МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ЕКОНОМІКО-ПРАВНИЧИЙ ФАХОВИЙ КОЛЕДЖ
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ»

Циклова комісія математичних дисциплін та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ПРОЄКТУВАННЯ ТА РОЗРОБКА ГАЛЕРЕЇ ОНЛАЙН МАГАЗИНУ ОДЯГУ»

Виконала: здобувач освіти 4 курсу, групи К121-21
Спеціальність 121 Інженерія
програмного забезпечення
(шифр і спеціальність)
Каміла ГОРБ
(ім'я та ПРІЗВИЩЕ)
Керівник Юлія БОРИСОВСЬКА
(ім'я та ПРІЗВИЩЕ)
Рецензент доцент каф. програмної інженерії ЗНУ, к.т.н., доцент Юлія ЛИМАРЕНКО
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ)

Запоріжжя
2025

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ВСП «Економіко-правничий фаховий коледж ЗНУ»

Освітньо-професійний ступінь фаховий молодший бакалавр

Спеціальність 121 – Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ

Голова циклової комісії математичних дисциплін та інформаційних технологій

_____ Тетяна СМОЛЯНКОВА
(підпис)

“ 14 ” червня 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

ГОРБ Камілі Дмитрівні
(ПРІЗВИЩЕ ім’я та по–батькові)

1. Тема роботи «Проєктування та розробка інтернет галереї онлайн магазину одягу»

Керівник роботи викладач першої категорії БОРИСОВСЬКА Юлія Олександрівна
(посада, науковий ступінь, вчене звання, ПРІЗВИЩЕ ім’я по-батькові)

Затверджені наказом ВСП ЕПФК ЗНУ від « 26 » листопада 2024 р. № 1852–с

2. Строк подання студентом роботи 13.06.2024

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково– пояснювальної записки (перелік питань, які потрібно розробити)
1. Основи веброзробки та аналіз існуючих застосунків
2. Розробка проєкту сервісу
3. Програмна реалізація

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)
презентація до захисту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01.12.2024

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи:		
	Постановка задачі	Грудень 2024	виконано
2.	Збір вихідних даних, обробка методичних		
	та теоретичних джерел	Січень 2025	виконано
3.	Розробка першого розділу:		
	Огляд інформаційних систем та технологій для їхнього створення	Лютий 2025	виконано
4.	Розробка другого розділу:		
	Розробка проєкту інформаційної системи та вибір технологій	Квітень 2025	виконано
5.	Розробка третього розділу:		
	Розробка сервісу	Травень 2025	виконано
6.	Оформлення і нормоконтроль кваліфікаційної роботи та перевірка на плагіат	Червень 2025	виконано
7.	Захист кваліфікаційної роботи	20.06.2025	виконано

Здобувач освіти _____ Камала ГОРБ
(підпис) (ім'я ПРІЗВИЩЕ)

Керівник роботи _____ Юлія БОРИСОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

Нормоконтроль пройдено

Нормоконтролер _____ Юлія БОРИСОВСЬКА
(підпис) (ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Кваліфікаційна робота: 50 сторінок, 3 таблиці, 17 ілюстрацій.

Об’єкт дослідження – процес представлення, фільтрації, пошуку та взаємодії користувачів із асортиментом товарів в онлайн-галереях інтернет–магазинів одягу. Він охоплює способи візуального та текстового представлення товарів, навігаційні механізми, а також технічні рішення, що забезпечують ефективний вибір товару для користувача.

Предмет дослідження – концепція та програмна реалізація інтерактивної галереї інтернет-магазину одягу "Kemmi_Store_Ua". Досліджується структура даних, алгоритми фільтрації, пошуку та сортування товарів, методи динамічного відображення контенту (зокрема, використання модальних вікон для деталей товару), а також реалізація функціоналу "Обране" з локальним зберіганням даних та форми зворотного зв'язку.

Мета дослідження – розробка інтерактивної галереї інтернет-магазину одягу "Kemmi_Store_Ua", яка забезпечує користувачам ефективний та інтуїтивно зрозумілий доступ до асортименту товарів. Це включає:

- Створення гнучкої системи фільтрації та пошуку товарів за різними критеріями (стать, категорія, ціна, популярність) та пошуковим запитом.
- Реалізацію динамічного відображення детальної інформації про кожен товар (розміри, кольори, опис) через модальні вікна.
- Впровадження функціоналу "Обране" для персоналізації досвіду користувача зі зберіганням даних на стороні клієнта (localStorage).
- Забезпечення адаптивного дизайну інтерфейсу для інтуїтивного коректного функціонування.
- Інтеграцію механізму зворотного зв'язку для комунікації з користувачами.

ВЕБСАЙТ, HTML, CSS, JavaScript, PHP, MySQL.

SUMMARY

Qualification work: 50 pages, 3 tables, 17 illustrations.

The object of the study is the process of displaying, filtering, searching and interacting with the assortment of goods in online galleries of online clothing stores. This includes methods of presenting visual and textual information about goods, methods of navigation and interaction, as well as mechanisms that ensure the convenience and efficiency of product selection for the user.

The subject of the study is the concept and software implementation of the interactive gallery of the online clothing store "Kemmi_Store_Ua". The data structure, algorithms for filtering, searching and sorting goods, methods for dynamically displaying content (in particular, the use of modal windows for product details), as well as the implementation of the "Favorites" functionality with local data storage and feedback forms are studied.

The purpose of the study is to develop an interactive gallery of the online clothing store "Kemmi_Store_Ua", which provides users with effective and intuitive access to the assortment of goods. This includes:

• Creating a flexible system for filtering and searching for products by various criteria (gender, category, price, popularity) and search query.

• Implementing dynamic display of detailed information about each product (sizes, colors, description) through modal windows.

• Implementing the "Favorites" functionality to personalize the user experience with data storage on the client side (localStorage).

• Providing adaptive interface design for intuitive correct functioning.

• Integration of a feedback mechanism for communication with users.

WEBSITE, HTML, CSS, JavaScript, PHP, MySQL

ЗМІСТ

Завдання на кваліфікаційну роботу студентці 2
Реферат 4
Summary 5
Вступ 7
1 Основи веброзробки та аналіз існуючих застосунків 9
1.1 Веброзробка та її складові 9
1.2 Сучасні тенденції веброзробки 13
1.3 Аналіз предметної області 15
1.4 Огляд існуючих інтернет–магазинів 16
2 Розробка проєкту вебсайту 21
2.1 Технічне завдання 21
2.2 Діаграма прецендентів 24
2.3 Проєктування бази даних 26
3 Програмна реалізація 28
3.1.Архітектура та технології реалізації 28
3.2. Реалізації функціональних вимог 29
3.3. Аналіз результатів та тестування 38
Висновки 40
Перелік використаних джерел 42
Додаток А 44
Додаток Б 45
Додаток В 47
Додаток Г 50

ВСТУП

У сучасному світі цифрових технологій вебсайти стали невід'ємною складовою сучасного бізнесу, особливо в сфері онлайн-торгівлі. Інтернет–магазини активно витісняють традиційні точки продажу, оскільки дають змогу клієнтам швидко переглядати асортимент, знаходити потрібні товари, здійснювати покупки та оформлювати замовлення без обмежень у часі чи місці. У зв'язку з цим, все більшого значення набувають інтерфейси, які є зручними, адаптивними, привабливими візуально та мають інтеграцію з базами даних.

Головною метою кваліфікаційної роботи є створення вебсайту «Kemmi_store_ua» – інтерактивної галереї для демонстрації асортименту одягу. Проєкт охоплює розробку як клієнтської частини, орієнтованої на зручність користувача, так і серверної частини, що забезпечує функціональність ресурсу. Основне завдання системи – забезпечити зрозуміле та естетичне представлення продукції з можливістю перегляду, пошуку, фільтрації товарів за категоріями та доступу до актуальної інформації.

Для реалізації поставлених цілей використано сучасний набір інструментів веброзробки HTML, CSS і JavaScript – для побудови інтерфейсу користувача. Для реалізації логіки на боці сервера було обрано PHP, а як систему управління базами даних – реляційну базу даних MySQL. Програмний код розроблявся та підтримувався за допомогою текстового редактора Notepad++. Обрані технології дали змогу створити гнучке рішення, яке легко адаптувати під потреби конкретного бізнесу та забезпечити ефективну взаємодію між клієнтською частиною та сховищем даних.

Процес реалізації включає повний цикл проєктування та розробки від постановки завдання і вивчення аналогічних систем до створення архітектури, моделювання бази даних на MySQL, реалізації вебінтерфейсу з використанням

HTML, CSS та JavaScript, налаштування API на PHP для взаємодії з базою даних та підготовки супровідної документації. Особлива увага в роботі приділена принципам UX/UI дизайну.

Практична значимість роботи полягає в розробці готового до впровадження вебпродукту, що може застосовуватись у реальних комерційних умовах, а також бути базою для подальших бізнес-проєктів. Результат може стати основою для повноцінного онлайн-магазину з можливістю подальшого розширення – додаванням функцій реєстрації користувачів, інтеграції платіжних систем та ведення аналітики.

1 ОСНОВИ ВЕБРОЗРОБКИ ТА АНАЛІЗ ІСНУЮЧИХ ЗАСТОСУНКІВ

1.1 Веброзробка та її складові

Веброзробка – це процес створення вебзастосунків, вебсайтів або окремих функціональних блоків, що доступні через мережу Інтернет. Вона охоплює як технічну сторону проєктування, так і творчу складову дизайну користувацького інтерфейсу [1].

Основні складові веброзробки поділяються на два головні напрями: клієнтська (фронтенд) і серверна (бекенд) частина. Кожен із них виконує свої функції, взаємодіючи між собою для забезпечення цілісного користувацького досвіду [1].

Клієнтська частина розробки (фронтенд) – це частина, яку бачить і з якою безпосередньо взаємодіє користувач [2]. До основних технологій фронтенду належать HTML (мова розмітки структури сторінки), CSS (мова стилізації) та JavaScript (мова логіки і динаміки сторінки). Сучасні етапи веброзробки, в залежності від проєкту, також передбачають використання фреймворків, наприклад, React, Vue.js або Angular. Це дозволяє створювати інтерактивні інтерфейси з високим рівнем зручності.

Серверна частина розробки (бекенд) – це невидима користувачу частина, яка відповідає за логіку обробки даних, зберігання інформації, авторизацію, адміністрування та забезпечення безпеки [2]. Бекенд взаємодіє з базами даних і клієнтським інтерфейсом через API (інтерфейси прикладного програмування). Найпоширенішими мовами для серверної частини є JavaScript (Node.js), Python, PHP, Java і C#.

Веброзробка є процесом, що охоплює широкий спектр технологій, методологій та інструментів, необхідних для створення та підтримки

вебзастосунків. Для чіткого розуміння цього процесу, компоненти веброзробки доцільно класифікувати за їхньою роллю та місцем у загальній архітектурі вебсайту.

До компонентів веброзробки можна віднести [2]:

Клієнтську частину, або Frontend, яка відповідає за інтерфейс користувача (User Interface, UI) та взаємодію з ним безпосередньо у веббраузері. Її головна мета – забезпечити інтуїтивно зрозумілий, естетично привабливий та функціональний користувацький досвід (User Experience, UX). Клієнтська частина вебсайту реалізується за допомогою фундаментальних технологій, зокрема HTML, CSS, та JavaScript. Ці технології утворюють основу фронтенд–розробки, сприяючи оптимізації користувацького досвіду та ефективній презентації інформації. А саме:

- HTML (HyperText Markup Language) є основою будь–якої вебсторінки. HTML визначає структуру та зміст вебдокумента, використовуючи теги для позначення заголовків, параграфів, зображень, посилань, списків, форм тощо. Це семантичний скелет, на якому будується весь вебконтент.

- CSS (Cascading Style Sheets) використовується для візуального оформлення вебсторінок, створених за допомогою HTML. CSS дозволяє контролювати кольори, шрифти, розміри, відступи, розташування елементів, анімації та адаптивність дизайну, забезпечуючи відображення контенту на різних пристроях.

- JavaScript динамічна мова програмування, що надає вебсторінкам інтерактивності. JavaScript дозволяє реалізовувати складні функції на стороні клієнта, а саме валідацію форм, анімації, динамічне оновлення контенту без перезавантаження сторінки. Бібліотеки та фреймворки JavaScript забезпечують структуру, інструменти та оптимізації для прискорення розробки та підвищення її ефективності [1].

Додаткові інструменти для реалізації клієнтської частини включають [2]:

- Visual Studio Code (VS Code) від Microsoft –це потужний безоплатний редактор із великою кількістю розширень і можливостями налаштування.
- Notepad++ – це безкоштовний редактор тексту та вихідного коду, який часто використовується у фронтенд–розробці. Він надає зручні інструменти для роботи з HTML, CSS та JavaScript, такими як підсвічування синтаксису, автоматичне форматування коду та згортання блоків коду.

Серверна частина (Backend Development) відповідає за логіку роботи вебсайту, обробку запитів від клієнта, взаємодію з базами даних, безпеку та управління даними. Користувачі не бачать код бекенду безпосередньо, але він є основою всієї функціональності.

Технології для серверної розробки включають [2]:

- Python популярний завдяки своїй простоті, читабельності та великій кількості бібліотек.
- Node.js (JavaScript на сервері) дозволяє використовувати JavaScript для розробки як на клієнтській, так і на серверній стороні.
- PHP одна з найстаріших та найпоширеніших мов для веброзробки, особливо для систем управління контентом (CMS) типу WordPress.
- Java використовується для розробки великих корпоративних застосунків, які вимагають високої продуктивності, надійності та масштабованості
- C# мова програмування для розробки вебзастосунків, десктопних програм та ігор, що пропонує потужні інструменти та інтеграцію з екосистемою Windows.

Реляційні бази даних зберігають дані у вигляді таблиць зі зв'язками. Прикладні реляційні СУБД, оптимізовані для вебзастосунків [1]:

- PostgreSQL потужна, відкрита, об'єктно–реляційна СУБД, що підтримує складні запити та великі обсяги даних.

- MySQL найпопулярніша відкрита реляційна СУБД, широко використовується для вебзастосунків.

Нереляційні бази даних (NoSQL) зберігають дані у гнучкіших форматах (документи, ключі–значення, графіки). Приклади нереляційних баз даних, що використовуються для веброзробки [2]:

- MongoDB документоорієнтована СУБД, популярна для швидкої розробки та масштабованих застосунків.
- Redis сховище даних типу "ключ–значення" у пам'яті, часто використовується для кешування, сесій та брокерів повідомлень.

Забезпечення безпеки вебзастосунків є комплексним процесом, що включає кілька критично важливих механізмів. Автентифікація дозволяє перевірити особу користувача, а авторизація—визначити рівень його доступу до системи відповідно до наданих прав. Одним із ключових методів захисту є хешування паролів, що передбачає зберігання паролів у криптографічно незворотній формі, запобігаючи їх компрометації у випадку витоку даних [1].

Додатковим рівнем безпеки є валідація вхідних даних, що передбачає ретельну перевірку всіх параметрів, отриманих від користувача або зовнішніх джерел, з метою запобігання атакам, таким як SQL–ін’єкції або XSS. Важливим етапом контролю якості вебсайту є тестування, яке гарантує його стабільність, продуктивність та коректну роботу в різних умовах. Особливу увагу приділяють кросбраузерному тестуванню, що дозволяє оцінити відображення та функціональність інтерфейсу у різних веббраузерах, забезпечуючи коректний користувацький досвід незалежно від платформи.

Такий підхід до безпеки та тестування дозволяє створювати стійкі, захищені та надійні вебсайти, відповідні сучасним вимогам кіберзахисту та якості розробки.

Веброзробка є комплексним процесом, що поєднує проєктування архітектури, програмування, тестування, налагодження та подальшу підтримку вебсайту.

1.2 Сучасні тенденції веброзробки

Сфера веброзробки перебуває у стані постійної еволюції, що зумовлено безперервним розвитком технологій та інструментів. Це призводить до динамічних змін у підходах до створення вебзастосунків. У сучасному світі простежуються кілька ключових тенденцій, що визначають вектор розвитку галузі, включаючи акцент на динамічності, інтерактивності, зрозумілості та адаптивності інтерфейсів [3].

Зокрема, ефективність онлайн-галереї одягу безпосередньо залежить від впровадження певних особливостей, що забезпечують привабливий та функціональний користувацький досвід. Ці особливості можна згрупувати за кількома основними напрямками.

По-перше, для успішної презентації одягу важливим є використання високоякісних візуальних матеріалів. Це охоплює зображення з високою роздільною здатністю, які підтримують детальне масштабування. Значну цінність додають 360–градусні огляди товарів або відеодемонстрації, що дозволяють потенційним покупцям всебічно оцінити одяг, імітуючи досвід фізичного магазину.

По-друге, зручна та інтуїтивно зрозуміла навігація є основою позитивного користувацького досвіду. Важливо забезпечити ефективну категоризацію товарів (за статтю, типом, сезоном) та наявність розширених фільтрів (за ціновим діапазоном, брендом, матеріалом), що дозволяє користувачам швидко знаходити потрібні позиції, скорочуючи час пошуку та підвищуючи ефективність взаємодії.

Окрім цього, інтеграція нових цифрових інтерактивних елементів значно покращує залученість користувачів. До них належать, наприклад, віртуальна примірка, що використовує технології доповненої.

По-третє, важливими тенденціями є функції, що спрощують процес перегляду товарів та вибору, такі як "список бажань" та можливість швидкого додавання товару до "обраного". Ці інструменти підвищують зручність користування та сприяють повторним візитам.

Четвертим, але не менш важливим аспектом є адаптивний та кросплатформний дизайн. Це означає, що онлайн-галерея одягу повинна чудово виглядати та бездоганно функціонувати на будь–якому пристрої – від великих моніторів до планшетів та смартфонів.

Ще однією важливою тенденцією, є оптимізація швидкості завантаження вебзастосунків. Досвід користувача (UX – User Experience) безпосередньо залежить від швидкості реакції сайту. Швидке завантаження збільшує ймовірність того, що відвідувач залишиться на сайті та здійснить покупку, тоді як затримки можуть призвести до його передчасного відтоку. Чим швидше завантажується сайт, тим приємніше та легше ним користуватися [4].

Загалом, сучасні тенденції веброзробки чітко вказують на пріоритет створення швидких, інтуїтивно зрозумілих, динамічних та зручних у використанні вебсайтів. Такі рішення повністю відповідають вимогам сучасного ринку, забезпечуючи найкращий можливий досвід для кінцевих користувачів. Важливо пам'ятати, що попри спільні тенденції, кожен вебсайт є унікальним проєктом, і його розробка має враховувати специфічні потреби, цілі та цільову аудиторію. Це дозволяє створити індивідуальне рішення, що максимально ефективно виконуватиме свої функції та виділятиметься серед конкурентів.

1.3 Аналіз предметної області

Аналіз предметної області є також важливим етапом у розробці будь–якого вебсайту, особливо при створенні галереї інтернет–магазину одягу. Цей процес дозволяє детально зрозуміти потреби кінцевих користувачів, а також специфіку функціонування онлайн-платформи для продажу одягу. Ключовою частиною є визначення всіх необхідних функцій та характеристик, які повинна мати галерея товарів для задоволення потреб користувачів і підвищення ефективності магазину.

Основні можливості галереї одягу, реалізованої через вебсайт, включають:

1. Перегляд каталогу товарів. Товар представлений у вигляді візуальних зображень, що допомагають користувачам швидко орієнтуватися в асортименті. Для кожного елемента одягу повинні бути представлені детальні зображення, короткий опис, інформація про розміри, кольори, ціну та доступність товару.

2. Можливість деталізації характеристик товарів. Це передбачає швидкий перегляд категорій товарів, їх фільтрацію за різними характеристиками (стать, стиль, тощо) та можливість перегляду товарів у збільшеному форматі для кращого огляду деталей. Інтерактивність галереї. Система має забезпечувати зручну навігацію по товарних категоріях.

3. Інтерактивні елементи, такі як анімації при наведенні на товар або зміна зображення при виборі кольору чи розміру, роблять досвід перегляду більш захоплюючим.

4. Керування галереєю. Для адміністративного персоналу магазину важливо мати функціонал для швидкого додавання, редагування або видалення товарів у галереї, що дозволяє підтримувати актуальність даних та ефективно керувати вмістом.

На основі цих можливостей формуються ключові вимоги до системи, саме можливість перегляду товарів з великою кількістю зображень та функцією

збільшення для детального огляду. Реалізація детальних фільтрів для сортування товарів за різними ознаками. Зручна система адміністрування для додавання, редагування та видалення товарів у галереї. Застосування сучасних технологій для забезпечення швидкого та зручного користувацького досвіду. Загалом, ретельний аналіз предметної області дозволяє чітко визначити функціональні вимоги для створення галереї інтернет–магазину одягу, що максимально відповідає потребам як кінцевих користувачів, так і бізнесу.

1.4 Огляд існуючих інтернет–магазинів

Огляд існуючих інтернет–магазинів дозволяє оцінити конкурентне середовище та зрозуміти найкращі практики, які використовуються в онлайн-торгівлі одягом. Це дозволяє визначити, які елементи функціональності та дизайну користуються найбільшою популярністю серед користувачів, а також як вони впливають на досвід покупця. Більшість інтернет–магазинів одягу мають галереї товарів, які є основною частиною їхніх вебсайтів. Галерея товарів зазвичай включає в себе зображення одягу, яке покупці можуть переглядати, а також різні інструменти для фільтрації та сортування товарів, щоб знайти те, що вони шукають. Розглянемо кілька прикладів, як це реалізовано в популярних інтернет-магазинах.

1. Zara (zara.com).

Zara є одним з найбільших гравців на ринку інтернет–торгівлі одягом. Вебсайт компанії пропонує простий і зручний інтерфейс, який забезпечує швидкий доступ до різних категорій товарів. Особливо виділяється функція "швидкого перегляду" товарів, що дозволяє покупцю бачити деталі без переходу на окрему сторінку. Також застосовано адаптивний дизайн, що робить сайт зручним для використання на мобільних пристроях. Галерея товарів на сайті Zara

дозволяє користувачам переглядати одяг із різних ракурсів, а також дає можливість збільшувати зображення для детальнішого огляду. Є фільтри для вибору категорій, розмірів, кольорів і цінових діапазонів (див. рис. 1.1).

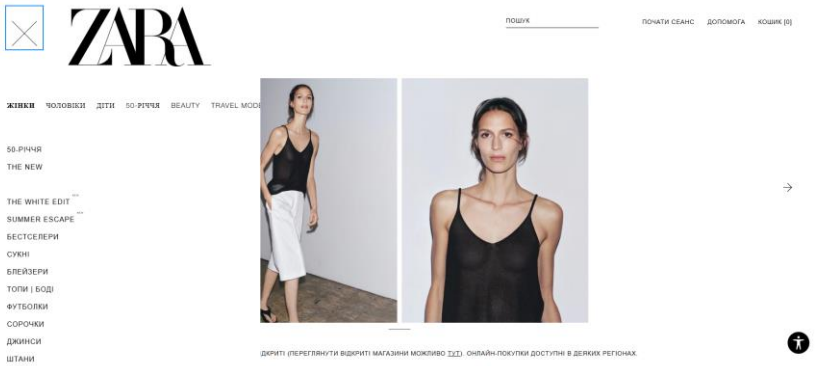


Рисунок 1.1 – Інтернет–магазин Zara

2. ASOS (asos.com).

ASOS – популярний інтернет-магазин одягу, який має великий асортимент товарів для різних вікових груп. Вебсайт включає функції фільтрації за категоріями, кольорами, цінами та брендами. Однією з інновацій є можливість переглядати товари за допомогою 360-градусних зображень, що дозволяє користувачам детально оглядати продукт з усіх боків. Крім того, на ASOS є інтеграція з соціальними мережами, де покупці можуть бачити відгуки про товари. Галерея ASOS також пропонує високу якість зображень і дає можливість користувачам швидко переглядати товар без необхідності відкривати окремі сторінки. Функції швидкого перегляду і фільтрації роблять процес покупки дуже зручним (див. рис. 1.2).

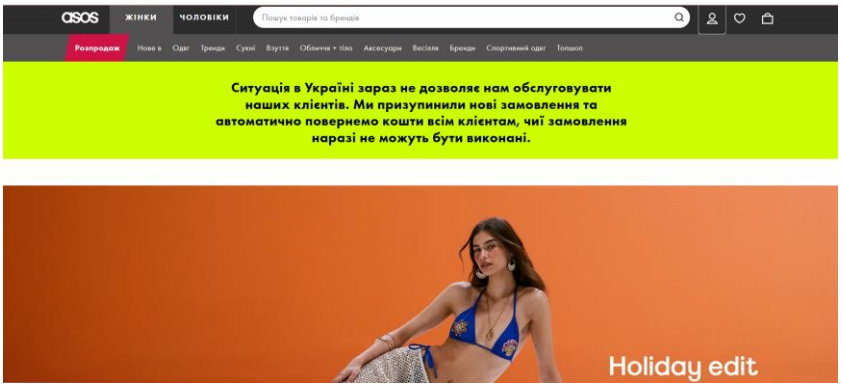


Рисунок 1.2 – Інтернет–магазину ASOS

3. H&M (hm.com).

H&M – ще один великий магазин, що продає одяг через інтернет. Його вебсайт дуже зручний і має чітку навігацію по категоріях товарів. Особливістю є можливість створювати списки бажаних товарів і надсилати їх друзям або родичам через електронну пошту чи соціальні мережі. Крім того, сайт підтримує функцію швидкої покупки через "один клік", що значно спрощує процес оформлення замовлення. Сайт H&M має просту і ефективну галерею товарів, що дозволяє користувачам переглядати одяг із чіткими фотографіями та фільтрами для пошуку потрібного товару за різними критеріями, наприклда, колір, розмір, тип (див. рис. 1.3).

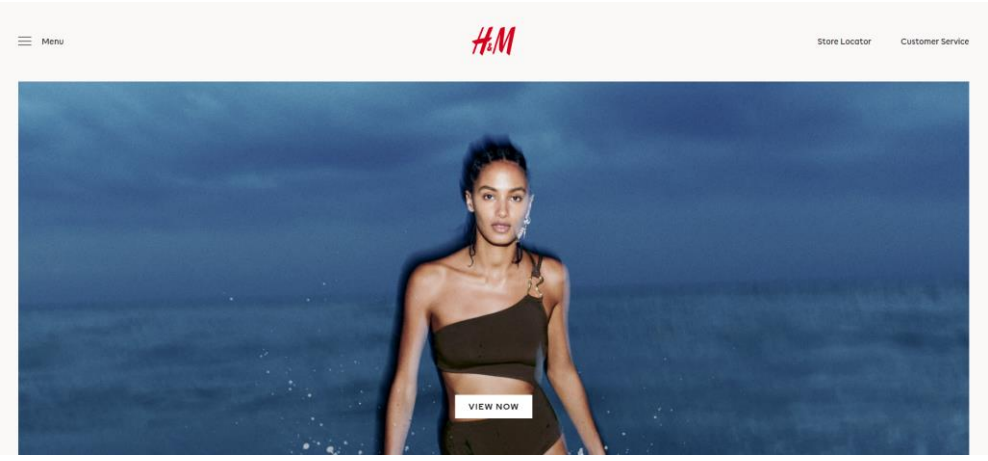


Рисунок 1.3 – Інтернет–магазин H&M

4. Mango (mango.com).

Інтернет-магазин Mango відомий своїм стильним та зручним інтерфейсом. На сайті представлено безліч фільтрів для сортування товарів за різними критеріями. Важливою перевагою є візуальні рекомендації, коли система пропонує додаткові товари, що підходять до вибраного товару. Це підвищує ймовірність покупки, оскільки користувач може одразу придбати всю необхідну комплектацію. Mango активно використовує інтерактивні елементи на своєму сайті, де можна переглядати товар не тільки за допомогою зображень, але й за допомогою відеооглядів. Галерея товарів має високу якість зображень та кілька опцій для перегляду (див. рис. 1.4).

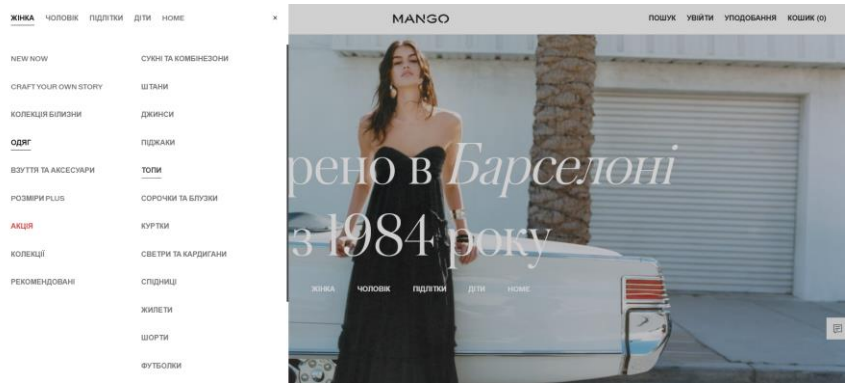


Рисунок 1.4 – Інтернет–магазин Mango

На основі проведеного аналізу існуючих рішень можна виділити кілька ключових тенденцій, що формують сучасний ринок онлайн-торгівлі одягом [5]:

1. Адаптивний дизайн. Вебсайти повинні бути оптимізовані для перегляду як на настільних комп’ютерах, так і на мобільних пристроях. Це забезпечує безперебійний доступ до галереї магазину з будь-якого пристрою, що є критично важливим для сучасних користувачів.

2. Персоналізація. Інтернет-магазини дедалі частіше впроваджують інструменти персоналізації, зокрема рекомендації товарів, ґрунтуючись на

попередніх покупках або переглядах. Це допомагає залучити користувачів, підвищуючи рівень конверсії.

3. Інтерактивні елементи. Багато сайтів використовують інтерактивні елементи, такі як 360-градусні огляди товарів або функції доповненої реальності, що дозволяють користувачам віртуально "приміряти" одяг. Ці функції допомагають підвищити рівень задоволеності покупців.

4. Швидкість завантаження. Вебсайти з високою швидкістю завантаження користуються більшою популярністю серед покупців. Дослідження показують, що користувачі зазвичай залишають сайт, якщо він не завантажується більше ніж за 3 секунди.

5. Інтеграція з соціальними мережами. Можливість поділитися покупками або переглянутими товарами у соціальних мережах стала важливим елементом маркетингової стратегії інтернет-магазинів, сприяючи збільшенню охоплення аудиторії та залученню нових клієнтів.

Проаналізувавши існуючі галереї інтернет-магазинів одягу, можна сказати, що для досягнення успіху вебресурсу з продажу одягу необхідно реалізувати ключові елементи, виявлені під час цього огляду. Інтерактивність, персоналізація, адаптивність дизайну та швидкість завантаження є важливими аспектами, які необхідно врахувати при створенні галереї товарів для онлайн-магазину одягу.

2 РОЗРОБКА ПРОЄКТУ ВЕБСАЙТУ

2.1 Технічне завдання

Створення технічного завдання (ТЗ) є одним із ключових етапів у життєвому циклі розробки програмного забезпечення. У проєкті «Розробка та проєктування галереї онлайн магазину одягу» технічне завдання було сформовано на основі комплексного аналізу ринку електронної комерції, дослідження існуючих демонстраційних платформ та вивчення потреб потенційних користувачів.

2.1.1 Найменування і область застосування

Розроблений програмний продукт отримав найменування "Галерея магазину одягу Kemmi_Store_Ua". Його основною метою є створення сучасної, інтерактивної та візуально привабливої онлайн-вітрини, що дозволяє магазину одягу Kemmi_Store_Ua ефективно демонструвати свій широкий асортимент продукції потенційним клієнтам.

Ключове функціональне призначення сервісу полягає в наданні користувачам максимально зручного та інтуїтивно зрозумілого інструменту для візуального та інформаційного ознайомлення з товарами. Це включає можливість легкого перегляду різноманітних колекцій, отримання вичерпної інформації про кожен окремий товар – від його детального опису до переліку доступних розмірів та кольорів. Особливою інтерактивною функцією є можливість для користувача формувати та керувати власним списком уподобаних позицій, що сприяє глибшій взаємодії з асортиментом магазину. Важливо підкреслити, що ця галерея свідомо

зосереджена на демонстрації та виборі, без інтеграції повноцінної платіжної системи на даному етапі, що дозволяє зосередити всі ресурси на створенні досконалого візуального досвіду та зручності ознайомлення з продукцією.

Область застосування "Галереї онлайн-магазину одягу Kemmi_Store_Ua" охоплює сектор роздрібної торгівлі одягом та аксесуарами. Гнучкість архітектури дозволяє використовувати цей вебсайт як самостійну демонстраційну платформу для представлення нових колекцій, сезонних пропозицій або ексклюзивних товарів. Крім того, він може бути безшовно інтегрований у структуру вже існуючих онлайн-магазинів як додатковий демонстраційний модуль, значно розширюючи їх функціональні можливості у частині візуалізації галереї асортименту. Вебсайт орієнтований на всіх користувачів, які здійснюють пошук та вибір одягу та аксесуарів онлайн, надаючи їм естетично привабливий та високофункціональний інструмент для попереднього відбору товарів перед потенційною покупкою.

2.1.2 Технічні вимоги до програмного продукту

Підрозділ описує ключові технічні аспекти проєктування та реалізації вебсайту, визначаючи функціональні й нефункціональні вимоги, що забезпечують працездатність, безпеку, зручність використання та відповідність сучасним стандартам веброзробки [3].

Функціональні вимоги в проєкті включають:

- Перегляд асортименту. Забезпечення зручного, інтуїтивного та динамічного перегляду товарів у форматі карток з основною інформацією (зображення, назва, ціна).

- Пошук та фільтрація. Реалізація можливостей фільтрації та сортування товарів за категоріями, а також універсального пошуку за назвою або ключовими словами з оновленням результатів в режимі реального часу.
- Детальний перегляд товару. Відображення розширеної інформації про товар через модальне вікно (опис, характеристики, розміри, кольори тощо) без перенаправлення на іншу сторінку.
- Персоналізований список "обраних". Можливість додавання товарів до особистого списку з миттєвим візуальним підтвердженням та збереженням даних у локальному сховищі браузера.
- Навігаційна структура. Інтуїтивно зрозуміле головне меню з основними розділами та функцією "Прокрутити нагору" для полегшення навігації.
- Комунікація: Інтегрована форма зворотного зв'язку для надсилання запитів та пропозицій, що підвищує рівень клієнтського сервісу.

Нефункціональні вимоги описують якісні характеристики та обмеження системи, які є не менш важливими за її безпосередні функції. Ці вимоги гарантують високу якість, стабільність та безпеку програмного продукту:

- UI/UX. Інтерфейс має бути інтуїтивно зрозумілим, логічно структурованим і естетично привабливим, що забезпечує позитивний користувацький досвід.
- Надійність. Система повинна працювати безперебійно з механізмами обробки виняткових ситуацій для гарантування стабільності.
- Безпека. Захист даних реалізується за допомогою сучасних методів (наприклад, використання підготовлених запитів), що мінімізує загрози типу SQL-ін’єкцій.
- Модульність. Архітектура системи повинна забезпечувати легкість додавання нового функціоналу без суттєвих змін у базовій логіці.

- Вибір технологій. Використовуються перевірені технології (HTML, CSS, JavaScript, PHP, MySQL, Notepad++ та XAMPP) для забезпечення актуальності та ефективності розробки.

2.2 Діаграма прецендентів

Проєктування інтерфейсу розпочалося з визначення ключових елементів, які формують візуальне сприйняття та функціональність галереї. За основу було взято підхід, орієнтований на чистий дизайн та зручність використання. Основні сторінки, такі як чоловічий, жіночий одяг та аксесуари, розроблені з акцентом на відображення асортименту товарів та надання користувачам можливості легко взаємодіяти з ними (див. рис. 2.1).

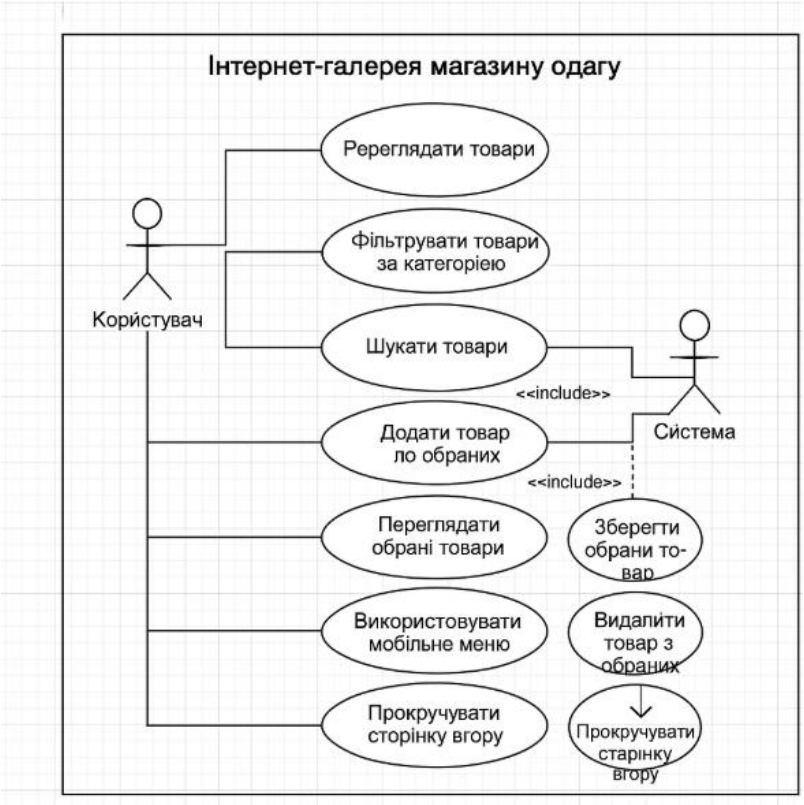


Рисунок 2.1 – Діаграма прецендентів

У контексті проєкту "Галерея онлайн-магазину одягу "Kemmi_Store_Ua"" виділяються два основні типи акторів, які взаємодіють із системою. Перший – це користувач, який є основним суб'єктом вебсайту. Його роль охоплює широкий спектр дій, від простого перегляду асортименту товарів до активної взаємодії з інтерфейсом для пошуку, фільтрації та персоналізації досвіду, наприклад, додавання товарів до списку обраних. Другий актор – це система, що уособлює всі автоматизовані процеси, які виконуються на фоні, зокрема, керування даними у локальному сховищі браузера (localStorage).

Функціональні можливості, що реалізуються системою, відображені у вигляді прецедентів, кожен з яких описує певну взаємодію з користувачем. Користувач має змогу переглядати всі доступні товари в галереї, а також фільтрувати їх за категорією, дозволяючи звузити вибірку до конкретних типів одягу, таких як "футболки" або "джинси" тощо. Крім того, передбачено функціонал пошуку товарів, де користувач може ввести запит для швидкого знаходження бажаних позицій за їх назвою чи описом.

Ключовою інтерактивною можливістю є перегляд деталей товару, що здійснюється шляхом відкриття модального вікна з розширеною інформацією про обрану річ. Особливою функцією є також додавання товарів до обраних користувач може зберегти вподобані позиції у свій персональний список. Цей прецедент обов'язково включає автоматичне збереження даних про обраний товар у localStorage системою. Аналогічно, існує можливість видалити товар з обраних, що також передбачає операцію оновлення даних у localStorage для видалення відповідного запису. Користувач може у будь–який момент переглядати свої обрані товари, відкриваючи спеціальне модальне вікно.

Окрім основних функцій, система підтримує додаткові прецеденти, що підвищують зручність використання. Інтегрована функція прокрутки сторінки вгору, яка надає користувачеві швидкий спосіб повернутися до початку довгої сторінки одним кліком.

2.3 Проєктування бази даних

У процесі реалізації вебсайту було розроблено та створено реляційну базу даних MySQL. Реляційна база даних – це тип бази даних, що зберігає інформацію в електронних таблицях і здійснює пошук даних в одній таблиці на підставі визначених ключових полів іншої таблиці. Програми, які призначені для структурування інформації, розміщення її в таблицях і маніпулювання даними, називаються системами управління базами даних (СУБД) [6]. У роботі була використана СУБД MySQL.

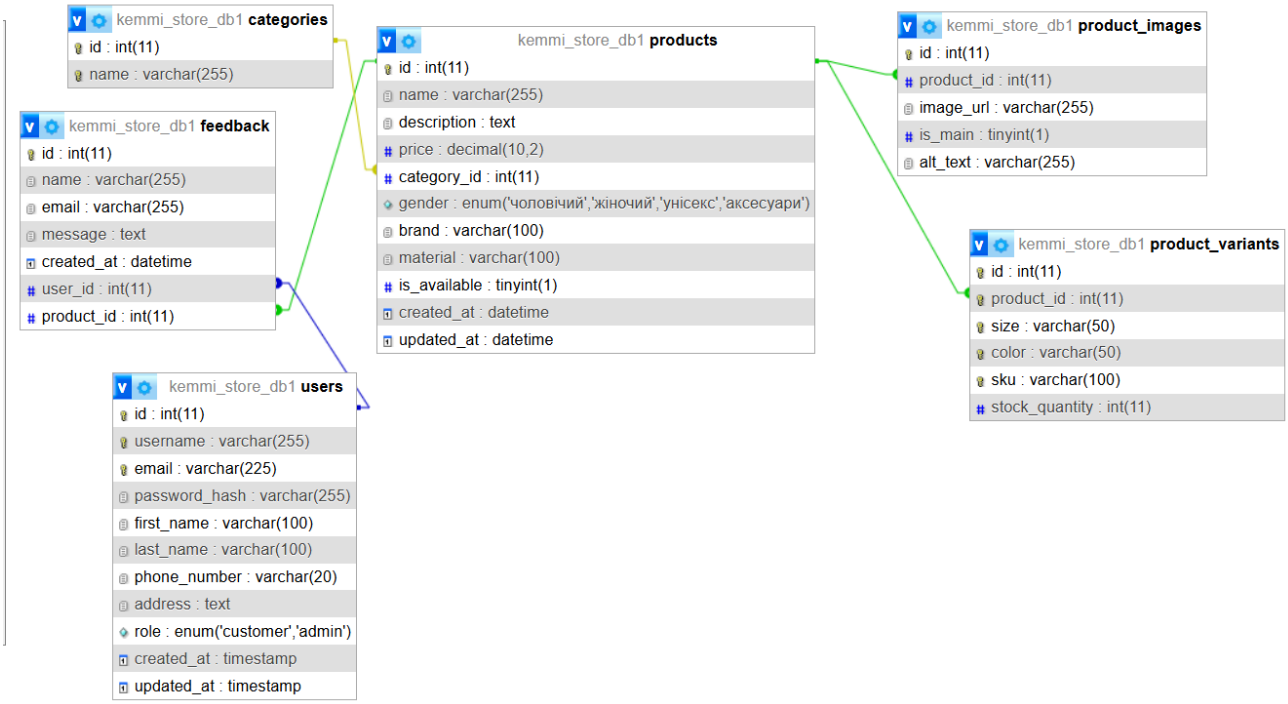
Визначені основні сутності, необхідні для реалізації функціоналу інтерактивної галереї:

- Продукти (Products) – основна сутність для зберігання інформації про товари.
- Зворотний зв'язок (Feedback) – сутність для збору повідомлень від користувачів.

У таблиці "Продукти" (products) зберігається вся необхідна інформація про товари, що представлені в галереї. До її стовпців належать: id (унікальний ідентифікатор товару), name (назва товару), description (детальний опис товару), price (ціна товару), image_url (шлях до зображення товару), category (категорія товару) та gender (стать, для якої призначений товар). Також зберігається created_at для відстеження часу додавання товару.

Таблиця "Зворотний зв'язок" (feedback) призначена для збору повідомлень відвідувачів вебсайту. Вона містить такі стовпці: id (унікальний ідентифікатор повідомлення), name (ім'я відправника), email (електронна пошта відправника), message (текст повідомлення) та created_at (дата та час відправлення повідомлення). Прямий зв'язок таблиці feedback з іншими таблицями не є необхідним для поточного функціоналу, оскільки вона слугує для збору загальних запитів, не пов'язаних з конкретними товарами чи користувачами.

Перед створенням бази даних важливо мати структурований огляд бази для кращого розуміння її організації. Таким чином, на рисунку 2.2 представлена схема бази даних, яка візуалізує структуру та взаємозв'язки між сутностями.



Рискнок 2.2 – ER-діаграми бази даних

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Архітектура та технології реалізації

Програмна реалізація галереї магазину одягу базується на класичній клієнт-серверній архітектурі [7]. Цей підхід забезпечує чітке розподілення відповідальності та функціоналу, тобто клієнтська частина відповідає за відображення інтерфейсу користувача та інтерактивну взаємодію з ним, тоді як серверна частина займається обробкою даних, їх зберіганням та реалізацією основної бізнес–логіки. Для досягнення поставлених цілей проєктування та розробки було обрана наступний стек вебтехнологій [8]:

- HTML – використовувався як фундаментальна технологія для створення семантичної та логічної структури всіх вебсторінок. HTML забезпечує коректне та структуроване розташування контенту, що є невід'ємною основою для подальшої стилізації та інтерактивності.
- CSS – відповідає за візуальне оформлення інтерфейсу користувача, включаючи макети, колірні схеми, типографіку та анімації. Застосування сучасних можливостей CSS у поєднанні з медіа–запитами, дозволило створити макети, які коректно відображаються та оптимально функціонують.
- JavaScript – застосовувався для реалізації всієї інтерактивної логіки на стороні клієнта. Це включає динамічне завантаження та оновлення товарів, функціонал фільтрації та пошуку, взаємодію з модальними вікнами, реалізацію функції "Обране" (з використанням localStorage) та обробку подій користувача. Завдяки JavaScript, інтерфейс є динамічним і забезпечує плавний та сучасний користувацький досвід без повного перезавантаження сторінок.
- PHP – була обрана як основна мова для розробки серверної логіки. PHP відповідає за обробку HTTP–запитів, що надходять від клієнтської частини,

взаємодію з базою даних MySQL, виконання операцій фільтрації, сортування та пошуку даних, а також формування відповідей у форматі JSON для клієнта.

- MySQL – використовується як система управління реляційними базами даних. У ній надійно зберігається вся інформація про товари (назва, опис, ціна, зображення, категорія, гендерна приналежність, доступні розміри та кольори), а також інша довідкова інформація, необхідна для функціонування галереї.

3.2. Реалізації функціональних вимог

Реалізація функціональних вимог системи була виконана з особливим акцентом на ефективність, розширюваність, модульність та максимальну зручність для кінцевого користувача.

3.2.1 Дизайн вебсайту.

Перед безпосереднім початком розробки постало завдання вибору концепції дизайну інтерфейсу. Було розглянуто два основні варіанти: розробка унікального власного дизайну з нуля або адаптація та використання існуючих безкоштовних макетів. Зважаючи на те, що кваліфікаційна робота сконцентрована переважно на програмній реалізації функціоналу, персоналізації роботи та демонстрації технічних рішень, було обрано перший варіант (розробка унікального власного дизайну). Саме для створення цього унікального візуального проєкту, що відповідав нашим вимогам, ефективним інструментом виявився хмарний сервіс Figma. Figma – це багатоплатформовий інструмент для дизайнерів інтерфейсів та веброзробників, який дозволяє працювати над

візуальною концепцією безпосередньо в браузері. На рисунку 3.1 зображено вигляд головної сторінки дизайну у середовищі Figma.

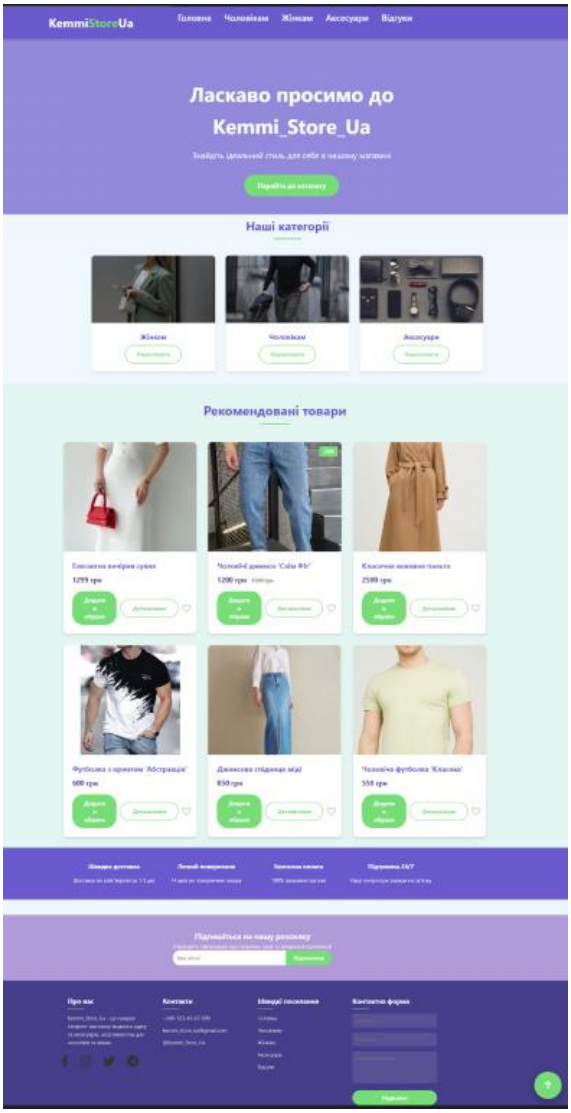


Рисунок 3.1 – Дизайн голового екрану вебсайту у середовищі Figma.

3.2.2 Створення клієнтської частини

Першим пріоритетом на етапі розробки була клієнтська частина вебсайту, яка відповідає за представлення даних користувачеві та всю інтерактивну

взаємодію [2]. Всі елементи інтерфейсу були реалізовані з використанням стандартних вебтехнологій (HTML, CSS, JavaScript) з акцентом на динамічне завантаження контенту та високу інтерактивність.

Наступним етапом була безпосередня верстка інтерфейсу сервісу. Важливо розуміти, що у сучасній веброзробці, особливо для динамічних вебсайтів, не створюється кожна статична HTML-сторінка окремо для кожного розділу сайту. Натомість, створюються гнучкі шаблони сторінок та повторювані компоненти, а основний контент (наприклад, дані про товари, їх описи, зображення) завантажується динамічно з бази даних.

Весь процес верстки здійснювався з урахуванням цієї парадигми. Кожна логічна частина інтерфейсу чи то ціла сторінка (наприклад, "Чоловічий одяг", "Жіночий одяг"), чи повторюваний елемент (наприклад, окрема картка товару) – була реалізована як окремий HTML–структурований блок. Ці блоки інкапсулюють у собі не тільки HTML–розмітку, а й відповідні CSS–стилі та JavaScript–логіку. Наприклад, компонент картки товару, який багаторазово використовується у галереях товарів, був винесений в окремий модуль, що значно пришвидшує процес розробки, спрощує підтримку коду та забезпечує його перевикористання. На рисунку 3.2 показано вигляд декількох ключових елементів у структурі проєкту.

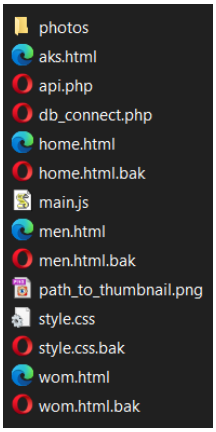


Рисунок 3.2 – Компоненти у структурі проєкту.

Кожен логічний блок у проєкті, містить у собі три основні секції [8]:

- HTML розмітка – відповідає за структурне відображення майбутньої сторінки або компонента.
- CSS стилі – використовуються для стилізації елементів цього конкретного блоку. Можливе використання як inline–стилів, так і підключення зовнішніх файлів CSS.
- JavaScript логіка – містить усю інтерактивну поведінку, обробку подій та динамічне завантаження даних, властиві даному блоку.

Для програмування інтерактивної логіки клієнтської частини сервісу використовувався JavaScript [2]. Ця мова програмування дозволяє динамічно маніпулювати об'єктною моделлю документа, обробляти події користувача (кліки, введення тексту), відправляти асинхронні запити до сервера та миттєво оновлювати інтерфейс без необхідності перезавантаження всієї сторінки. Це забезпечує швидкий та плавний користувацький досвід, характерний для сучасних вебсайтів.

Для навігації між сторінками сервісу без повного перезавантаження всієї сторінки, що характерно для Single Page Application (SPA), використовується також JavaScript [9]. Він динамічно керує URL-адресою та завантажує новий контент. Це забезпечує швидкі та плавні переходи між логічними "сторінками". Нижче наведено лістинг обробки навігації по фільтрах категорій.

```
    // Отримання всіх посилань фільтрів категорій у відповідному контейнері
    const categoryLinks = document.querySelectorAll('.category–filters .nav–link');

    categoryLinks.forEach(link => {
      link.addEventListener('click', (e) => {
        e.preventDefault(); // Запобігаємо стандартній дії посилання (переходу на іншу сторінку)
```

```
            // Видаляємо активний клас з усіх посилань фільтрів
            categoryLinks.forEach(item => item.classList.remove('active'));
            // Додаємо активний клас до посилання, на яке клікнув користувач
            link.classList.add('active');

            const category = link.dataset.category; // Отримуємо значення категорії з data–атрибута посилання
            const gender = document.body.dataset.gender; // Отримуємо гендер поточного розділу (наприклад, зі схованого data–атрибута body)
            // Викликаємо функцію завантаження товарів з новими параметрами фільтрації та поточним пошуковим запитом
            fetchProducts(gender, category, document.getElementById('searchInput').value);
        });
    });
```

Наведений код дозволяє фільтрувати товари за категоріями, динамічно оновлюючи список товарів без перезавантаження сторінки, що значно покращує користувацький досвід.

Важливою частиною фронтенду галереї інтернет-магазину є також створення фільтрів та сортування товарів, оскільки це значно підвищує користувацький досвід та зручність пошуку. Для цього використовувалися стандартні можливості JavaScript для маніпуляцій з масивами даних. Розглянемо, наприклад, функціонал сортування товарів за різними критеріями в додатку А, а саме за збільшенням та зменшенням ціни, за популярністю та новизною.

Даний фрагмент коду демонструє логіку сортування масиву currentProducts відповідно до обраного користувачем варіанта. Функція sortProducts динамічно визначає критерій сортування та застосовує його до масиву. Після сортування, функція displayProducts (описана вище) оновлює відображення товарів на сторінці, забезпечуючи миттєвий відгук на дії користувача.

3.2.3 Валідація та обробка помилок

При побудові будь-яких форм на сайті наприклад, контактна форма чи форма реєстрації або входу обов'язковим етапом було впровадження валідації полів вводу. Валідація – це процес перевірки вхідних даних на відповідність певним заздалегідь визначеним стандартам, вимогам або правилам [10]. Це є важливим для забезпечення коректності даних, що надходять на сервер, запобігання помилкам, захисту від зловмисних ін'єкцій та покращення загальної надійності системи.

Валідацію в проєкті було реалізовано на стороні клієнта за допомогою JavaScript, без використання сторонніх бібліотек. В додатку Б наведено лістинг коду, який перевіряє, чи всі обов'язкові поля контактної форми було заповнено перед її відправкою, а також перевіряє формат електронної пошти.

На рисунку 3.3 представлено відображення на вебсайті, якщо користувач намагається відправити форму з незаповненими полями, отримуючи відповідне повідомлення про помилку. На рисунку 3.3 зображенно повне правильне заповнення із повідомленням та на рисунку 3.4 правильні заповненні поля із повідомленням.

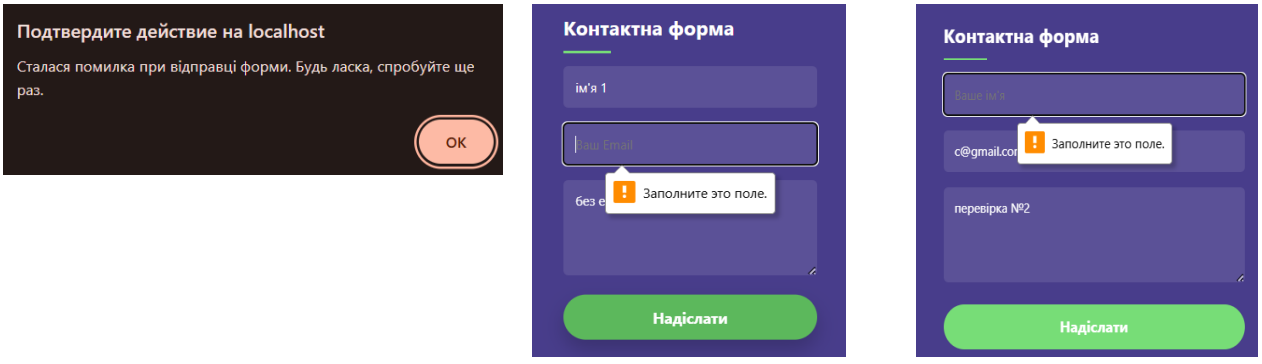


Рисунок 3.3 – Вигляд контактної форми із незаповненими полями та повідомленням

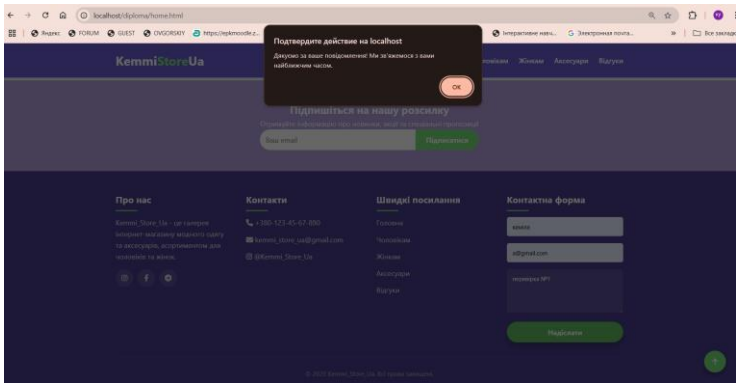


Рисунок 3.4 – Приклад правильної валідації

3.2.4 Створення серверної частини

Основу серверної частини сервісу складають технології PHP та MySQL. Ця комбінація забезпечує надійне зберігання даних та гнучку обробку запитів від клієнтської частини, дозволяючи динамічно керувати контентом галереї.

Для ефективної взаємодії з базою даних MySQL на серверній стороні використовується PHP. Всі запити до бази даних виконуються через спеціальний централізований файл api.php, який слугує єдиним ендпоінтом для отримання даних клієнтською частиною. Це забезпечує структурований підхід до взаємодії клієнта та сервера.

Приклад фрагменту PHP-коду наведено в додатку В з файлу api.php. Даний код обробляє запити для отримання товарів з урахуванням фільтрів за статтю, категорією та пошуковим запитом.

Наведений PHP-скрипт динамічно формує та виконує SQL–запити на основі параметрів, отриманих від клієнта. Він використовує підготовлені запити (mysqli_prepare, mysqli_stmt_bind_param), що є важливим для запобігання SQL–ін'єкціям та підвищення безпеки сервісу.

Для розробки та тестування серверної частини та взаємодії з базою даних використовувався локальний вебсервер. Створення підключення до бази даних

MySQL здійснювалося за допомогою окремого PHP-скрипту, який містить конфігураційні дані, необхідні для встановлення з'єднання з сервером бази даних (наприклад, адреса сервера localhost, ім'я користувача root, порожній пароль для локальної розробки).

Для створення таблиць у базі даних та заповнення їх початковими (тестовими) даними активно використовувався phpMyAdmin. phpMyAdmin – це безкоштовний програмний інструмент, написаний на PHP, який надає зручний вебінтерфейс для адміністрування сервера баз даних MySQL [11].

На рисунку 3.5 можна побачити структуру таблиці сутності товарів з кількома тестовими записами у phpMyAdmin, що демонструє її наповнення.

id	name	description	price	category_id	gender	brand	material	is_available	created_at
1	Чоловіча футболка "Класика"	Класична чоловіча футболка з високоякісної бавовни...	499.99	1	чоловічий	Kemmi_UA	100% Бавовна	1	2025-05-27 11:26:29
2	Джинсова спідниця міді	Класична джинсова спідниця міді - ідеальний варіан...	850.00	2	жіночий	Kemmi_UA	100% Денім	1	2025-05-27 11:26:29
3	Міський рюкзак "Подорожній"	Стильний та міцний рюкзак з багатьма відділеннями,...	1299.00	3	унісекс	PETERSON	Водонепроникний поліестер	1	2025-05-27 11:26:29

Рисунок 3.5 – Таблиця товарів у phpMyAdmin.

Така модель даних дозволяє ефективно зберігати та керувати інформацією про продукти, включаючи їх назви, категорії, ціни, описи, зображення та зв'язані дані, такі як доступні розміри та кольори (які отримуються за допомогою JOIN операцій в SQL-запиті).

Для налаштування маршрутизації серверної частини використовувався PHP. Кожен тип запиту (наприклад, отримання товарів, відправка контактної форми) обробляється відповідним блоком коду в центральному api.php файлі.

Як вже було зазначено раніше файл api.php слугує універсальним маршрутизатором, який обробляє різні запити на отримання даних про товари (всіх, відфільтрованих за статтю або категорією, або знайдених за пошуковим запитом). Використання асинхронних запитів (AJAX) з клієнтської частини до

цього файлу дозволяє отримати необхідні дані у форматі JSON без перезавантаження сторінки.

Приклад обробки POST-запиту для контактної форми наведено в додатку Г.

Цей приклад показує, як PHP-скрипт api.php може обробляти різні типи запитів (GET для отримання даних, POST для відправки форм) та відповідно реагувати, повертаючи JSON–відповіді.

На рисунку 3.9 представлено вигляд на сайті при відображенні усіх товарів наприклад, розділу "Чоловічий одяг", на рисунку 3.10 можна побачити результат з використаними фільтрами.

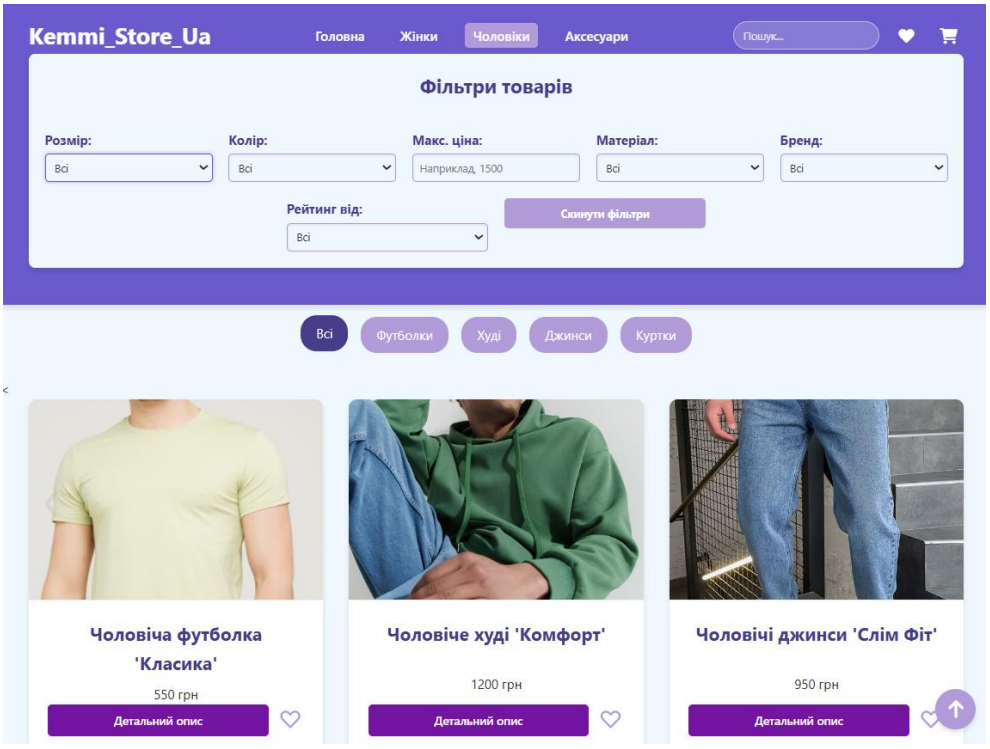


Рисунок 3.9 – Відображення усіх товарів

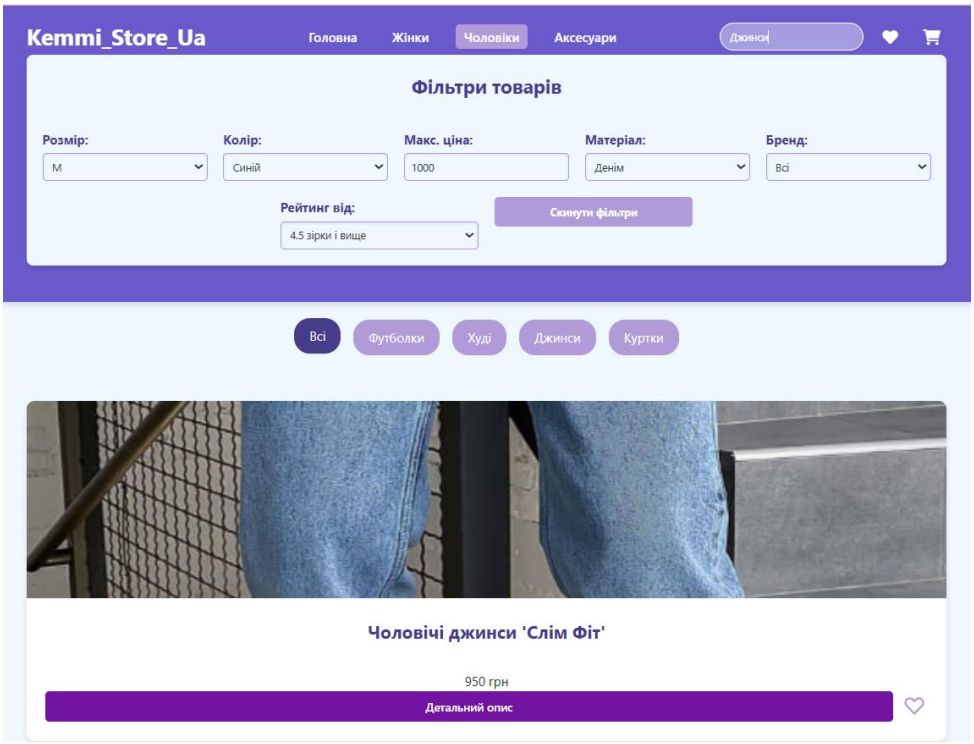


Рисунок 3.10 – Відображення товарів за фільтрами

3.3. Аналіз результатів та тестування

Після завершення етапу програмної реалізації всіх функціональних можливостей було проведено комплексне тестування системи. Під час тестування особливу увагу було приділено перевірці коректної роботи всіх елементів інтерфейсу та функціоналу (таких як завантаження контенту, фільтрація товарів, відображення модальних вікон та функція "Обране") в різних браузерах. Результати підтвердили повну сумісність та ідентичне відображення системи незалежно від використаного веббраузера, а також оцінка зручності використання (UX).

Результати тестування:

1. Завантаження та відображення товарів: підтверджено коректне завантаження товарів з бази даних MySQL через серверний API (api.php) для всіх

категорій (чоловічий одяг, жіночий одяг, аксесуари). Картки товарів відображаються коректно, містять усю необхідну інформацію: зображення, назву, опис, ціну, а також динамічно завантажувані розміри та кольори.

2. Фільтрація та пошук: функціонал фільтрації за категоріями (наприклад, "футболки", "джинси") працює відповідно до очікувань: список товарів динамічно оновлюється, відображаючи тільки релевантні позиції. Пошук за назвою або описом товару також функціонує бездоганно, надаючи точні результати на основі введеного запиту.

3. Перегляд деталей товару в модальному вікні: відкриття та закриття модального вікна з детальним описом товару реалізовано коректно. Динамічне завантаження розширеної інформації про товар (повний опис, галерея зображень, доступні розміри та кольори) функціонує без збоїв. Перемикання мініатюр зображень у модальному вікні відбувається плавно. Можливість закриття вікна натисканням кнопки "Закрити" та клавіші Escape також підтверджена.

4. Функціонал "Обране": додавання та видалення товарів зі списку "Обране" працює стабільно та без помилок. Дані про обрані товари надійно зберігаються у localStorage браузера, забезпечуючи їх збереження між сесіями користувача. Модальне вікно "Обране" коректно відображає додані товари та дозволяє їх видаляти зі списку.

5. Адаптивніст: кнопка "Прокрутити нагору" з'являється та функціонує коректно, забезпечуючи швидке повернення до початку сторінки.

6. Зручність використання: загальний користувацький досвід визнано позитивним. Інтерфейс є інтуїтивно зрозумілим, навігація по сайту легка та прозора. Швидкість завантаження сторінок та динамічного контенту (завдяки асинхронним запитам) є задовільною, що сприяє приємній взаємодії з системою. Валідація форм на стороні клієнта забезпечує миттєвий зворотний зв'язок користувачу, що підвищує зручність заповнення форм.

ВИСНОВКИ

У кваліфікаційній роботі було розглянуто проектування та розробку інтерактивної галереї магазину одягу "Kemmi_Store_Ua". Було проведено аналіз сучасних тенденцій у сфері вебдизайну та електронної комерції, визначено необхідні функціональні та нефункціональні вимоги до майбутнього вебзастосунку. На основі цього було сформульоване технічне завдання та встановлено технічні вимоги до розробки.

Під час виконання кваліфікаційної роботи було зроблено:

1. Дослідження поняття веброзробки та засобів для створення сучасних вебзастосунків. Було вивчено основні концепції веброзробки та інструменти, що використовуються для створення інтерактивних і функціональних вебзастосунків, зокрема, HTML5, CSS3 та JavaScript для клієнтської частини, PHP для серверної логіки та MySQL для зберігання даних.

2. Визначено архітектуру додатку та його компоненти. Було спроектовано клієнт-серверну архітектуру додатку, яка включає фронтенд на основі вебтехнологій, бекенд на PHP та реляційну базу даних MySQL. Визначено ключові компоненти системи та принципи їх взаємодії.

3. Реалізовано серверний API з використанням PHP. Серверна частина була розроблена з використанням PHP, яка надає API для ефективної взаємодії з базою даних MySQL. Було забезпечено обробку запитів на отримання списку товарів з можливістю фільтрації, пошуку та сортування.

4. Розроблено клієнтський інтерфейс з використанням HTML5, CSS3 та JavaScript. Було створено інтуїтивно зрозумілий та візуально привабливий інтерфейс користувача для галереї магазину одягу. Інтерфейс включає сторінки відображення каталогу товарів, деталізації продуктів у модальних вікнах та механізм зворотного зв'язку.

5. Забезпечено комунікацію між клієнтом та сервером за допомогою асинхронних HTTP-запитів. Комунікація між клієнтом і сервером здійснювалася

за допомогою асинхронних запитів, що забезпечило динамічне оновлення контенту без перезавантаження сторінки. Фронтенд надсилав запити на сервер для отримання даних товарів, фільтрації та відправки повідомлень зворотного зв'язку.

6. Реалізовано функціонал фільтрації, пошуку та відображення деталей товарів. Впроваджено гнучкі інструменти для фільтрації товарів за категоріями та статтю, а також ефективний пошук за назвою та описом. Деталі товарів динамічно завантажуються та відображаються у зручних модальних вікнах.

7. Впроваджено функціонал "Обране" з використанням локального сховища браузера. Реалізовано можливість додавання та видалення товарів до списку "Обране", дані якого надійно зберігаються на стороні клієнта за допомогою localStorage, забезпечуючи їх доступність між сесіями користувача.

8. Розроблено механізм зворотного зв’язку з клієнтською та серверною валідацією. Впроваджено форму зворотного зв'язку, яка дозволяє користувачам легко надсилати повідомлення. Забезпечено валідацію вхідних даних як на стороні клієнта, так і на сервері, для підвищення надійності та зручності.

9. Виконано комплексне тестування додатку. Було проведено тестування додатку для перевірки його коректної роботи, стабільності, функціональності та відповідності вимогам. Тестування включало перевірку роботи API, інтерфейсу користувача, взаємодії між компонентами, а також кросбраузерну сумісність.

Розроблена інтерактивна галерея магазину одягу "Kemmi_Store_Ua" відповідає сформованому технічному завданню та технічним вимогам. Було реалізовано повний цикл розробки від проектування архітектури до тестування і забезпечення якості програмного забезпечення. Використання сучасних вебтехнологій та підходу до взаємодії з базою даних дозволило створити функціональний, зручний у використанні, візуально привабливий та легко масштабований вебзастосунок, повністю готовий до потенційного впровадження у реальних комерційних умовах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-розробка - з чого розпочати навчання. URL: https://kiev.itstep.org/blog/web-development-where-to-start-learning (дата звернення: 28.05.2025).

2. Чим відрізняється Frontend та Backend розробка? Що обрати? URL: https://frontend.lviv.ua/chym-vidriznyayetsya-frontend-ta-backend-rozrobka-shho-obraty (дата звернення: 28.05.2025).

3. Тенденції технологій веб-розробки. URL: https://www.kulikoff.com.ua/articles_view/tendencii-tehnologij-veb-rozrobki-2024/ (дата звернення: 28.05.2025).

4. Досліджуємо ринок, аналізуємо тренди та ділимося досвідом. URL: https://gl.ua/blog/shvydkist-zavantazhennya-saytu-yak-odyn-z-faktoriv-ranzhuvannya?page=9 (дата звернення: 28.05.2025).

5. Top fashion ecommerce trends for 2025 - The future of fashion retail. URL: https://www.channelengine.com/en/blog/fashion-ecommerce-trends (дата звернення: 28.05.2025).

6. База даних MySQL. URL: https://promoter.net.ua/articles/baza-danix-mysql.html (дата звернення: 30.05.2025).

7. Клієнт-серверна архітектура. URL: https://training.qatestlab.com/blog/technical-articles/client-server-architecture/ (дата звернення: 01.06.2025).

8. Джонс, Ф. Н. Веб-розробка. Посібник із застосування HTML, XHTML, CSS та XML. Санкт-Петербург: «БХВ-Петербург», 2015. 766 с.

9. SPA у програмуванні: занурення у світ односторінкових додатків. URL: https://foxminded.ua/spa-u-prohramuvanni/ (дата звернення: 01.06.2025).

10. Валідація. Тестування валідації. URL: https://training.qatestlab.com/blog/technical-articles/validation-testing/ (дата звернення: 01.06.2025).

11. Основні можливості та вхід в phpMyAdmin. URL: https://thehost.ua/ua/wiki/ispmanager/database/phpmyadmin (дата звернення: 01.06.2025).

ДОДАТОК А

Стандартні можливості JavaScript функціоналу сортування товарів за різними критеріями, а саме за збільшенням та зменшенням ціни, за популярністю та новизною

```
// Є масив `currentProducts`, який містить поточні товари, отримані з сервера.
// Цей масив оновлюється щоразу при завантаженні або фільтрації товарів.
let currentProducts = [];
/**
 * Сортує масив товарів за заданим критерієм.
 * @param {string} sortOption – Опція сортування (наприклад, 'price–low–to–high', 'newest', 'popularity').
 */
function sortProducts(sortOption) {
    if (!currentProducts || currentProducts.length === 0) {
        return; // Якщо товарів немає, нічого не сортуємо
    }
    currentProducts.sort((a, b) => {
        if (sortOption === 'price–low–to–high') {
            return a.price – b.price; // Сортування за зростанням ціни
        } else if (sortOption === 'price–high–to–low') {
            return b.price – a.price; // Сортування за спаданням ціни
        } else if (sortOption === 'newest') {
            // Припускаємо, що у об'єктів товарів є поле 'created_at' у форматі ISO–8601 (наприклад, '2025–05–20T10:00:00Z')
            return new Date(b.created_at).getTime() – new Date(a.created_at).getTime(); // Сортування від новіших до старіших
        } else if (sortOption === 'popularity') { // Додано для явності
            // За замовчуванням або якщо опція не визначена, сортуємо за популярністю
            // Припускаємо, що є поле 'popularity_score' (числове значення)
            return b.popularity_score – a.popularity_score; // Сортування від найпопулярніших до менш популярних
        } else {
            // Якщо опція не розпізнана, можливо, повертаємося до початкового порядку або іншого дефолтного
            return 0; // Зберігаємо початковий порядок, якщо опція не збігається
        }
    });
    displayProducts(currentProducts); // Після сортування, знову відображаємо оновлений список товарів
}
// Приклад прив'язки цієї функції до елемента вибору сортування (наприклад, <select id="sortSelect">)
// document.getElementById('sortSelect').addEventListener('change', (e) => {
//     sortProducts(e.target.value);// });
```

ДОДАТОК Б

Лістинг коду JavaScript, який перевіряє, чи всі обов'язкові поля контактної форми було заповнено перед її відправкою, а також перевіряє формат електронної пошти

```
// Обробник події submit для контактної форми
document.getElementById('contactForm').addEventListener('submit', async function(e) {
  e.preventDefault(); // Запобігаємо стандартній відправці форми, щоб обробити її JavaScript'ом

  // Отримуємо значення полів та обрізаємо зайві пробіли
  const name = document.getElementById('contactName').value.trim();
  const email = document.getElementById('contactEmail').value.trim();
  const message = document.getElementById('contactMessage').value.trim();

  let errorMessage = ''; // Змінна для зберігання повідомлення про помилку

  // Виконуємо послідовну валідацію полів
  if (!name) {
    errorMessage = 'Будь ласка, введіть ваше ім\'я.';
  } else if (!email) {
    errorMessage = 'Будь ласка, введіть вашу електронну пошту.';
  } else if (!isValidEmail(email)) { // Виклик допоміжної функції для перевірки формату email
    errorMessage = 'Будь ласка, введіть коректний формат електронної пошти.';
  } else if (!message) {
    errorMessage = 'Будь ласка, введіть ваше повідомлення.';
  }

  if (errorMessage) {
    alert(errorMessage); // Відображаємо повідомлення про помилку користувачу (можна замінити на більш елегантне відображення)
    return; // Зупиняємо виконання функції, якщо є помилка
  }

  // Якщо валідація пройшла успішно, відправляємо дані форми асинхронно (AJAX–запитом)
```

```
        try {
            const formData = { name, email, message };
            const response = await fetch('/api/contact', { // Припустимо, на сервері є ендпоінт для обробки контактних форм
                method: 'POST', // Використовуємо метод POST для відправки даних
                headers: {
                    'Content–Type': 'application/json' // Вказуємо тип контенту як JSON
                },
                body: JSON.stringify(formData) // Перетворюємо об'єкт даних на JSON–рядок
            });

            if (response.ok) {
                alert('Дякуємо за ваше повідомлення! Ми зв\'яжемося з вами найближчим часом.');
                this.reset(); // Очищуємо форму після успішної відправки
            } else {
                throw new Error(`Помилка при відправці форми на сервер. Статус: ${response.status}`);
            }
        } catch (error) {
            console.error('Помилка відправки форми:', error);
            alert('Виникла помилка при відправці повідомлення. Будь ласка, спробуйте ще раз.');
        }
    });

    /**
     * Допоміжна функція для простої перевірки формату електронної пошти.
     * @param {string} email – Електронна пошта для перевірки.
     * @returns {boolean} – True, якщо формат email коректний, інакше False.
     */
    function isValidEmail(email) {
        return /^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email);
    }
```

ДОДАТОК В

Даний php код обробляє запити для отримання товарів з урахуванням фільтрів за статтю, категорією та пошуковим запитом

```
<?php
// api.php – Центральний API–ендпоінт для отримання даних про товари

// Підключення до файлу, що встановлює з'єднання з базою даних.
// `require_once` гарантує, що файл буде підключений лише один раз.
require_once 'db_connect.php';

// Встановлення HTTP–заголовка, що відповідь сервера буде у форматі JSON.
// Це необхідно для коректної інтерпретації даних на клієнтській стороні.
header('Content–Type: application/json');

$products_data = []; // Ініціалізація порожнього масиву для зберігання даних про товари

// Отримання параметрів фільтрації та пошуку з URL–запиту (GET–параметри).
// Використовується тернарний оператор для безпечного отримання значень або порожнього рядка, якщо параметр відсутній.
$gender = isset($_GET['gender']) ? $_GET['gender'] : '';           // Наприклад, 'чоловічий', 'жіночий', 'аксесуари'
$category = isset($_GET['category']) ? $_GET['category'] : '';    // Наприклад, 'футболки', 'джинси'
$search_query = isset($_GET['search']) ? $_GET['search'] : '';    // Рядок для пошуку за назвою/описом

// Базовий SQL–запит з JOIN–операторами для отримання додаткових даних (розмірів, кольорів)
// з пов'язаних таблиць. `GROUP_CONCAT` агрегує розміри та кольори в один рядок.
$sql_query = "SELECT p.id, p.name, p.description, p.price, p.image_url, p.gender,
                     GROUP_CONCAT(DISTINCT s.size_name) AS sizes,
                     GROUP_CONCAT(DISTINCT c.color_name) AS colors,
                     p.created_at, p.popularity_score  /* Додано created_at та popularity_score для сортування */
              FROM products p
              LEFT JOIN product_sizes ps ON p.id = ps.product_id
```

```
            LEFT JOIN sizes s ON ps.size_id = s.id
            LEFT JOIN product_colors pc ON p.id = pc.product_id
            LEFT JOIN colors c ON pc.color_id = c.id
            WHERE 1=1"; // Завжди істинна умова для зручного динамічного додавання фільтрів

        $params = []; // Масив для зберігання значень параметрів, які будуть прив'язані до запиту
        $param_types = ''; // Рядок, що визначає типи даних параметрів ('s' для string, 'i' для integer тощо)

        // Динамічне додавання умов фільтрації до SQL–запиту, якщо відповідні GET–параметри присутні.
        // Використовується підготовлений запит (`?`) для безпеки.
        if (!empty($gender)) {
            $sql_query .= " AND p.gender = ?";
            $param_types .= 's';
            $params[] = $gender;
        }

        if (!empty($category) && $category !== 'all') { // Якщо категорія не 'all' (тобто не всі), то фільтруємо
            $sql_query .= " AND p.category = ?";
            $param_types .= 's';
            $params[] = $category;
        }

        if (!empty($search_query)) {
            $sql_query .= " AND (p.name LIKE ? OR p.description LIKE ?)"; // Пошук за назвою або описом
            $param_types .= 'ss'; // Два параметри типу string
            $params[] = '%' . $search_query . '%'; // Додаємо символи % для пошуку підрядків
            $params[] = '%' . $search_query . '%';
        }

        // Завершення SQL–запиту: групування результатів та сортування
        // `GROUP BY p.id` агрегує дані для кожного продукту, щоб отримати всі розміри/кольори.
        // `ORDER BY p.created_at DESC` сортує товари від нових до старих (за замовчуванням)
        $sql_query .= " GROUP BY p.id ORDER BY p.created_at DESC";

        // Використання підготовлених запитів (`mysqli_prepare`) для запобігання SQL–ін'єкціям.
```

```
if ($stmt = mysqli_prepare($link, $sql_query)) {
    if (!empty($params)) {
        // Динамічна прив'язка параметрів до підготовленого запиту.
        // Оператор `...` (splat operator) розпаковує масив $params в окремі аргументи.
        mysqli_stmt_bind_param($stmt, $param_types, ...$params);
    }
    mysqli_stmt_execute($stmt); // Виконання підготовленого запиту
    $result = mysqli_stmt_get_result($stmt); // Отримання результату запиту

    // Обробка отриманих рядків даних
    while ($row = mysqli_fetch_assoc($result)) {
        // Перетворюємо рядки розмірів та кольорів (отримані через GROUP_CONCAT) на PHP–масиви.
        $row['sizes'] = !empty($row['sizes']) ? explode(',', $row['sizes']) : [];
        $row['colors'] = !empty($row['colors']) ? explode(',', $row['colors']) : [];
        $products_data[] = $row; // Додаємо оброблений рядок до основного масиву даних про товари
    }
    mysqli_stmt_close($stmt); // Закриття підготовленого запиту для звільнення ресурсів
} else {
    // Обробка помилок у разі, якщо підготовка SQL–запиту не вдалася.
    echo json_encode(["error" => "Помилка підготовки запиту до бази даних: " . mysqli_error($link)]);
    exit(); // Зупиняємо виконання скрипта
}

echo json_encode($products_data); // Відправляємо остаточні дані про товари у форматі JSON
mysqli_close($link); // Закриваємо з'єднання з базою даних
?>
```

ДОДАТОК Г

Обробка POST-запиту для контактної форми

```
<?php
// ... (початок api.php, підключення до БД, заголовки) ...

// Обробка POST–запитів
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Отримуємо JSON дані з тіла запиту
    $input = file_get_contents('php://input');
    $data = json_decode($input, true);

    // Перевірка, чи це запит для контактної форми
    if (isset($data['action']) && $data['action'] === 'submitContactForm') {
        $name = $data['name'] ?? '';
        $email = $data['email'] ?? '';
        $message = $data['message'] ?? '';
        // Проста серверна валідація (додатково до клієнтської)
        if (empty($name) || empty($email) || empty($message) || !filter_var($email, FILTER_VALIDATE_EMAIL)) {
            http_response_code(400); // Bad Request
            echo json_encode(['success' => false, 'message' => 'Некоректні дані форми.']);
            exit();
        }
        // Тут може бути логіка збереження повідомлення в БД, відправки email тощо
        // Наприклад, збереження у таблицю 'contact_messages'
        $stmt = mysqli_prepare($link, "INSERT INTO contact_messages (name, email, message, created_at) VALUES (?, ?, ?, NOW())");
        mysqli_stmt_bind_param($stmt, "sss", $name, $email, $message);

        if (mysqli_stmt_execute($stmt)) {
            http_response_code(200); // OK
            echo json_encode(['success' => true, 'message' => 'Повідомлення успішно відправлено!']);
        } else {
            http_response_code(500); // Internal Server Error
            echo json_encode(['success' => false, 'message' => 'Помилка сервера при збереженні повідомлення.']);
        }
        mysqli_stmt_close($stmt);
        exit();
    }
}

// ... (обробка GET–запитів, як показано вище) ...?>
```

Декларація

академічної доброчесності

здобувача освіти ВСП «Економіко-правничого фахового коледжу ЗНУ»

Я, Горб Каміла Дмитрівна, здобувач освіти 4 курсу, спеціальності/освітньо-професійної програми 121 Інженерія програмного забезпечення/Розробка програмного забезпечення, групи К121-21, адреса електронної пошти epfk.6121gorbkamila@gmail.com

- підтверджую, що написана мною кваліфікаційна робота на тему «Проєктування та розробка галереї онлайн магазину одягу» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст. 42 Закону України «Про освіту», зі змістом яких ознайомлена;

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

- згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 13.06.2025 Підпис _____ Каміла ГОРБ

Дата 13.06.2025 Підпис  Юлія БОРИСОВСЬКА