

Міністерство освіти і науки України Харківський
національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА ВЕБСАЙТУ ОНЛАЙН-МАГАЗИНУ З ПРОДАЖУ
ОДЯГУ

(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-20-1

Паламаренко А. І.

(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Тітова О.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки Факультет

Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«__» _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Паламаренко Анастасії Ігорівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебсайту онлайн-магазину з продажу одягу

затверджена наказом університету від 20 травня 2024 року № 464 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 6 червня 2024 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека з відкритим кодом React.

4. Перелік питань, що потрібно опрацювати в роботі

1. Аналітичний огляд мов програмування та бібліотек для розробки.

2. Моделювання інтернет магазину з продажу одягу.

3. Програмна реалізація застосунку.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми інтернет-магазину з продажу одягу, постановка задачі, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	08.04.2024	
2	Аналіз завдання, підбір літератури	08.04.24-15.04.24	
3	Аналіз літератури з досліджуваної проблеми	16.04.24-18.04.24	
4	Аналіз технічних засобів	19.04.24-25.04.24	
5	Моделювання застосунку	26.04.24-14.05.24	
6	Програмна реалізація	15.05.24-23.05.24	
7	Оформлення пояснювальної записки	24.05.24-31.05.24	
8	Перевірка на плагіат	01.06.24-02.06.24	
9	Рецензування	03.06.24-04.06.24	
10	Підготовка презентації та доповіді	05.06.24-09.06.24	
11	Занесення роботи в електронний архів	10.06.24-11.06.24	
12	Попередній захист кваліфікаційної роботи	12.06.2024	

Дата видачі завдання 8 квітня 2024 р.

Студент _____
(підпис)

Керівник роботи _____
(підпис)

_____ доц. Тітова О.В.
(посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 48 с., 32 рис., 30 джерел.

ВЕБСАЙТ, МОВА ГІПЕРТЕКСТОВОЇ РОЗМІТКИ HTML5, КАСКАДНІ ТАБЛИЦІ СТИЛІВ CSS, МОВА ПРОГРАМУВАННЯ JAVASCRIPT, JAVASCRIPT-БІБЛІОТЕКА REACT, JAVASCRIPT ОТОЧЕННЯ NODE.JS, ДОКУМЕНТНО-ОРІЄНТОВНА СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАННИХ MONGODB.

Об'єктом роботи є створення вебсайту онлайн-магазину з продажу одягу. Буде використовуватися JavaScript, Node.js, Express, React, MongoDB для розробки системи.

Метою роботи є створення надійної та продуктивної платформи, яка спрощує процес пошуку та придбання бажаного одягу для клієнтів.

Сайт має два рівні доступу: незареєстрований та зареєстрований користувач. Незареєстрований користувач може переглядати каталог і опис кожного товару, зареєструватися або увійти. Зареєстрований користувач може робити все те саме, плюс додавати товари у кошик та здійснювати покупки.

У результаті роботи створено вебсайт для продажу одягу з усіма базовими функціями, які зазвичай ми можемо побачити у подібних вебсайтах.

WEBSITE, HYPERTEXT MARKUP LANGUAGE HTML5, CASCADING STYLE SHEETS CSS, PROGRAMMING LANGUAGE JAVASCRIPT, JAVASCRIPT LIBRARY REACT, JAVASCRIPT ENVIRONMENT NODE.JS, DOCUMENT-ORIENTED DATABASE MANAGEMENT SYSTEM MONGODB.

The object of work is to create a website for an online clothing store. Will use JavaScript, Node.js, Express, React, and MongoDB to develop the system.

The aim of the work is to create a reliable and productive platform that simplifies the process of finding and purchasing the desired clothes for customers.

The site has two levels of access: unregistered and registered user. The unregistered user can view the catalog and description of each product? Register or login. A registered user can do all the same, plus add items to the cart and make purchases.

As a result, a website for the sale of clothes has been created with all the basic functions that we can usually see in such websites.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	6
Вступ	7
1 Теоретичний огляд предметної області.....	8
1.1 Актуальність інтернет-магазинів	8
1.2 Вибір методів розробки вебсайтів	9
1.2.1 Конструктор.....	9
1.2.2 CMS	11
1.2.3 Самостійна розробка	12
1.3 Постановка задачі	13
2 Моделювання структури вебсайту	15
2.1 Загальний огляд концепції роботи застосунку	15
2.2 Структура дизайну вебсайт «магазину з продажу одягу»	15
2.3 Розробка бізнес правил	16
2.4 Моделювання UML-діаграми для вебсайту	17
2.5 Методики Front-end частини.....	17
2.6 Методики Back-end частини.....	20
2.7 Обґрунтування обраної бази даних та її розробка.....	22
3 Програмна реалізація.....	28
3.1 Вибір засобів для реалізації поставленої задачі	28
3.2 Тестування реалізованого вебсайту	29
3.3 Перспективи подальшої роботи	42
Висновки	44
Перелік джерел посилання	45

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІМ – інтернет-магазин

HTML – HyperText Markup Language (мова розмітки гіпертексту) CSS – Cascading Style Sheets (каскадні таблиці стилів)

DOM – Document Object Model (об'єктна модель документа) MVC – Model-View-Controller (модель-вид-контролер)

UDF – Unidirectional Data Flow (односпрямований потік даних)

CBP – Component-Based Programming (компонентно-орієнтоване програмування)

CBFS – Component-Based Folder Structure (структура папок на основі компонентів)

REST – Representational State Transfer (передача репрезентативного стану)

WEB – World Wide Web (всесвітня павутина)

ВСТУП

В сучасному цифровому світі, де інформація швидко поширюється, а взаємодія з клієнтами та аудиторією відбувається переважно в онлайн-режимі, вебсайти виконують роль віртуальних вітрин, які пропонують доступ до найрізноманітніших продуктів, послуг та інформації з будь-якої точки світу, 24/7.

З розвитком мережі Інтернет, вебсайти стають невід'ємною частиною щоденного життя людей, незалежно від їхніх потреб та інтересів. Вони не лише забезпечують можливість знайти потрібну інформацію або придбати товари та послуги, але й стають центральним майданчиком для взаємодії з брендом, компанією чи організацією.

Якщо казати про ІМ, наразі вони стали ключовим інструментом для здійснення покупок. Завдяки розвитку сучасних технологій та зростання впливу інтернету на споживачів, відбулось перетворення, зручного раніше для нас способу покупок. ІМ відкрили нам набагато більше можливостей для придбання різноманітних товарів та послуг, що більше не обмежує нас, нашою локацією або часовими рамками.

Актуальність роботи, полягає у тому, що ІМ відкрили перед нами безмежні можливості для того, щоб ми швидко та зручно отримували необхідні товари та послуги. Неможливо описати їхнє значення в сучасному світі торгівлі, ІМ стали не просто містом, де споживач може придбати необхідну річ, а й важливим інструментом для закриття потреб сучасного споживача.

1 ТЕОРЕТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність інтернет-магазинів

Головна мета будь-якого ІМ, полягає в залученні більшого числа потенційних клієнтів, що допомагає в отриманні більшої кількості продажів. ІМ забезпечує клієнта усією необхідною йому інформацією про товар, починаючи з наявних розмірів, кольорів, складу тканини та закінчуючи вартістю, користувач має змогу порівняти ціни в різних магазинах, не виходячи з дому, ознайомитися з відгуками до товарів, які його цікавлять, та саме приємне, вони дають можливість оформити покупку всього за декілька хвилин, таким чином заощадивши час на відвіданні фізичних магазинів.

Ще однією вагомою перевагою ІМ є широкий асортимент. У порівнянні з офлайн магазинами, ІМ пропонують більший перелік товарів та послуг, оскільки вони не обмежені фізичним простором. Це дозволяє користувачам знайти все, що може їм знадобитися, без виходу з дома. З ще одних не очевидних плюсів, ІМ пропонують знижки та акції, які покупець може переглянути на сайті одразу всі, та дасть можливість зекономити гроші на покупках. Також за часту у ІМ зберігається історія перегляду користувача, що дає можливість швидко повернутись до товару, на який до цього покупець звернув увагу.

Хоча важлива інформація часто подається візуально, емоційні аспекти також можуть мати значний вплив на споживачів. Проте з кожним роком кількість ІМ збільшується, оскільки вони відповідають потребам покупців у зручності та вигоді. Більшість ІМ працюють цілодобово, бо не потребують участі продавця у купівлі товару. Щодо ІМ одягу, цей ринок наразі є доволі конкурентним, але саме це змушує нас вигадувати все більше зручних функцій для покупців, щоб полегшити їм покупку потрібного їм товару. Крім того, створення ІМ одягу може бути вигідним бізнесом для підприємців. За

допомогою вірно побудованого ІМ можна привертати нових клієнтів, розширювати аудиторію та збільшувати продажі. Зростання популярності онлайн-покупок, а також розвиток електронної комерції взагалі, створюють сприятливі умови для успішного функціонування інтернет-магазинів одягу, тому я вважаю цю область актуальною.

1.2 Вибір методів розробки вебсайтів

Вибір методів розробки вебсайтів зазвичай залежить від різноманітних факторів, включаючи обсяг проекту, доступність ресурсів, терміни виконання, технічні вимоги та бізнес-цілі. Усі ці методи ми можемо віднести до трьох категорій:

- розробка за допомогою конструктора сайтів;
- розробка за допомогою CMS (найпопулярніші WordPress або Tilda);
- самостійна розробка з нуля, з використання додаткових інструментів.

Кожен з цих методів має свої переваги та недоліки, і вибір залежить від конкретних вимог та обставин проекту.

1.2.1 Конструктор

Метод розробки вебсайтів за допомогою конструкторів є одним з найпоширеніших серед нефахівців у веб-розробці, оскільки він дозволяє створювати вебсайти без глибоких знань програмування. Підійде для тих, хто хоче зробити не складний вебсайт, можливо для себе та не розраховує на велику швидкість роботи вебсайту. Ось перелічено деякі переваги та недоліки використання конструкторів для розробки вебсайтів:

Плюси:

- економія коштів. Використання конструкторів дозволяє створювати сайти безкоштовно або за доступну ціну. Це особливо вигідно для невеликих бізнесів та особистих проєктів з обмеженим бюджетом;

- швидкість розробки. За допомогою конструкторів можна створити сайт лише за кілька хвилин. Виберіть шаблон, додайте контент і ваш сайт готовий. Це значно швидше, ніж розробка сайту з нуля, яка може займати тижні та навіть місяці;

- різноманітність шаблонів. Конструктори надають великий вибір шаблонів різних тематик, що задовольняють будь-який смак. Деякі з них навіть дозволяють частково редагувати шаблон;

- готова структура сайту. Вам не потрібно розмірковувати, як організувати сайт і розмістити на ньому інформацію. Конструктори надають готову структуру з прикладами, вам лише Потрібно додати свій контент.

Мінуси:

- шаблонний дизайн. Багато сайтів, створених за допомогою конструкторів, мають схожий дизайн, що може зменшити унікальність вашого сайту. І хоча деякі конструктори дозволяють змінювати шаблони, це може бути обмежено;

- обмежені можливості редагування. Конструктори можуть генерувати великий HTML код, що може негативно вплинути на швидкість завантаження сайту. Крім того, ви залежите від функціональності, яку пропонує обраний вами конструктор;

- обмежена структура сайту. Ви можете бути обмежені у зміні структури сайту, а також у функціоналі, який можна додати. Конструктори пропонують лише обмежений набір модулів;

- проблеми з пошуковою оптимізацією. Сайти, створені за допомогою конструкторів, мають проблеми з оптимізацією для пошукових систем, пошукові системи ставляться до них з великою недовірою, що може вплинути на їхнє ранжування в результатах пошуку.

1.2.2 CMS

Розглянемо також, такий метод розробки вебсайтів, як CMS. Цей метод базується на використанні спеціалізованих платформ для керування вмістом. Ці системи надають зручний інтерфейс для створення, редагування та управління вебсайтами, та не потребує великих знань у програмуванні. Завдяки CMS можна створювати, як легкі, так і складні рішення, змінювати контент та взаємодіяти з користувачем, керувати структурою сайту, а також використовувати різноманітні розширення для покращення функціональності. CMS підходить для абсолютно різних категорій користувачів, від блогерів до великих корпорацій, завдяки своїй гнучкості та простоті використання. Але навіть в такому варіанті є свої особливості.

Плюси:

- ціна. Більшість CMS безкоштовні, що дає змогу обрати шаблон, який сподобався та просто переробити під себе;
- швидкість розгортання. CMS дозволяють швидко створювати та розгортати сайти завдяки готовим шаблонам і функціоналу. Це особливо важливо для проектів з обмеженим часом;
- простота використання. CMS мають інтуїтивно зрозумілий інтерфейс, що дозволяє навіть неспеціалістам легко керувати вмістом сайту, додавати нові сторінки, редагувати тексти та зображення;
- гнучкість та розширюваність. Більшість CMS мають велику кількість додатків і плагінів, які дозволяють розширити функціонал сайту. Ви можете вибрати лише необхідні функції та легко інтегрувати їх у ваш сайт;
- багатофункціональність. CMS часто мають вбудовані модулі для різних потреб, таких як блог, форум, галерея зображень тощо. Це дозволяє створювати різноманітні типи сайтів без додаткового програмування.

Мінуси:

- обмежена гнучкість. Використання готових CMS може обмежити вас в реалізації складних або нестандартних функціональностей. Ви можете стикнутися з обмеженнями, якщо ваші потреби відхиляються від стандартних

шаблонів;

– залежність від розробників CMS. Ви залежите від виробників CMS для виправлення помилок, оновлень безпеки. Це може призвести до ситуації, коли ваш сайт може вплинути на оновлення або помилки CMS;

– безпека. Більшість атак на вебсайти спрямовані на CMS через їх популярність. Якщо ваша CMS не оновлюється регулярно або ви не вживаєте заходів безпеки, ваш сайт може бути уразливим для атак;

– швидкість завантаження. Неправильне використання CMS та додатків може призвести до повільної швидкості завантаження сайту, що може вплинути на користувацький досвід та рейтинг в пошукових системах.

1.2.3 Самостійна розробка

Самостійна розробка вебсайтів - це процес, коли особа або група людей розробляють сайт без залучення зовнішніх спеціалістів або використання готових платформ. У цьому випадку, розробники повинні мати різні навички, такі як програмування (HTML, CSS, JavaScript, тощо), дизайн, адміністрування серверів тощо. Самостійна розробка дозволяє повністю контролювати кожен аспект сайту і створити унікальний продукт, але вона також вимагає значних зусиль та знань. Такий підхід часто використовується тими, хто має достатні технічні навички і бажання вкласти час і зусилля у створення свого веб-проекту. У цьому варіанті теж є, як свої переваги так і недоліки [1-2].

Плюси:

– контроль. Розробник має повний контроль над усіма аспектами проекту, включаючи дизайн, функціональність та безпеку;

– унікальність. Здатність створити унікальний продукт, який повністю відповідає вимогам та потребам клієнта або власного проекту;

– навички. Можливість розвивати та покращувати технічні навички

веб-розробки, програмування та дизайну;

– ефективність. Здатність адаптувати сайт до специфічних потреб та вимог клієнта без обмежень, які можуть існувати при використанні готових платформ.

Мінуси:

– час. Самостійна розробка може зайняти набагато більше часу, тижні або навіть місяці, особливо якщо розробник не має достатньої кількості досвіду;

– складність. Процес розробки може бути складним, особливо для початківців, і вимагати великої кількості зусиль та досліджень;

– брак підтримки. Відповідальність за підтримку та обслуговування сайту покладається на розробника, що може стати проблемою, наприклад у разі виникнення технічних проблем;

– обмежені можливості. Відсутність доступу до готових рішень та плагінів, які пропонуються готовими платформами, може обмежити функціональність та можливості сайту;

– ціна. Створення власного вебсайту може виявитися витратним процесом через необхідність оплати ресурсів, таких як хостинг, доменне ім'я та інструменти розробки, які можуть додатково обтяжувати бюджет проекту. Також потрібно враховувати витрати на підтримку, оновлення та розвиток сайту у майбутньому.

1.3 Постановка задачі

Об'єктом роботи є створення вебсайту інтернет-магазину для продажу одягу. Буде використовуватися JavaScript, Node.js, Express, React та MongoDB для розробки системи.

Метою роботи є створення надійного та ефективного магазину, що дозволяє користувачам легко та швидко знайти та замовити бажану гру.

Для досягнення мети необхідно вирішити такі завдання:

- забезпечити можливість реєстрації та авторизації користувачів для здійснення покупок;
- створити зручний та естетичний інтерфейс, який відображає асортимент одягу;
- забезпечити можливість перегляду докладної інформації про кожен товар, включаючи фотографії, розміри, ціни тощо;
- забезпечити можливості переглядати категорії товарів;
- оптимізувати сайт для швидкого завантаження та коректної роботи на різних пристроях;
- забезпечити пошуковий функціонал для швидкого знаходження певного товару або категорії;
- реалізувати функціонал додавання товарів у кошик;
- оформлення замовлення та оплати.

2 МОДЕЛЮВАННЯ СТРУКТУРИ ВЕБСАЙТУ

2.1 Загальний огляд концепції роботи застосунку

Основною ідеєю цієї роботи є створення ІМ різного одягу, який буде поділений на декілька категорій: жіночий одяг, чоловічий одяг та дитячий одяг. Вебзастосунок буде поєднувати в собі інформаційну частину з оглядом вмісту кожного пункту.

В даному вебзастосунку буде використовуватися три варіанти осіб, які можуть їм користуватися: адміністратор, користувач, який не створював аккаунт та зареєстрований користувач. В кожного з цих варіантів будуть свої можливості та особливості використання, давайте розглянемо кожен з них.

Користувач, який не проходив реєстрацію, буде мати змогу переглядати асортимент сайту, переглядати відгуки, додавати сподобавшийся товар до кошика, проте здійснити замовлення змоги в нього не буде.

Зареєстрований користувач, буде мати точно такі самі можливості, як і користувач, який не реєструвався на сайті, за винятком, що в нього буде змога оформити замовлення.

Якщо казати вже про адміністратора, цей користувач, буде мати можливість додавати та видаляти товари ІМ, також в нього буде змога редагувати вже існуючі товари ІМ.

2.2 Структура дизайну вебсайт «магазину з продажу одягу»

Даний вебсайт будемо вибудовувати методом блочної верстки (рис. 2.1), такий метод включає в себе декілька секцій, на які поділяється вебсайт: header, footer, main та article [3-5].

Завичай, використовуючи даний метод, на сайтах також додають aside, інколи один, а інколи два, в протилежних кутках вебсайту. Проте на даному

вебсайті, здалось додавати їх недоречним.

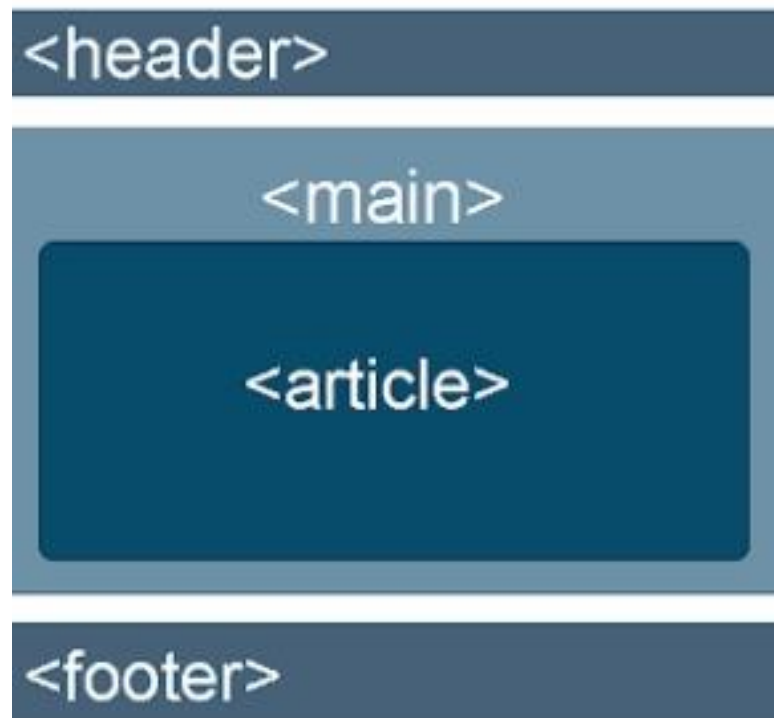


Рисунок 2.1 - Макет головної сторінки вебсайту «магазину з продажу одягу»

2.3 Розробка бізнес правил

Для ефективної роботи застосунка, потрібно створити бізнес правила для даного вебсайту:

- користувач, що не реєструвався на сайті, може переглядати усі сторінки сайту;
- користувач, що не реєструвався, може додавати товари до кошика;
- зареєстрований користувач, може переглядати усі сторінки ІМ;
- зареєстрований користувач, може додавати товари до кошика;
- зареєстрований користувач, може оформлювати замовлення того, що додав до свого кошика;
- адміністратор, може додавати нові товари до сайту, видаляти товари з сайту, а також редагувати вже існуючі товари.

2.4 Моделювання UML-діаграми для вебсайту

Виходячи з тих бізнес-правил, що ми аналізували, можна створити таку UML-діаграму для вебсайту «магазину з продажу одягу» (рис. 2.2).

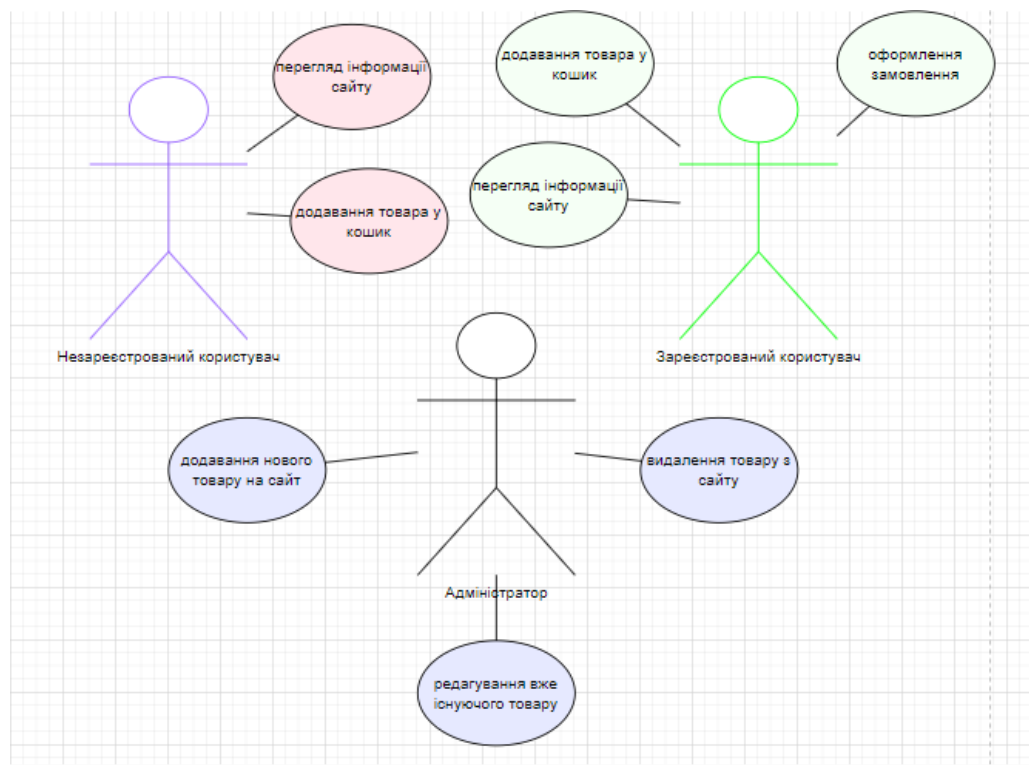


Рисунок 2.2 - UML-діаграма користувачів «магазину з продажу одягу»

2.5 Методики Front-end частини

Бібліотека React є основою для створення вебсайту магазину одягу, це одна з найпопулярніших JavaScript бібліотек для створення користувацьких інтерфейсів. Він має широкий спектр застосування, включаючи розробку ІМ. Роль React у створенні вебсайту з продажу одягу можна детально розглянути з різних аспектів, включаючи архітектуру компонентів, продуктивність, управління станом, інтеграцію з бекендом та багато іншого [6-10].

React відрізняється від інших фреймворків своїм компонентним підходом, який дозволяє розробникам створювати модульні та повторно використовувані компоненти. Це особливо важливо для великих та складних

додатків, таких як IM. Основні переваги React включають:

- компонентна архітектура. Можливість створення незалежних, багаторазових компонентів;
- віртуальний DOM. Швидке оновлення та рендеринг користувацького інтерфейсу;
- односторонній потік даних. Простота у відстеженні та налагодженні стану застосунку.

Основною одиницею в React є компонент. Кожен компонент відповідає за певну частину інтерфейсу, що дозволяє розробникам розділити складний інтерфейс на менші, керовані частини. Наприклад, у вебсайті з продажу одягу можуть бути такі компоненти:

- Header. Компонент, що відображає заголовок сайту, навігацію та кошик;
- ProductList. Компонент, який відповідає за відображення списку товарів;
- ProductItem. Компонент, що представляє окремий товар з детальною інформацією;
- Cart. Компонент, що управляє кошиком користувача;
- Checkout. Компонент, що забезпечує оформлення замовлення.

Одна з ключових переваг React – можливість повторного використання компонентів. Наприклад, компонент ProductItem може використовуватися як у списку товарів, так і на сторінці детальної інформації про товар.

React використовує віртуальний DOM для підвищення продуктивності. Віртуальний DOM є копією реального DOM, яка знаходиться в пам'яті. Коли стан компонента змінюється, React оновлює віртуальний DOM, а потім порівнює його з реальним DOM. Це дозволяє React мінімізувати кількість маніпуляцій з реальним DOM, що суттєво покращує швидкість рендерингу.

React оптимізує процес оновлення DOM, оновлюючи тільки ті частини інтерфейсу, які дійсно змінилися. Це особливо важливо для великих додатків, де часті оновлення DOM можуть призвести до зниження

продуктивності.

React використовує односторонній потік даних, що робить управління станом додатку більш передбачуваним та легшим для відстеження. Дані передаються від батьківських компонентів до дочірніх через пропси, що забезпечує чітку структуру даних.

Односторонній потік даних дозволяє легко інтегрувати React з іншими бібліотеками та фреймворками, забезпечуючи зручну роботу з глобальним станом додатку.

Для управління станом великого додатку, такого як ІМ, можуть використовувати Redux або Context API. Вони дозволяють централізовано керувати станом додатку, спрощуючи передачу даних між компонентами та зменшуючи кількість пропсів.

У React кожен компонент може мати свій власний стан. Це особливо корисно для управління локальним станом, наприклад, у формі для оформлення замовлення.

React легко інтегрується з бекенд-сервісами для отримання та відправки даних. Наприклад, для ІМ можна використовувати API для отримання списку товарів, даних про користувачів та управління замовленнями.

Для взаємодії API у React можуть використовуватися бібліотеки Axios або вбудований метод Fetch. Вони дозволяють легко здійснювати HTTP-запити та обробляти відповіді сервера.

React дозволяє створити динамічний каталог товарів, який автоматично оновлюється на основі даних з бекенду. Це включає можливість фільтрації та сортування товарів з різними критеріями (ціна, бренд, категорія).

Компоненти React дозволяють легко реалізувати кошик покупок з можливістю додавання, видалення та оновлення товарів. Стан кошика можна зберігати локально або використовувати глобальний стан, наприклад, через Redux.

Інтерактивна форма для оформлення замовлення з перевіркою даних у

реальному часі забезпечує зручний процес покупки для користувачів. React дозволяє динамічно оновлювати форму та перевіряти введені дані.

Реалізація особистого кабінету користувача включає можливість перегляду та редагування профілю, а також перегляду редагування профілю, а також перегляду історії замовлень. React компоненти забезпечують динамічне оновлення інтерфейсу на основі дій користувача.

2.6 Методики Back-end частини

Як платформа для серверного програмування на Javascript, Node.js стає все більш популярною серед розробників завдяки своїй гнучкості, високій продуктивності та масштабованості. У сучасному світі веброзробки, де швидкість, ефективність і можливість масштабування є ключовими факторами успіху, Node.js надає розробникам інструменти, необхідні для створення динамічних та інтерактивних вебдодатків. Зокрема, для ІМ з продажу одягу, Node.js пропонує безліч переваг, які сприяють покращенню користувацького досвіду, зниженню витрат на розробку та обслуговування, а також забезпеченню високої надійності та швидкості роботи сайту. Розглянемо детально роль Node.js у створенні вебсайтів для ІМ і ті переваги, які він надає [11-16].

Node.js дозволяє використовувати JavaScript як клієнтський, так і на серверній стороні. Це означає, що розробники можуть використовувати одну мову для написання як фронтенд, так і бекенд частин додатку. Це значно спрощує процес розробки та обслуговування додатку, оскільки всі частини проекту написані однією мовою.

Node.js базується на асинхронній архітектурі, що дозволяє обробляти багато запитів одночасно без блокування. Завдяки цьому сервер може обробляти великий обсяг запитів з високою швидкістю. Для ІМ це означає, що сайт зможе обробляти велику кількість користувачів та запитів до бази даних одночасно, не знижуючи продуктивності.

Node.js підтримує горизонтальне та вертикальне масштабування. Це означає, що можна легко додавати нові ресурси (сервери або ядра процесора) для обробки зростаючого навантаження. Для ІМ це особливо важливо під час пікових навантажень, таких як сезонні розпродажі чи акції.

Завдяки асинхронній обробці запитів та швидкому виконанню JavaScript на сервері, Node.js забезпечує високу швидкість завантаження сторінок. Це важливо для користувачів, оскільки швидкість завантаження безпосередньо впливає на їх задоволеність та ймовірність завершення покупки.

Node.js має величезну екосистему модулів та бібліотек, доступних через NPM (Node Package Manager). Це дозволяє швидко інтегрувати різноманітні функції до вашого додатку, такі як обробка платежів, аутентифікація, управління базою даних та багато іншого. Для ІМ це означає, що розробники можуть легко додавати нові функціональні можливості, не витрачаючи багато часу на їх реалізацію з нуля.

Node.js підтримує WebSockets, що дозволяє творити додатки з функціональністю реального часу. Це може бути корисно для інтерактивних елементів ІМ, таких як чати підтримки клієнтів, оновлення стану замовлень у реальному часі або нотифікації про зміну статусу товарів.

Node.js легко інтегрується з багатьма хмарними платформами, такими як AWS, Google Cloud Platform та Microsoft Azure. Це дозволяє швидко розгорнути та масштабувати вебсайт без значних зусиль. Для ІМ це означає, що ви можете швидко запускати нові версії додатку та легко масштабувати інфраструктуру відповідно до зростаючих потреб [17-20].

Node.js має велику та активну спільноту розробників, яка постійно працює над удосконаленням платформи, виправленням помилок та розробкою нових модулів. Це означає, що ви завжди можете знайти підтримку, приклади коду та рішення для будь-яких проблем, з якими можете зіткнутися під час розробки.

Node.js надає потужні інструменти для розробки високопродуктивних

та масштабованих веб-додатків. Для ІМ з продажу одягу це означає можливість створення швидкого, надійного та масштабованого сайту, який здатний обробляти велику кількість користувачів одночасно. Використання однієї мови на клієнтській та серверній стороні, легка інтеграція з різними сервісами, підтримка реального часу та велика екосистема модулів роблять Node.js відмінним вибором для сучасних ВЗ.

2.7 Обґрунтування обраної бази даних та її розробка

Бази даних були створені через необхідність обробки та зберігання великої кількості інформації. База даних вебсайту для продажу одягу відіграє ключову роль у забезпеченні ефективної роботи та зберіганні великих обсягів інформації. Інформація про клієнтів, їх замовлення та інші важливі дані повинні бути належним чином організовані та захищені. Для цього часто використовують документно-орієнтовану базу даних, яка відповідає потребам сучасного онлайн-бізнесу.

Однією з особливостей документно-орієнтованої бази даних є можливість зберігати дані у вигляді документів, які містять інформацію у вигляді пар «ключ-значення». Наприклад, кожен клієнт може мати свій документ, який містить його особисті дані, історію замовлень та іншу інформацію [21-26].

Також важливою перевагою документно-орієнтованих баз даних є можливість гнучкої моделі даних, яка дозволяє зберігати різноманітні дані в одній колекції. Наприклад, дані про товари, категорії, замовлення та клієнтів можуть знаходитися в різних документах, але легко взаємодіяти між собою.

Щодо безпеки, документно-орієнтовані бази даних можуть забезпечити шифрування даних на рівні бази даних, а також контроль доступу до даних на рівні документів. Це дозволяє захистити конфіденційні дані клієнтів та іншу важливу інформацію від несанкціонованого доступу.

Таким чином, така база даних є ефективним і надійним рішенням для

забезпечення функціональності та безпеки вебсайту для продажу одягу.

MongoDB – це система баз даних NoSQL з відкритим вихідним кодом, яка призначена для зберігання та обробки великих обсягів неструктурованих даних. Відмінною рисою таких баз даних є їх гнучкість та можливість зберігати дані у вигляді JSON-подібних документів, що дозволяють зберігати дані з різною структурою без потреби встановлення схеми даних.

MongoDB є безкоштовною для використання та має відкритий вихідний код, що означає, що користувачі можуть завантажити та використовувати її безкоштовно без обмежень. В порівнянні з комерційними системами баз даних, MongoDB набагато дешевша, проте це не впливає на її функціональність та надійність.

MongoDB відома своєю високою продуктивністю, гнучкістю та можливістю масштабування. Вона підтримує розподілені системи, які дозволяють легко розширювати обсяги даних та завантаження. Також MongoDB має потужний механізм запитів та агрегацій, який дозволяє ефективно виконувати складні запити до бази даних [27-30].

Для того, щоб розробити базу даних даної програми, треба знати які сутності ми будемо використовувати.

Так як це ІМ для продажу одягу, товар який перелічен на сайті, потрібно десь зберігати. Для даної задачі створимо таблицю products:

- `_id` (унікальний ідентифікатор товару);
- `name` (назва товару);
- `description` (опис товару);
- `price` (ціна товару);
- `category` (категорія товару);
- `image` (посилання на зображення товару).

Також у застосунку будуть присутні користувачі, з різними ролями (адміністратор, зареєстрований користувач та незареєстрований користувач), для них ми створимо таблицю users:

- `_id` (унікальний ідентифікатор користувача);

- username (ім'я користувача);
- email (електронна пошта користувача);
- password (хеш пароля користувача);
- role (роль користувача: admin, registered, unregistered).

Також в будь-якого користувача, буде можливість додати товар для кошика. Треба зберігати ці дані, для цього буде створена таблиця carts:

- _id (унікальний ідентифікатор кошика);
- user_id (ідентифікатор користувача);
- products (масив ідентифікаторів товарів у кошику).

Товари які потрапляють до кошика ми виведемо в окрему таблицю, назвемо її orders:

- _id (унікальний ідентифікатор замовлення);
- user_id (ідентифікатор користувача);
- products (масив ідентифікаторів товарів у замовленні);
- total_price (повна вартість замовлення);
- status (статус замовлення: pending, processing, completed).

На основі визначених раніше даних, створимо структуру бази даних застосунку.

Лістинг 2.1 Таблиця «Products»:

```
"products": [
  {
    "_id": ObjectId(),
    "name": "T-shirt",
    "description": "Cotton T-shirt with logo print",
    "price": 200,
    "category": "men",
    "image": "https://example.com/tshirt.jpg"
  },
  {
    "_id": ObjectId(),
```



```

    "name": "Jeans",
    "description": "Blue denim jeans with skinny fit",
    "price": 800,
    "category": "women",
    "image": "https://example.com/jeans.jpg"
  }
],

```

ЛІСТИНГ 2.2 Таблиця «Users»:

```

"users": [
  {
    "_id": ObjectId(),
    "username": "user123",
    "email": "user123@example.com",
    "password": "hashed_password",
    "role": "admin"
  },
  {
    "_id": ObjectId(),
    "username": "user456",
    "email": "user456@example.com",
    "password": "hashed_password",
    "role": "registered"
  },
  {
    "_id": ObjectId(),
    "username": "guest",
    "email": "guest@example.com",
    "password": "hashed_password",
    "role": "unregistered"
  }
]

```

```
]

```

Лістинг 2.3 Таблиця «Carts»:

```
"carts": [
  {
    "_id": ObjectId(),
    "user_id": ObjectId("user_id_here"),
    "products": [
      ObjectId("product_id1"),
      ObjectId("product_id2")
    ]
  }
],

```

Лістинг 2.4 Таблиця «Orders»:

```
"orders": [
  {
    "_id": ObjectId(),
    "user_id": ObjectId("user_id_here"),
    "products": [
      ObjectId("product_id1"),
      ObjectId("product_id2")
    ],
    "total_price": 1200,
    "status": "completed"
  }
]
}

```

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір засобів для реалізації поставленої задачі

Під час розробки сайту було створено діаграми прецедентів та послідовностей, які організують цілі розробки вебсайту. Це допомогло визначити ролі та характеристики вебсайту.

Діаграма прецедентів показує взаємодію користувача з застосунком. Основна ідея цього методу полягає в зображенні дійових осіб або сутностей, які взаємодіють з застосунком через варіанти використання. Це допомагає краще розуміти, які функції потрібно реалізувати.

Для розробки сайту з продажу одягу використовувався React. React – це бібліотека для створення інтерфейсів користувача, яка дозволяє створювати ефективні ВЗ. Цей підхід використовується для створення багатосторінкових вебсайтів з динамічними інтерфесами. Це робить сайт набагато швидшим у реагуванні на дії користувача порівняно зі звичайними методами.

React підходить для великих проектів, оскільки ця бібліотека дозволяє створювати масштабовані та надійні вебсайти. Однією з основних переваг React є його компонентна структура, яка дозволяє задіяти тільки необхідні частини, не перевантажуючи код. React має багато інструментів, які допомагають реалізувати складні задачі. Постійне вдосконалення продукту є важливою перевагою в сфері програмування. Для спрощення роботи з React можна використовувати різні бібліотеки.

Visual Studio Code – це інтегроване середовище для розробки на JavaScript та пов'язаних технологіях. Перевагами цього сервісу є простий і зручний інтерфейс, підсвічування та аналіз коду під час його редагування. Visual Studio Code підтримує фреймворки React, Angular та інші.

MongoDB – це потужна документо-орієнтована система баз даних з відкритим вихідним кодом. Основними перевагами MongoDB є саме відкритий код, потужність та велика кількість функцій.

Node.js – забезпечує високу продуктивність завдяки використанню V8 JavaScript двигуна і асинхронної, неблокуючої архітектури, що підвищує ефективність обробки запитів. Це середовище дозволяє створювати масштабовані мережеві додатки, використовуючи JavaScript як на клієнтській, так і на серверній стороні, що спрощує розробку. Крім того, Node.js має велику екосистему модулів і пакетів через npm.

Загалом, застосування сучасних технологій і підходів у розробці вебсайту магазину з продажу одягу забезпечило високий рівень функціональності, продуктивності та зручності користування. Це дозволило створити ефективний і конкурентоспроможний продукт, який відповідає потребам користувачів і вимогам ринку.

3.2 Тестування реалізованого вебсайту

Тестування вебсайту є критично важливим етапом у процесі розробки, оскільки воно забезпечує функціональність, стабільність і безпеку кінцевого продукту. Відсутність належного тестування може призвести до появи помилок і вразливостей, які можуть негативно вплинути на користувацький досвід і довіру клієнтів. Тестування дозволяє виявити та виправити помилки на ранніх стадіях, що зменшує ризик виникнення серйозних проблем під час роботи сайту.

Іншою важливою причиною тестування є забезпечення сумісності сайту з різними браузерами та пристроями. Користувачі можуть заходити на сайт з різних пристроїв, таких як комп'ютери, планшети чи смартфони, і з різних веббраузерів. Тестування допомагає переконатися, що сайт виглядає та функціонує коректно незалежно від платформи, що значно покращує користувацький досвід.

Безпека вебсайту також є ключовим аспектом, який необхідно ретельно перевіряти. Тестування на вразливості, допомагає захистити дані

користувачів і забезпечити цілісність системи. Регулярне тестування безпеки допомагає підтримувати високий рівень захисту та запобігати можливим атакам.

Таким чином, тестування є невід’ємною частиною процесу розробки, що забезпечує надійність, сумісність і безпеку вебсайту. Воно сприяє створенню позитивного користувацького досвіду та підвищує довіру клієнтів до вебсайту, що є важливим для успішного функціонування будь-якого ІМ.

Головна сторінка даного вебсайту, знаходиться на рисунку 3.1, вона виглядає однаково, як для зареєстрованого користувача, так і для незареєстрованого. На цій сторінці є можливість перемикачів між категоріями та увійти в свій аккаунт або переглянути корзину своїх замовлень.

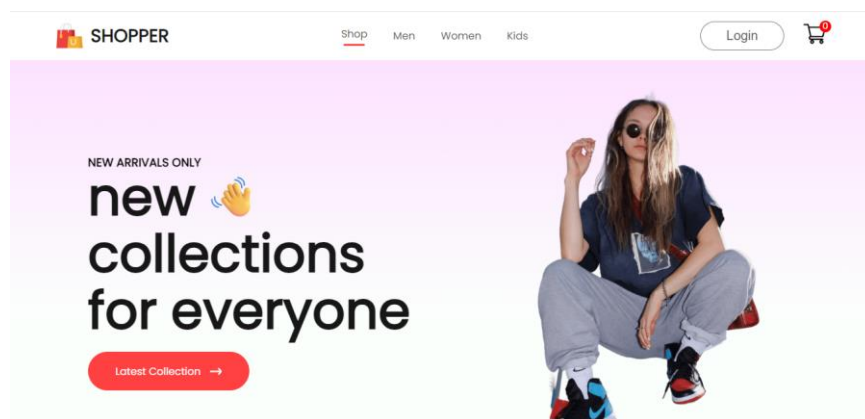


Рисунок 3.1 – Головна сторінка вебсайту для зареєстрованих та незареєстрованих користувачів

На головній сторінці також зверху предоставлене меню, де користувач може обрати потрібну йому категорію, що представлено на рисунку 3.2.



Рисунок 3.2 – Головне меню вебсайту

Обираючи категорію «men», користувач потрапляє на сторінку, де може переглянути каталог чоловічого одягу (рис. 3.3).

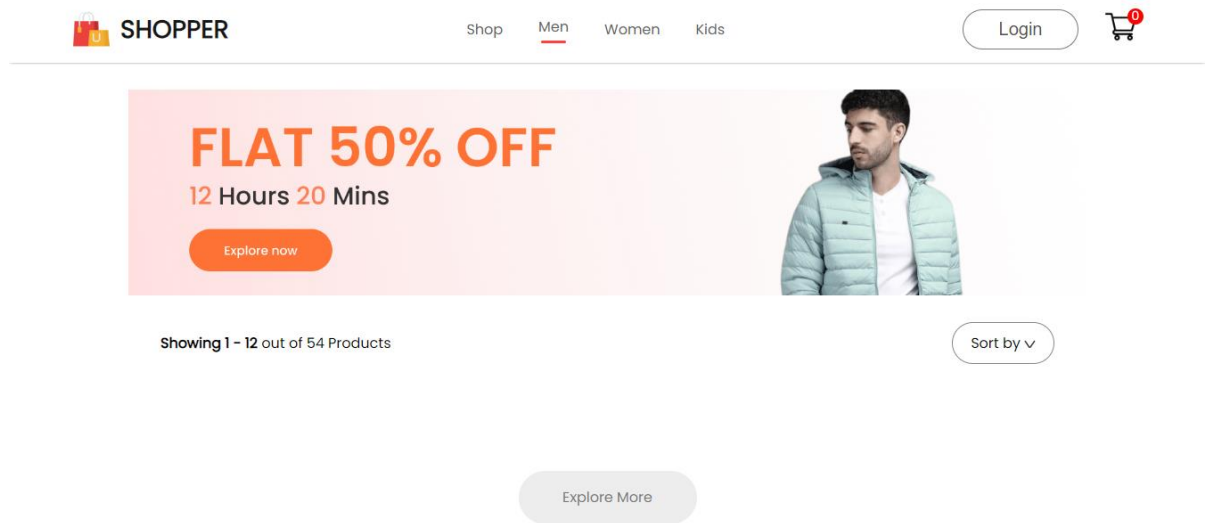


Рисунок 3.3 – Сторінка чоловічого каталогу

Обираючи категорію «women», користувач потрапляє на сторінку, де може переглянути каталог жіночого одягу (рис. 3.4).

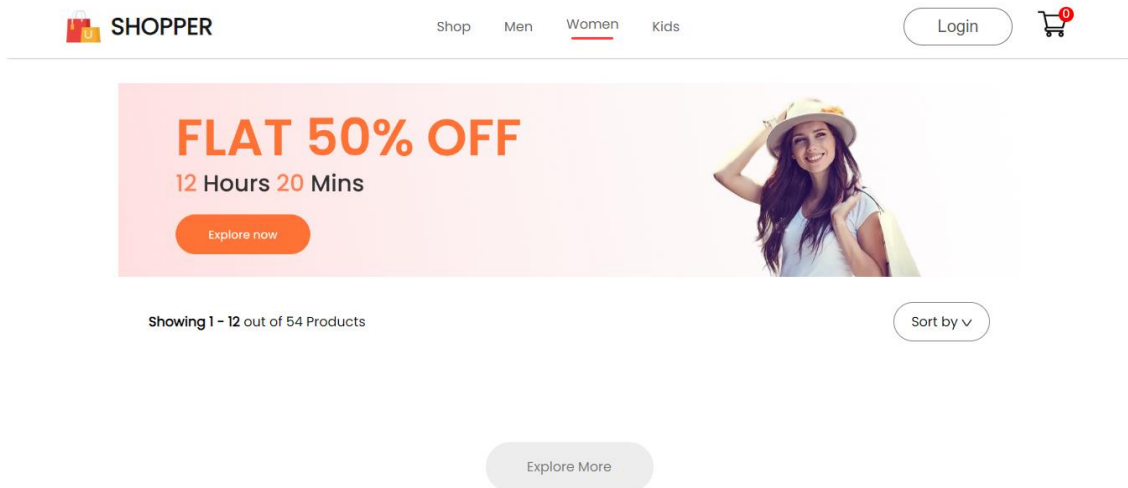


Рисунок 3.4 – Сторінка жіночого каталогу

Обираючи категорію «kids», користувач потрапляє на сторінку, де може переглянути каталог дитячого одягу (рис. 3.5).

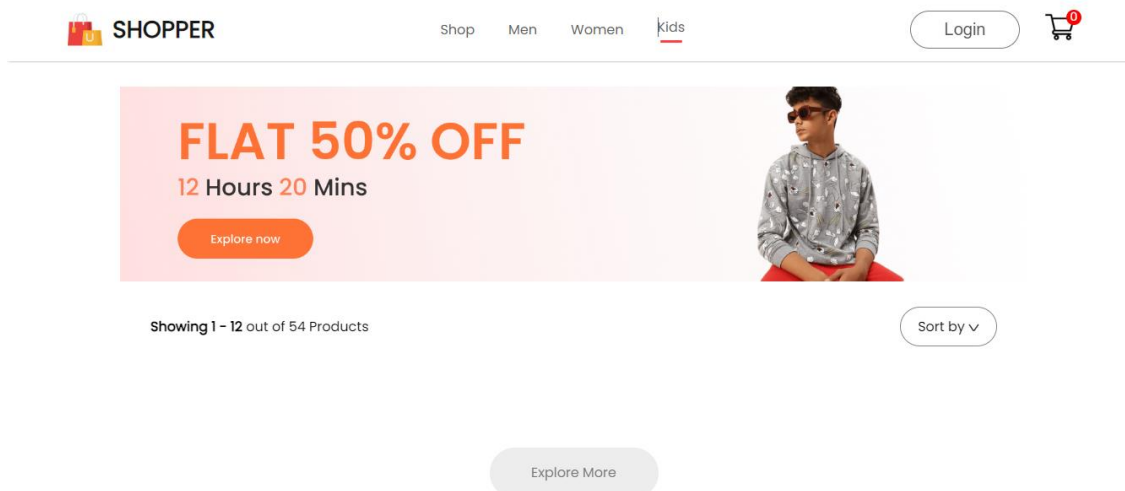


Рисунок 3.5 – Сторінка дитячого каталогу

Користувач, для зручності, також має можливість переглянути сторінку конкретного товару, який його зацікавив (рис. 3.6).

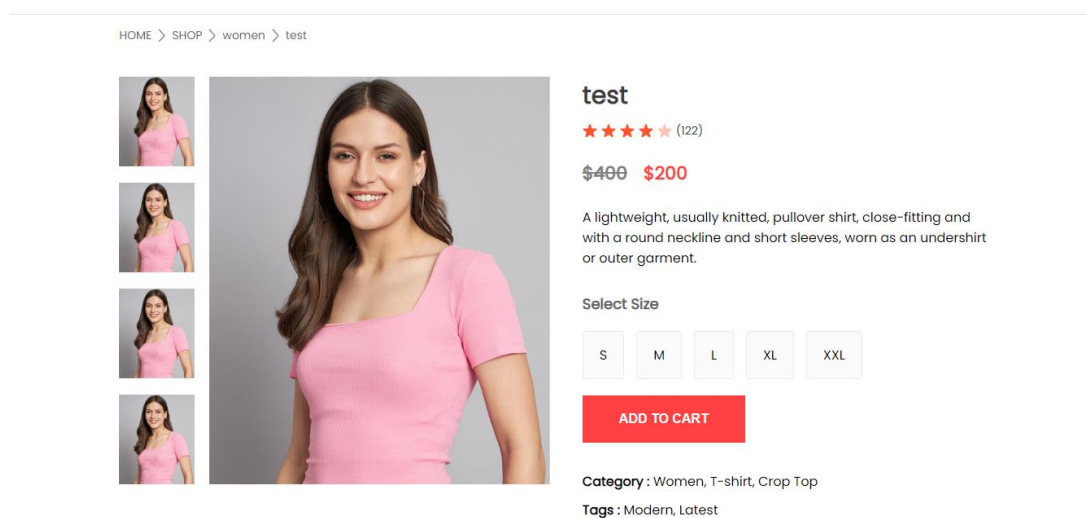
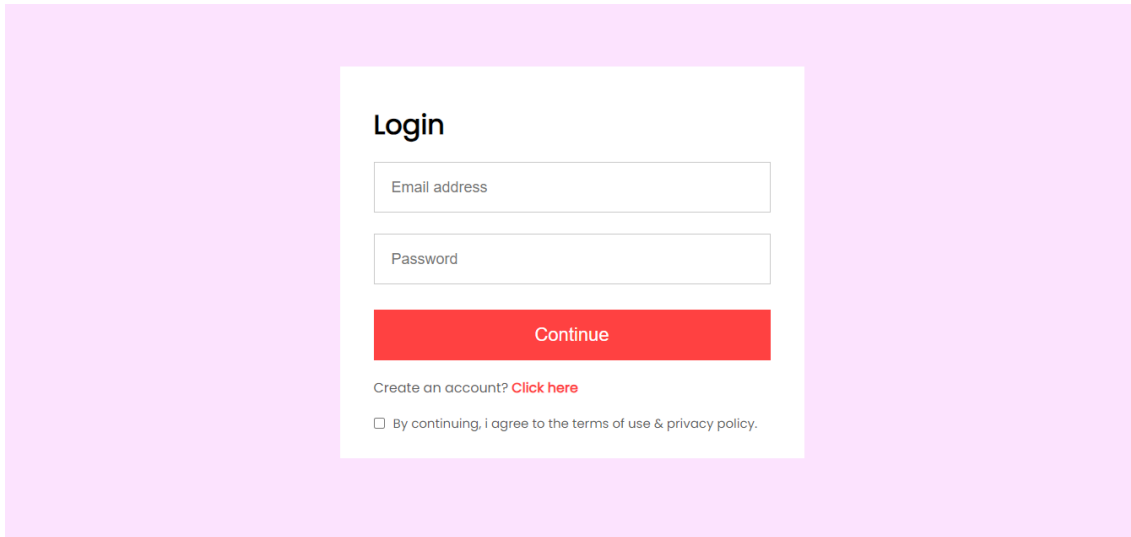


Рисунок 3.6 – Сторінка обраного товару

В меню зверху також представлена кнопка входу в аккаунт, натискаючи куди, користувач може увійти в аккаунт, який вже створював до цього, на даному вебсайті (рис. 3.7).



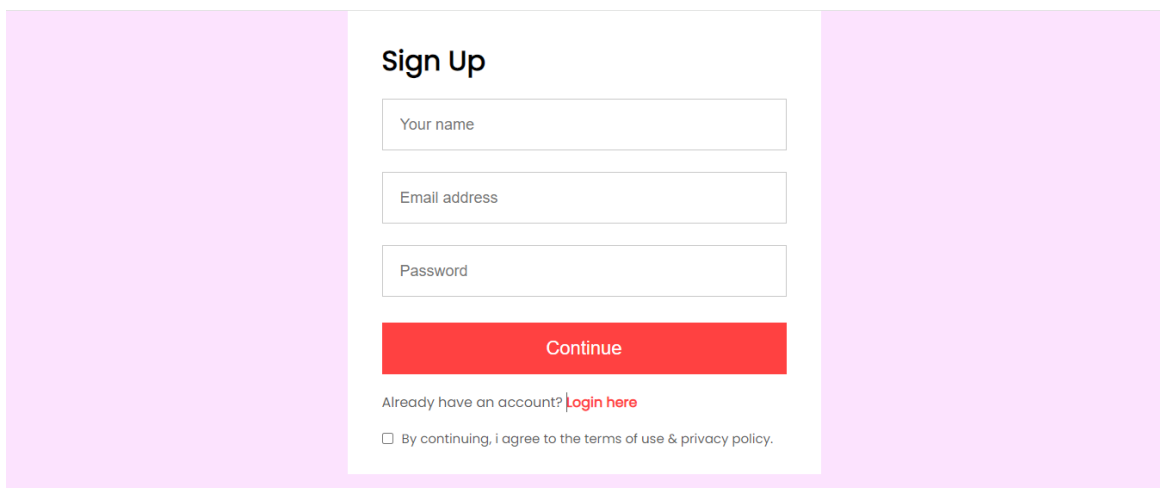
The image shows a login form on a light purple background. The form is titled "Login" and contains the following elements: a text input field for "Email address", a text input field for "Password", a red button labeled "Continue", a link "Create an account? [Click here](#)", and a checkbox with the text "By continuing, i agree to the terms of use & privacy policy."

Рисунок 3.7 – Сторінка входу в аккаунт користувача

Також в користувача є можливість створити новий аккаунт, якщо до цього він це не робив, для чого він повинен натиснути на кнопку «Click here», після чого, для нього відкриється нова сторінка реєстрації (рис. 3.8-3.9).

Create an account? [Click here](#)

Рисунок 3.8 – Кнопка для переходу на сторінку реєстрації



The image shows a sign-up form on a light purple background. The form is titled "Sign Up" and contains the following elements: a text input field for "Your name", a text input field for "Email address", a text input field for "Password", a red button labeled "Continue", a link "Already have an account? [Login here](#)", and a checkbox with the text "By continuing, i agree to the terms of use & privacy policy."



Рисунок 3.9 – Сторінка для реєстрації користувача

Після того, як користувач вводить потрібні дані для реєстрації, йому потрібно натиснути кнопку «Continue», (рис. 3.10), після чого користувача автоматично перекидає на головну сторінку сайту (рис. 3.11).

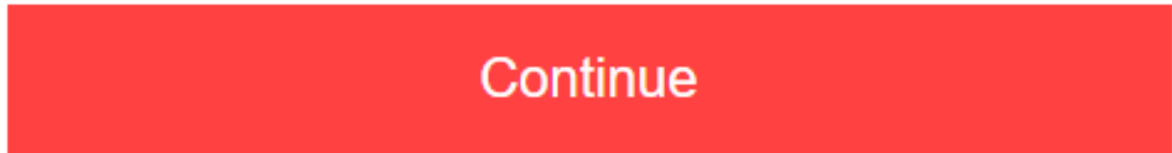


Рисунок 3.10 – Кнопка для збереження даних реєстрації

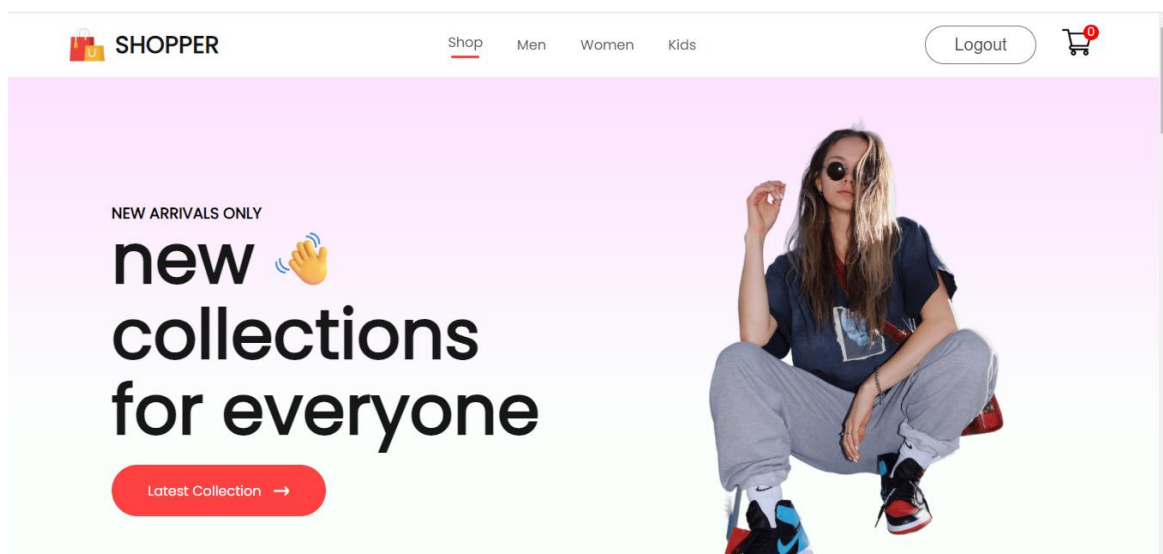


Рисунок 3.11 – Сторінка, на яку користувача перекидає після реєстрації

На головній сторінці вебсайту, користувач також має можливість, перейти на соцмережі даного магазину одягу (рис. 3.12).

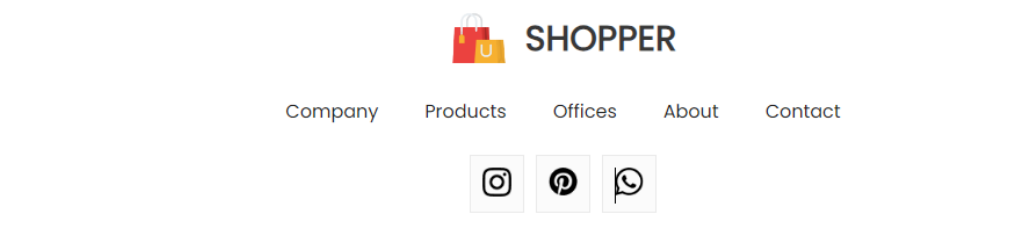


Рисунок 3.12 – Посилання на соцмережі магазину

При обиранні конкретного товару, який зацікавив клієнта, він має змогу обрати потрібний йому розмір (рис. 3.12), та додати даній товар до свого кошика, натискаючи кнопку «add to cart» (рис. 3.13).

Select Size

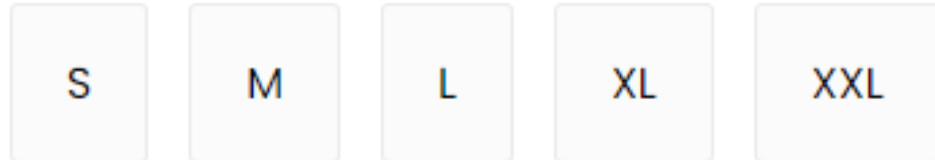


Рисунок 3.12 – Таблиця розмірів



Рисунок 3.13 – Кнопка для додавання товару для кошика

Після натискання кнопки «add to cart», користувач може переглянути, що товар був доданий до його кошика (рис. 3.14), натискаючи у правому верхньому куті на кнопку кошика (рис. 3.15).

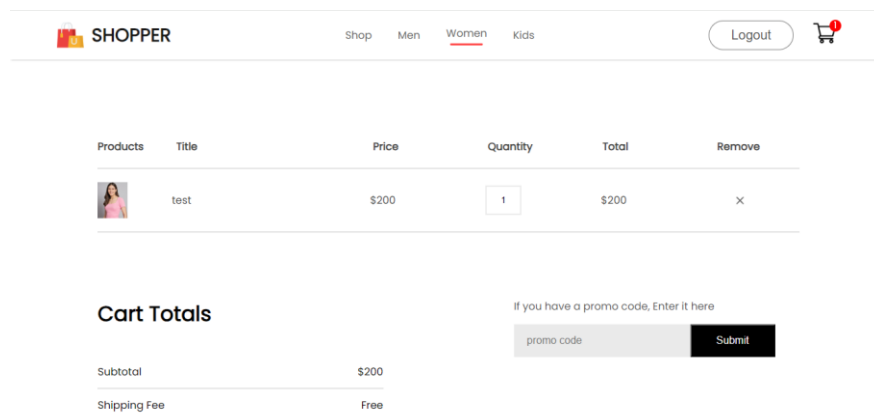


Рисунок 3.14 – Сторінка кошика користувача



Рисунок 3.15 – Кнопка переходу на корзину користувача

Після того, як користувач додав корзину до кошику, він може видалити товар (рис. 3.16) , також він має ввести промокод на знижку, якщо вона в нього є (рис. 3.17), після чого покупець може перейти до оформлення свого замовлення, натискаючи на кнопку «proceed to checkout» (рис. 3.18).

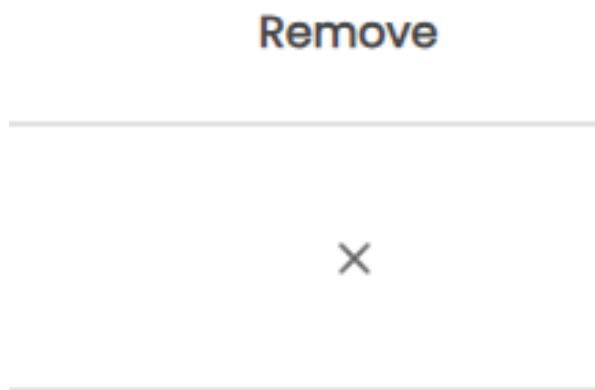


Рисунок 3.16 – Кнопка видалення товару з кошика

If you have a promo code, Enter it here

promo code	Submit
------------	--------

Рисунок 3.17 – Поле для введення промокоду

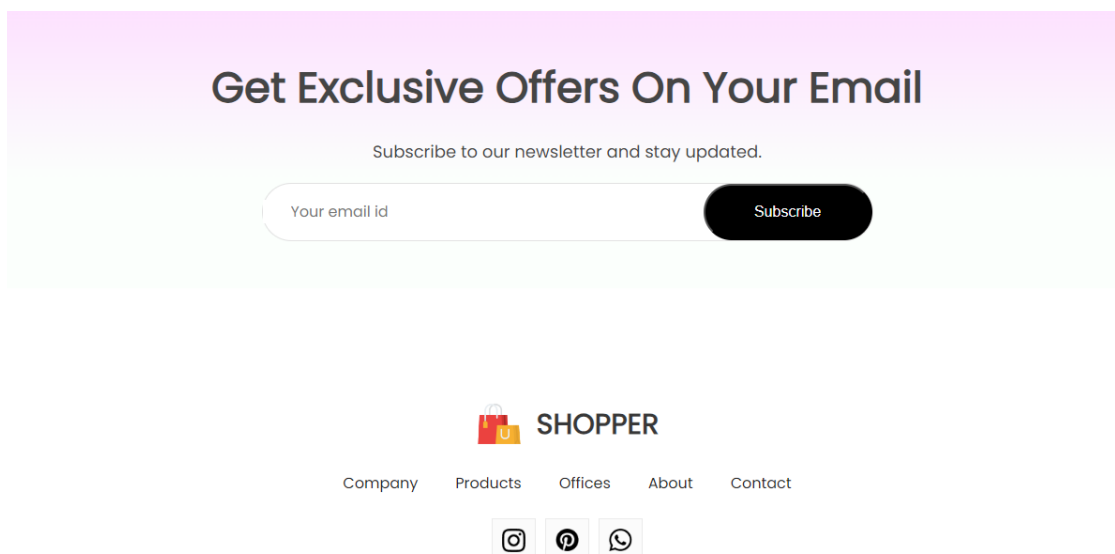
Cart Totals

Subtotal	\$200
Shipping Fee	Free
Total	\$200

[PROCEED TO CHECKOUT](#)

Рисунок 3.18 – Кнопка для оформлення замовлення

Повертаючись до головної сторінки, якщо прогортати вниз, користувач може побачити поле, де в нього є можливість ввести свою пошту, щоб отримувати листи про індивідуальні знижки та акційні пропозиції від магазину (рис. 3.19).




Get Exclusive Offers On Your Email

Subscribe to our newsletter and stay updated.

Your email id

[Subscribe](#)

 **SHOPPER**

[Company](#) [Products](#) [Offices](#) [About](#) [Contact](#)

[!\[\]\(d2f52ea40706e22ec583c717643b2243_img.jpg\)](#) [!\[\]\(5c0d77435f38eb04d6ec91190bc38832_img.jpg\)](#) [!\[\]\(a67322d9bb747733e8759e24b51ae157_img.jpg\)](#)

Рисунок 3.19 – Поле для отримання листів про знижки

Щоб повернутись швидко до головної сторінки сайту, користувач також може натиснути на логотип магазину, який знаходиться у лівому верхньому куті (рис. 3.20).

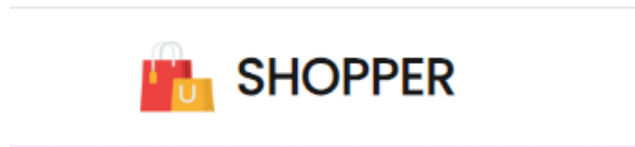


Рисунок 3.20 – Логотип-посилання для переходу на головну сторінку

Якщо після здійснення покупок, у користувача винекне бажання вийти зі свого облікового запису, він може легко це зробити, натиснувши на верхньому меню кнопку «Logout» (рис. 3.21).

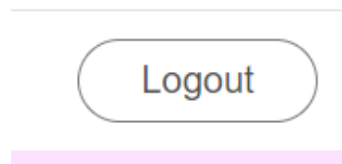


Рисунок 3.21 – Кнопка для виходу зі свого облікового запису

У випадку, якщо з'явились нові позиції одягу, адміністратор може перейти на свою особисту сторінку (рис. 3.22).

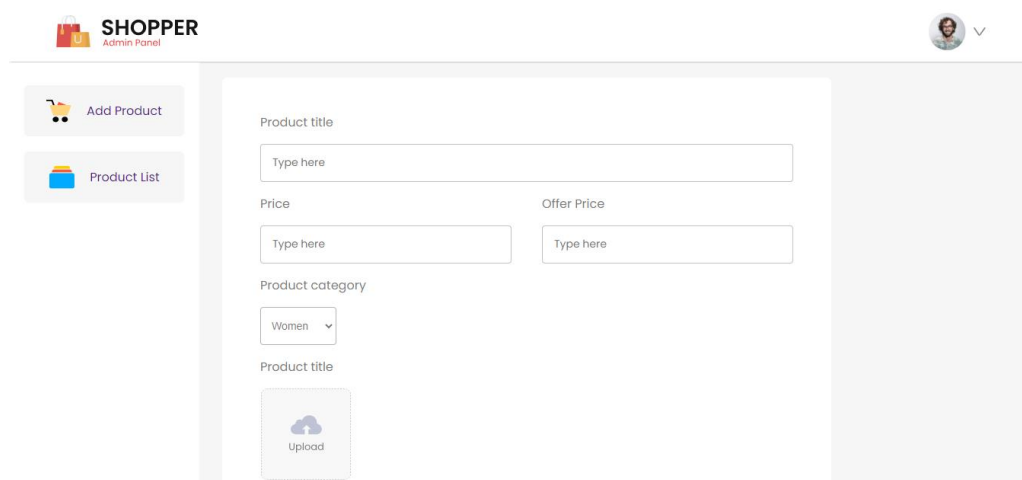


Рисунок 3.22 – Сторінка для адміністратора

Після того як адміністратор вебсайту магазину одягу, перейшов на свою сторінку, в нього є можливість додати новий товар на сайт, заповнюючи відповідні поля (рис. 3.23).

Product title

Type here

Price

Type here

Offer Price

Type here

Product category

Women ▾

Product title

Upload

Рисунок 3.23 – Поля для додавання нового товару

Після того як адміністратор заповнив назву товару, його ціну та ціну зі знижкою, якщо вона наявна, він повинен обрати категорію одягу, в яку буде додавати потрібний товар (рис. 3.24).

Product category

Women ▾

Women

Men

Kid

Рисунок 3.24 – Список категорій, в яку можна додати товар

Також адміністратор має можливість завантажити будь-яку фотографію товару до вебсайту, натискаючи на вікно «upload», після чого його перекидає до провідника його комп'ютеру, та він обирає потрібну фотографію (рис. 3.25).

Product title

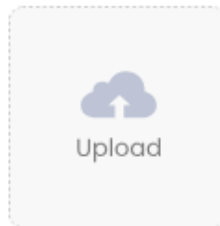


Рисунок 3.25 – Кнопка для додавання фотографії товару

Після того, як адміністратор заповнив усі відповідні поля, для того щоб додати товар до вебсайту, йому треба натиснути кнопку «add» (рис. 3.26), після чого товар буде успішно додано до потрібної категорії на вебсайті, а адміністратору висвітиться відповідне повідомлення, щоб підтвердити дану дію (рис. 3.27).

ADD

Рисунок 3.26 – Кнопка для додавання товару на вебсайт

Подтвердите действие на localhost:3003

Product Added

OK

Рисунок 3.27 – Повідомлення для підтвердження додавання товару до вебсайту

Також в адміністратора є можливість переглядати та видаляти, усі товари, що знаходяться на вебсайті, для цього йому потрібно перейти до відповідної сторінки, натискаючи кнопку «product list» (рис. 3.28), після чого адміністратора перенаправить на відповідну сторінку (рис. 3.29).



Рисунок 3.28 – Кнопка для переходу на сторінку, де є перелік усіх товарів вебсайту

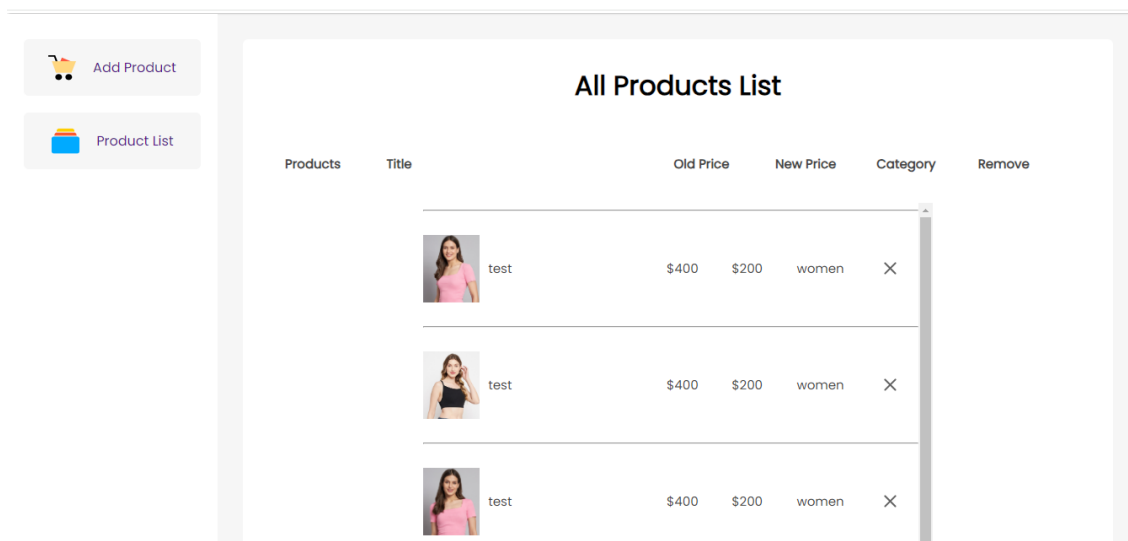


Рисунок 3.29 – Сторінка де знаходиться перелік усіх товарів вебсайту, для адміністратора

На сторінці, що представлена на рисунку 3.29, у адміністратора з'являється можливість видаляти неактуальні позиції з вебсайту, натискаючи кнопку хрестика (рис. 3.30) на обраному товарі, після чого товар одразу ж видаляється.



Рисунок 3.30 – Кнопка видалення товару з вебсайту

3.3 Перспективи подальшої роботи

Перспективи подальшої роботи над вебсайтом магазину з продажу одягу є надзвичайно важливими для забезпечення його конкурентоспроможності та задоволення потреб клієнтів. На даний момент сайт вже має базові функції, наступні етапи розвитку можуть включати:

- розширення функціоналу сайту з метою покращення взаємодії з користувачем. Це може включати впровадження персоналізованих рекомендацій на основі історії покупок та переглядів;
- зменшення кількості кроків під час оформлення замовлення може значно підвищити конверсію;
- додавання нових платіжних методів, таких як Apple Pay, Google Pay, PayPal, а також інтеграція з різними службами доставки для надання більшого вибору користувачам;
- розробка та впровадження системи лояльності для постійних клієнтів, що може включати накопичення бонусів на наступні покупки;
- впровадження сучасних методів захисту даних користувачів, таких як двофакторна автентифікація;
- додавання можливості вибору мови для зручності користувачів з різних країн;
- інтеграція сучасних технологій, таких як штучний інтелект для персоналізації пропозицій, чат-ботів для підтримки клієнтів.

Розширення та вдосконалення функціоналу вебсайту магазину з

продажу одягу є важливим кроком для залучення нових клієнтів, підвищення задоволеності існуючих користувачів та забезпечення стабільного зростання бізнесу в умовах конкурентного ринку.

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений та реалізований вебсайт для магазину з продажу одягу.

У процесі розробки було створено функціональний вебсайт для магазину з продажу одягу, який надає користувачам можливість переглядати асортимент товарів, перемикатися між категоріями, реєструватися, додавати товари до кошика та оформлювати замовлення. Розробка цього вебсайту включала використання сучасних технологій для реалізації як клієнтської, так і серверної частини проєкту.

Для розробки Front-end частини були застосовані HTML, CSS, Javascript та React, що дозволило створити адаптивний та зручний інтерфейс користувача. На Back-end використовувався Node.js, що забезпечило швидку та ефективну обробку запитів та управління даними. Для зберігання даних була обрана документно-орієнтована система MongoDB, яка забезпечує гнучкість і масштабованість бази даних.

Проєктування та реалізація бази даних були виконані з урахуванням специфіки інтернет-магазину, що дозволило забезпечити зручне управління товарами, користувачами та замовленнями.

Під час тестування системи були виявлені та виправлені помилки, що дозволило забезпечити високу якість продукту. Тестування також виявило можливості для подальшого вдосконалення вебсайту, такі як інтеграція з додатковими платіжними системами, оптимізація процесу оформлення замовлень та покращення мобільної версії вебсайту.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИ

1. Ю. Козміна, Р. Харроп, К. Шефер, К. Хо. (2019) Spring 5 для професіоналів Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools, 1120 с.
2. Богданенко Д. А. (2018) Підходи до архітектурного проектування веб додатків, 195 с.
3. Бекірова Е. А., Халілова З. Е. (2019) Основні етапи створення вебзастосунків (укр.) - Державна бюджетна освітня установа вищої освіти Республіки Крим "Кримський інженерно-педагогічний університет імені Февзі Якубова" (Сімферополь), С. 84-91. - ISSN 2658-5944.
4. Волков А. С., Волкова К. А. (2019) Огляд архітектурних компонентів сучасного вебзастосунку, 961 с.
5. Jake Spurlock (2013) Bootstrap. Responsive Web-Development, 128 с.
6. David Cochran, Ian Whitley (2014) Bootstrap Site Blueprints., 304 с.
7. David Cochran. (2012) Twitter Bootstrap Web Development., 68 с.
8. Фрімен Ерік, Фрімен Елізабет (2010) Head First HTML with CSS & XHTML., 656 с.
9. Пітер Лабберс, Брайан Олберс, Френк Салім (2011) HTML5 для професіоналів: потужні інструменти для розробки сучасних вебдодатків., 272 с.
10. Гороховатський, В. О., & Творошенко, І. С. (2022). Аналіз багатовимірних даних за описом у формі множини компонент.
11. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, Сучасні інформаційні системи, 7(1), С. 5-13.
12. Kinoshenko, D., Mashtalir, V., Yegorova, E., & Vinarsky, V. (2005). Hierarchical partitions for content image retrieval from large-scale database. In

Machine Learning and Data Mining in Pattern Recognition: 4th International Conference, MLDM 2005, Leipzig, Germany, July 9-11, 2005. Proceedings 4 (pp. 445-455). Springer Berlin Heidelberg.

13. Єльчанінов, Д. Б., Косіло, М. С., & Белова, Н. В. (2014). Технології автоматизації проектування програмних систем. Системи обробки інформації, (6), 135-140.

14. Iryna, T., & Maksym, K. (2021). Research results of functional, white box and smoke testing methods for mobile applications. Trends in science and practice of today, 5, 418.

15. Mohammadi, R. Performance, and Efficiency based Comparison of Angular and React in a case study of Single page application (SPA) (Doctoral dissertation, Dissertation 2020, Herat University, <https://scholar.google.com/scholar>).

16. Nguyen, H. (2021). Front end architecture for a single page web application.

17. Kornienko, D. V., Mishina, S. V., & Melnikov, M. O. (2021, November). The Single Page Application architecture when developing secure Web services. In Journal of Physics: Conference Series (Vol. 2091, No. 1, p. 012065). IOP Publishing.

18. Ситніков, Д. Е., & Тітова, О. В. (2013). Аналіз веб-сайтів органів місцевої влади як механізму забезпечення права доступу до публічної інформації. Вісник Харківської державної академії культури, (41), 134-142.

19. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. Харків: ХНУРЕ.

20. Тітов, С. В., & Тітова, О. В. (2015). Оцінка юзабіліті освітніх сайтів: методи і технології. Вісник Харківської державної академії культури. Серія: Соціальні комунікації, (47), 127-134.

21. Iryna, T., & Heorhii, M. (2021). Research of regression and modular testing of web applications. Editorial Board, 406.

22. Tvoroshenko, I., & Andrieieva, A. (2021). Development of web applications for remote learning of English.
23. Tvoroshenko, I. S., & Maksimenko, H. (2021). To the question of analysis of existing mechanisms of web application testing.
24. Kuzomin, O., Tolmachova, T., & Astappiev, O. (2017). Analysis of Web user activity data. *International Journal of Information Models and Analyses*, 6(2), 108-118.
25. Osmani, A. (2023). *Learning JavaScript design patterns*. " O'Reilly Media, Inc."
26. Rawat, P., & Mahajan, A. N. (2020). ReactJS: A modern web development framework. *International Journal of Innovative Science and Research Technology*, 5(11), 698-702.
27. Saks, E. (2019). *JavaScript Frameworks: Angular vs React vs Vue*.
28. Miell, I., & Sayers, A. (2019). *Docker in practice*. Simon and Schuster.
29. Potdar, A. M., Narayan, D. G., Kengond, S., & Mulla, M. M. (2020). Performance evaluation of docker container and virtual machine. *Procedia Computer Science*, 171, 1419-1428.
30. Ranjan, A., Sinha, A., & Battewad, R. (2020). *JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript*. BPB Publications.