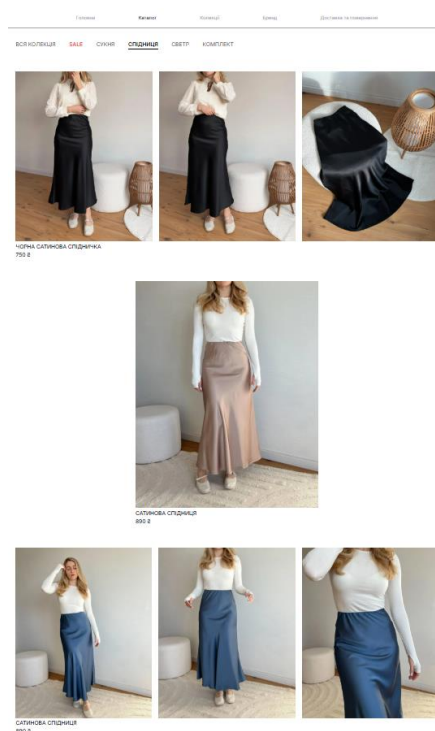


Рисунок 3.6 – Відображення товарів відповідно до обраної категорії

Натиснувши на відповідний товар у каталозі, користувач потрапляє на сторінку детального перегляду, де відображається повноцінна картка товару. У ній міститься велике зображення, назва, ціна, кнопка додавання до кошика, інформація про розмір, склад тканини, параметри моделі та контактні дані. Такий підхід забезпечує зручність ознайомлення з товаром і сприяє прийняттю рішення про покупку (рис. 3.7).

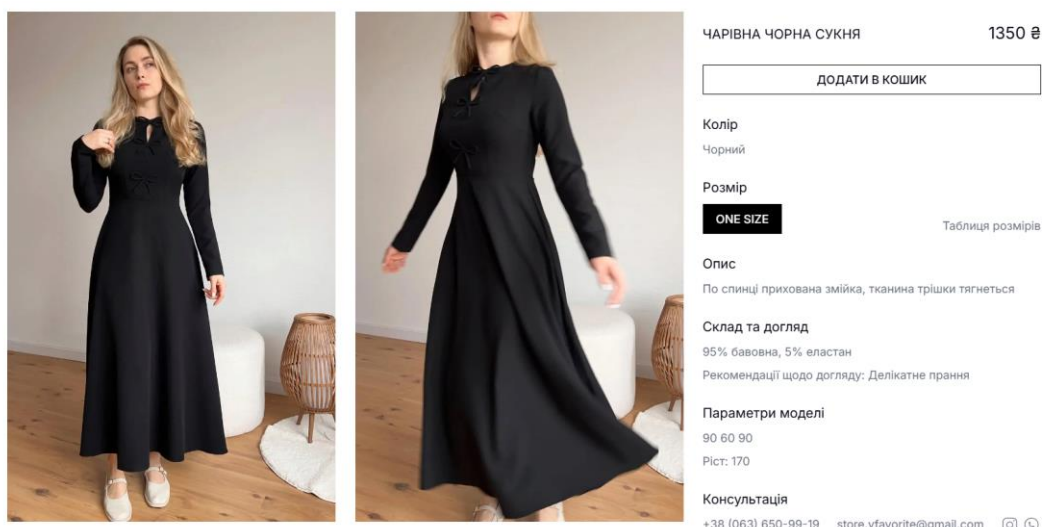


Рисунок 3.7 – Картка товару

Після переходу на сторінку оформлення замовлення користувач взаємодіє з продуманим, інтуїтивно зрозумілим інтерфейсом, який логічно поділений на два основні функціональні блоки – форму введення даних та список товарів у кошику (рис. 3.8). У формі введення даних передбачено послідовне заповнення особистої інформації: ім'я, прізвище, номер телефону, електронна пошта. Далі користувач переходить до блоку доставки, де необхідно вибрати місто та відділення служби доставки, які реалізовані за допомогою автозаповнення через сторонній API.

Елементи форми мають чітку структуру, логічне розміщення та супровідні підказки, що зменшує ймовірність помилок при заповненні. Такий підхід забезпечує зручність для користувача та точність даних, що надходять до системи обробки замовлень.

ОФОРМЛЕННЯ ЗАМОВЛЕННЯ

СПИСОК ПОКУПОК

ОСОБИСТІ ДАНІ

Ім'я

Ліза

Прізвище

Приходько

Номер телефона

+38 (098) 000 00 00

E-mail

user@gmail.com

ДОСТАВКА

Місто

Введіть назву

Відділення

Введіть відділення

ОФОРМИТИ ЗАМОВЛЕННЯ

×

ЧАРІВНА ЧОРНА СУКНЯ

ONE SIZE / Чорний

−

1

+

1350 ₴

×

ЧАРІВНА БОРДОВА СУКНЯ

ONE SIZE / Бордовий

−

1

+

1450 ₴

Сума товарів:

2800 ₴

Рисунок 3.8 – Сторінка «Оформлення замовлення»

Для перевірки коректності введених даних використано схему валідації на основі бібліотеки «zod». Для кожного поля задано правила – мінімальна кількість символів, обов'язковість заповнення, формат (наприклад, email). У разі помилки система автоматично генерує повідомлення. Це зменшує ймовірність некоректних даних і підвищує якість введення (ліст. 3.7), що позитивно впливає на користувацький досвід та процес оформлення замовлення.

Лістинг 3.7 – Схема валідації форми

```
const formSchema = z.object({
  firstName: z.string().min(2, {
    message: 'Ім'я повинно містити щонайменше 2 символи',}),
  lastName: z.string().min(2, {
    message: 'Прізвище повинно містити щонайменше 2 символи',}),
  phoneNumber: z.string().min(10, {
    message: 'Номер телефону занадто короткий',}),
  email: z.string().email({
    message: 'Невірний формат електронної пошти',}),
  city: z.string().min(2, {
    message: 'Назва міста повинна містити щонайменше 2 символи',}),
  buildingNumber: z.string().min(1, {
    message: 'Необхідно вказати номер будинку або будівлі',}),
  DATA: z.string().optional(),});
```

кінець лістингу 3.7

Щоб забезпечити зручний огляд замовлення перед підтвердженням, праву частину сторінки займає блок зі списком доданих товарів. Усі позиції відображаються через компонент «ProductList», що дозволяє змінювати кількість товарів або повністю видаляти їх зі списку. Лістинг 3.8 демонструє реалізацію цього динамічного елемента. У нижній частині блоку автоматично підраховується загальна сума замовлення. Завдяки використанню класів Tailwind CSS, верстка компонента є адаптивною, що забезпечує коректне відображення як на мобільних пристроях, так і на екранах з великою роздільною здатністю.

Лістинг 3.8 – Відображення списку покупок

```

<div className="md:order-2 md:w-1/2">
  <h2
    className="hidden md:block md:font-semibold md:uppercase md:mb-
8 md:pb-4 md:border-b md:text-xl xl:text-2xl">
    Список покупок
  </h2>
  <div
    className={` ${
      isOpened ? 'flex' : 'hidden'
    } fixed z-50 top-0 right-0 bottom-0 left-0 bg-white overflow-y-auto flex-col
min-h-screen md:flex md:static md:min-h-0`}>
    <div
      className="w-full h-[58px] px-4 border-b border-gray-400 flex
items-center justify-between flex-shrink-0 md:hidden">
      <button type="button" title="Закрити список"
onClick={handleToggle}>
        <Cross1Icon />
      </button>
    </div>
    <div className="flex flex-col justify-between gap-4 px-4 py-5
md:p-0 md:divide-y">
      <ProductList
        products={products}
        setProducts={setProducts}
        onRemoveProduct={handleRemoveProduct}/>
      <div className="flex flex-col gap-4 md:py-5">
        <p className="flex justify-between">
          <span>Сума товарів:</span> <span>{totalAmount} €</span>
        </p>
      </div>
    </div>
  </div>
</div>

```

кінець лістингу 3.8

Для автоматичного підбору відділень доставки у формі реалізовано інтеграцію з API Нової пошти. Лістинг 3.9 демонструє функцію «handleAddressChangeBuilding», яка виконує асинхронний запит до сервера з метою отримання актуального списку відділень на основі вибраного міста. Запит надсилається методом «POST», а відповідь обробляється та зберігається у стані компонента для подальшого відображення підказок користувачу. Такий підхід забезпечує зручність при введенні адреси та мінімізує ризик помилок при виборі пункту доставки.

Лістинг 3.9 – Пошук адресу через API нової пошти

```

async function handleAddressChangeBuilding(
  event: React.ChangeEvent<HTMLInputElement>,
  suggestionProp?: any,
) {
  const query = event.target.value;
  const url = 'https://api.novaposhta.ua/v2.0/json/getWarehouses';
  const options = {
    method: 'POST',
    mode: 'cors',
    headers: {
      'Content-Type': 'application/json',
      Accept: 'application/json',
    },
    body: JSON.stringify({
      "apiKey": "e260c4c0c7c5b3cb0eff584d20b54511",
      "modelName": "AddressGeneral",
      "calledMethod": "getWarehouses",
      "methodProperties": {
        "SettlementRef": suggestionProp?.Ref || suggestion.Ref,
        "Page": "1"
      }
    }),
  } as const;
  try {
    const response = await fetch(url, options);
    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }
    const { data } = await response.json();
    setSuggestionsBuilding(data);
  } catch (error) {
    console.log('error', error);
  }
}

```

кінець лістингу 3.9

На завершальному етапі оформлення замовлення формується структура зібраних даних, яка надсилається на сервер для подальшої обробки (ліст. 3.10). Усі поля, зокрема ім'я, електронна пошта, телефон, номер замовлення, сума, а також серіалізовані деталі доставки, упаковуються у вигляді об'єкта «dataS» та передаються через запит «axios.post» на відповідний внутрішній маршрут «/api/checkPayment». Незважаючи на відсутність повноцінної платіжної інтеграції, така структура є готовою для подальшого підключення до платіжного сервісу за потреби.

Лістинг 3.10 – Відправка ведених даних на сервер

```
const dataS = {
  terminalkey: { value: '1695905921618' },
  frame: { value: 'false' },
  language: { value: 'ua' },
  amount: { value: `${amountTotal}` },
  order: { value: `${orderNumber}` },
  description: { value: '' },
  name: { value: fio },
  email: { value: email },
  phone: { value: phone },
  receipt: { value: receiptValue },
  DATA: { value: JSON.stringify(DATA) },
};
axios.post('/api/checkPayment', dataS)
```

кінець лістингу 3.10

У веб-застосунку «Your Favorite» окрім функціональних сторінок каталогу та оформлення замовлення передбачено також додаткові інформаційні розділи, які виконують важливу роль у формуванні повноцінного користувацького досвіду. До таких сторінок належать «Колекції», «Бренд» та «Доставка та повернення».

Сторінка «Колекції» виступає вітриною магазину, де представлено актуальні модні капсули (рис. 3.9). Користувач може обрати одну з декількох капсул (наприклад, «1st Capsule», «2nd Capsule», «Trophy») і переглянути пов'язану з нею продукцію. Кожна колекція подається у візуально привабливому форматі – з якісними зображеннями моделей і вбрання, що сприяє натхненню та

допомагає користувачу швидко зорієнтуватися у стилістичних пропозиціях бренду. Цей розділ є ключовим для просування тематичних товарів і створення естетичної атмосфери навколо продукту.



Рисунок 3.9 – Сторінка «Колекції»

Сторінка «Бренд» містить інформацію про філософію компанії та її місію. У цьому розділі розповідається про унікальний погляд бренду на моду, орієнтований на витонченість, жіночність і зручність. Тут також пояснюється прагнення зробити кожен етап взаємодії з сайтом – від перегляду до доставки – максимально комфортним. Розділ оформлено з використанням елегантних зображень та стислих текстових блоків, що дозволяє ефективно передати цінності бренду.

На сторінці «Доставка та повернення» надається користувачу вся необхідна інформація про логістику товарів. Тут описано умови доставки, терміни, правила повернення й обміну, а також можливі способи зв'язку зі службою підтримки. Це дозволяє забезпечити прозорість процесу оформлення замовлень і підвищити довіру з боку клієнтів.

У межах реалізації сайту також була розроблена адміністративна панель, яка забезпечує зручне керування контентом інтернет-магазину. Вона складається з кількох основних розділів, доступних через бокове меню: «Головна», «Товари», «Каталоги» та «Замовлення».

Головна сторінка адміністративної панелі містить дашборд (рис. 3.10), який відображає основні показники роботи магазину: кількість замовлень і суму прибутку за сьогодні, тиждень і місяць. Таке представлення даних у вигляді окремих блоків дозволяє адміністратору оперативно оцінювати динаміку продажів та фінансову ефективність, що сприяє прийняттю своєчасних управлінських рішень. Інтерфейс реалізований у зрозумілому візуальному форматі, що забезпечує швидкий доступ до ключової аналітики.

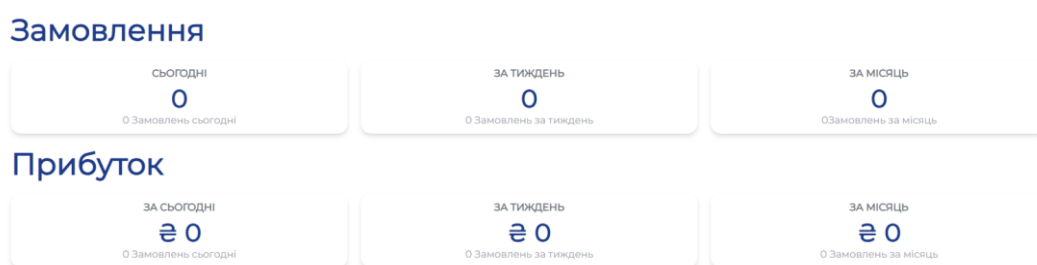


Рисунок 3.10 – Інформаційний дашборд головної сторінки адміністративної панелі

У розділі «Товари» реалізовано зручний інтерфейс керування асортиментом у вигляді візуальних карток, кожна з яких містить зображення товару, його назву, ціну в гривнях та кнопку видалення у вигляді іконки червоного кошика. Такий підхід дозволяє швидко орієнтуватися у списку доступних позицій, візуально ідентифікувати продукти та оперативно реагувати на необхідність їх редагування або вилучення. У верхній частині сторінки розміщено кнопку з іконкою «+», яка відкриває інтерфейс для додавання нового товару до бази, що забезпечує ефективне оновлення асортименту без зайвих переходів або складних форм (рис. 3.11). Візуальне оформлення з чіткою

структурою та розміткою підвищує зручність роботи адміністратора та забезпечує позитивний користувацький досвід.

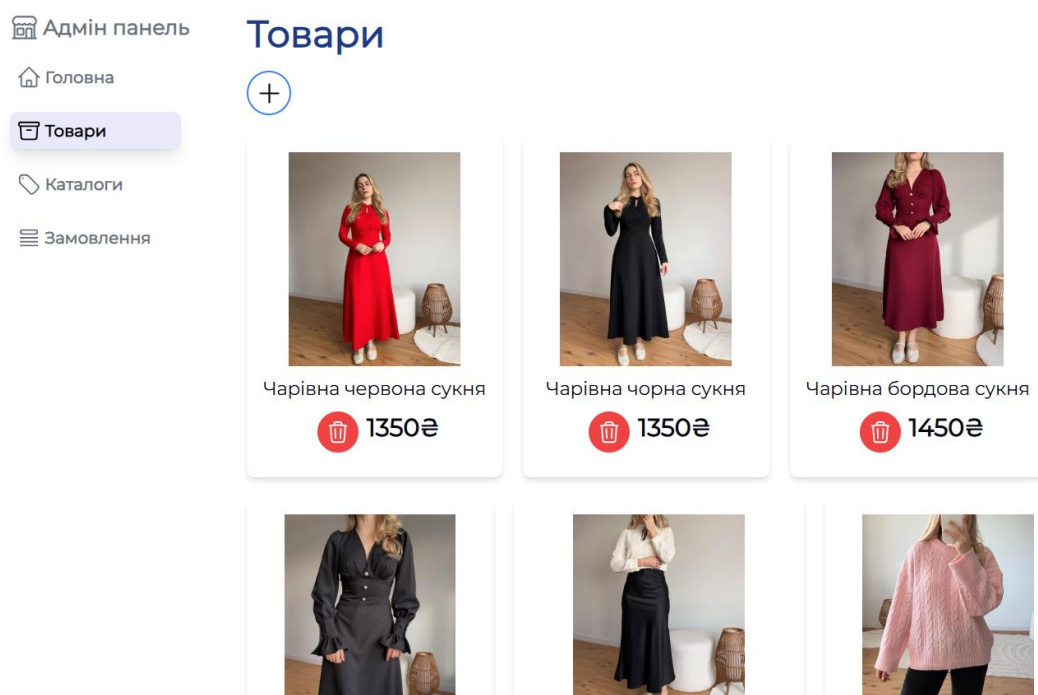


Рисунок 3.11 – Інтерфейс управління товарами в адміністративній панелі

У вкладці «Каталоги» реалізовано функціонал створення, редагування та видалення категорій товарів. Інтерфейс включає текстове поле для введення назви нового каталогу та таблицю з переліком уже створених категорій, що дозволяє зручно керувати товарною структурою магазину (рис. 3.12).

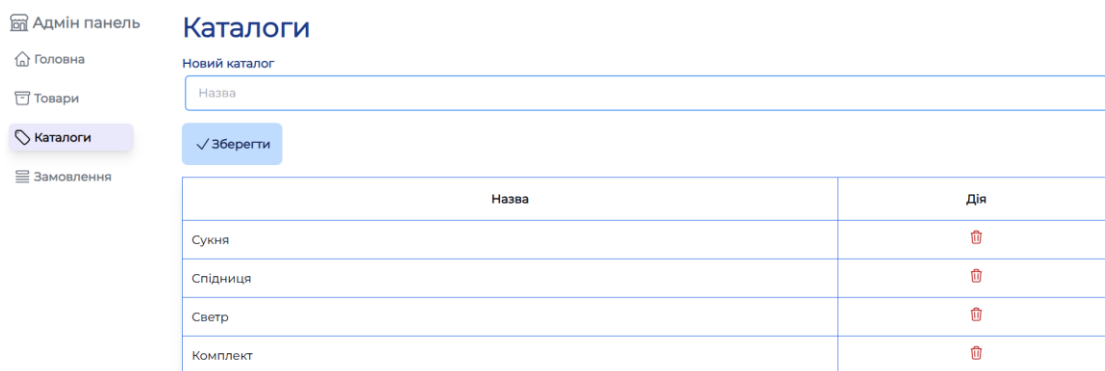


Рисунок 3.12 – Інтерфейс керування каталогами товарів

У результаті створення сайту «Your Favorite» вдалося реалізувати повноцінний інтернет-магазин із зручним інтерфейсом як для користувачів, так і для адміністратора. Магазин містить каталог товарів, кошик, форму оформлення замовлення, а також адміністративну панель для керування товарами, категоріями й замовленнями. Завдяки сучасним технологіям сайт працює швидко, адаптується до різних пристроїв і забезпечує комфортний досвід для відвідувачів.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Для перевірки якості пошукової оптимізації та загальної продуктивності веб-застосунку було проведено аудит за допомогою інструмента Lighthouse у Google Chrome. Цей інструмент дозволяє оцінити сайт за кількома критеріями: ефективність «Performance», доступність «Accessibility», дотримання найкращих практик «Best Practices» та оптимізація для пошукових систем «SEO».

Під час основного тестування комп'ютерної версії сайту було зафіксовано задовільні результати. Зокрема, ефективність становила 79 балів, що вважається прийнятним значенням для веб-застосунків із великою кількістю графічного контенту. Показник доступності склав 95, відповідність найкращим практикам – 96, а «SEO» – 100, що є високими показниками (рис. 3.13). Такі результати свідчать про якісну реалізацію та технічну відповідність основним вимогам, водночас вказуючи на потенціал для подальшої оптимізації завантаження ресурсів і стилів.

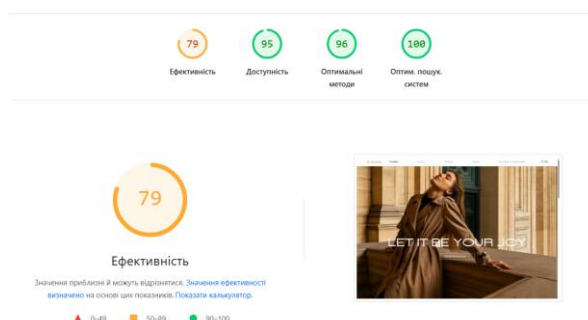


Рисунок 3.13 – Тестування комп'ютерної версії

Окрім тестування десктопної версії, аналогічне дослідження було проведено і для мобільної версії сайту. За результатами тестування мобільна версія отримала високі оцінки: ефективність – 91, доступність – 93, «Best Practices» – 96, «SEO» – 100 (рис. 3.14), що свідчить про якісну адаптацію інтерфейсу та відповідність сучасним стандартам веб-розробки.

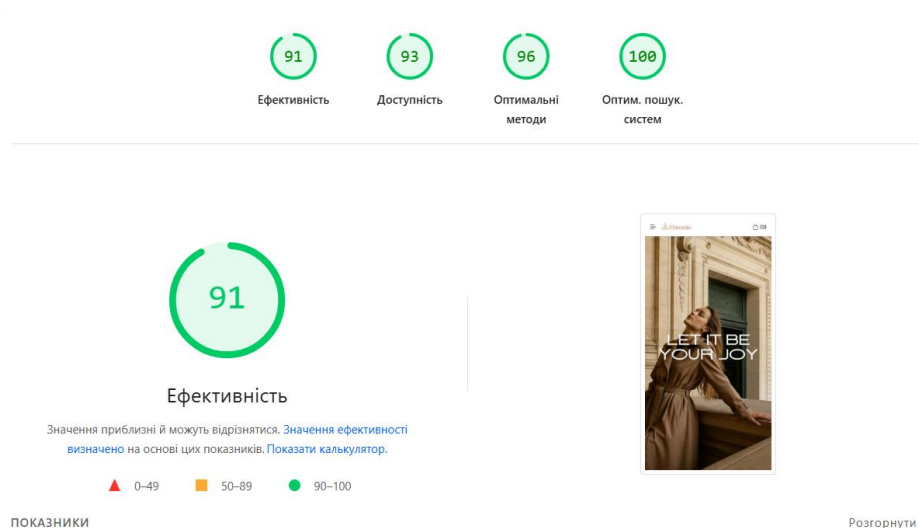


Рисунок 3.14 – Тестування мобільної версії

Було проведено ручне тестування коректного відображення веб-сайту «Your Favorite» на пристроях із різною шириною екрана з метою перевірки адаптивності інтерфейсу. Перевірка здійснювалася на кількох типах пристроїв: планшеті з шириною 768 пікселів, ноутбучі з шириною 1024 пікселі та мобільному телефоні з шириною 360 пікселів. У результаті тестування встановлено, що всі основні елементи інтерфейсу (меню, каталог товарів, кошик, форма оформлення замовлення) коректно масштабується відповідно до розмірів екрана. Особливу увагу приділено відображенню мобільної версії (рис. 3.15), оскільки вона є найважливішою для більшості користувачів. Інтерфейс на смартфоні адаптується до вертикального перегляду, основні кнопки залишаються зручними для натискання, а блоки автоматично перебудовуються у вертикальні стеки. Це підтверджує, що сайт є повністю адаптивним і придатним для використання на різних пристроях.

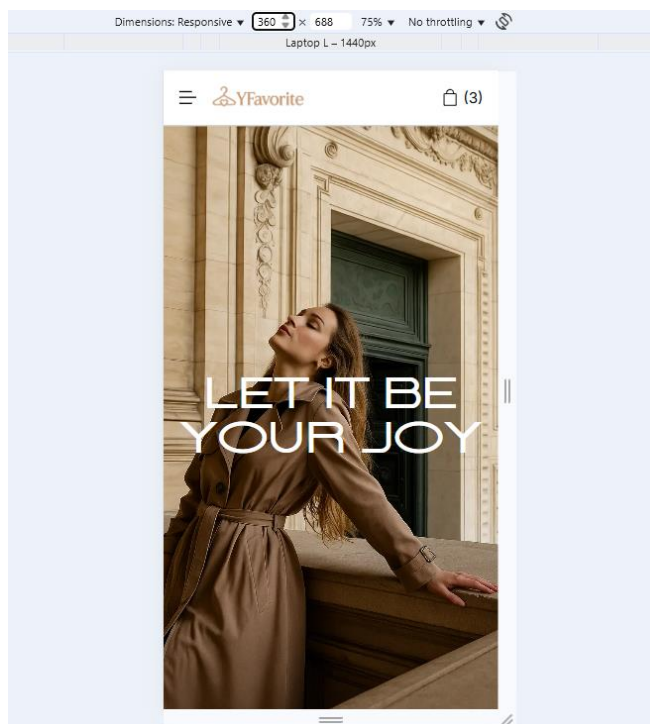


Рисунок 3.15 – Відображення сайту на екрані шириною 360 пікселів

У результаті проведеного тестування підтверджено, що всі елементи сайту «Your Favorite» коректно адаптуються до різних типів пристроїв та ширини екранів. Інтерфейс масштабувався без спотворень, усі блоки зберігали логіку компонування та відображалися зручно як на ноутбучі, так і на планшеті чи смартфоні. Це свідчить про якісну реалізацію адаптивного дизайну з використанням Tailwind CSS, що забезпечує гнучке налаштування інтерфейсу.

Крім візуального аналізу, було також вручну протестовано функціональність основних елементів сайту. Перевірці підлягали навігаційне меню, картки товарів, бічна панель кошика, форма оформлення замовлення та інтеграція з API Нової пошти. Усі функції працюють стабільно та без збоїв: додаються та видаляються товари з кошика, коректно обраховується загальна сума, валідація форми на основі бібліотеки «Zod» виконується згідно з заданими умовами, а динамічне завантаження списку відділень здійснюється згідно з обраним містом. Помилки у роботі ключових компонентів не виявлено, що підтверджує надійність реалізованої логіки та високий рівень готовності проєкту до реального використання.

3.3 Розробка бази даних

У процесі створення веб-сайту «Your Favorite» для зберігання, обробки та подальшого використання інформації про товари, замовлення та категорії була реалізована база даних на основі MongoDB. MongoDB є документно-орієнтованою системою, що дозволяє гнучко оперувати структурами типу «JSON» без суворого визначення схем, що особливо зручно для веб-розробки з динамічно змінюваним контентом.

Для підключення до бази даних у Next.js застосовується функція «mongooseConnect», яка перевіряє стан з'єднання та ініціалізує підключення при потребі (ліст. 3.11). Адреса з'єднання зберігається у змінній середовища «MONGODB_URI».

Лістинг 3.11 – Функція підключення до бази даних

```
export function mongooseConnect() {
  if (mongoose.connection.readyState === 1) {
    return Promise.resolve(mongoose.connection);
  } else {
    const uri = process.env.MONGODB_URI;
    if (!uri) {
      console.error("MongoDB URI is not provided");
      return Promise.reject("MongoDB URI is not provided");
    }
    return mongoose.connect(uri).catch(error => {
      console.error("Failed to connect to MongoDB", error);
      return Promise.reject(error);
    });
  }
}
```

кінець лістингу 3.11

Далі було створено схему «ProductSchema» (ліст. 3.12), яка чітко визначає структуру документів для зберігання повної інформації про товари. У цій схемі зазначено обов'язкові поля, типи даних, масиви зображень, категорії, розміри, знижки, склад тканини та інші ключові характеристики, необхідні для правильного відображення товарів на сайті.

Лістинг 3.12 – Схема таблиці для зберігання замовлень

```

import mongoose, {Schema, model, models} from "mongoose";
const ProductSchema = new Schema(
  {
    name: {
      type: String,
      required: true,},
    description: {
      type: String,},
    images: [{type: String}],
    price: {
      type: Number,
      required: true,},
    url: {
      type: String,
      required: true},
    color: {
      type: String,
      required: true},
    composition: {
      type: String,
      required: true},
    care: {
      type: String,
      required: true},
    modelParam: {
      type: String,
      required: true},
    modelHeight: {
      type: String,
      required: true},
    sale: {
      type: Number,
      required: true},
    sizeList: [{type: String}],
    count: {
      type: Number,
      required: true },
    similar: [{
      _id: Schema.Types.ObjectId,
      name: String }],
    justify: {
      type: String,
      required: true},
    heightRow: {type: Number,required: true},
    countImageRow: {type: Number,required: true},
    category: [{_id: mongoose.Types.ObjectId, name: String}], },
  {timestamps: true,});
export const Product = models.Product || model("Product", ProductSchema);

```

кінець лістингу 3.12

Робота з даними здійснюється за допомогою «Mongoose» – бібліотеки, яка забезпечує зв'язок між MongoDB та JavaScript-кодом. Завдяки їй можна легко створювати, оновлювати, читати й видаляти записи через методи моделі «Product», а також використовувати фільтрацію, сортування чи агрегацію даних.

Використання MongoDB у поєднанні з Next.js та TypeScript дозволило досягти високої швидкодії, простоти у масштабуванні структури даних і зручності подальшого розширення функціоналу веб-застосунку.

3.4 Розміщення сайту на вибраній платформі хостингу

Після завершення етапів розробки та тестування веб-застосунку постала потреба в його розгортанні для публічного доступу. Для цього було обрано сучасну хостинг-платформу Vercel, яка ідеально інтегрується з фреймворком Next.js, підтримує CI/CD-підхід та дозволяє зручно керувати деплоєм через GitHub.

Процес розміщення проєкту на Vercel розпочинається зі створення облікового запису та переходу до головної панелі інтерфейсу, де пропонується імпортувати проєкт з Git-репозиторію (рис. 3.16).

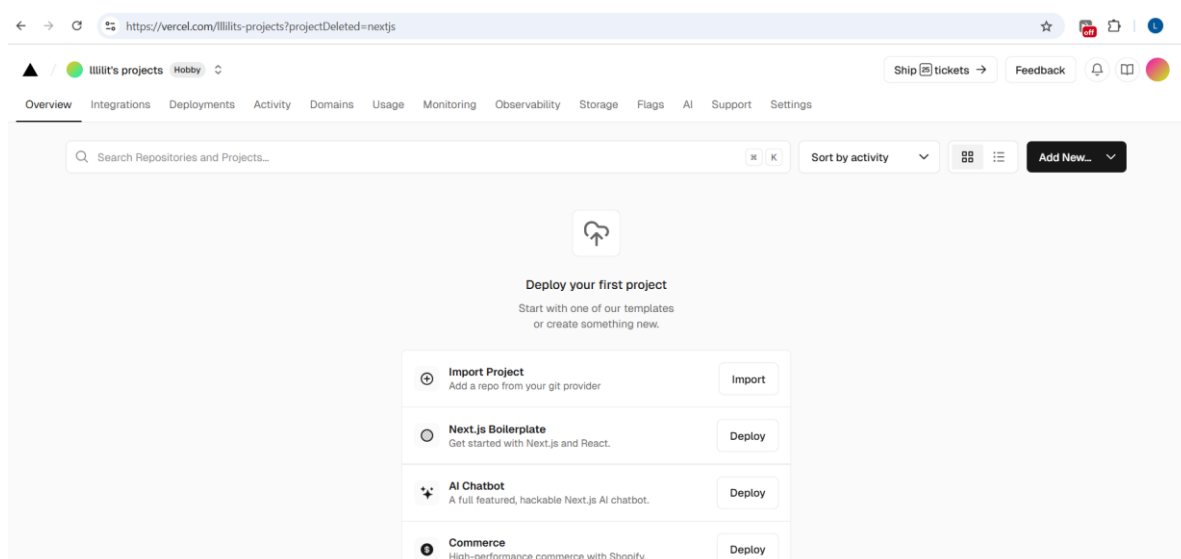


Рисунок 3.16 – Інтерфейс платформи Vercel для початку деплою

Далі користувач обирає конкретний проєкт із власного GitHub-акаунту, після чого система автоматично визначає тип фреймворку (у цьому випадку – Next.js) та надає можливість встановити параметри збірки, змінні середовища та директорію проєкту. Після підтвердження налаштувань розпочинається процес деплою (рис. 3.17).

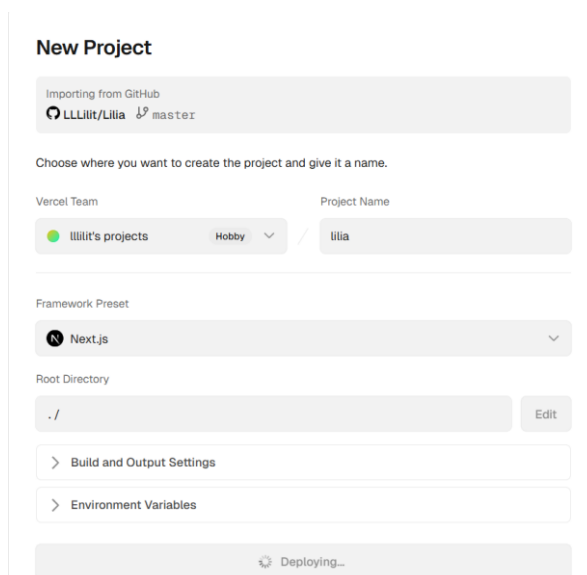


Рисунок 3.17 – Запуск процесу деплою проєкту

Успішний деплой проєкту відображається в панелі керування розгортанням, де зазначено статус, гілку репозиторію, дату останнього оновлення та посилання на активну версію сайту (рис. 3.18).

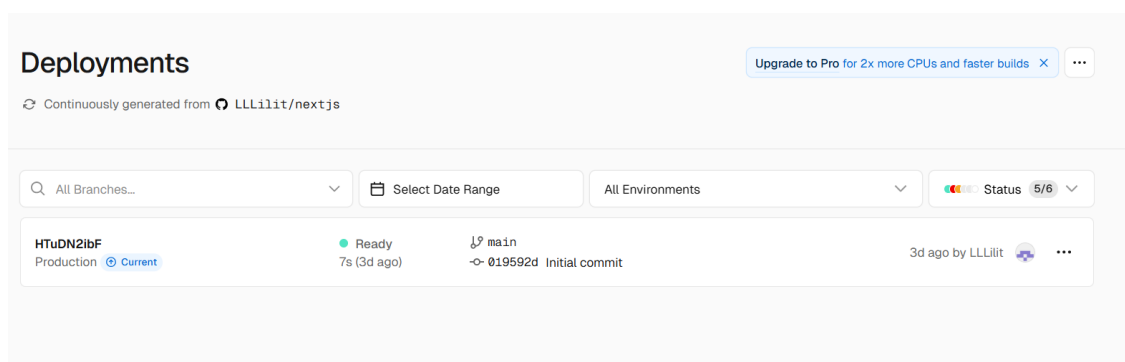


Рисунок 3.18 – Інтерфейс панелі керування деплоєм на Vercel

Таким чином, завдяки зручності платформи Vercel та її глибокій інтеграції

з GitHub, вдалося швидко та безпомилково розгорнути готовий веб-застосунок на хостингу, що забезпечує його стабільну та безперебійну роботу в онлайновому режимі.

3.5 Безпека розроблюваного веб-сайту

У веб-сайті «Your Favorite» реалізовано комплекс заходів, спрямованих на захист як користувацьких даних, так і внутрішніх компонентів системи. Першочергову роль у безпеці відіграє структура архітектури: взаємодія з базою даних здійснюється виключно через серверну частину, без прямого підключення з клієнтської сторони. Це дозволяє уникнути витоку чутливих даних, таких як URI з обліковими даними або токени доступу. Усі параметри підключення до MongoDB зберігаються в захищених змінних середовища на сервері, що унеможливорює їх вивід або перехоплення через браузер.

Адміністративна панель, яка надає можливість керування контентом та обробки замовлень, також захищена надійним механізмом авторизації. Доступ до цієї частини сайту мають лише користувачі з роллю адміністратора. Паролі зберігаються у зашефрованому вигляді з використанням сучасних алгоритмів шифрування, що запобігає їх розшифруванню навіть у разі витоку даних. Крім того, при невірному введенні паролю користувач перенаправляється на головну сторінку без розголошення причин відмови у доступі, що знижує ризик атак типу brute force.

Додаткову безпеку веб-сайту забезпечує хостинг-платформа Vercel, на якій розміщено проєкт. Однією з ключових переваг цієї платформи є автоматичне застосування SSL/TLS сертифікатів, що забезпечує передачу даних по захищеному протоколу HTTPS. Це дозволяє уникнути ризиків перехоплення або підміни інформації під час комунікації між клієнтським пристроєм та сервером. Наявність SSL гарантує автентичність сервера, захист даних від змін у процесі передавання та їх шифрування.

Усі дії, що відбуваються на клієнтській стороні – введення даних у форму,

взаємодія з кошиком, перегляд товарів – передбачають додаткову валідацію на сервері. Наприклад, валідація замовлення перед його відправкою на обробку, обмеження доступу до критичних API-ендпоінтів, а також контроль прав доступу.

Загалом, розроблений веб-сайт відповідає сучасним стандартам безпеки та забезпечує захист на всіх основних рівнях: мережевому, серверному та клієнтському. Це дозволяє гарантувати конфіденційність персональних даних користувачів, стабільність функціонування системи та надійність адміністрування.

Висновки до розділу 3

У третьому розділі було детально описано процес розробки, тестування, розгортання та забезпечення безпеки веб-застосунку «Your Favorite» – інтернет-магазину жіночого одягу. Було реалізовано повноцінний функціонал, що охоплює всі етапи взаємодії користувача з сайтом: від навігації, перегляду товарів і фільтрації каталогу до оформлення замовлення та інтеграції з API служби доставки. Завдяки використанню сучасного технологічного стеку – Next.js, React, TypeScript, Tailwind CSS, MongoDB та Vercel – вдалося досягти високої продуктивності, адаптивності та зручності у користуванні.

Було проведено тестування працездатності веб-застосунку за допомогою Lighthouse, яке підтвердило високі показники ефективності, доступності, дотримання найкращих практик і оптимізації для пошукових систем як у десктопній, так і мобільній версіях. Ручне тестування на різних пристроях підтвердило коректність адаптивного рендерингу та стабільність функціонування ключових елементів інтерфейсу.

Також було створено надійну базу даних на основі MongoDB, яка забезпечує ефективне зберігання та обробку даних про товари, категорії та замовлення. Розгортання веб-застосунку на платформі Vercel забезпечило

швидке й стабільне підключення сайту до продакшн-середовища з можливістю CI/CD-оновлень.

Особливу увагу приділено безпеці: усі конфіденційні дані ізольовано на серверному рівні, передача даних здійснюється через захищений протокол HTTPS, а доступ до адміністративної панелі обмежений механізмами автентифікації. Усі ці заходи дозволяють гарантувати захист персональної інформації користувачів і стабільність роботи сайту.

Таким чином, розроблений інтернет-магазин повністю відповідає вимогам до сучасного e-commerce ресурсу, є технічно досконалим, зручним для користувачів і готовим до подальшого масштабування та вдосконалення.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було виконано всі поставлені у вступі завдання, що дозволило досягнути основної мети – створення повнофункціонального веб-сайту інтернет-магазину жіночого одягу «Your Favorite».

На першому етапі було обґрунтовано актуальність теми розробки. Проведений аналіз показав стрімкий розвиток онлайн-торгівлі, особливо у сфері продажу одягу. Було встановлено, що існуючі інтернет-магазини часто мають недоліки, пов'язані з функціональністю, адаптивністю, персоналізацією та користувацьким досвідом, що підкреслило необхідність створення нової якісної платформи.

Далі були визначені функціональні та нефункціональні вимоги до веб-ресурсу. Серед функціональних вимог – реалізація перегляду товарів, кошика та оформлення замовлення. Нефункціональні вимоги включали адаптивність інтерфейсу, високу продуктивність, безпеку передачі даних і зручність користування.

Наступним кроком став огляд сучасних методів і підходів до розробки веб-сайтів. Було проаналізовано використання CMS, конструкторів сайтів, ручного програмування та фреймворків. У результаті порівняння обґрунтовано доцільність використання технологій React, Next.js, Tailwind CSS і MongoDB для реалізації даного проєкту.

Відповідно до визначених вимог було реалізовано веб-сайт «Your Favorite». Розроблено адаптивний інтерфейс користувача з навігаційною панеллю, каталогом товарів, інтерактивним кошиком і формою оформлення замовлення. Усі компоненти виконано із застосуванням JSX, React-хуків та стилізовано за допомогою Tailwind CSS. Крім клієнтської частини, було також створено адміністративну панель, яка дозволяє керувати товарами, категоріями та замовленнями в зручному візуальному інтерфейсі.

Здійснено тестування працездатності веб-сайту. Було проведено як автоматизовану перевірку за допомогою інструменту Lighthouse, так і ручне тестування на різних пристроях. Результати показали високу ефективність роботи сайту, коректну адаптацію до різних екранів та відсутність критичних помилок у функціонуванні.

Окремо було спроектовано та реалізовано базу даних на основі MongoDB. Створено схеми для зберігання інформації про товари, категорії та замовлення. Для взаємодії з базою застосовано бібліотеку «Mongoose», що забезпечило зручну та безпечну роботу з документами.

Після завершення розробки веб-сайт було успішно розгорнуто на хостинг-платформі Vercel. Весь процес розгортання автоматизовано завдяки інтеграції з GitHub, що дозволило забезпечити зручне керування версіями, CI/CD-підхід і швидке оновлення застосунку.

З метою забезпечення безпеки веб-застосунку було впроваджено комплекс заходів. Ключі доступу до бази даних зберігаються у захищених змінних середовища, взаємодія з базою здійснюється виключно з серверної частини. Адмінпанель захищено механізмом авторизації, дані передаються через протокол HTTPS із застосуванням SSL-сертифіката, а валідація форм реалізована як на клієнтському, так і на серверному рівнях.

Таким чином, усі завдання, визначені на початковому етапі кваліфікаційної роботи, були повністю реалізовані. Створений веб-сайту «Your Favorite» є готовим інструментом для впровадження в електронну торгівлю, який поєднує технічну досконалість, зручність використання та високий рівень безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Вплив технологій на інтернет-продажі: огляд новітніх інструментів та рішень. URL: https://webpromoexperts.net/ua/blog/vpliv-tehnologiy-na-internet-prodazhi-oglyad-novitnih-instrumentiv-ta-rishen/?utm_source=chatgpt.com (дата звернення: 06.02.2025).
2. Аналіз електронної комерції як драйвера глобалізації: можливості для бізнесу та ризику. URL: https://www.researchgate.net/publication/386114595_Analiz_elektronnoi_komercii_ak_drajvera_globalizacii_mozlivosti_dla_biznesu_ta_riziki (дата звернення: 06.02.2025).
3. Новітні тренди та перспективи розвитку електронної комерції в міжнародному бізнесі. URL: https://economyandsociety.in.ua/index.php/journal/article/view/584?utm_source=chatgpt.com (дата звернення: 08.02.2025).
4. Технології для розробки сайту: що це, які найпопулярніші та як вибрати. URL: <https://iampm.club/ua/blog/tehnologiyi-dlya-rozrobki-sajtu-shho-cze-yaki-najpopulyarnishi-ta-yak-vibrati/> (дата звернення: 19.02.2025).
5. Wix vs Weebly in 2025. Discover Why Users Prefer Wix. URL: <https://www.websitebuilderexpert.com/website-builders/comparisons/wix-vs-weebly/> (date of access: 19.02.2025).
6. Weebly Website Builder Review. URL: <https://adwservice.com.ua/en/weebly-website-builder-review> (date of access: 25.02.2025).
7. What are the pros and cons of a Wix website. URL: <https://www.seaglow.co.uk/news/what-are-the-pros-and-cons-of-a-wix-website> (date of access: 28.02.2025).
8. Joomla vs WordPress: which CMS is best for you. URL: <https://illustrate.digital/wordpress/joomla-vs-wordpress/> (date of access: 28.02.2025).
9. Django vs Next.js: Choosing the Right Framework Between Django and Next.js for Your Web Development. URL: <https://nomadicsoft.io/next-js-vs-django-choosing-between-django-and-nextjs-for-your-project/> (date of access: 28.02.2025).

10. Next.js on Vercel. URL: <https://vercel.com/docs/frameworks/nextjs> (date of access: 28.02.2025).
11. Переваги фреймворку Laravel. URL: <https://brander.ua/blog/perevahy-freymvorku-laravel> (дата звернення: 01.03.2025).
12. Використання TypeScript. URL: <https://uk.react.dev/learn/typescript> (дата звернення: 10.03.2025).
13. Introduction to Tailwind CSS. URL: <https://www.geeksforgeeks.org/introduction-to-tailwind-css/> (date of access: 10.03.2025).
14. Основи MongoDB. URL: <https://devzone.org.ua/post/osnovy-mongodb> (дата звернення: 10.03.2025).
15. Опис продукту JetBrains WebStorm. URL: <https://itpro.ua/catalog/catalogarticle/view/jetbrains-webstorm/?tab=description> (дата звернення: 10.03.2025).

ДОДАТКИ

Додаток А

Апробація на XIX Міжнародній науково-практичній конференції «Current Problems of the Development of Science: Trends and Innovations»



CURRENT PROBLEMS OF THE DEVELOPMENT OF SCIENCE: TRENDS AND
INNOVATIONS

9.	Яцемірська Н.П. ЗБЕРЕЖЕННЯ ЕКОСИСТЕМ НАЦІОНАЛЬНОГО ПРИРОДНОГО ПАРКУ "ПОДІЛЬСЬКІ ТОВТРИ" ТА ЗАБЕЗПЕЧЕННЯ ЕКОЛОГІЧНОЇ РІВНОВАГИ НАВКОЛИШНЬОГО СЕРЕДОВИЩА ЧЕРЕЗ РОБОТУ НА ОСВІТНІХ ПЛАТФОРМАХ	45
CHEMISTRY		
10.	Таран А., Клебанська М., Пожіткова Л., Велічко Т. КАТАБОЛІЗМ ГЛЮКОЗИ ТА АДАПТАЦІЙНІ СТРАТЕГІЇ У БАКТЕРІЙ З ВИСОКИМ БІОТЕХНОЛОГІЧНИМ ПОТЕНЦІАЛОМ	47
COMPUTER SCIENCE		
11.	Andrushchak I., Shemeiko L. FUNCTIONAL AND TECHNICAL FEATURES OF MODERN WEB DEVELOPMENT	50
12.	Kiyo Yakishima HYBRID RELAXED BROAD LEARNING SYSTEM FOR COMMUNITY-BASED DENTAL SCREENING UNDER DATA IMBALANCE	55
13.	Melnykova N., Marikutsa U., Ptashynskyi M. APPLICATION OF MACHINE LEARNING TECHNIQUES FOR ALZHEIMER'S DISEASE STAGE CLASSIFICATION	58
14.	Xiaobin Liu ADAPTIVE BROAD LEARNING FOR IMBALANCED ORAL DISEASE DIAGNOSIS USING STRUCTURED CLINICAL DATA	63
15.	Zhipeng Hong, Paul Gutsch ADAPTIVE RELAXED BROAD LEARNING FOR IMBALANCED FINANCIAL RISK DETECTION	66
16.	Мельник М.Є. ВІДКРИТА НАУКА ТА КІБЕРБЕЗПЕКА: ВИКЛИКИ ЦИФРОВОЇ ДОБИ	70
17.	Співак І.Я., Крепич С.Я., Стадник С.О. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СИСТЕМИ ОЦІНКИ РІВНЯ КОМПЕТЕНТНОСТІ ЕКСПЕРТІВ	73

FUNCTIONAL AND TECHNICAL FEATURES OF MODERN WEB DEVELOPMENT

Andrushchak Igor,

Doctor of technical sciences, Professor

Shemeiko Lilia,

Student of the SEGs-21

Lutsk National Technical University

This article reveals such a concept as WEB development, the procedure for creating a WEB application or WEB site, also describes the main stages of website development, methods of training using an existing web resource, and touches on the specifics of choosing a CMS.

Keywords: WEB development, WEB applications, WEB sites, programming, Internet site.

WEB development is a process of professional website programming that regulates all subsequent stages related to the formation of HTML code, adding various functional components and scripts that affect usability and technical stability.

The developer's task is to create an infrastructure for the implementation of the optimal functionality of the site, to which visual components are attached in the future, providing interactive capabilities.

In other words, in more accessible language, WEB development is the procedure for creating a WEB application or a WEB site [1].

Let's highlight the main stages of WEB site development:

- designing a WEB application or the site itself, that is, collecting and subsequently analyzing all requirements, developing technical specifications, drafting an interface project;
- developing a site concept taking into account creativity;
- developing a design concept for an Internet resource;
- developing site page layouts;
- creating and implementing FLASH elements and multimedia;
- layout of templates and pages;
- software work, such as creating functional tools, or integrating into an existing content management system;
- posting on the site and optimizing its text materials;
- testing the site and making adjustments if necessary;
- launching the created project on a public platform on the Internet;
- maintenance work on an existing portal or its software.

However, depending on the required task, some of the above stages in the WEB development process can be used or be closely interconnected with each other.

COMPUTER SCIENCE
CURRENT PROBLEMS OF THE DEVELOPMENT OF SCIENCE: TRENDS AND
INNOVATIONS

It should be noted that all stages of a WEB site project are quite dependent on many different factors, such as the size of the Internet portal, its functionality, as well as the tasks for which the created Internet resource is intended and much, much more. But, nevertheless, there are also a number of stages that are certainly present when planning absolutely any future project.

Any work on an Internet site begins with creating its design, usually using a graphic editor for this. A WEB designer usually creates several such options, but in strict accordance with the technical specifications [2].

In this case, the design of the "Main" page of the site is developed separately. And then - the design of the other standard pages (news, articles, about us, catalog). In fact, the "design" itself is a graphic file, like a layered drawing, including smaller pictures in the form of layers in the overall picture (Pic. 1).



Picture 1. – Step guide to WEB-development

At the same time, the specialist must take into account all the limitations of the HTML standard, that is, he does not develop a design that cannot be subsequently implemented by standard HTML tools. The only exception is Flash design.

The number of sketches themselves and the order of their presentation to the customer are agreed upon in advance with the managers of the entire project, who monitor the planned deadlines. In addition, in large WEB companies, the Art Director also takes part in the process, monitoring the quality of the graphics. This stage, just like the previous one, ends with its approval by the customer.

The design approved by the client is then transferred to a layout specialist, who "cuts" the graphic image into separate pictures, from which the HTML page will later be composed. In the course of such work, a program code is created, which can already be viewed using any browser (Internet browser). Well, and as such, these typical pages will subsequently be used as HTML templates. After the above-mentioned activities have been carried out, the finished files in HTML format are transferred to the WEB programmer for work. The development of software for an Internet site can be carried out both "from scratch" and on the basis of a CMS system, often the so-called "CMS engine".

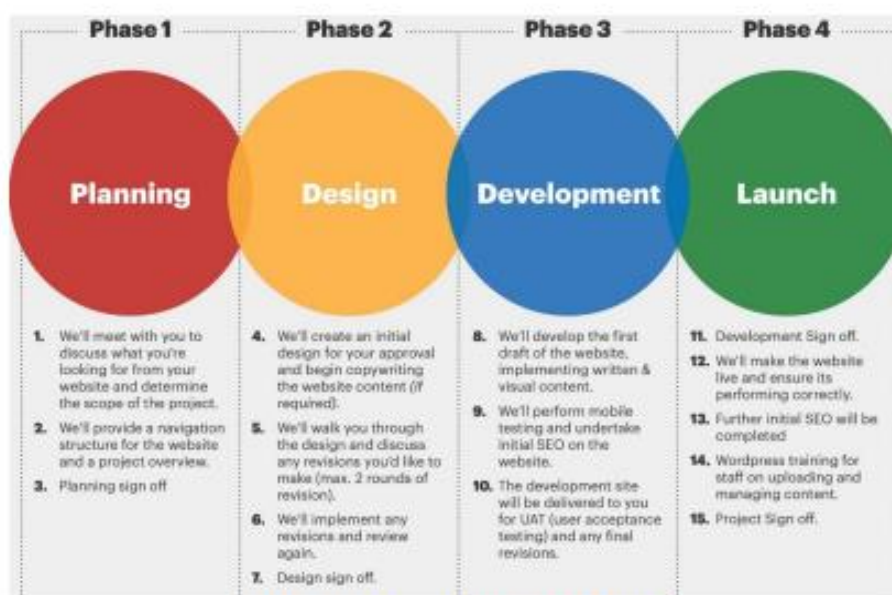
COMPUTER SCIENCE
CURRENT PROBLEMS OF THE DEVELOPMENT OF SCIENCE: TRENDS AND
INNOVATIONS

In case of using a content management system, it should be noted that it itself, in a sense of the word, is already a ready-made site, including replaceable blocks. Well, and the programmer himself, in this case, will be more correctly called a "CMS specialist".

He must replace the existing standard template with a new original one, developed on the basis of the initial WEB design, taking into account the individual wishes of the customer [3].

When developing software for an Internet site, a CMS specialist is also given deadlines for completing the work.

This process itself may well contain a variety of checks, such as, for example: the appearance of a site page with enlarged fonts, with different browser window sizes, or due to the lack of a Flash player, and much more. User testing, the so-called usability, is also used (Pic. 2).



Picture 2. – Our website development process

Detected errors in the site's operation are sent for correction until the contractor eliminates them. In this case, the same project manager controls the work deadlines. Although, at the testing stage, the designer himself is also involved in the work so that he can carry out author's supervision.

The files of the developed WEB-site are placed on the server, for example, of the provider, where the necessary settings are made. It should be noted that at this stage the Internet site is still closed to a wide range of users [4].

COMPUTER SCIENCE
CURRENT PROBLEMS OF THE DEVELOPMENT OF SCIENCE: TRENDS AND
INNOVATIONS

The new Internet site is filled with content, that is, various texts, images, files for downloading, and the like. Sometimes the texts themselves are written by a copyright specialist of the company that developed the site, sometimes the client himself does it, hiring an author from outside. This issue is resolved at the stage of drawing up the TOR itself, since if the content is developed by the developer's copywriter, then this must be agreed with the customer simultaneously with other stages of the project being implemented.

Internal optimization is associated with certain changes made to the Internet site itself.

The optimization process begins with the compilation of the so-called semantic core, for which keywords are selected that will capture the greatest attention of portal users, and with which it is easier to bypass competitors. Subsequently, such "keywords" are entered into the Internet portal itself and all hyperlinks, texts and meta tags are adapted in such a way that Internet search engines can quickly detect them using the compiled "keywords".

External SEO, as a rule, comes down to building a certain structure for incoming links to a new site. In principle, this is the promotion of a new portal itself and external optimization has nothing to do with the creation of a site. SEO optimization itself is divided into the so-called: "white" and "black", after the first, the Internet portal gets into the TOP, and after the second, it is "banned" by search engines.

It should be noted that "white" optimization is a rather long and labor-intensive process, the cost of which can exceed several times the material costs of WEB development of the site itself. The customer gets acquainted with the fully completed order and, if he is completely satisfied with everything, then the relevant documents are signed, such as the Certificate of Completion of Work, on the delivery of the entire project to him. In addition, at this stage of work, the client or his representative is also trained in the necessary skills for administering the new site.

As a side note, it should be noted that the WEB design of the website being developed must certainly look attractive when users use different browsers, especially browsers such as Chrome, Internet Explorer, Safari, Firefox and Opera [5].

Let's highlight the advantages of web applications compared to conventional programs:

- ease of use. Providing instant access to the application, you only need to have an Internet connection and a working browser;
- multi-user system. Several users can work in the web application at the same time; low cost of installation and maintenance. The web application is installed only on the server. There is no longer a need to separately install and configure on each computer, since further settings are made through a remote server;
- centralized storage and protection of information. All data is in one place, where changes are instantly saved and backup copies are created;
- the ability to adjust information by users;
- constant revision and development.

Thus, a rich arsenal of technologies, the ability to choose a cooperation model depending on the project features and customer wishes, the creation of complex sites

COMPUTER SCIENCE
CURRENT PROBLEMS OF THE DEVELOPMENT OF SCIENCE: TRENDS AND
INNOVATIONS

of any subject, flexibility and transparency in project management allows you to get a high-quality and effective project as a result of development.

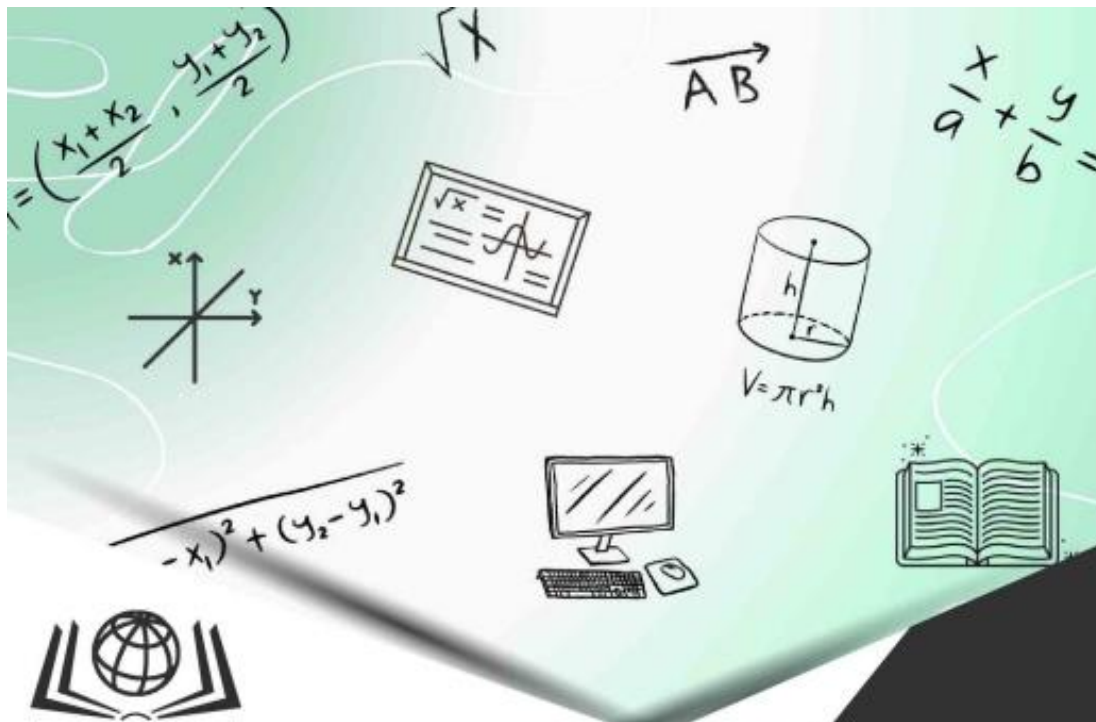
References:

1. Andrushchak I., Koshelyuk V., Yasashny D. Security Audit of Orchestration of Light Container Platforms. Modern Trends in the Development of Economy, Technology and Industry: Collection of Scientific Papers "International Scientific Unity" with Proceedings of the 3International Scientific and Practical Conference. April 9-11, 2025. Toronto, Canada. 170-174 p. ISBN 979-8-89704-989-9 (series) DOI 10.70286/ISU-09.04.2025 <https://isu-conference.com/en/archive/modern-trends-in-the-development-of-economy-technology-and-industry-09-04-25/>
2. Martsenyuk V., Andrushchak I., Kit N., Kravchik Yu., Sverstyuk A., Palyanytsya Yu. Comparison of the results of numerical analysis of modeling of cyberphysical biosensor systems /Bulletin of the Khmelnytsky National University. 2023. Vol. 1, No. 2, 2023 (319). P. 102–108. DOI: <https://www.doi.org/10.31891/2307-5732-2023-319-2> (Index Copernicus, Google Scholar).
3. Nakonechnyi A., Martsenyuk V., Sverstiuk A., Andrushchak I., Loiko I. Network modeling of coexistence of virus strains admitting chaotic behavior / / Ceur Workshop Proceedings / 2020, 2753, pp. 56-61.
4. Stewart J.M., Kinsey D. Network security, firewalls, and VPNs. Burlington: Jones & Bartlett Learning, 2021. 482 p.
5. Vermeir Nico. Introducing .NET 6. Getting Started with Blazor, MAUI, App Windows SDK, Development Desktop, and Containers. Apress, 2022. 319 p.



Додаток Б

Апробація на XX Міжнародній науково-практичній конференції «Innovative Scientific Research: Latest Theories, Modern Methods and Practices»



International Science Group
ISG-KONF.COM

XX
INTERNATIONAL SCIENTIFIC
AND PRACTICAL CONFERENCE
«INNOVATIVE SCIENTIFIC RESEARCH: LATEST
THEORIES, MODERN METHODS AND PRACTICES»
Seville, Spain
May 20 – 23, 2025

ISBN 979-8-89692-711-2
DOI 10.46299/ISG.2025.1.20

INNOVATIVE SCIENTIFIC RESEARCH: LATEST THEORIES, MODERN METHODS AND PRACTICES

10.	Коц В.П., Коц С.М., Коц В.В. АНТИТІЛА ТА АНТИГЕНИ І МЕХАНІЗМ ЇХ УТВОРЕННЯ	48
11.	Манжай Ю.А., Шейко В.І. ПОКАЗНИКИ НЕСПЕЦИФІЧНОЇ ЛАНКИ СИСТЕМОГО ІМУНІТЕТУ СПОРТСМЕНІВ ІГРОВИХ ВИДІВ СПОРТУ ПРИ ВЖИВАНІ АФЛУБІНУ	52
CHEMISTRY		
12.	Klimko Y., Levandovskii S. REACTION OF ACYLIMINIUM SALTS WITH AMINO ACID ESTERS	55
13.	Кизим О. ВИЗНАЧЕННЯ ГЛЮКОЗИ У БАД З ІНУЛІНОМ МЕТОДОМ ПОТЕЦЕНЦІОМЕТРИЧНОГО ТИТРУВАННЯ	57
COMPUTER SCIENCE		
14.	Andrushchak I., Shemeiko L. OPTIONS FOR TECHNICAL SECURITY IN THE OPERATION OF MODERN WEB SITES	59
15.	Andrushchak I., Malsky V. STEP-BY-STEP ANALYSIS OF SECURITY FEATURES OF MODERN CONTENT MANAGEMENT SYSTEMS	64
16.	Kiyo Yakishima MULTI-SOURCE BROAD LEARNING INTEGRATION FOR AUTOMATED DENTAL RADIOGRAPH DIAGNOSIS UNDER CLASS IMBALANCE	71
17.	Kiyo Yakishima ADAPTIVE BROAD LEARNING SYSTEM FOR DENTAL IMPLANT FAILURE RISK PREDICTION USING CLINICAL AND RADIOGRAPHIC INDICATORS	73
18.	Mirzagalikova B., Kenzhebekova R., Kovaleva K., Bakhytzhan N. PROGRAMMING-DRIVEN ANALYSIS OF GENETIC SUSCEPTIBILITY TO ASTHMA: A CASE-CONTROL STUDY USING PYTHON TOOLS	76

OPTIONS FOR TECHNICAL SECURITY IN THE OPERATION OF MODERN WEB SITES

Andrushchak Igor,

Doctor of technical sciences, Professor

Shemeiko Lilia,

Student of the SEGs-21

Lutsk National Technical University

The article describes the most common types of attacks and various vulnerabilities with the help of which hackers manage to penetrate and hack a software product. The relevance of this topic is due to the fact that, despite the high progress in the field of system security, many systems, including websites, can still be hacked attackers.

Keywords: vulnerability, protection, information protection, Dos-attack, server, information security.

Most modern information systems are created in the form of websites, so security must be given special attention. As soon as a device with an operating system and software application appears, it immediately attracts the interest of intruders. They begin to study it and hack it. In the last year, many reports at major hacker conferences have been devoted to this trend.

Developers do not always pay special attention to the complete protection and security of the created sites, although it plays a very important role, because each site can be hacked by hackers in one way or another, given that sites can be very diverse: electronic libraries, social networks, web portals of various educational institutions, online stores, official sites of various organizations, banks [1].

First of all, it is necessary to pay attention to the security of the web server, since it is responsible for receiving and processing HTTP requests from clients to the website. It is he who ensures the functioning of numerous websites around the world, and is also responsible for basic services and stores personal data of both users and visitors to the site. The need to protect servers is one of the most important tasks of any organization.

Nowadays, it would seem that such threats should become much less or not exist at all, but, alas, this is far from the case: as before, a fairly large number of cyber attacks are now being carried out, mainly on such global organizations as government agencies and banks. This is due to the fact that on such content one can "earn" a large amount of money or obtain/change information that can somehow influence events important for the state. A rather important point is that the geographic location of the server does not affect its protection in any way. An attack on it can be carried out from any access point. This is due to the fact that web servers, due to their openness, are designed to transfer information between users and that is why they have many vulnerabilities. For example, an attacker can make some changes (modifications) to the code of the HTTP

server or database server, or the pages of the website themselves, changing its original functionality [2].

The most common actions of cyber attackers on hacked sites, which are often used to this day:

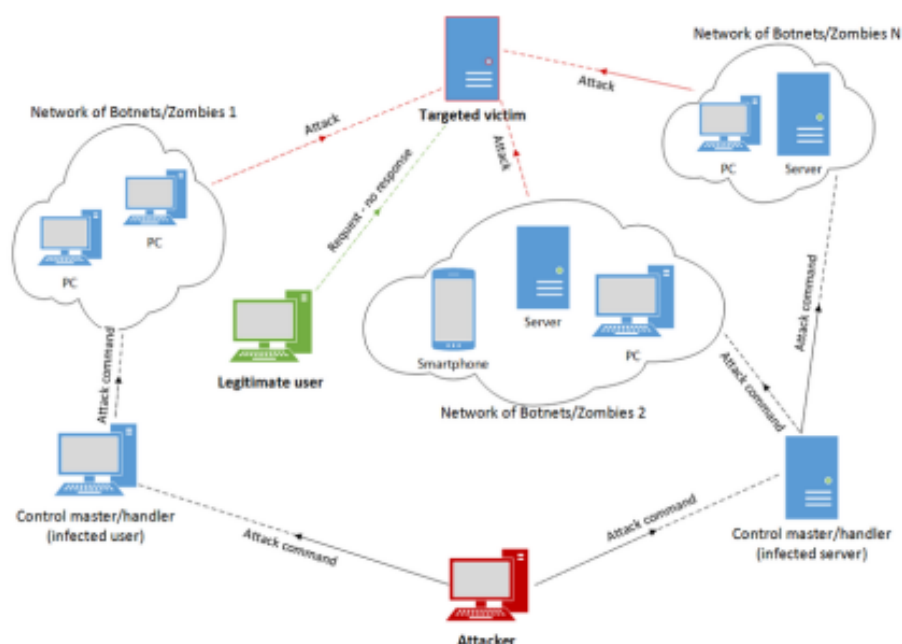
- published false information on the site;
- gained unauthorized access to the server, where very valuable information was collected;
- attacked the database of bank card holders; stole data concerning the account numbers of the main bank card holders, their names, surnames, contact information, and then stole a large amount of money;
- created viruses that easily cracked passwords, and then used the infected machines to send spam or as databases for storing stolen information;
- launched attacks on the websites of official (government) institutions of various countries;
- distributed viruses that led to a significant slowdown in the speed of the Internet (there were cases when, as a result of such viruses, some regions were completely cut off from the network);
- Dos-attacks (Denial of Service) on the network with the purpose of flooding it with additional requests so that useful traffic is either reduced to a minimum or interrupted altogether;
- gained access to information related to SecurID technology, which is used to ensure the security of corporate computer networks (the leakage of this information can lead to a "reduction in the effectiveness" of the current implementation of SecurID, which, in turn, can become the basis for attacks on protected resources).

Currently, there are various programs, as well as special Web testing, which can determine the approximate security and reliability factor of a site. Thanks to such testing, it is possible to assess the security of a site and its vulnerability in advance [3].

It is assumed that every third site is monitored by third parties. This is a rather unpleasant and alarming fact for web developers. Having all the necessary information about the site's vulnerabilities, it can be easily hacked, so the main task of developers in the issue of reliability and protection of the site is to prevent as many of these vulnerabilities as possible or at least hide them properly.

Laboratories for the prevention of various viruses have begun to pay most attention to the development of means to counteract DDoS attacks (Denial of Service - "denial of service") (Pic. 1).

COMPUTER SCIENCE
INNOVATIVE SCIENTIFIC RESEARCH: LATEST THEORIES, MODERN METHODS AND
PRACTICES



Picture 1. – The principle of implementation of a Dos attack

There is also cyber espionage, when attackers do not attack an information system or website, but simply steal all the necessary information for their own benefit.

Larger enterprises approach information security with much greater responsibility than small and medium-sized businesses, which is why these companies suffer the most. According to statistics, they are the ones that have more incidents in the context of information security.

According to research, about 90% of detected malware are banking Trojans that steal information about bank cards.

There are also special programs that block the contents of a device (in this context, specifically a website) until a “ransom” is paid.

When creating a new website, developers can use either existing platforms, or more precisely CMS (content management system), or write everything from scratch, but most often small companies prefer to use ready-made tools.

Main functions of CMS:

- providing tools for creating content, organizing collaborative work on content;
- content management: storage, version control, compliance with access mode, document flow management;
- publishing content;
- presenting information in a form convenient for navigation and search.

The main disadvantage of CMS is that they also have their own vulnerabilities, which can usually be found on the Internet.

The main victims of DDoS attacks are mainly online stores and trading platforms (Pic.2).



Picture 2. – Process Dos attack

Every year, more and more new CMS updates are released, and at the same time, new vulnerabilities appear: although it is also necessary (to note) to pay attention to the fact that some old vulnerabilities are prevented.

But, despite all the concerns, one should not think that it is completely impossible to protect a web server, you just need to make some efforts to ensure its security. This will require joint actions of website administrators, programmers and designers, and it is also necessary to take into account that antivirus software, operating systems and access rights constantly require special attention [4].

The most popular methods of protecting CMSs from various attacks:

1. Protection from XSS injections.
2. Hiding unnecessary information.
3. Forced use of SSL.
4. Root file protection.
5. Methods of preventing spammers and bots.
6. Using various plugins to protect against malicious URL requests.
7. Using methods to prevent leechers.
8. "Destroying" the admin.
9. Protecting directories on the server from viewing.

The implementation of modern security tools is an integral part of information security measures [5].

Protecting your site is one of the most important tasks for web developers. If you do not pay special attention to it, you can acquire (create) a lot of losses, both materially and politically. Due to the fact that the interest of intruders in attacks on web sites does

not cease even now (considering that all malicious acts are punishable by law), developers must first of all think about its security. There are many means of protecting web sites, but no matter how effective they are, it is still necessary to periodically test web sites for vulnerabilities that a hacker can use at any time. Thus, the main advice to developers and administrators of web sites is to repeatedly check the level of its reliability and security.

References:

1. Andrushchak I., Shemeiko L. Functional and technical features of modern web development. Proceedings of the XIX International Scientific and Practical Conference. Oslo, Norway. 2025. Pp. 50-55. URL: <https://isg-konf.com/current-problems-of-the-development-of-science-trends-and-innovations/>
2. Martsenyuk V., Andrushchak I., Kit N., Kravchik Yu., Sverstyuk A., Palyanytsya Yu. Comparison of the results of numerical analysis of modeling of cyberphysical biosensor systems /Bulletin of the Khmelnytsky National University. 2023. Vol. 1, No. 2, 2023 (319). P. 102–108. DOI: <https://www.doi.org/10.31891/2307-5732-2023-319-2> (Index Copernicus, Google Scholar).
3. Mitchell C., Konev A. Ensuring the security of websites. // Australia: SophosLabs. [Electronic resource]. URL: <http://help/webmaster/protecting-sites/contents.xml>
4. Smirnov S.N. Security of database systems. Helios ARV, 2007. – 352 p.
5. Vitaken R. DDoS attacks and small business [Electronic resource]. URL: http://bezshablona.ospfmon.com/2014/02/ddos_4866.html?m=0

