

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)
на тему:

«Інтернет магазин спортивного одягу»

Виконав студент групи ІПЗ-20-3
Бардук Владислав Геннадійович

Керівник роботи:
Локтікова Тамара Миколаївна

Рецензент:
Чижмотря Олексій Володимирович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії програмного
забезпечення

Тетяна Вакалюк

«26» лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **Бардук Владислав Геннадійович**

Керівник роботи: **Локтікова Тамара Миколаївна**

Тема роботи: **«Інтернет магазин спортивного одягу»**,

затверджена Наказом закладу вищої освіти від **«23» лютого 2024 р., №74/с**

Вихідні дані для роботи: **об'єктом дослідження є інформаційні технології для реалізації інтернет магазину спортивного одягу, а предметом дослідження - використання сучасних технологій розробки веб-додатків та організації їхньої інформаційної взаємодії, а також створення їхнього функціоналу.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Локтікова Т.М	04.03.2024р.	04.03.2024р.
2	Локтікова Т.М	08.04.2024р.	08.04.2024р.
3	Локтікова Т.М	13.05.2024р.	13.05.2024р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з веб-додатку інтернет-магазину спортивного одягу та пояснювальної записки. Пояснювальна записка до випускної роботи містить 80 сторінок, 40 ілюстрацій та 11 таблиць. Список використаних джерел нараховує 15 найменувань та займає 2 сторінки. В роботі налічується 16 сторінок додатків.

Метою роботи є створення веб-додатку інтернет-магазину спортивного одягу з використанням сучасних технологій та методів збору, створення та оновлення даних.

Архітектура веб-додатку - систематизація інформації та навігація по ній з метою допомогти відвідувачам більш успішно знаходити потрібні їм дані, це логічний поділ сайту на блоки, розташовані в ієрархічному порядку.

Програмний продукт реалізований із використанням мов програмування PHP, JavaScript, бібліотеки Chart.js, системи керування базами даними MySQL.

Додаток має зручний інтерфейс, який дозволяє користувачам легко знаходити та замовляти обраний ними товар.

Ключові слова: ВЕБ-ДОДАТОК, ІНТЕРНЕТ-МАГАЗИН, JAVASCRIPT, MYSQL, PHP.

					ІПЗ.КР.Б-121-24-ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Інтернет магазин спортивного одягу. Пояснювальна записка	Літ.	Арк.	Аркушів
Розроб.	Бардук В.Г.							
Керівник	Локтікова Т.М.						3	80
						Житомирська політехніка, група ІПЗ-20-3		
Рецензент	Чижмотря О.В.							
Зав. каф.	Вакалюк Т.А.							

ABSTRACT

The bachelor's thesis consists of a web application of an online sportswear store and an explanatory note. The explanatory note to the final thesis contains 80 pages, 40 illustrations and 11 tables. The list of used sources contains 15 names and occupies 2 pages. The work includes 16 pages of appendices.

The method of operation is the creation of a web application of an online sportswear store using modern technologies and methods of collecting, creating and updating data.

The architecture of the web application - the systematization of information and navigation through it with an indication helps visitors to more successfully place the data they need, it is a logical division of the site into blocks located in a hierarchical order.

The software product is implemented using the programming languages PHP, JavaScript, the Chart.js library, and the MySQL database management system.

The application has a user-friendly interface that allows users to easily place and order the product of their choice.

Keywords: WEB APPLICATION, INTERNET STORE, JAVASCRIPT, MYSQL, PHP.

					ИПЗ.КР.Б-121-24-ИЗ	Апр.
						4

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ НАПРЯМІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНТЕРНЕТ МАГАЗИНУ	9
1.1 Постановка задачі	9
1.2 Аналіз аналогів програмного продукту	13
1.3 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	18
Висновки до першого розділу.	20
РОЗДІЛ 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	21
2.1 Вимоги до системи та визначення варіантів використання	21
2.2 Алгоритм роботи та стани програмної системи	24
2.3 Розробка бази даних системи.....	31
2.4 Реалізація функціоналу системи інтернет-магазину	38
Висновки до другого розділу	44
РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ	46
3.1 Користувацький інтерфейс	46
3.2 Інтерфейс для адміністратора та власника інтернет-магазину	54
3.3 Тестування програмного забезпечення.....	56
Висновки до третього розділу	60
ВИСНОВКИ	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ.....	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – прикладний програмний інтерфейс

БД – база даних

ПЗ – програмне забезпечення

СКБД – система керування базами даних

SDK – набір для розробки програмного забезпечення (software development kit)

CMS – (Content Management System) система управління контентом

ЗМІ – засоби масової інформації

CRM - це аббревіатура від Customer (клієнт) Relationship (відносини) Management (управління). Дослівно - це управління відносинами з клієнтами і вашими співробітниками. Це софт, який допомагає оптимізувати роботу відділу продажів і внутрішні процеси в компанії, що в кінцевому підсумку призводить до збільшення прибутку

MVC – схема розділення даних додатків і керуючої логіки на три окремі компоненти Model-View-Controller

CRUD – основні операції, що використовуються для управління даними (Create , Read , Update , Delete)

DApps – децентралізований додаток

URL – (Uniform Resource Locator) – єдиний вказівник на ресурс

WAMP – акронім від «Windows, Apache, MySQL і PHP» – скорочення, що описує комплект пакетів програм під відповідну ОС, позначену першою літерою

ВСТУП

Актуальність теми підкреслює розвиток інформаційних технологій у світі і їх велика роль у буденному житті людини в цілому. В наші дні майже неможливо знайти людину, яка б не мала пристрою з доступом до мережі, що набагато спрощує її життя, та зберігає її час для більш важливих справ.

В наші дні не потрібно навіть виходити з дому, щоб придбати щось в магазині, а достатньо лише декілька кліків, та невеликий проміжок часу, щоб ваші покупки вже були біля ваших дверей. Крім комфорту споживачів, інтернет-комерція - це значне полегшення роботи для власників магазинів. Це як і приваблення та заохочення нових клієнтів, так і значна економія в оренді чи будівництві нових приміщень.

Але серед великої кількості веб-ресурсів знайти дійсно зручну платформу стає все більш важче. Крім того, на сьогоднішній день дуже важко знайти спеціаліста, який би зміг підтримувати створений веб-ресурс, якщо він написаний на будь-якій мові програмування.

Саме тому стоїть необхідність розробити веб-ресурс за допомогою такого методу, який би в подальшому дозволяв легко та швидко редагувати контент на сайті без додаткових знань та професійних навичок.

Створити інтернет магазин - означає вирішити кілька завдань одночасно: вплинути на певну цільову аудиторію, залучити постачальників, почати реалізацію своїх товарів і послуг по всій країні. Структура, спрямованість, тематика сайту може бути найрізноманітнішою.

Метою даної роботи є створення інтернет-магазину спортивного одягу. Сервіс має забезпечувати функціонал для повної і безперебійної роботи інтернет-магазину та взаємодіяти з веб-додатками.

Для досягнення поставленої мети необхідне виконання таких **завдань**:

1. Постановити план реалізації проекту.

					ІПЗ.КР.Б-121-24-ІПЗ	Арк.
						7

2. Провести порівняльні характеристики аналогів програмної продукції.
3. Сформулювати інструменти та методи реалізації програмного забезпечення.
4. Визначити вимоги до системи.
5. Спроектувати інтерфейс та алгоритми роботи програми.
6. Спроектувати та реалізувати базу даних та функціонал системи.
7. Протестувати роботу додатку.

Об'єктом дослідження в роботі є інформаційні технології для реалізації інтернет магазину спортивного одягу.

Предмет дослідження - використання сучасних технологій розробки веб-додатків та організації їхньої інформаційної взаємодії, а також створення їхнього функціоналу.

РОЗДІЛ 1 АНАЛІЗ НАПРЯМІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ІНТЕРНЕТ МАГАЗИНУ

1.1 Постановка задачі

Метою кваліфікаційної роботи бакалавра є створення інтернет-магазину спортивного одягу, тобто веб-додатку для електронної комерції, що має у собі відношення до купівлі та продажу товарів або послуг через Інтернет.

У той же час, як електронний бізнес ставиться до всіх аспектів ведення онлайн-бізнесу, електронна комерція відноситься конкретно до транзакції товарів і послуг.

З кожним роком цей тип комерційних відносин займає все більшу частку на ринку, постійно збільшуючи обсяги реалізованих товарів, а також підключаючи нові області і сфери діяльності. Під електронною комерцією можна мати на увазі бронювання і реалізацію замовлень, транзакції через банківські сервіси або системи електронних грошей.

Існує чотири основних типи моделей електронної комерції, які можуть описувати практично кожну транзакцію, між споживачами і підприємствами:

- бізнес для споживача (B2C): коли підприємство продає товар або послугу окремому споживачу (наприклад, ви купуєте пару взуття в інтернет магазині);
- бізнес для бізнесу (B2B): коли бізнес продає товар або послугу іншому бізнесу (наприклад, бізнес продає програмне забезпечення як послугу для використання іншими підприємствами);
- від споживача до споживача (C2C): коли споживач продає товар або послугу іншому замовнику (наприклад, ви продаєте свої старі меблі на eBay іншому споживачеві);

- споживач до бізнесу (C2B): коли споживач продає свої власні продукти або послуги бізнесу, або організації (наприклад, впливова людина пропонує свою аудиторію онлайн в обмін на плату, або фотограф ліцензує свою фотографію для використання в бізнесі).

Сьогодні більшу частину займають ніші B2B і B2C.

Так чи інакше, у всіх випадках, дана сфера має включати в себе такі аспекти:

- торгівельний майданчик (веб-сайт, аккаунт на онлайн-ринку, інтернет-магазин);
- канали залучення трафіку (контекстна реклама, таргет-реклама, ЗМІ);
- система оброблення замовлень та робота з клієнтом (CRM, служби підтримки);
- закупівля, постачання, доставка та повернення продукції.

Інтернет-магазин повинен бути реалізований як веб-сайт, доступний в Інтернеті. Сайт має складатися із взаємопов'язаних частин, а функції кожної частини мають бути чітко розділені.

Ця програма має включати в себе наступні модулі:

1. Модуль пошуку товару. Дає користувачу можливість швидкого знаходження потрібного чи зацікавленого ним товару. Тобто пошук має здійснюватись наступним чином:
 - 1.1. За всіма категоріями.
 - 1.2. За певними категоріями.
2. Модуль фільтрації товарів. Змога підібрати користувачу продукцію за потрібними йому параметрами значно полегшить роботу та скоротить час пошуку товару. Фільтри накладаються за такими категоріями:
 - 2.1. За ціною.
 - 2.2. За популярністю.
 - 2.3. За категорією.
 - 2.4. За розміром.

2.5. За сезоном використання.

3. Модуль кошика з товарами. Кошик з товарами має бути для того, щоб користувач незалежно від стану авторизації, міг зберігати товар який йому потрібен, щоб потім замовити його.
4. Модуль оформлення замовлення. Надає користувачеві змогу замовити товар з кошика, проте тільки після погодження замовлення адміністратором.
5. Модуль адміністрування. Модуль, що надає адміністратору можливості додавання товару, зміни інформації про товар, змінювати статуси замовлення, перегляд статистики.

Сайт повинен складатись з двох частин, а саме інтерфейсу користувача та адміністративної частини.

Інтерфейс користувача - це набір фізичних та технічних елементів, з якими користувач взаємодіє з технологією чи продуктом, елементами керування, кнопками, прапорцями тощо.

Користувальницький інтерфейс сайту має бути:

- зрозумілим, легко представляти інформацію, логічно та швидко переходити до сторінок та розділів сайту;
- графічні елементи повинні бути чіткими, виразними і швидко завантажуватися;
- на сайті магазину неприпустимо використовувати анімацію і звук, які довго завантажуються і відволікають увагу покупця;
- покупець повинен без зусиль знаходити товар, що цікавить його, і мати можливість отримати про нього інформацію;
- наявність навігації на сторінках сайту. Правильна структура дає можливість користувачам без проблем користуватися сайтом, будучи впевненими, що вони завжди можуть легко повернутися на раніше відвідувані сторінки.

За допомогою адміністративної частини сайту адміністратор має можливість повністю управляти вмістом інтернет-магазину:

- додавати або видаляти товари, описи і фотографії до них, змінювати їх вартість і рівень знижок;
- редагувати розділи магазину;
- редагувати контактну інформацію інтернет-магазину;
- переглядати історію замовлень та інше.

Для того, щоб досягти мети даної роботи необхідно вирішити такі завдання:

- вивчити процеси організації інтернет торгівлі;
- провести аналіз сайтів-аналогів для розробки технічного завдання;
- вибрати та налаштувати інструменти для розробки;
- розробити веб-додаток;
- розробити інструмент для оплати;
- провести тестування програмного продукту;
- розробити інструкції для користувачів.

Отже, план створення веб-додатку виглядає так:

1. Розробка дизайну застосунку:
 - створення загальної концепції дизайну;
 - створення UI компонентів;
 - розробка дизайну сторінок.
2. Створення клієнтської частини додатку.
3. Створення серверної частини додатку.
4. Система авторизації, ролей та створення паролів.
5. Розробка адміністративної частини додатку.
6. Основний функціонал:
 - логіка авторизації;
 - створення БД для зберігання інформації;
 - організування бекенду;

- зв'язок клієнтської частини з серверною частиною додатку;
- автоматизація процесів.

7. Тестування додатку.

Результатом виконання плану дій є створення платформи, за допомогою якої користувачі можуть створювати замовлення, які після підтвердження адміністраторами будуть відображатись в загальному списку, що дозволить будь-якому користувачу повноцінно отримувати замовлену продукцію.

1.2 Аналіз аналогів програмного продукту

Розробка такого продукту як інтернет-магазин, потребує глибокого аналізу конкурентів на ринку. Не обов'язково мати відомий світовий бренд, для того, щоб знайти свого покупця, для цього достатньо зробити зручний інтерфейс, мати чим більшу цільову аудиторію, та розуміти який товар найбільше матиме попит. Серед аналогів були проаналізовані такі популярні інтернет-магазини українського ринку:

- <https://ua.puma.com>;
- <https://megasport.ua>;
- <https://sportcity.ua>.

PUMA – інтернет-магазин світового бренду в Україні, який дозволяє місцевим жителям купити оригінал спортивного одягу від реального виробника. На офіційному інтернет-порталі запропоновані всі колекції спорттоварів, які сьогодні доступні в Україні.

Головна сторінка інтернет-магазину «Puma» представлена на рис. 1.1.

Перевагами сайту є:

- сучасний дизайн та зрозумілий інтерфейс;
- навігаційна система;
- наявність чат-боту;

- сторінка товару здається мені зручною у використанні за змістом, але чітко виражає всю потрібну інформацію для замовлення і також має функцію обрахування загальної ціни замовлення (з урахуванням адреси);
- наявність багатьох мов інтерфейсу.

Недоліком є занадто зжата сторінка користувача, тобто забагато інформації різного виду відображається всього на одній сторінці.

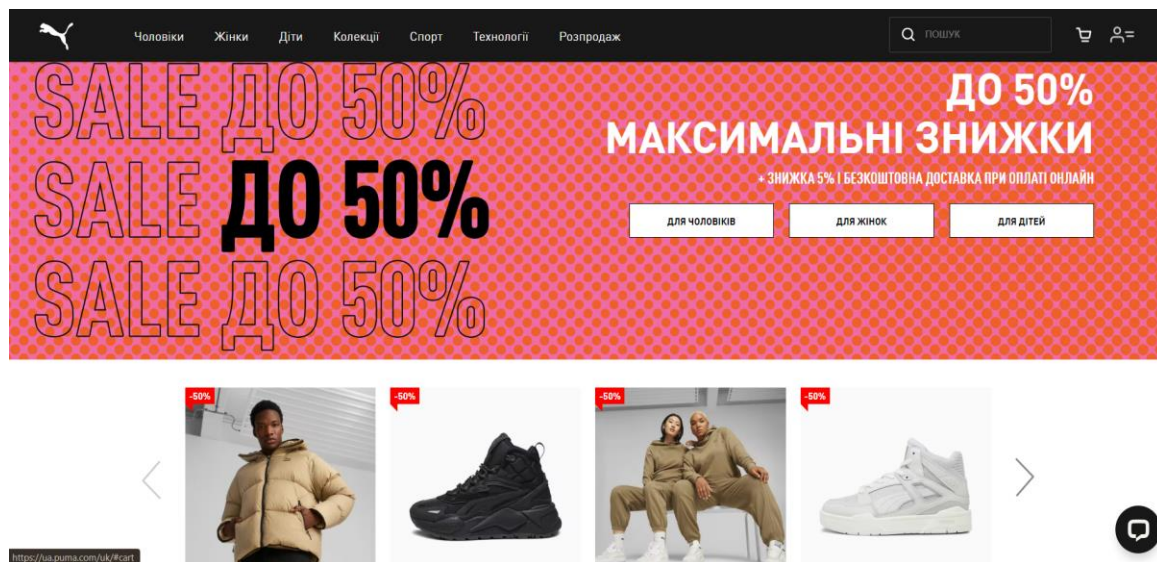


Рисунок 1.1 – Головна сторінка інтернет-магазину «Puma»

MEGASPORT - це найбільша національна мережа мультибрендових магазинів спортивного одягу, взуття та аксесуарів провідних світових виробників.

Головна сторінка інтернет-магазину «Megasport» представлена на рис. 1.2.

На відміну від PUMA, сайт MEGASPORT має:

- більш зручніший інтерфейс за рахунок компактності та чіткості;
- відсутній чат-бот, але сайт має свого телеграм-бота;
- вхід на сайт використовується не через логін та пароль, а підлягає способу аутентифікації через номер телефону, що не дуже зручно, адже це може зайняти більше часу;
- усього дві мови інтерфейсу.

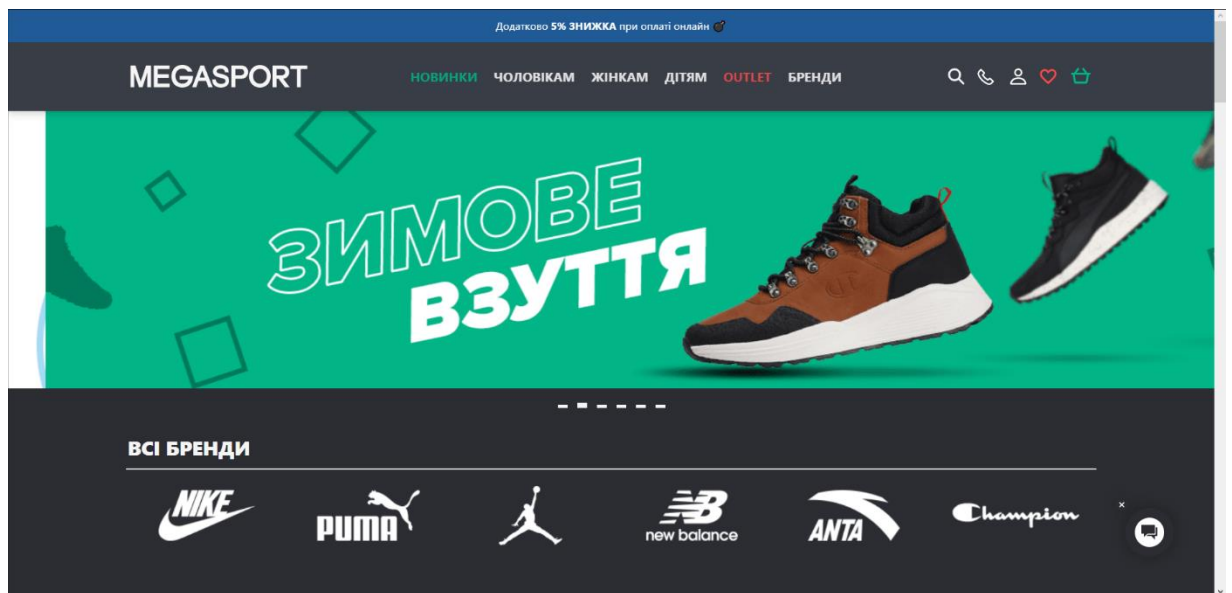


Рисунок 1.2 – Головна сторінка інтернет-магазину «Megasport»
SPORTCITY ще одна мережа мультибрендових магазинів спортивних товарів.

Головна сторінка інтернет-магазину «Sportcity» представлена на рис. 1.3.
 Сайт має трохи застарілий дизайн у порівнянні з попередніми аналогами.
 Недоліками також є:

- не дуже практична система навігації;
- відсутній розширений опис багатьох товарів, що не дає змоги повністю оцінити продукт;
- не зручний інтерфейс.

Перевагами сайту є лише:

- великий асортимент товарів;
- багато компактних елементів що спрощують візуальну взаємодію з додатком.

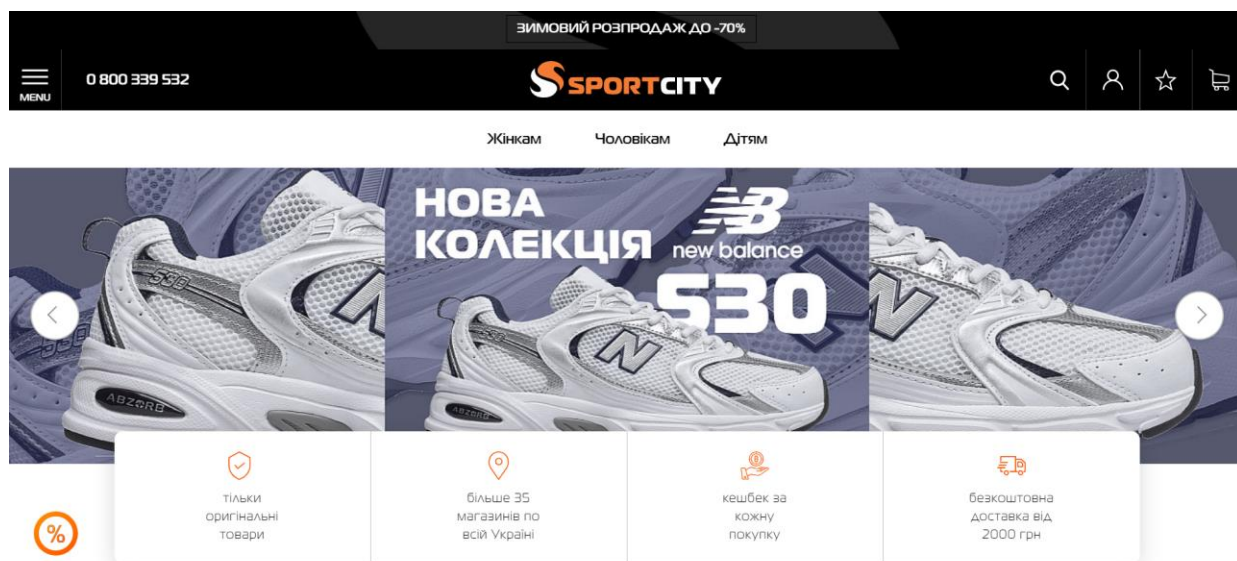


Рисунок 1.3 – Головна сторінка інтернет-магазину «Sportcity»

В даний час створено велику кількість додатків такого роду, проте вони мають деякі недоліки.

Після аналізу інтернет-магазинів можна виділити кілька недоліків, таких як:

- неактуальна інформація. Щодня змінюється асортимент і наявність товарів на складі, а у веб-додатку не завжди оновлюється інформація;
- простота використання. Потенційний клієнт не може самостійно розібратися, як оформити замовлення, куди натиснути, щоб переглянути умови оплати та доставки;
- несучасний дизайн;
- повільна обробка замовлень.

Виконаємо загальний аналіз, в табл. 1.1 більш детально представлено аналіз обраних інтернет-магазинів.

Таблиця 1.1

Порівняльна характеристика аналогів програмного продукту

Критерії	PUMA	MEGASPORT	SPORTCITY
Сучасний дизайн	+	+	-

Зручність використання	+	+	+
Практична навігація	+	+	-
Неактуальна інформація	+	+	-
Масштабованість зображень	-	+	-
Зручність авторизації	+	-	+
Статус замовлення	-	+	+
Зворотній зв'язок	+	+	+

Отже, проаналізувавши інтернет-магазини зі схожою тематикою, можна виділити основні відмінності: деякі з них мають зручні для користування інтерфейси, інші дозволяють отримати замовлення лише після погодження з адміністратором, великий асортимент, невеликий збір та конфіденційність інформації про користувачів.

Враховуючи цю інформацію було прийняте рішення створити веб-додаток з зручним, компактним та зрозумілим користувачу інтерфейсом, цільовим пошуком продукції, без обов'язкової оплати товару для замовлення.

Таким чином, даний додаток має бути дуже простим у використанні, з інтуїтивним дизайном; бути децентралізованим, тобто усі платежі будуть здійснюватися за допомогою банківських карток, та у випадку відміни замовлення, кошти мають повертатись користувачу назад.

1.3 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Зараз перед веб-розробником стоїть безліч різноманітних завдань – від створення інтерактивних розважальних сайтів до серйозних бізнес-проектів, які вимагають підвищеної надійності та безпеки від несанкціонованого доступу. Щоб їх реалізувати, потрібні правильні інструменти – мови програмування фреймворки або системи керування контентом, які стають все більше актуальними.

Для створення інтернет-магазину ми використовуємо мову PHP, середовище програмування PhpStorm та локальний веб-сервер Open-Server.

PHP [5,11] - мова програмування загального призначення, що використовується для розробки Web-орієнтованого програмного забезпечення. На сьогодні PHP є лідером серед мов програмування, що використовуються для створення динамічних Web-сайтів. PHP використовується для розробки проектів різних масштабів — від найпростіших сайтів до великих багатофункціональних Web-сервісів.

Ця мова програмування має ряд значних переваг, які роблять її основним інструментом для розробки серверної частини додатків та сайтів:

1. PHP підтримується за замовчуванням майже будь-яким сучасним хостинг-провайдером, що є запорукою доступної вартості серверів для додатку або сайту, написаного цією мовою.
2. З використанням PHP створено велику кількість систем управління контентом (CMS) для сайтів, таких як WordPress, Joomla, Drupal — системи загального призначення, OpenCart, Magento — системи для управління Інтернет-магазинами. Це лише найбільш розповсюджені та безкоштовні CMS, їх реальна кількість — безкоштовних та платних — дуже велика.

3. Створено значну кількість високоефективних інструментів та фреймворків, що засновуються на PHP. Вони дозволяють розробникам створювати додатки швидше, при цьому підтримуючи код програми в чистоті. До таких інструментів належать Symfony, Laravel, Yii2, CodeIgniter 4 — це лише найбільш розповсюджені.
4. Висока швидкість роботи. Останні версії PHP вражають швидкістю відпрацювання скриптів у порівнянні з іншими мовами програмування.

Проте, як і будь-який інструмент, PHP як мова програмування має ряд недоліків. Справа в тому, що мова має дуже низький поріг входження. Це обумовлено тим, що спочатку він створювався як інструмент для розробки домашніх сторінок — такий, який кожен зміг би освоїти. На перший погляд, в цьому немає нічого поганого. Але насправді ця особливість дозволяє входити в PHP розробку тим, хто не має достатнього досвіду в програмуванні. А створення програмного забезпечення недосвідченим розробником може призвести до проблем з безпекою та технічною підтримкою проекту в подальшому.

Таким чином, PHP — це чудовий інструмент для розробки серверного програмного забезпечення, який може забезпечити високу швидкодію та безпеку, а також простоту подальшої технічної підтримки, але лише в тому випадку, коли цей інструмент знаходиться в руках справжнього майстра.

PhpStorm - комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains [3] на основі платформи IntelliJ IDEA. PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript [6,7,13] з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

OpenServer - це портативне програмне середовище, створене спеціально для веб-розробників з урахуванням їх рекомендацій та побажань.

Даний програмний комплекс включає ретельно підібраний набір серверного програмного забезпечення, а так само неймовірно зручну і продуману керуючу утиліту, яка володіє потужними можливостями з адміністрування і налаштування всіх доступних компонентів.

MySQL [9,15] – вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця СКБД з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних вебсторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

Висновки до першого розділу

У цьому розділі було проведено аналіз предметної області, за допомогою якого були визначені основні вимоги до веб-додатку, основною задачею яких буде створення зручного та зрозумілого користувацького інтерфейсу з легким та безперешкодним замовленням товарів.

В процесі дослідження аналогічних продуктів були проаналізовані та порівняні різні платформи з різними принципами роботи, основним функціоналом, яких є зазначені вище аспекти.

Крім цього, були сформовані основні вимоги для власної платформи. На основі відповідних вимог та критеріїв був сформований набір основних технологій і засобів реалізації для розробки веб-додатку: мова програмування PHP, середовище розробки PhpStorm, OpenServer та СКБД MySQL.

РОЗДІЛ 2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вимоги до системи та визначення варіантів використання

Для забезпечення неперервного виконання необхідних функцій і відповідності встановленим стандартам та специфікаціям, наведено високорівневі вимоги, які повинні задовольняти веб-орієнтований магазин спортивного одягу. Ці вимоги допомагають забезпечити безперебійну роботу системи та виконання всіх необхідних стандартів і специфікацій.

Бізнес вимоги

1. Основні цілі: проект створюється з метою спростити взаємодію між власниками магазинів та автоматизації певних процесів.
2. Представлення проекту: проект буде реалізовано у вигляді сайту, що містить структуровану інформацію по даній тематичній області.

Вимоги користувачів

1. Можливість створення облікового запису, редагування особистих даних як адміністратора, так і відвідувачів магазину;
2. Можливість зручного внесення інформації;
3. Можливість вподобання (збереження) товару;
4. Можливість додавання товарів до кошику;
5. Можливість зміни інформації або її видалення;
6. Можливість переглядати список як товарів, відгуків, так і заходів;
7. Можливість пошуку та фільтрації товару;
8. Можливість робити замовлення товару;
9. Можливість оплати онлайн та самовивозу;
10. Можливість перегляду історії та статусу замовлень;
11. Можливість створення відгуків та коментарів.

Вимоги персоналу веб-додатку (власники / адміністратори магазину):

					ІПЗ.КР.Б-121-24-ПЗ	Арк.
						21

1. Можливість перегляду списку, створення, редагування та видалення усіх даних, які стосуються інформації про товар та події.
2. Можливість зв'язку з користувачами, коли вони замовляють товар, та останні вже будуть відправлятися у чат-бот як замовлення.
3. Можливість перегляду інформації про інших користувачів.
4. Можливість надавати знижки та купони користувачам для використання.
5. Можливість видалення коментарів.
6. Можливість перегляду статистики.
7. Можливість підтвердження та скасування замовлень.
8. Можливість блокування товару.

Характеристика об'єкта комп'ютеризації:

Користувач додатку буде мати можливість переглядати інформацію про активні заходи, мати можливість пошуку та фільтрації товару, перегляду вподобаних товарів, а також легко працювати з додатком через простоту та зручність.

Функціональні вимоги:

1. Авторизація у системі: в системі повинна бути представлена можливість реєстрації користувача та присвоєння йому відповідної ролі (користувач, власник / адміністратор готелю).
2. Можливість збереження інформації: система повинна зберігати інформацію і надавати можливість користувачу керувати нею.
3. Можливість перегляду попередньо оглянутих товарів.
4. Можливість зворотнього зв'язку.

Нефункціональні вимоги:

1. Сприйняття
 - час, необхідний для навчання звичайних користувачів – 2-3 години, а для навчання просунутих користувачів – 1 година; – час відгуку для

типових задач – не більше 15 секунд, для складних завдань – не більше 30 секунд.

- інтерфейс має бути зрозумілим та з відповідними підказками.

2. Надійність

- доступність – час, що витрачається на обслуговування системи не повинно перевищувати 6% від загального часу роботи.
- середній час безвідмовної роботи – 30 робочих днів.
- максимальна норма помилок та дефектів в роботі системи – 1 помилка на десять тисяч рядків коду.

3. Продуктивність

- система повинна підтримувати мінімум 50 одночасно працюючих користувачів, пов'язаних із загальною базою даних.

4. Можливість експлуатації

- масштабування
- система повинна мати можливість збільшувати потужності (продуктивність), зі збільшенням користувачів таким чином, щоб це не вплинуло негативно на її роботу.

Аналіз вимог користувачів дав підстави для визначення варіантів використання додатку інтернет магазину спортивного одягу. На рис. 2.1 наведено діаграму використання програмної системи.

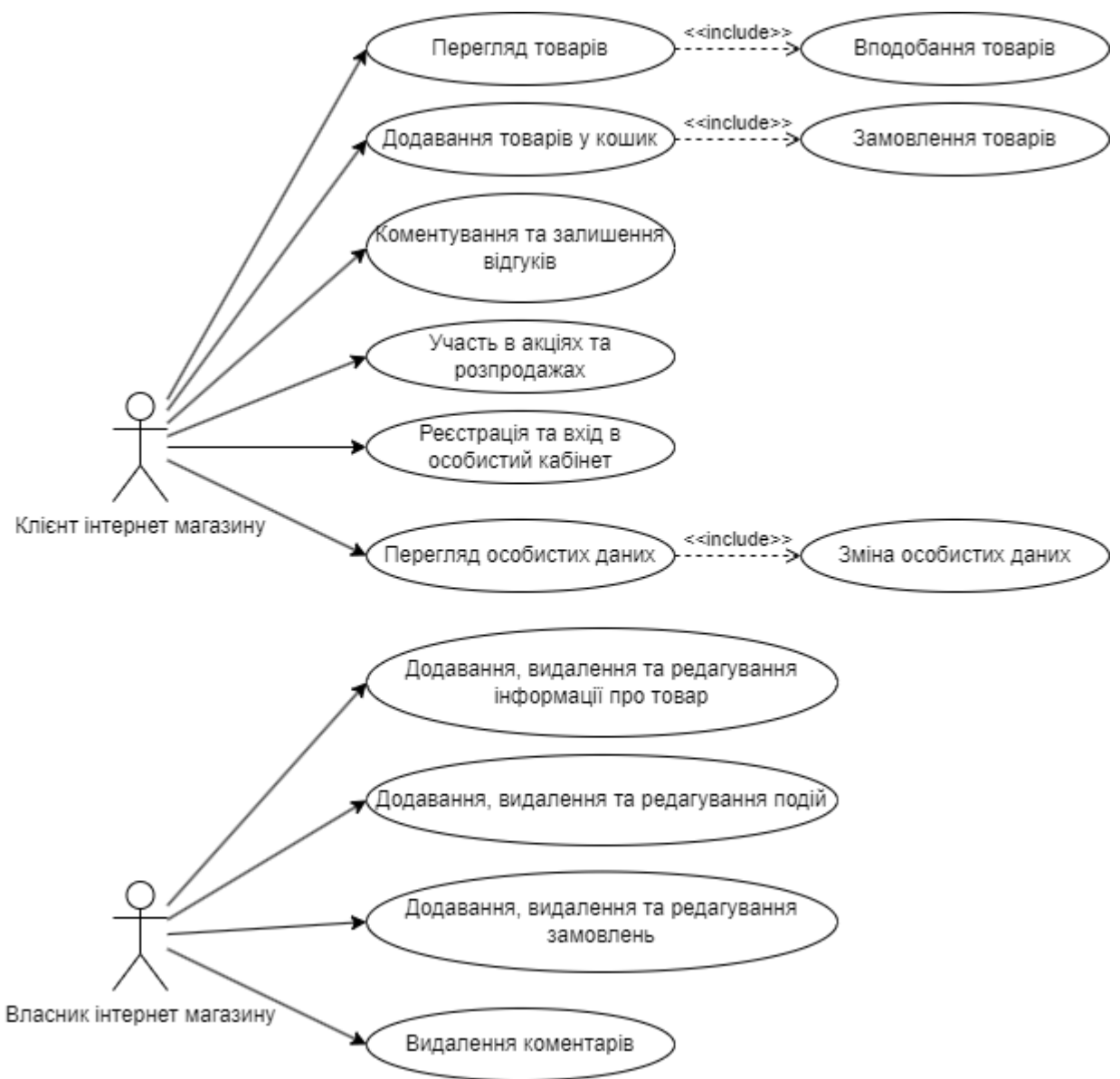


Рисунок 2.1 - Діаграма використання програмної системи.

2.2 Алгоритм роботи та стани програмної системи

Після отриманої нами теоретичної інформації про потрібну нам галузь, ми можемо перейти до етапу проектування програмного забезпечення. Складемо алгоритм запуску роботи програми, що надасть можливість оцінити користувачем алгоритм роботи з цією програмою. Опис алгоритму: Після запуску сайту у нас з'являється головне меню з вибором подальших дій. У нашому випадку це:

- перегляд товару;

- перегляд каталогу;
- перегляд подій;
- вхід/реєстрація на сайт;
- перегляд кошику.

Загальний вигляд алгоритму для подальших дій зображений на діаграмі активності для авторизації\реєстрації на рис. 2.2.

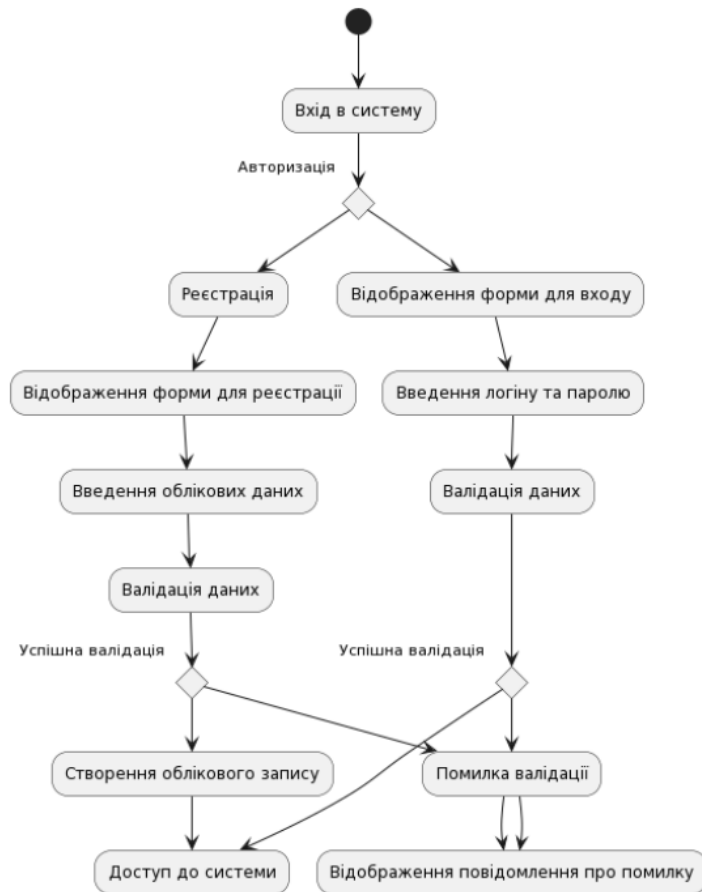


Рисунок 2.2 – Діаграма активності для авторизації\реєстрації

Діаграма авторизації відображає функцію обробки для входу або створення облікового запису додатку. Це необхідно користувачеві для збереження персональних даних, списку обраних товарів, перегляду історії замовлень і написання відгуків товарам. Замовлення товару може здійснюватись без авторизації, і його можна буде додати до історії замовлень після входу в обліковий запис.

Алгоритм замовлення товарів виглядає наступним чином:

1. На сторінці каталогу обираємо потрібний нам товар;
2. Додаємо товар у кошик;
3. Обираємо ще товари, або переходимо одразу до кошика;
4. Обираємо спосіб доставки: якщо доставка то вводимо адресу доставки та контактну інформацію, якщо самовивіз то тільки контактна інформація;
5. Обираємо спосіб оплати;
6. Підтверджуємо замовлення;
7. Переглядаємо замовлення в історії замовлень.

Діаграма активності для замовлення товару зображена на рис. 2.3 та діаграма діяльності для замовлення товару між усіма акторами зображена на рис. 2.4.



Рисунок 2.3 – Діаграма активності для замовлення товару

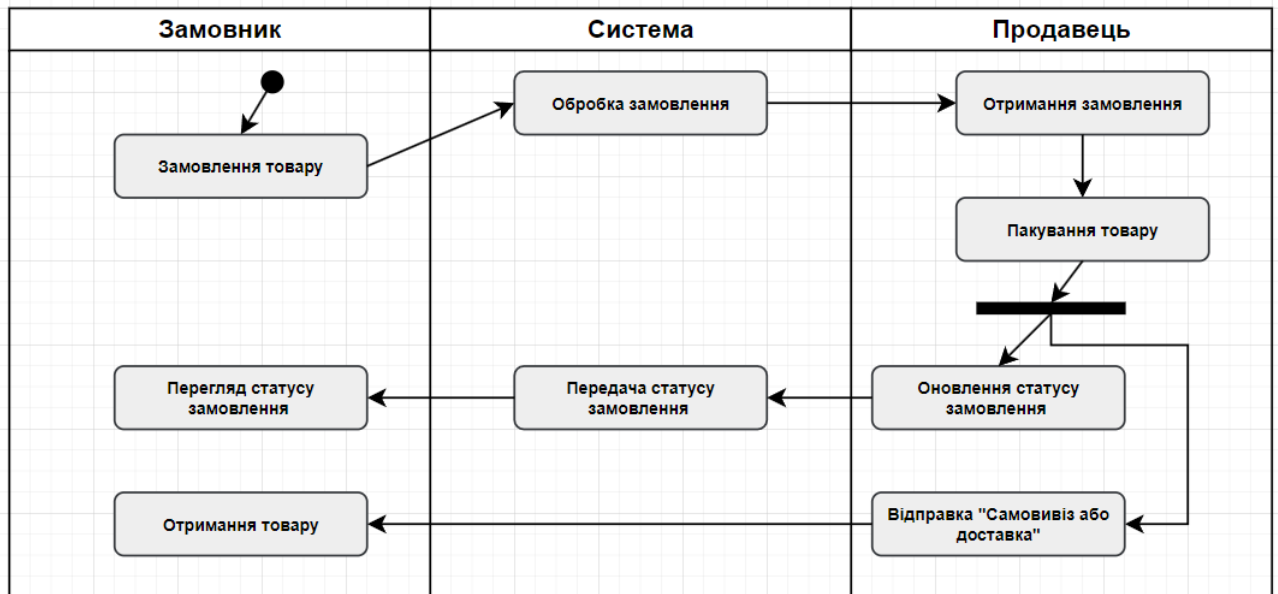


Рисунок 2.4 – Діграма діяльності для замовлення товару

1. Під час створення нового замовлення стан замовлення переходить в **стан очікування**.
2. В стані очікування система чекає наступної відповіді про подальші дії: або користувач оплатив замовлення, або відмінив. Отже якщо користувач відмінив замовлення, система виходить зі стану очікування та скасовує створення нового замовлення, якщо ні – створюється нове замовлення із статусом «**замовлення оплачено**».
3. Далі при підготовленні до передачі та відправленні замовлення статус оновлюється до «**підготовка до відправлення**».
4. Після до вставлення замовлення воно переходить в стан «**замовлення доставлено**».
5. Підтвердження отримання замовлення означає перехід статусу замовлення у «**замовлення виконане**».

Діаграма станів для замовлення товару зображена на рис. 2.5.

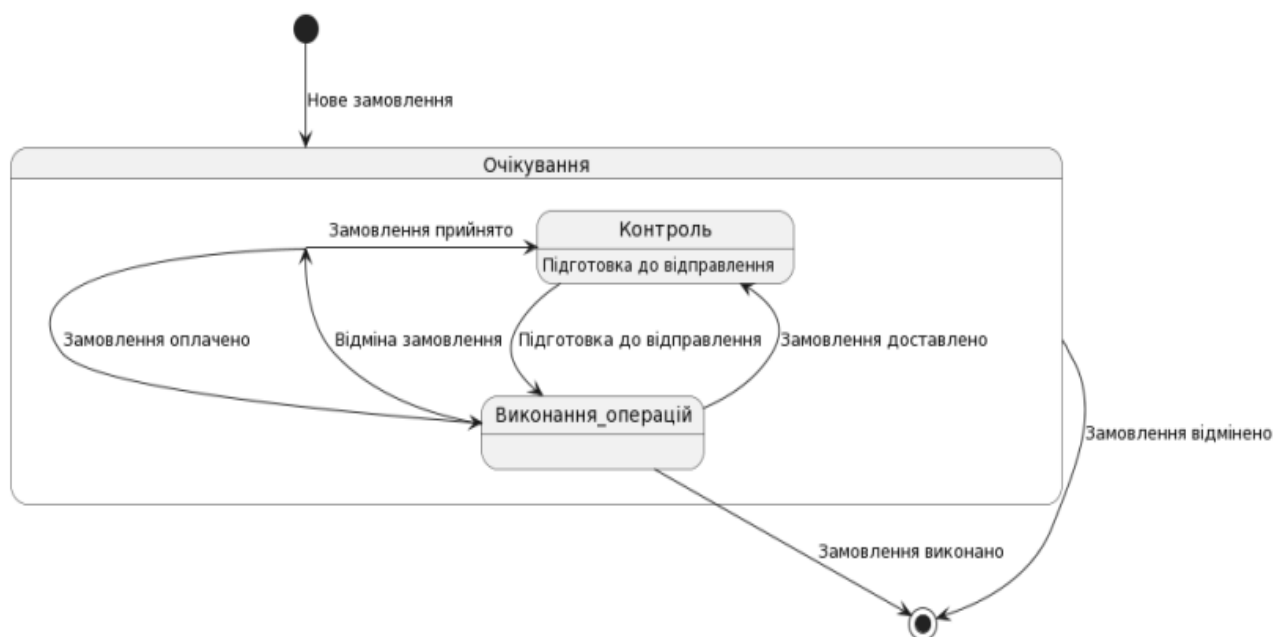


Рисунок 2.5 – Діграма станів для замовлення товару

Загальний алгоритм роботи системи зображений на діаграмі послідовності дій системи на рис. 2.6.

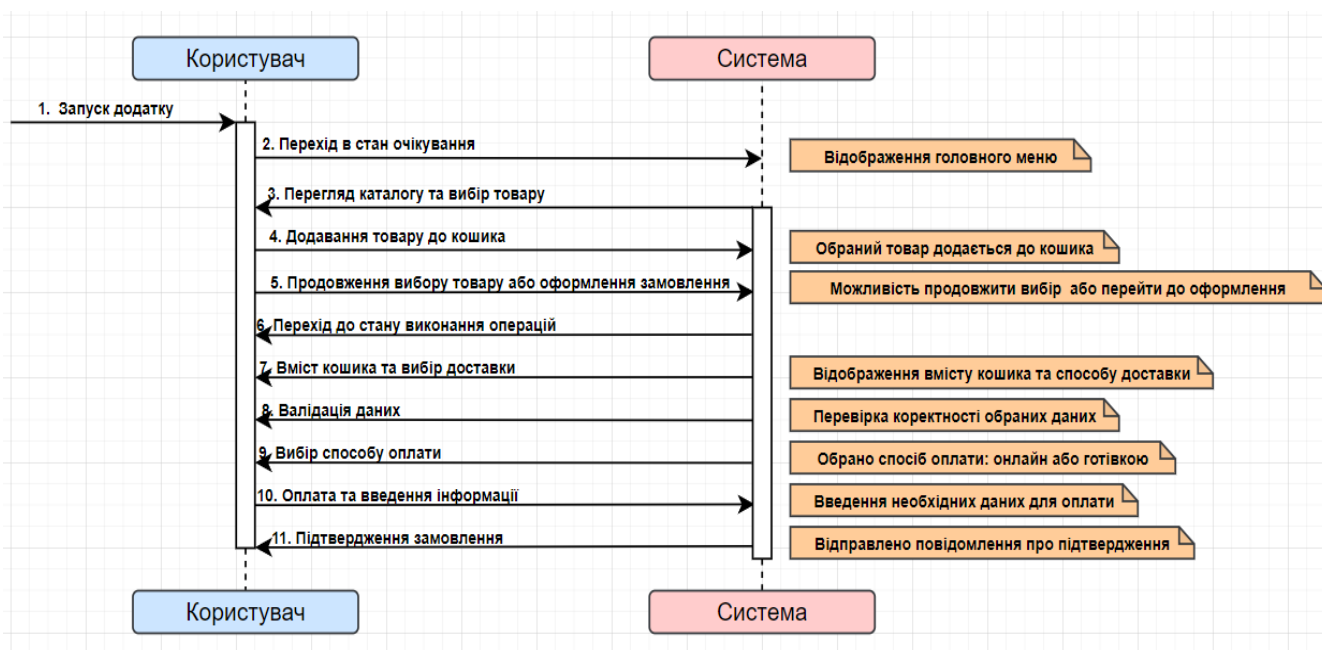


Рисунок 2.6 – Діграма послідовності дій системи

Сценарій входу в обліковий запис:



Рисунок 2.7 - Діаграма взаємодії акторів при вході в обліковий запис користувача

Дана діаграма (рис.2.7) кооперації показує взаємодію між акторами сценарію входу користувача в обліковий запис. Клієнтська частина отримує дані від користувача, відправляє запит з даними до серверної частини, та в свою чергу перевіряє дані у базі даних і звіряється з базою даних блокчейну, після чого отримує відповідь з блокчейну та передає відповідь до клієнтської частини.

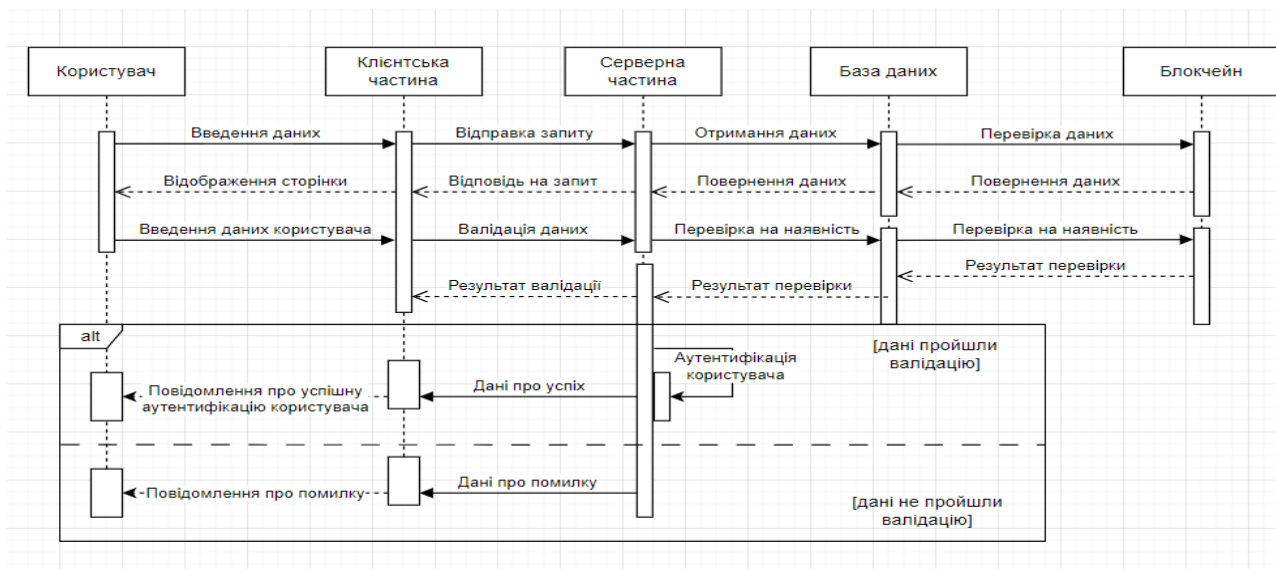


Рисунок 2.8 - Діаграма послідовності дій в системі при аутентифікації

Дана діаграма послідовності (рис. 2.8) показує послідовність дій в сценарії аутентифікації акаунта. Користувач вводить дані для входу. Клієнтська частина додатку отримує ці дані та передає їх серверній частині. Серверна частина виконує обробку запита, перевіряє передані дані в БД, якщо дані не були знайдені, перевіряє їх на блокчейні.

В разі успішної перевірки, сервера частина передає отриманий статус валідації до клієнтської частини, а вона в свою чергу повертає відповідь користувачу.

Сценарій створення нового замовлення:



Рисунок 2.9 - Діаграма взаємодії акторів при замовленні товару

Дана діаграма кооперації (рис. 2.9) показує взаємодію між акторами сценарію створення нового замовлення. В даній діаграмі відсутня клієнтська частина, адже передбачається що користувач вже взаємодіє з нею. Дані для запиту передаються до замовлення, він в свою чергу генерує запит до серверу, сервер оброблює запит та передає його до блокчейну. Блокчейн генерує номер та відправляє результат до серверної частини, далі на замовлення відправляється відповідь на запит.

Діаграма послідовності показує послідовність дій в сценарії замовлення товару (рис. 2.10). Користувач обирає товари для покупки. Серверна частина виконує обробку запиту, перевіряє передані дані в Базі даних, якщо дані не були знайдені, перевіряє їх на блокчейні. Далі при успішній обробці запиту, серверна частина генерує унікальні номери, створює нові замовлення та відправляє їх у стан очікування підтвердження.

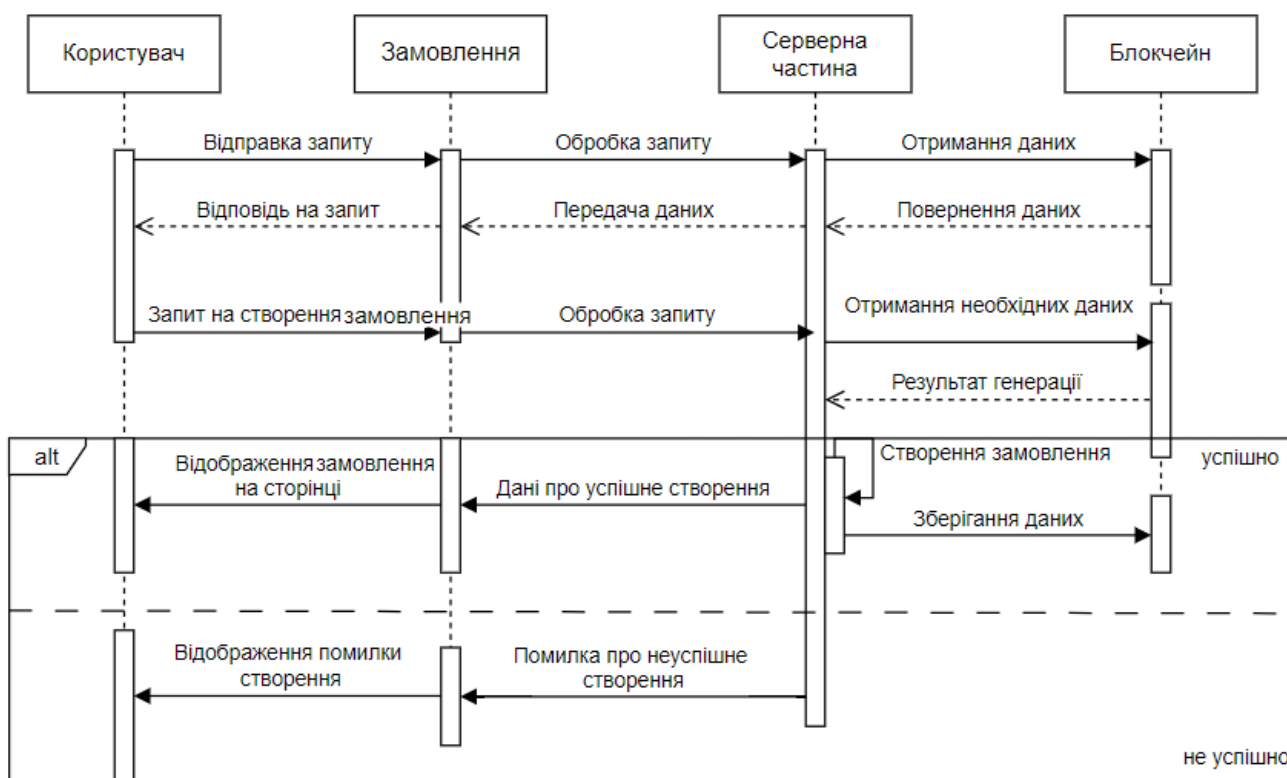


Рисунок 2.10 - Діаграма послідовності дій в системі при замовленні товару

2.3 Розробка бази даних системи

База даних (БД) - це сукупність пов'язаних даних, які зберігаються в системі за допомогою таблиць. Програмне забезпечення, яке дозволяє створювати, редагувати, ділитися та використовувати базу даних, називають системою керування базою даних (СКБД). У базі даних досить вдало створений набір таблиць. Будь-яка таблиця - це масив однорідних частин, який називаються записами. Запис

також може містити один або кілька іменовані поля. Імена та номери полів вказуються під час створення таблиць. Кожне поле також має певний тип даних.

Для зберігання різних даних на сайті використовується MySQL.

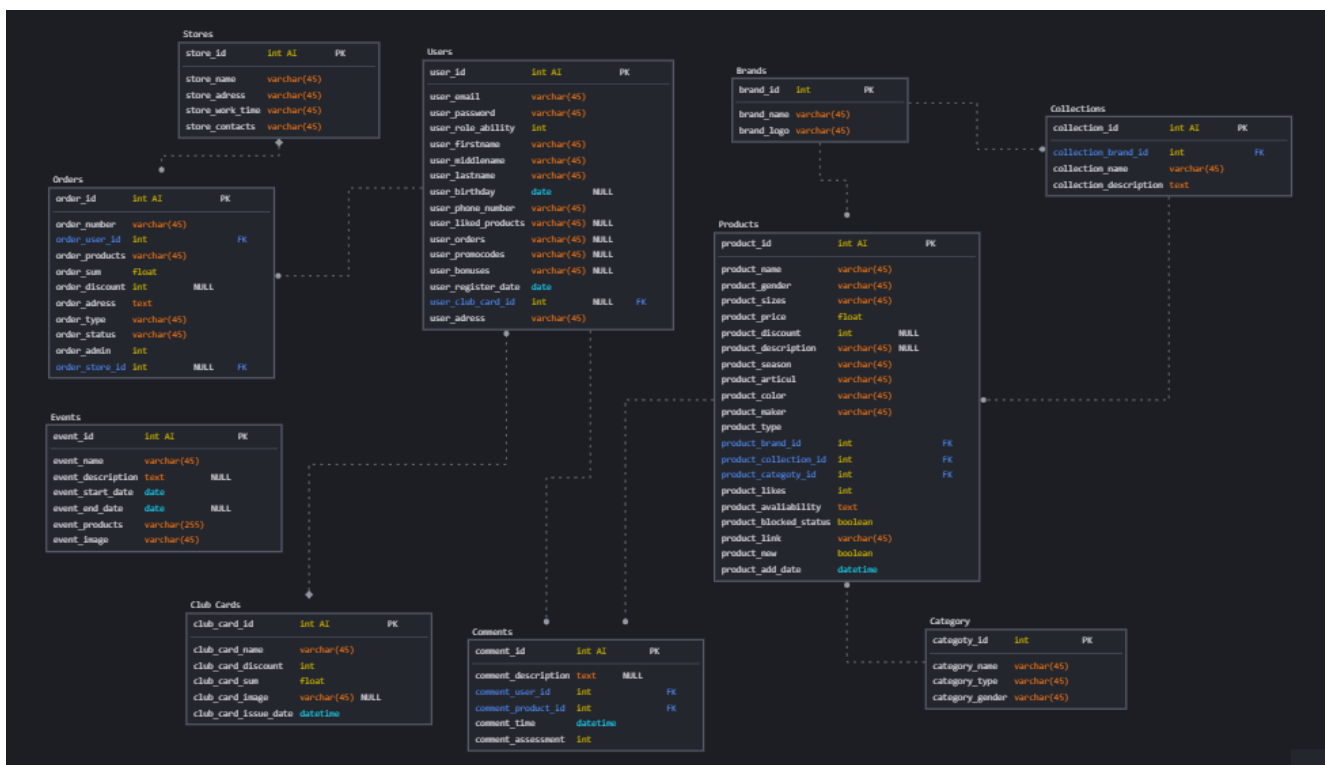


Рисунок 2.11 – Схема бази даних

На рис. 2.11 зображено структуру основних таблиць бази даних системи. БД складається з 10 основних таблиць:

- Users - таблиця користувачів системи;
- Club Cards – таблиця клубних карток з бонусами користувачів;
- Products – таблиця товарів;
- Category – таблиця категорій товару;
- Brands – таблиця брендів товару;
- Collections – таблиця колекцій товарів;
- Comments – таблиця коментарів товарів;
- Stores – таблиця фізичних магазинів;
- Events – таблиця подій, що відбуваються у магазині (розпродажі, акції, аукціони і т.д.);

- Orders – таблиця замовлень.

У табл. 2.1 – 2.10 описані таблиці бази даних, які зображені на схемі з якими найбільше працюватимуть користувачі та їх поля.

Таблиця 2.1

Опис таблиці Users

Поле	Опис
user_id	ідентифікатор користувача
user_email	електронна пошта користувача
user_password_	пароль для входу
user_role_ability	роль користувача
user_firstname	прізвище
user_middlename	ім'я
user_lastname	по батькові
user_birthday	дата народження
user_phone_number	номер телефону
user_liked_products	список вподобаних товарів
user_orders	список замовлень
user_promocodes	промокоди
user_bonuses	бонуси
user_club_card	клубна картка для знижок
user_city_id	місто проживання
user_register_date	дата реєстрації

Таблиця 2.2

Опис таблиці Products

Поле	Опис
product_id	ідентифікатор товару

product_name	назва товару
product_gender	стать
product_sizes	наявні розміри
product_price	ціна
product_discount	знижка
product_description	опис
product_season	сезон для використання
product_articul	артикул
product_color	колір
product_maker	виробник
product_type	тип товару
product_brand_id	бренд
product_collection_id	колекція
product_category_id	категорія
product_likes	кількість збережень
product_avaliability	наявність в магазинах
product_blocked_status	статус блокування
product_image	зображення товару

Таблиця 2.3

Опис таблиці Orders

Поле	Опис
order_id	ідентифікатор замовлення
order_number	номер замовлення
order_user_id	замовник
order_user_name	ім'я, на яке створено замовлення

order_user_email	електронна пошта, на яку оформлено замовлення
order_user_phone_num	номер телефону замовника
order_products	список замовлених товарів
order_sum	сума замовлення
order_discount	знижка на замовлення
order_adress	адреса доставки
order_type	тип замовлення (доставка або самовивіз)
order_status	статус замовлення
order_admin	адміністратор, який виконав замовлення
order_store_id	магазин, який виконав замовлення
order_time	час зміни стану замовлення

Таблиця 2.4

Опис таблиці Events

Поле	Опис
event_id	ідентифікатор події
event_name	назва події
event_description	опис події
event_start_date	дата початку
event_end_date	дата завершення
event_products	товари на які діє подія
event_image	зображення

Таблиця 2.5

Опис таблиці Comments

Поле	Опис
comment_id	ідентифікатор коментаря
comment_description	опис коментаря
comment_user_id	автор
comment_product_id	прокоментований товар
comment_time	час та дата опублікування
comment_assessment	оцінка товару

Таблиця 2.6

Опис таблиці Brands

Поле	Опис
brand_id	ідентифікатор бренду
brand_name	назва бренду
brand_logo	логотип бренду

Таблиця 2.7

Опис таблиці Collections

Поле	Опис
collection_id	ідентифікатор бренду
collection_brand_id	бренд колекції
collection_name	назва бренду
collection_description	логотип бренду

Таблиця 2.8

Опис таблиці Categories

Поле	Опис
category_id	ідентифікатор катеєгорії
category_name	тип категорії (взуття, одяг, аксесуари)
category_type_en	тип одягу (кросівки, кофти, штани і т.д.)
category_type_ua	тип одягу на українській мові
category_gender	стать

Таблиця 2.9

Опис таблиці Stores

Поле	Опис
store_id	ідентифікатор магазину
store_name	назва магазину
store_address	адреса магазину
store_contacts	контактні дані магазину
store_work_time	час відкриття та закриття

Таблиця 2.10

Опис таблиці Club Cards

Поле	Опис
club_card_id	ідентифікатор картки
club_card_name	назва картки
club_card_image	фото картки
club_card_discount	знижка, яку дає картка
club_card_sum	сума накопичених бонусів
club_card_issue_date	термін дії картки

У контексті блокчейн-технологій, дані після створення запитів зберігаються в самому блокчейні. Блокчейн є розподіленою базою даних, яка зберігається на різних вузлах (комп'ютерах) в мережі. Кожен блок у блокчейні містить код, які включають в себе дані, пов'язані зі створенням, оновленням або виконанням запитів.

2.4 Реалізація функціоналу системи інтернет-магазину

Було складено загальний алгоритм роботи, спроектовано та створено базу даних програми. Таким чином можна оцінити його та розбити на більш малі та прості модулі. В результаті будуть отримані алгоритми, які для зручності описують роботу деяких функцій цієї програми. На підставі даних загального алгоритму та функціональних алгоритмів роботи програми можна сформулювати більш реалістичне уявлення про потрібний функціонал та методи вирішення поставлених у технічних завданнях задач.

Необхідність створення сайту з простим інтерфейсом вказує на те що не потрібно намагатися створити занадто красивий інтерфейс користувача.

Для створення проекту було вирішено використовувати патерни : MVC та Repository.

MVC (model-view-controller) - архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення. Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд та модуль керування.

Repository - це патерн проектування, який використовується при розробці програмного забезпечення, що дозволяє централізовано керувати логікою доступу до даних. Він відокремлює логіку, яка отримує дані і зіставляє їх з сутнісною моделлю, від бізнес-логіки, яка оперує цією моделлю.

Ядру додатку, відповідаючим за навігацію, збір параметрів та виконання запитів відповідатиме клас **Route**. Лістинг даного класу наведений в додатку А.

Отже, клас отримує URL-запит введений користувачем, розбиває рядок запиту на навігаційні класи та викликає потрібний контролер, далі викликає функцію контролера для створення потрібної сторінки і за наявності приймає параметри. Якщо користувач вводить рядок на запит з неіснуючою сторінкою чи хибні параметри, роутер автоматично переходить на сторінку з пошуковою помилкою.

Клас головного контролера виглядає наступним чином:

```
<?php

namespace application\core;

class Controller
{
    public $model;
    public $view;
    public $params;

    function __construct($params)
    {
        $this->view = new View();
        $this->params = $params;
    }

    function action_index()
    {
    }

    public static function throwMessage($type, $text){
        echo '<div class="mt-3 alert alert-' . $type . '" role="alert">' . $text . '</div>';
    }
}
```

Після виклику класу контролера він автоматично підшукує свою модель. Без наявності моделі, контролер не зможе працювати. Лістинг класу моделі:

```
<?php

namespace application\core;
```

```

use application\db\Db;

class Model
{
    public Db $db;

    public function __construct()
    {
        $this->db = new Db;
    }
}

```

Стає очевидним, що моделі контролерів, тільки збирають та оновлюють дані, що будуть відображатися на сторінці, пов'язані з БД. У ході створення сторінки підключається клас виду:

```

<?php

namespace application\core;

class View
{
    public $template_view; // вид - по замовчуванню .

    function generate($content_view, $template_view, $title, $data = null )
    {
        if(is_array($data)) {
            // перетворення елементів в масиву в змінні
            extract($data);
        }
        include 'application/views/'.$template_view;
    }
}

```

Клас виду поєднує 2 сторінки одночасно: вид по замовчуванню з шапкою та підвалом сайту та сторінкою контенту.

Розберемо приклад запиту:

<https://fitwear/main/catalog/categories-all/genders-all>

- fitwear – назва домену;
- main – назва контролера, який відповідатиме за клас виконуючий певні запити;
- catalog – сторінка на якій відображаються результати запитів;

- categories-all та genders-all – параметри запитів, що передаються на сторінку.

Лістинг класу контролера та функції моделі Main наведений в додатку Б.

Розберемо інші головні функції:

1. Функція відображення товару (conroller_product->action_index()).

Спочатку витягується інформація про зацікавлений нами товар, далі витягується інформація про інші товари, які б могли сподобатись користувачеві.

```
function action_index(){
    if (!empty($this->params)){
        $data['product'] = $this->model->GetProd(explode('/', $this->params)[0]);
        if ($data['product']['product_collection_id'] != null){
            $data['collection'] = $this->model->getCollection($data['product']['product_collection_id']);
        }
        $data['rec_products'] = $this->model->GetOtherProds($this->params);
        $this->view->generate('product/product.php', 'layouts/template_view.php',
        $data['product']['product_category_type_ua'].'.'.$data['product']['product_brand'].'
        '.$data['product']['product_name'], $data);
    }
}
```

2. Функція додавання товару до кошика (Controller_basket->action-add()).

Через запит передається ідентифікатор та розмір обраного товару до кошику. Кошик зберігається в сховищі сесії (сеансу).

```
function action_add(){
    session_start();
    if (!empty($this->params && !empty($_POST))){
        $checker = false;
        foreach ($_SESSION['basket'] as $item){
            if ($item['product_id'] == $this->params && $item['product_size'] == $_POST['size']){
                $checker = true;
            }
        }
        if ($checker){
            $_SESSION['basket'][$this->params.$_POST['size']]['basket_product_count'] += 1;
        }
        else{
            $_SESSION['basket'][$this->params.$_POST['size']] = $this->model->getProduct($this->params);
        }
    }
}
```

```

$_SESSION['basket'][$this->params.$_POST['size']]['basket_product_count'] = 1;

$_SESSION['basket'][$this->params.$_POST['size']]['product_size'] = $_POST['size'];
}
}
else{
    $_SESSION['message'] = [
        'type' => 'danger',
        'text' => 'Помилка додавання товару до кошика!',
    ];
}
header('location: https://fitwear/basket');
}

```

3. Функція аутентифікації (Controller_user->action_sign_in()). Форма приймає дані користувача для входу, та в залежності від успішності входу надає доступ до сховища сеансу та багатьох функцій для використання.

```

function action_signin(){
    if(!empty($_POST)) {
        session_start();
        $result = $this->model->Login($_POST['login'], $_POST['pass']);
        if($result != false){
            $_SESSION['message'] = [
                'type'=>'success',
                'text' => 'Користувач успішно увійшов!',
            ];
            $_SESSION['user'] = $result;
            if ($_SESSION['user']['user_role_ability'] == 1)
                header('Location: account');
            else
                header('location: https://fitwear/admin/account');
        }
        else{
            $_SESSION['message'] = [
                'type' => 'danger',
                'text' => 'Неправильний логін або пароль!',
            ];
            header('Location: login');
        }
    }
}
}

```


4. Функція створення замовлення (Controller_order->action_add()). Функція приймає дані з форми для замовлення, витягує товари зі сховища, створює унікальний номер замовлення та переходить в стан очікування.

```
function action_add(){
    session_start();
    if(!empty($_POST) && isset($_SESSION['user']) && isset($_SESSION['basket']))){
        $uniq_num = '';
        $count = 0;
        while ($count < 10){
            $uniq_num .= rand(0, 9);
            if ($count == 2 || $count == 5)
                $uniq_num .= '-';
            $count += 1;
        }
        $sum = 0;
        foreach ($_SESSION['basket'] as $item) {
            if ($item['product_discount'] != 0){
                $sum += round($item['product_price'] * (100 - $item['product_discount'])/100) *
                $item['basket_product_count'];
            }
            else{
                $sum += $item['product_price'] * $item['basket_product_count'];
            }
        }
        $products = '';
        foreach ($_SESSION['basket'] as $item){
            $products .= $item['product_id'] . '-' . $item['product_size'] . '-' . $item['basket_product_count'] . ',';
        }
        $products = substr($products, 0, -2);
        $order = [
            'order_number' => $uniq_num,
            'order_user_id' => $_SESSION['user']['user_id'],
            'order_user_name' => $_POST['firstName'] . ' ' . $_POST['middleName'] . ' ' . $_POST['lastName'],
            'order_user_email' => $_POST['email'],
            'order_user_phone_num' => $_POST['phone_num'],
            'order_products' => $products,
            'order_sum' => $sum,
            'order_discount' => 0,
            'order_type' => $_POST['delivery'] == 'by_own' ? 'Самовиніс' : 'Доставка',
            'order_adress' => $_POST['delivery'] == 'by_own' ? '' : $_POST['address'] . ', ' . $_POST['city'] . ', ' .
            $_POST['region'] . ', ' . $_POST['index'],
            'order_status' => $_POST['payment'] == 'now' && isset($_POST['cc-number']) && isset($_POST['cc-
            expiration']) && isset($_POST['cc-cvv']) ? 'Замовлення сплачено' : 'Замовлення створено',
        ];
    }
```

$$\left. \begin{array}{l} \{ \\ \} \end{array} \right\}$$

компонентів застосунку (рис. 2.12).

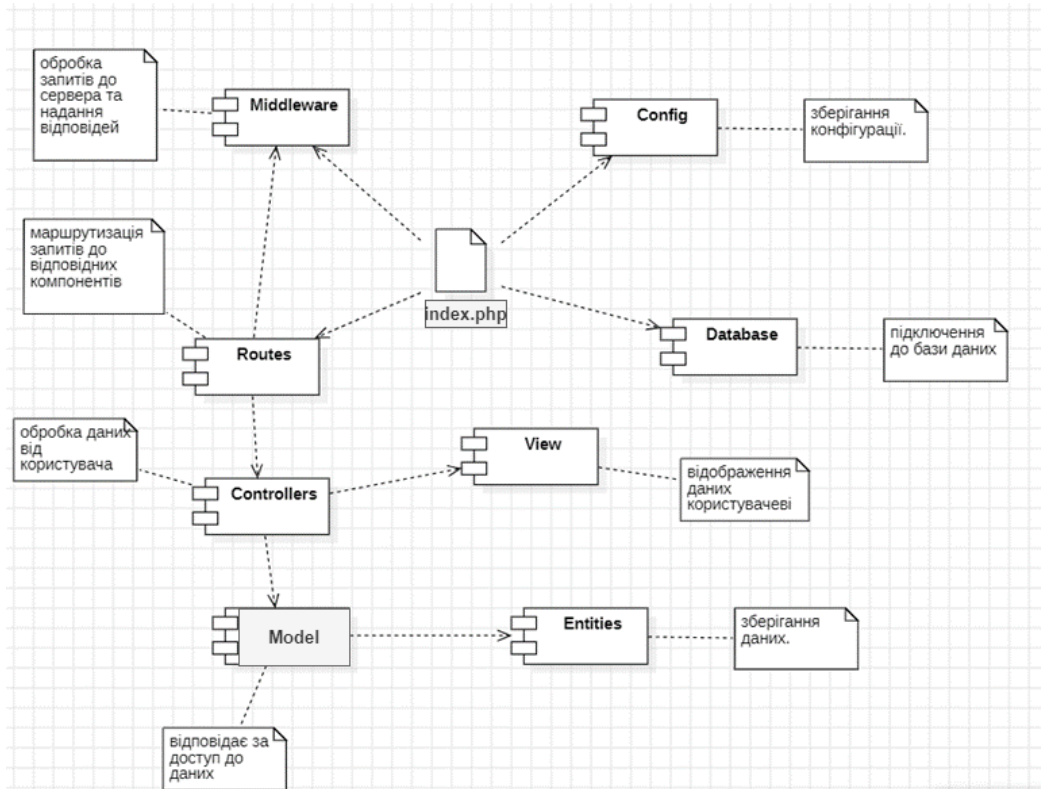


Рисунок 2.12 - Діаграма компонентів додатку

Висновки до другого розділу

В процесі проектування веб-додатку висунуті вимоги для створення веб-додатку та проаналізовані варіанти використання відповідно до вимог. Підсумувавши, було спроектовано структуру проекту.

Було спроектовано та створено базу даних системи, для зберігання інформації пов'язаних з товарами, замовленнями та користувачами інтернет-магазину.

Описані алгоритми роботи веб-додатку та послідовні пункти його використання.

Було описано основний функціонал додатку, спроектовано класи та головні функції для взаємодії користувача з додатком та необхідних для виконання замовлень в застосунку.

					ІІЗ.КР.Б-121-24-ІІЗ	Арк.
						45

РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ

3.1 Користувацький інтерфейс

Для створення користувацького інтерфейсу використовувалась верстка Bootstrap.

Bootstrap — це безкоштовний набір інструментів з відкритим кодом, призначений для створення вебсайтів та вебзастосунків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних вебсайтів і вебзастосунків.

Bootstrap — це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері.

У цій роботі використовувалась найостанніша версія v.5.3.3

Взаємодія користувачів відбувається через вебдодаток. На головній сторінці користувач може дивитися актуальні події, популярні товари, та товари нових колекцій (рис 3.1– 3.3).

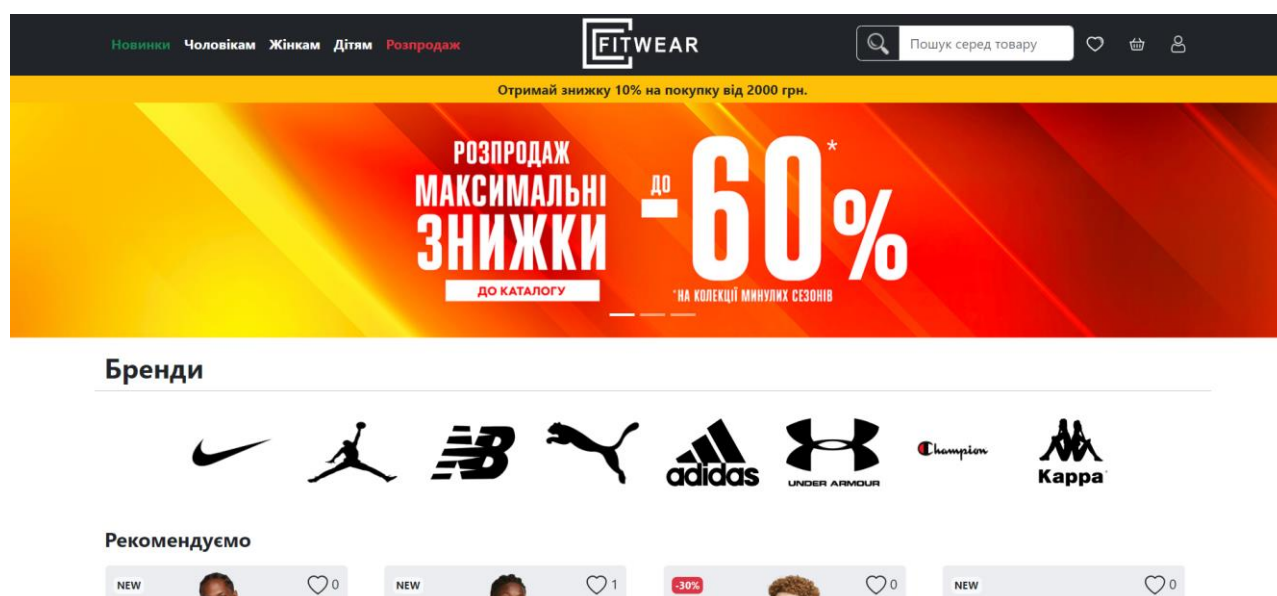


Рисунок 3.1 - Головна сторінка інтернет-магазину

					ІПЗ.КР.Б-121-24-ІПЗ	Арк.
						46

Рекомендуємо

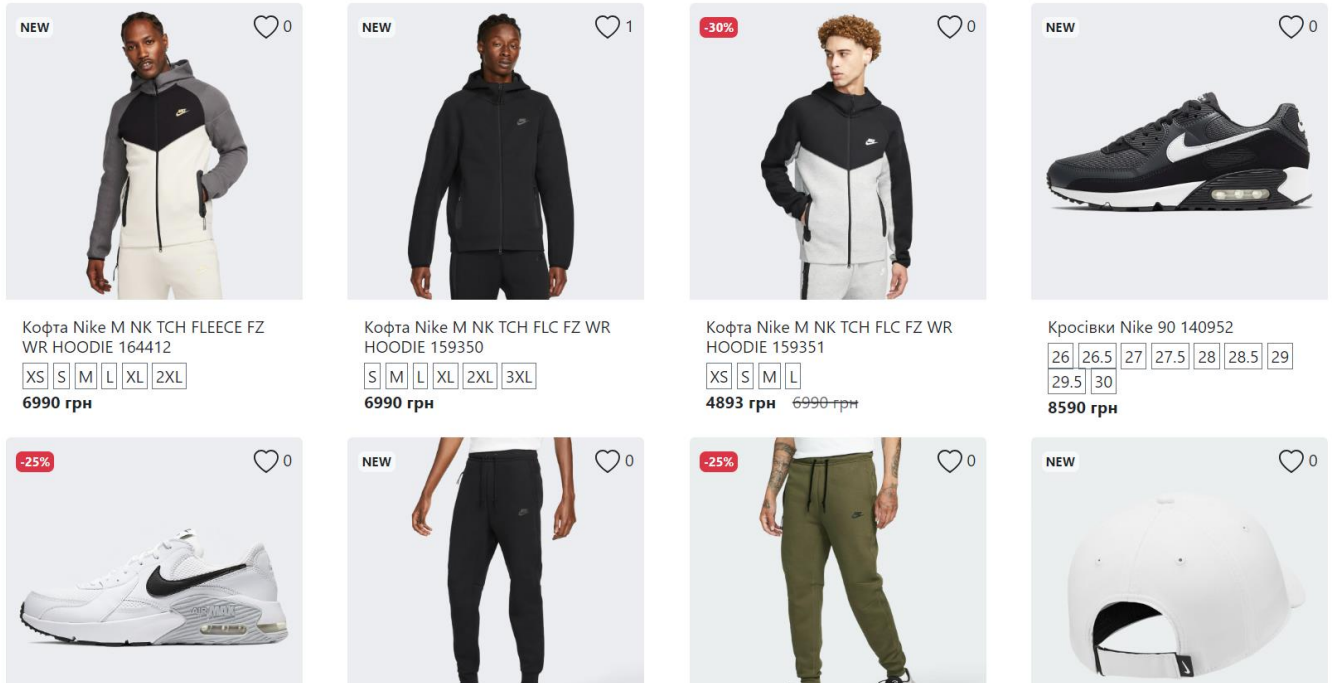


Рисунок 3.2 - Головна сторінка інтернет-магазину

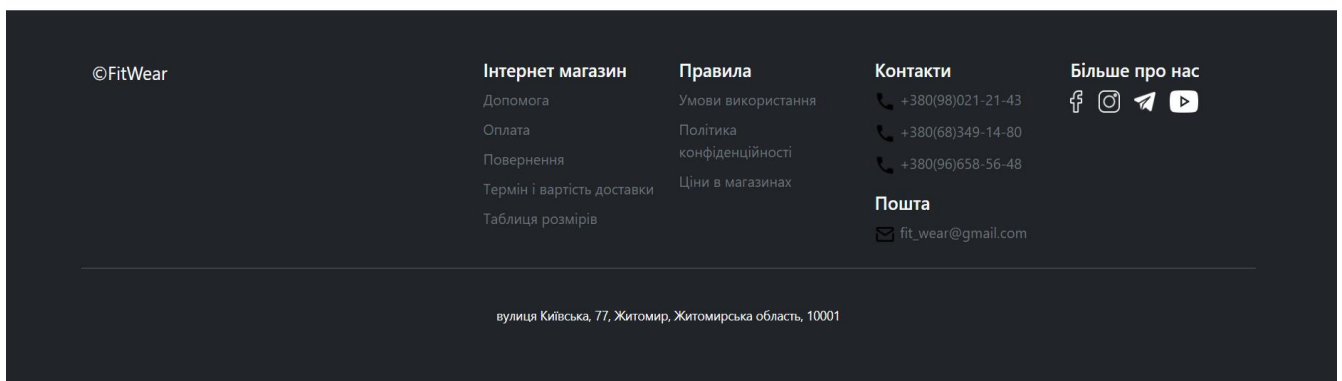
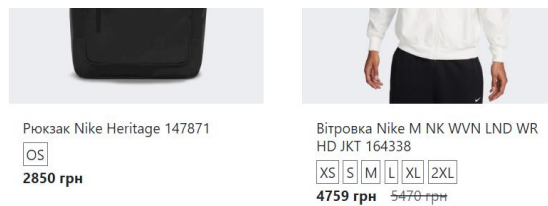


Рисунок 3.3 - Головна сторінка інтернет-магазину

У верхній частині сторінки знаходиться навігаційна панель (рис. 3.4) за допомогою якої користувач може перейти до сторінки каталогу (рис. 3.5).



Рисунок 3.4 - Навігаційна панель шапки сайту

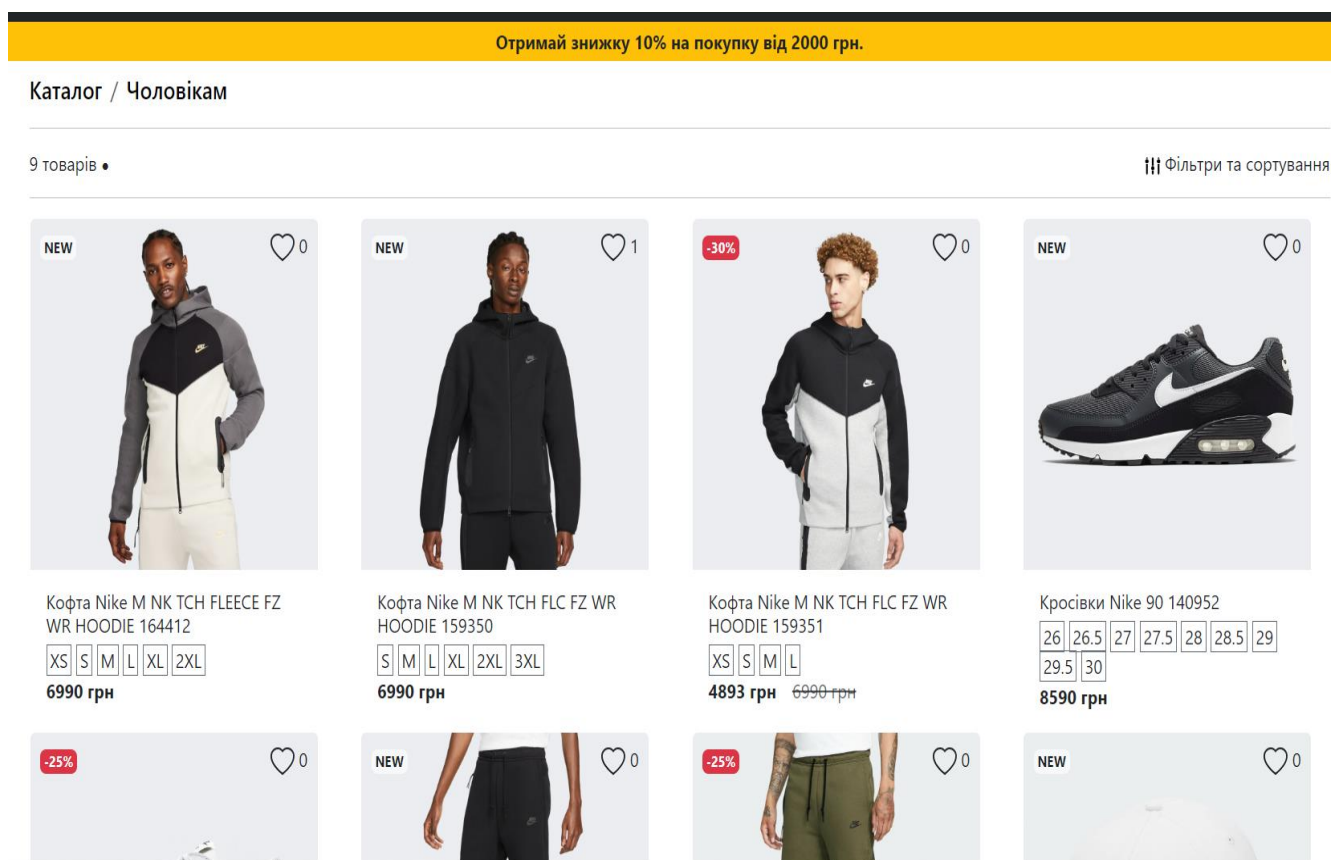


Рисунок 3.5 - Сторінка каталогу

У каталозі користувач має змогу підібрати фільтри та обрати спосіб сортування товару. Для цього потрібно натиснути на кнопку «Фільтри та сортування», для того, щоб вплило модальне вікно з вибором параметрів (рис. 3.6)

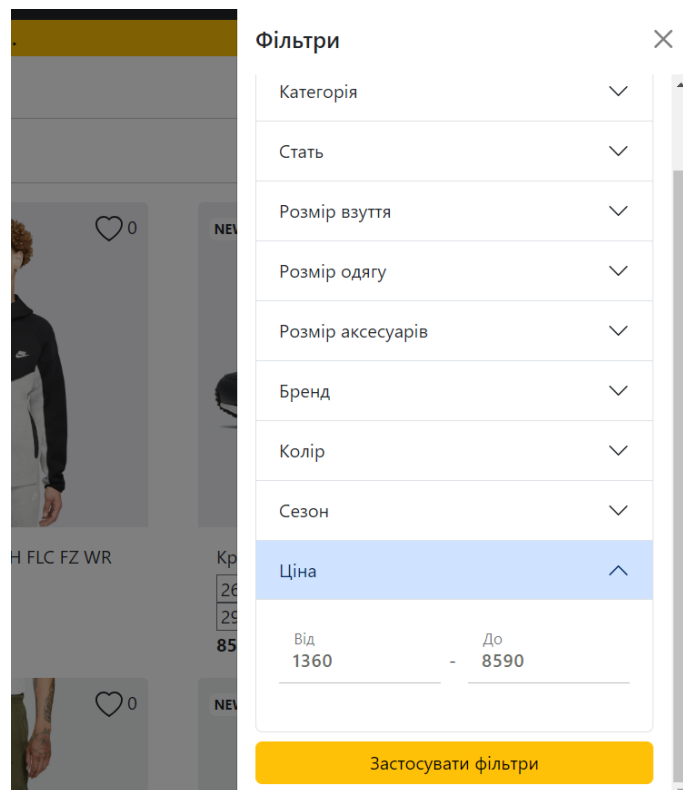


Рисунок 3.6 - Вікно застосування фільтрів у каталозі

У каталозі та на головній сторінці ми можемо переглянути товар, натиснувши на нього. Загальний вигляд сторінки товару зображений на рис. 3.7 – 3.8.

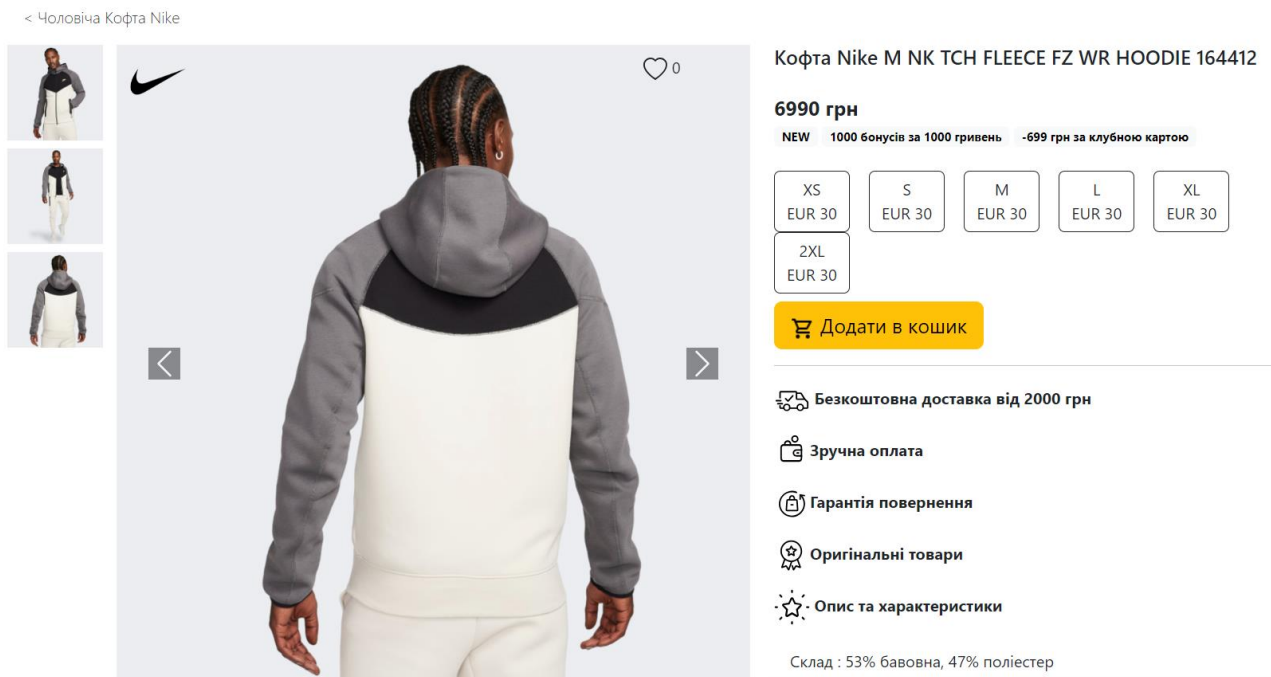


Рисунок. 3.7 - Сторінка товару

Вам може сподобатись



Кофта Nike M NK TCH FLC FZ WR HOODIE 159350

S M L XL 2XL 3XL

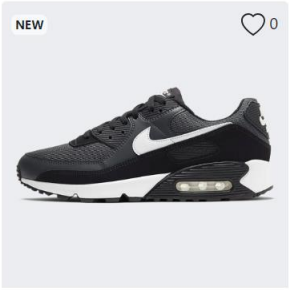
6990 грн



Кофта Nike M NK TCH FLC FZ WR HOODIE 159351

XS S M L

4893 грн 6990 грн



Кросівки Nike 90 140952

26 26.5 27 27.5 28 28.5 29 29.5 30

8590 грн



Кросівки Nike Excee 123982

26 26.5 27 27.5 28 28.5 29 29.5 30

4718 грн 6290 грн

Залиште відгук

admin@fitwear.com

Admin_m

Ваша оцінка

1 2 3 4 5

Рисунок 3.8 - Сторінка товару

На сторінці товару відображається основне фото товару, додаткові фото, інформація про доставку, товари, схожі до поточного, та коротка інформація про товар, але можна отримати більш детальну, натиснувши на кнопку «Опис та характеристики», в такому випадку з'явиться спливаюче вікно з повним описом та характеристикою товару (рис. 3.9)

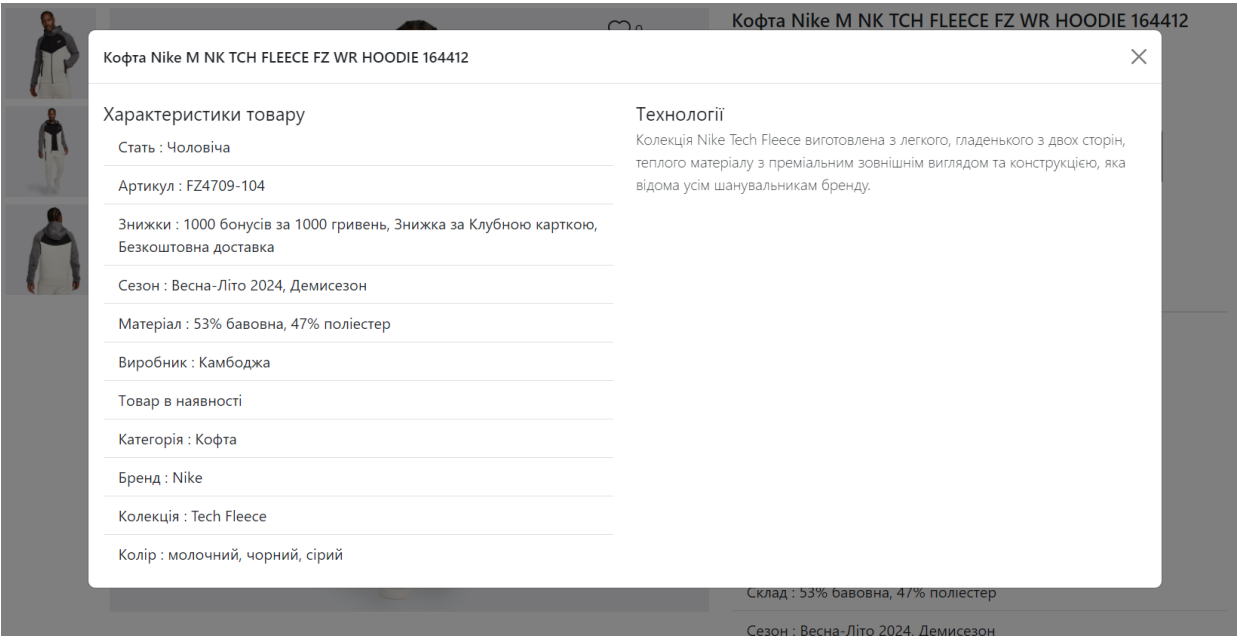


Рисунок 3.9 - Детальна інформація про товар

Також є можливість додати товар до обраного. Для цього потрібно натиснути на сердечко у правому верхньому кутку. Товар додається до списку бажаного та лєсистема автоматично поверне вас на сторінку того ж товару. Для додовання товару в кошик необхідно обрати розмір товару та натиснути на кнопку «Додати до кошика». Система додасть товар з обраним розміром до кошика та перенесе на його сторінку (рис.3.10)

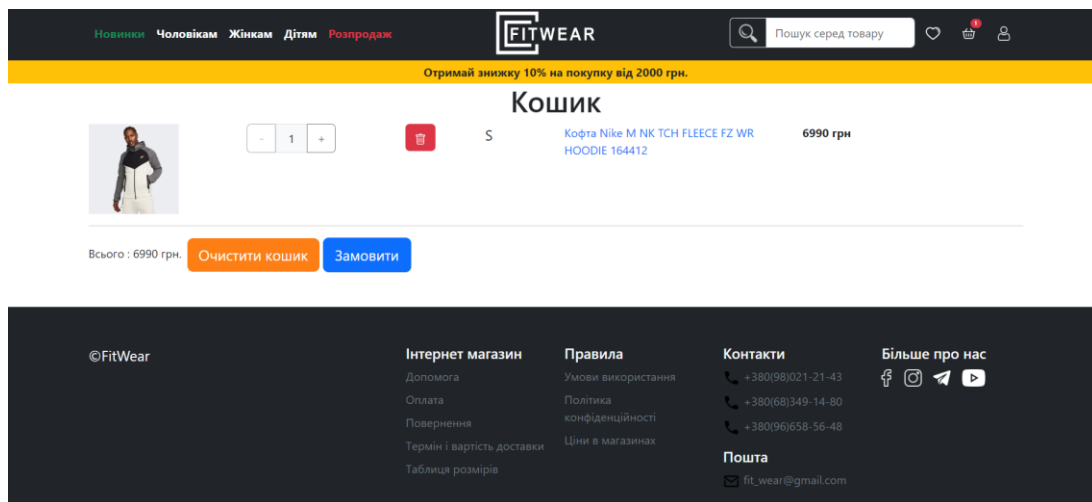


Рисунок 3.10 - Сторінка кошика

На сторінці кошику можна змінити кількість товару, що користувач хоче замовити, видити елемент з корзини очистити корзину повністю, та перейти до замовлення товару з кошика. Сторінка замовлення зображена на рис. 3.11.

Замовлення

Платіжна адреса

Прізвище: Ім'я: По-батькові:

Електронна пошта: Номер телефону:

Доставка

☒ Самовиніс ☐ Доставка

Адреса:

Місто: Область: Поштовий індекс:

Ваш кошик 1

Кофта Nike M NK TCH FLEECE FZ WR HOODIE 1 x 6990 грн.
164412

Всього: **6990 грн.**

Промо-код Активувати

Рисунок 3.11 - Сторінка замовлення

Після внесення даних, створюється замовлення і переходить в стан очікування, а користувача переносить до сторінки з історією замовлень (рис. 3.12).

Історія замовлень

Замовлення №788-118-5042, Статус - Замовлення створено



Кофта Nike M NK TCH FLEECE FZ WR HOODIE 164412

6990 грн. - 1 шт.

Колір: молочний, чорний, сірий, Артикул: FZ4709-104

Розмір S

Замовник: Admin_f Admin_m Admin_l, +380000000000

Тип доставки: Самовиніс,

Загальна сума: 6990 грн.

Скасувати замовлення

Рисунок 3.12 - Історія замовлень користувача

Але перш, ніж користувача перенесе до історії замовлень, він вже має бути авторизований на сайті. Для цього він має здійснити вхід на сторінці входу та реєстрації (рис 3.13, 3.14).

Вхід

Реєстрація

E-mail

Пароль

Увійти

Рисунок 3.13 - Сторінка входу (форма авторизації)

Вхід

Реєстрація

E-mail

Прізвище

Ім'я

По-батькові

Пароль

Повторіть пароль

Зареєструватися

Рисунок 3.14 - Сторінка входу (форма реєстрації)

Вже після входу користувач може переглядати інформацію про себе та діяльність пов'язану з його обліковим записом. Сторінка користувача зображена на рис. 3.15.

Обліковий запис

Іvanov1 Ivan

Клубна картка

Історія замовлень

Статус замовлення

Список бажаного

Промокоди

Бонуси

Вихід

Оновлення даних

Прізвище:

Ivanov1

Ім'я:

Ivan

По Батькові:

Ivanovich

Електронна пошта:

ivan_ivanov@gmail.com

Дата народження:

2000-01-01

Адреса:

Номер телефону:

+380000000000

Оновити дані

Зміна паролю

Новий пароль:

Повторіть новий пароль:

Змінити пароль

Користувач успішно увійшов!

Рисунок 3.15 - Сторінка користувача

					ІПЗ.КР.Б-121-24-ІПЗ	Арк.
						53

3.2 Інтерфейс для адміністратора та власника інтернет-магазину

Для доступу адміністративної панелі та інших можливостей адміністратора, користувачу необхідно мати спеціальний рівень доступу, який привласнюється іншим адміністратором або власником інтернет-магазину.

Головна сторінка адміністратора зображена на рис. 3.16.

Admin_f Admin_m

Коментарі

Замовлення

Користувачі

Список бажаного

Промокоди

Бонуси

Заходи

Товари

Статистика

Вихід

Оновлення даних

Прізвище: Admin_f

Ім'я: Admin_m

По Батькові: Admin_I

Електронна пошта: admin@fitwear.com

Дата народження: 2000-01-01

Адреса:

Номер телефону: +380000000000

Оновити дані

Зміна паролю

Новий пароль:

Повторіть новий пароль:

Змінити пароль

Користувач успішно увійшов!

Рисунок 3.16 - Сторінка адміністратора

Адміністратор має такі можливості як узгодження опублікування коментарів, погодження замовлень, ведення обліку користувачів, промокодів та бонусів, створення заходів, додавання, видалення та зміни інформації про товар, перегляд статистики (рис. 3.17 – 3.18).

Коментарі

Нові коментарі

Оцінка: 5

Кофта Nike M NK TCH FLC FZ WR HOODIE 159351

Відгук: Відмінна річ


ivan_ivanov@gmail.com, Ivan

2024-06-10 12:16:15

Рисунок 3.17 - Узгодження відгуків користувачів


Історія замовлень

Замовлення №122-933-1794, Статус - Замовлення створено, 2024-06-12



Кофта Nike M NK TCH FLC FZ WR HOODIE 159350
Колір: чорний, Артикул: FB7921-010
Розмір S

6990 грн. - 2 шт.



Кофта Nike M NK TCH FLC FZ WR HOODIE 159351
Колір: чорний, сірий, Артикул: FB7921-064
Розмір S


4893 грн. - 1 шт.

Замовник: Ivanov Ivan Ivanovich, +380000000000
Тип доставки: Самовиніс,
Загальна сума: 30756 грн.

Прийняти замовлення


Відмінити замовлення

Замовлення №093-872-6455, Статус - Замовлення створено, 2024-06-13



Кофта Nike M NK TCH FLC FZ WR HOODIE 159350
Колір: чорний, Артикул: FB7921-010
Розмір S

6990 грн. - 2 шт.



Кофта Nike M NK TCH FLC FZ WR HOODIE 159351
Колір: чорний, сірий, Артикул: FB7921-064
Розмір S

4893 грн. - 1 шт.

Замовник: Ivanov Ivan Ivanovich, +380000000000
Тип доставки: Самовиніс,
Загальна сума: 30756 грн.

Прийняти замовлення

Відмінити замовлення

Рисунок 3.18 - Управління замовленнями

Користувачі

Ivanov1 Ivan Ivanovich, ivan_ivanov@gmail.com Адреса: Дата народження: 2000-01-01 Зареєстровано: 2024-05-01	+380000000000
Петрук Вадим Павлович, petruk_vadim@meta.com Адреса: Дата народження: 1997-08-24 Зареєстровано: 2024-06-13	+380963267289
Кириychук Антон Петрович, toha777@gmail.com Адреса: Дата народження: 2004-03-13 Зареєстровано: 2024-05-17	+380630785672
Ніколайчук Ярослав Сергійович, yaroslav_nik@gmail.com Адреса: Дата народження: 2001-12-02 Зареєстровано: 2024-06-02	+380954317492
Ivanov Ivan Ivanovich, ivan_ivanov1@gmail.com Адреса: Дата народження: 2003-09-29 Зареєстровано: 2024-06-08	+380980256162

Рисунок 3.19 - Сторінка ведення обліку користувачів

Статистика

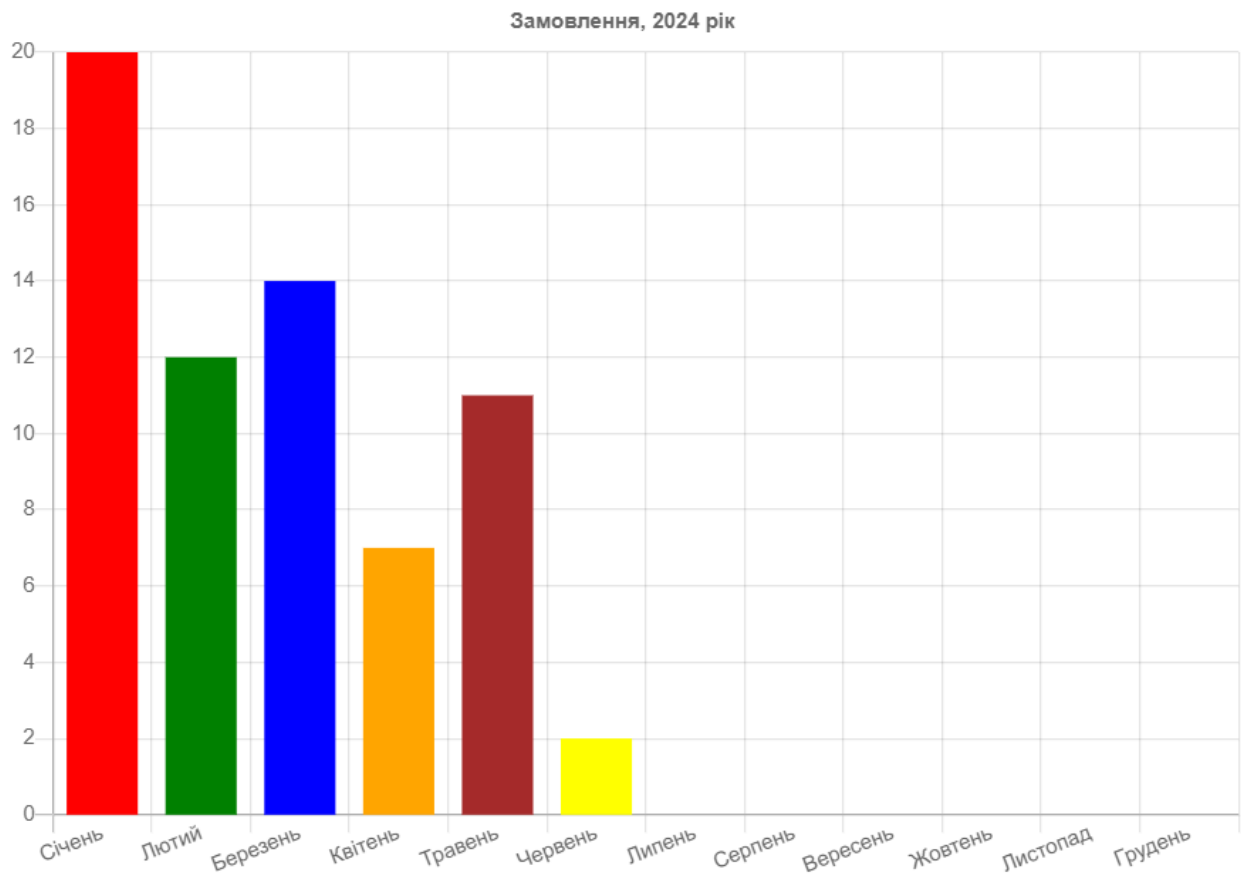


Рисунок 3.20 - Сторінка перегляду статистики

3.3 Тестування програмного забезпечення

При розробці програмного забезпечення все було створено, щоб при використанні веб-додатку виникала мінімальна кількість помилок. Тому серед усіх помилок у програмі було виявлено лише одну помилку, при якій іноді не змінюються у налаштуваннях як там записано «основні дані». Скоріше всього помилка може виникати або через неправильно записаний SQL- запит або через неправильний запис POST-методу.

В інших випадках програма методи контролерів самі перевіряють наявність помилок при передачі даних або при їх відсутності.

Спочатку протестуємо всі виключення при реєстрації:

1) Реєстрація з уже використаним email-ом (рис.3.21):

Вхід

Реєстрація

E-mail

ivan_ivanov@gmail.com

Прізвище

Ivanov

Ім'я

Ivan

По-батькові

Ivanovich

Пароль

....

Повторіть пароль

....

Зареєструватися

Користувач під цією поштою вже зареєстрований!

Рисунок 3.21 - Тест (реєстрація з використовуваним логіном)

2) Неправильно підтверджений пароль при реєстрації (рис.3.22):

Вхід

Реєстрація

E-mail

ivan_ivanov@gmail.com

Прізвище

Ivanov

Ім'я

Ivan

По-батькові

Ivanovich

Пароль

Повторіть пароль

Зареєструватися

Невірно підтверджений пароль!

Рисунок 3.22 - Тест (реєстрація з непідтвердженим паролем)

Отже все працює, але в ієрархії виключень, що задані в коді , виключення з паролем стоїть вище за логін, тобто при введенні існуючого логіну та непідтвердженого паролю повідомлення повідомить нас про непідтверджений пароль.

Тепер про виключення при вході. Було створено одне виключення при якому нас повідомляють про неправильно введені дані:

The screenshot shows a login interface with two tabs: 'Вхід' (Login) and 'Реєстрація' (Registration). Under the 'Вхід' tab, there are two input fields: 'E-mail' and 'Пароль' (Password). Below these fields is an orange button labeled 'Увійти' (Login). A pink error message box is displayed below the button, containing the text 'Неправильний логін або пароль!' (Incorrect login or password!).

Рис.3.23 – Тест (неправильні дані для входу)

Ще одне з виключень, що виникають вже при знімі інформації користувача - невірне підтвердження паролю:

The screenshot shows a form titled 'Обліковий запис' (User Profile). It is divided into two sections: 'Оновлення даних' (Update data) and 'Зміна паролю' (Change password). The 'Оновлення даних' section contains several input fields with pre-filled values: 'Прізвище:' (Ivanov1), 'Ім'я:' (Ivan), 'По Батькові:' (Ivanovich), 'Електронна пошта:' (ivan_ivanov@gmail.com), 'Дата народження:' (2000-01-01), 'Адреса:', and 'Номер телефону:' (+380000000000). There is an orange 'Оновити дані' (Update data) button at the bottom of this section. The 'Зміна паролю' section has two input fields: 'Новий пароль:' and 'Повторіть новий пароль:'. The 'Повторіть новий пароль:' field contains '....|', indicating a mismatch. A yellow 'Змінити пароль' (Change password) button is below these fields. A pink error message box at the bottom of the form reads 'Помилка! Невірно підтверджений пароль!' (Error! Password not confirmed correctly!).

Рисунок 3.24 – Тест(неправильний поточний пароль)

Також виникають виключення при відсутності авторизації для певних дій з її потребою. Наприклад перегляд вподобаних товарів (рис.3.25).

Вхід

Реєстрація

E-mail

Пароль

Увійти

Ви не увійшли в обліковий запис!

Рисунок 3.25 – Тест (перегляд вподобаних товарів без авторизації)

При переході на неіснуючі сторінки, система переходить на сторінку 404 (рис.3.26).

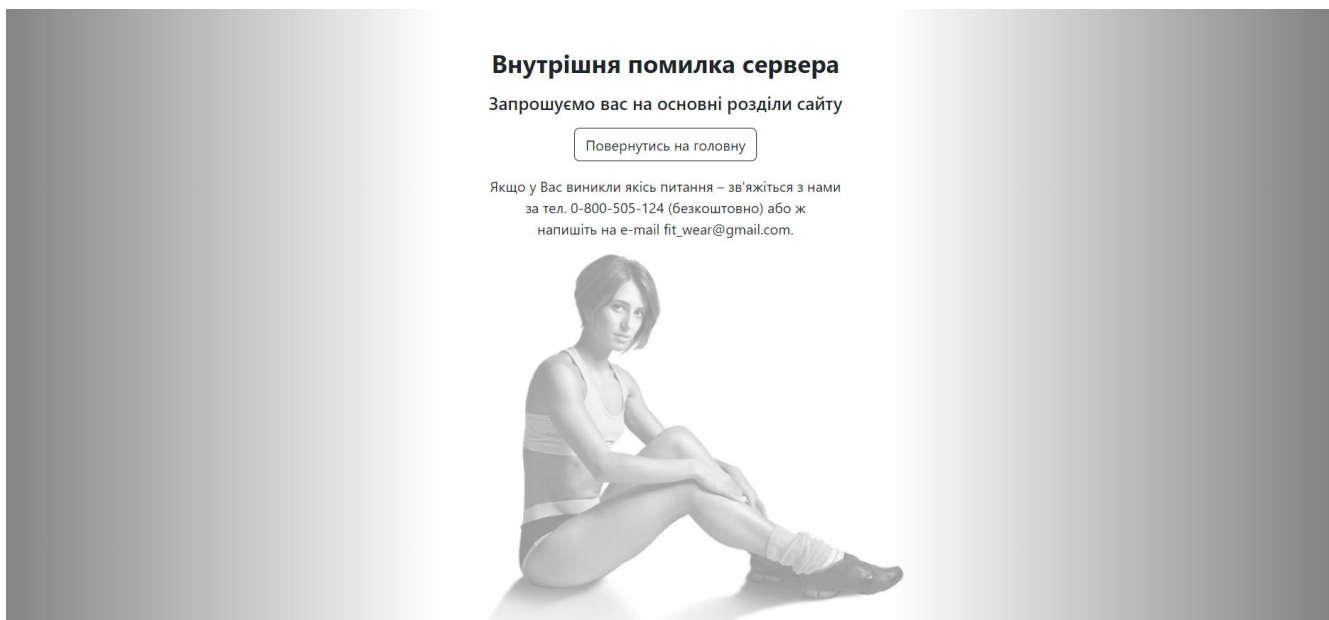


Рисунок 3.25 - Сторінка 404

Висновки до третього розділу

В цьому розділі було продемонстровано користувацький інтерфейс по сторінкам користувача, за алгоритмом замовлення товару на сайті. Було описано вигляд та можливості користувачів на цих сторінках. Це дало зрозуміти зв'язки між системою та архітектуру проекту.

Наступним було описано інтерфейс адміністратора, та власника інтернет-магазину, де будуть прийматися та узгоджуватися дії користувачів. Також це дало зрозуміти можливості користувачів за їхніми рівнями доступу. Це допомагає кожному користувачу зорієнтуватись у функціоналі додатку та використовувати його відповідно до своїх потреб.

Також було проведено тестування додатку. На проведення тестування було висунуто основні моменти, які потребують виникнення виключень. Загалом в процесі тестування було засвідчено, що програма виконує те, що від неї очікувалося на початку.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано веб-додаток інтернет-магазину, проведено аналіз предметної області та порівняно різні платформи. На основі цього аналізу були визначені основні вимоги до застосунку. В процесі дослідження була розроблена архітектура, підходи та основні технології реалізації веб-додатку.

Проектування платформи включало аналіз варіантів використання програмного продукту та проектування структури, включаючи моделювання взаємодії між складовими частинами. Також було спроектовано базу даних для зберігання інформації перед її записом на блокчейн, а також для даних, що зберігаються на серверній частині додатку.

Описано алгоритм роботи з додатком та послідовні пункти варіантів його використання. Розглянуто основний функціонал, включаючи створення замовлень, підтвердження адміністрацією, а також основні функції взаємодії користувачів замовленням.

Також була надана детальна схема розгортання веб-додатку, описано взаємодію структурних елементів системи та наведено опис інтерфейсу розробленого програмного продукту для різних типів користувачів. Також показано роботу статистики для адміністрації та звичайних користувачів, що надає інформацію про статус збору коштів, активність донорів та статистику за категоріями.

Додаток має інтуїтивний інтерфейс, який дозволяє користувачам легко переміщатися між сторінками в логічному порядку, знаходити потрібні їм товари, створювати облікові записи. Користувачі також можуть взаємодіяти з товарами, зберігати свої вподобання, створювати замовлення і спостерігати за їх станом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>.
2. Колірна гармонія [Електронний ресурс] – Режим доступу до ресурсу: <https://libdiz.com/uk/knygy/mystetstvo-kolioru/3-kolirnagarmoniya/>.
3. JetBrains [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/>.
4. Юрасов А. В. Електронна комерція навчальний посібник / А. В. Юрасов., 2003. – 482 с. – (Справа).
5. PHP.net [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/>.
6. Javascript.info [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.javascript.info/>.
7. Підручник по Javascript [Електронний ресурс] – Режим доступу до ресурсу: <https://w3schoolsua.github.io/js/index.html#gsc.tab=0>.
8. Іванкевич О. В. Інформаційні системи та структури даних / О. В. Іванкевич, Г. М. Кременецький, В. І. Мазур. – Київ: НАУ, 2006. – 232 с.
9. MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mysql.com/>.
10. Charts.js [Електронний ресурс] – Режим доступу до ресурсу: <https://www.chartjs.org/>.
11. Васильєв О. Програмування мовою PHP / О. Васильєв. – Київ: Ліра-К, 2022. – 368 с.
12. Елізабет Р. Head First. Програмування на JavaScript / Р. Елізабет, Ф. Ерік. – Харків: Фабула, 2014. – 700 с. – (1 видання).
13. Марк М. Розумніший спосіб вивчати JavaScript / Майєрс Марк. – Південна Кароліна, США: CreateSpace Independent Publishing Platform, 2014. – 254 с.

14. PhpMyAdmin [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.phpmyadmin.net/>.

15. Балик Н. Книга MySQL: лабораторний практикум / Н. Балик, В. Мандзюк.
– Київ: Навчальна книга - Богдан, 2008. – 88 с.

ДОДАТКИ

					ІІЗ.КР.Б-121-24-ІІЗ	Арк.
						64

Лістинг класу Route

```

<?php
namespace application\core;
use PDO;
class Route
{
    static function start()
    {
        // контролер та дія за замовчуванням
        $controller_name = 'Main';
        $action_name = 'index';
        $params = [];

        $routes = explode('/', trim($_SERVER['REQUEST_URI'], '/'));

        // отримуємо ім'я контролера
        if ( !empty($routes[0]) )
        {
            $controller_name = $routes[0];
        }

        // отримуємо ім'я екшена
        if ( !empty($routes[1]) )
        {
            $action_name = $routes[1];
        }
        if ( !empty($routes[2]) )
        {
            if (empty($routes[3])){
                $params = $routes[2];
            }
            else{
                $params = "";
                for ($i = 2; $i < count($routes); $i++){
                    $params = $params . $routes[$i] . '/';
                }
                $params = substr($params, 0, -1);
            }
        }

        // додаємо префікси
        $model_name = 'Model_'. $controller_name;
        $controller_name = 'Controller_'. $controller_name;
        $action_name = 'action_'. $action_name;
    }
}

```

					ІІЗ.КР.Б-121-24-ІІЗ	Арх.
						65

```

// підчеплюємо файл з класом моделі (файлу моделі може і не бути)

$model_file = strtolower($model_name).'.php';
$model_path = "application/models/".$model_file;
if(file_exists($model_path))
{
    include "application/models/".$model_file;
}
// підчеплюємо файл з класом контролера
$controller_file = strtolower($controller_name).'.php';
$controller_path = "application/controllers/".$controller_file;
if(file_exists($controller_path))
{
    include "application/controllers/".$controller_file;
}
else
{
    /*Тут вірно записати виключення, але ми одразу робимо
    переключення на сторінку 404*/
    Route::ErrorPage404();
}

// створюємо контролер
$controller = new $controller_name($params);
$action = $action_name;

if(method_exists($controller, $action))
{
    $controller->$action();
}
else
{
    // тут потрібно записати виключення
    Route::ErrorPage404();
}
}

static function connect(){
    $user = "root";
    $password = "";
    $dsn = "mysql:host=localhost;dbname=fitwear-db;port=3306;charset=utf8";
    $connection = new PDO($dsn, $user, $password);
    $connection->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
    return $connection;
}

```


```

    }

    static function ErrorPage404(){
        $host = 'https://'.$_SERVER['HTTP_HOST'].'/';
        header('HTTP/1.1 404 Not Found');
        header("Status: 404 Not Found");
        header('Location:'.$host.'404');
    }
}

```

Додаток Б

Лістинг класу контролера Main (controller_main)

```

<?php

use application\core\Controller;
use application\core\View;
use application\models\model_main;

include 'application/instruments/functions.php';

class Controller_Main extends Controller
{
    function __construct($params)
    {
        $this->view = new View;
        $this->model = new model_main();
        $this->params = $params;
    }

    function action_index()
    {
        $data['products'] = $this->model->getProducts();
        $data['brands'] = $this->model->getBrands();
        $this->view->generate('main/main_view.php', 'layouts/template_view.php', 'Головна', $data);
    }

    function action_catalog(){
        if(empty($this->params)){
            header('location: https://fitwear');
        }else{
            $parameters = explode('/', $this->params);

            $filters = [
                'category' => $parameters[0] == 'categories-all' ? null : $parameters[0],
            ];

```


```

        if ($parameters[1] == 'male'){
            $filters['gender'] = 'Чоловіча';
        }elseif ($parameters[1] == 'female'){
            $filters['gender'] = 'Жіноча';
        }elseif ($parameters[1] == 'child'){
            $filters['gender'] = 'Дитяча';
        }
        if (count($parameters) == 3){
            $parameters = trim($parameters[2], '?');
            $arr = explode('&', $parameters);
            foreach ($arr as $item){
                $filters[explode('=', $item)[0]] = explode('=', $item)[1];
            }
        }
    }

    $data['products'] = $this->model->getProducts($filters);
    $data['count_products'] = $this->model->getCountProducts($filters)['count'];
    $data['categories'] = $this->model->getCategories();
    $data['brands'] = $this->model->getBrands();
    $data['prices'] = $this->model->getMinAndMaxPrices();

    $data['f'] = $filters;
    $this->view->generate('main/catalog.php', 'layouts/template_view.php', 'Каталог', $data);
}
}
}

```

Лістинг класу моделі Main (model_main)

```

<?php

namespace application\models;

class model_main extends \application\core\Model
{
    function getProducts($params = []){
        $sql = "SELECT `product_id`, `product_name`, `product_gender`, `product_sizes`, `product_price`,
        `product_discount`, `product_description`, `product_season`, `product_articul`, `product_color`,
        `product_maker`, `product_content`, `product_type`, `brands`.`brand_name` AS `product_brand`,
        `product_collection_id`, `category`.`category_name` AS `product_category_name`,
        `category`.`category_gender` AS `product_category_gender`, `category`.`category_type_en` AS
        `product_category_type`, `category`.`category_type_ua` AS `product_category_type_ua`, `product_likes`,
        `product_avaliability`, `product_blocked_status`, `product_link`, `product_new`, `product_add_date`,
        `product_main_image`, `product_images` FROM `products`, `category`, `brands` WHERE
        `products`.`product_category_id` = `category`.`category_id` AND `products`.`product_brand_id` =
        `brands`.`brand_id`;
        if (!empty($params)){

```

```

$count = 0;
$filter = 'rec';
$price_min = -1;
$price_max = -1;
$sql .= ' AND ';
foreach ($params as $key => $value){
    $count += 1;
    if ($key == 'category'){
        if ($value != null){
            $sql .= '(`category`.`category_type_en` = ' . "\"\"\"\" . $value . "\"\"\"\" . ' OR `category`.`category_name` = '
            . "\"\"\"\" . $value . "\"\"\"\" . ')'. ($count == count($params) ? '' : ' AND ');
        }

        continue;
    }elseif ($key == 'f'){
        $filter = $value;
        continue;
    }elseif ($key == 'price_min'){
        $price_min = $value;
        continue;
    }elseif ($key == 'price_max'){
        $price_max = $value;
        continue;
    }
    $sql .= '`products`.`product_` . $key . '` = ' . "\"\"\"\" . $value . "\"\"\"\" . ($count == count($params) ? '' : ' AND ');
}
$sql .= ($price_min == -1 ? '' : ' AND `products.product_price` > ' . $price_min) .
($price_max == -1 ? '' : ' AND `products.product_price` < ' . $price_max);

}
$result = $this->db->row($sql);
foreach ($result as $item){
    if ($item['product_type'] == 0){
        $item['product_type'] = 'clothes';
    }elseif ($item['product_type'] == 1){
        $item['product_type'] = 'shoes';
    }elseif ($item['product_type'] == 2){
        $item['product_type'] = 'accessories';
    }
}
return $result;
}

function getCountProducts($params = []){
    $sql = "SELECT COUNT(*) AS `count` FROM `products`, `category`, `brands` WHERE
`products`.`product_category_id` = `category`.`category_id` AND `products`.`product_brand_id` =

```

```

`brands`.`brand_id`;
    if (!empty($params)) {
        $count = 0;
        $price_min = -1;
        $price_max = -1;
        $sql .= ' AND ';
        foreach ($params as $key => $value) {
            $count += 1;
            if ($key == 'category') {
                if ($value != null){
                    $sql .= '('`category`.`category_type_en` = ' . "\"" . $value . "\"" . ' OR `category`.`category_name` = ' .
                    . "\"" . $value . "\"" . ') ' . ($count == count($params) ? '' : ' AND ');
                }
                continue;
            } elseif ($key == 'price_min') {
                $price_min = $value;
                continue;
            } elseif ($key == 'price_max') {
                $price_max = $value;
                continue;
            }
            $sql .= '`products`.`product_` . $key . ` = ' . "\"" . $value . "\"" . ($count == count($params) ? '' : ' AND
        ');
    }
    $sql .= ($price_min == -1 ? '' : ' AND `products.product_price` > ' . $price_min) .
    ($price_max == -1 ? '' : ' AND `products.product_price` < ' . $price_max);
}
$result = $this->db->column($sql);
return $result;
}

function getBrands(){
    $result = $this->db->row("SELECT * FROM brands");
    return $result;
}

function getCategories(){
    $result = [];
    $shoes = $this->db->row("SELECT DISTINCT category_type_ua, category_type_en FROM `category` WHERE
category_name = 'shoes'");
    $clothes = $this->db->row("SELECT DISTINCT category_type_ua, category_type_en FROM `category`
WHERE category_name = 'clothes'");
    $accessories = $this->db->row("SELECT DISTINCT category_type_ua, category_type_en FROM `category`
WHERE category_name = 'accessories'");
    array_push($result, $shoes, $clothes, $accessories);
    return $result;
}

```

```

function getMinAndMaxPrices(){
    $result = $this->db->column("SELECT MIN(`product_price`) AS `min`, MAX(`product_price`) AS `max` FROM
products");
    return $result;
}
}

```

Додаток В

Лістинг вигляду сторінки продукту (product.php)

```

<div class="container mt-2">
    <ul class="nav mb-2">
        <li class="nav-item">
            <a class="nav-link text-dark fw-light" href="#"><?='< ' . $data['product']['product_gender']. ' ' .
$data['product']['product_category_type_ua'] . ' ' . $data['product']['product_brand']?></a>
        </li>
    </ul>
    <div class="row">
        <div class="col-1">
            <div class="img-small">
                
            </div>
            <?php if ($data['product']['product_images'] != null && $data['product']['product_images'] != ''):?>
            <?php
                $images = explode(' ', $data['product']['product_images']);
                foreach ($images as $item): ?>
                    <div class="img-small">
                        
                    </div>
                <?php endforeach; endif;?>
            </div>
        <div class="col-6">
            <div id="carouselExampleAutoplaying" class="carousel slide mb-2" data-bs-ride="carousel"
style="position: relative">
                
                <div class="carousel-inner">
                    <div class="carousel-item active">

```

```


    </div>
    <?php if ($data['product']['product_images'] != null && $data['product']['product_images'] != ""):?>
    <?php
    $images = explode(';', $data['product']['product_images']);
    foreach ($images as $item): ?>
        <div class="carousel-item">
            
            </div>
        <?php endforeach; endif;?>
    </div>
    <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleAutoplaying"
data-bs-slide="prev">
        <span class="carousel-control-prev-icon bg-dark" aria-hidden="true"></span>
        <span class="visually-hidden">Previous</span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleAutoplaying"
data-bs-slide="next">
        <span class="carousel-control-next-icon bg-dark" aria-hidden="true"></span>
        <span class="visually-hidden">Next</span>
    </button>
    <div class="text-dark" style="display: inline-block;position: absolute; right: 70px; top: 10px">

    <?php session_start();
    $is_liked = false;
    if (isset($_SESSION['user'])) {
        foreach (explode(';', substr($_SESSION['user']['user_liked_products'], 0, -1)) as $item) {
            if ($item == $data['product']['product_id'])
                $is_liked = true;
        }
    }
    ?>
    <a style="z-index: 10" href="https://fitwear/product/like/<?=$data['product']['product_id']?>"
onclick="function () {return false;}"
class="link-dark link-offset-2 link-underline link-underline-opacity-0">
    <?php if (!$is_liked):?>
        <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" fill="currentColor"
class="bi bi-heart" viewBox="0 0 16 16">
            <path d="m8 2.748-.717-.737C5.6.281 2.514.878 1.4 3.053c-.523 1.023-.641 2.5.314 4.385.92
1.815 2.834 3.989 6.286 6.357 3.452-2.368 5.365-4.542 6.286-6.357.955-1.886.838-3.362.314-

```

					ИЗ.КР.Б-121-24-ИЗ	Арк.
						72


```

4.385C13.486.878 10.4.28 8.717 2.01zM8 15C-7.333 4.868 3.279-3.04 7.824 1.143q.09.083.176.171a3 3 0 0 1
.176-.17C12.72-3.042 23.333 4.867 8 15"/>
    </svg>
    <?php else:?>
        <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" fill="currentColor"
class="bi bi-heart-fill" viewBox="0 0 16 16">
            <path fill-rule="evenodd" d="M8 1.314C12.438-3.248 23.534 4.735 8 15-7.534 4.736 3.562-
3.248 8 1.314"/>
        </svg>
    <?php endif; ?>
</a>
    <?=$data['product']['product_likes']?>
</div>
</div>
<div class="col-5">
    <h5 class="mb-4"><?=$data['product']['product_category_type_ua']. ' '.
$data['product']['product_brand']. ' '. $data['product']['product_name']?></h5>

    <?php if ($data['product']['product_discount'] != 0):?>
        <span class="fs-5 fw-bolder text-danger"><?=round($data['product']['product_price'] * (100 -
$data['product']['product_discount'])/100) . ' грн' ?></span>&nbsp;
        <span class="fs-5 text-muted text-decoration-line-through fw-light"><?=
$data['product']['product_price'] . ' грн' ?></span>
    <?php else:?>
        <span class="fs-5 fw-bolder"><?=$data['product']['product_price'] . " грн"?></span>
    <?php endif;?>
<br>

    <?php if ($data['product']['product_discount'] != 0):?>
        <span class="badge bg-danger mb-4"><?=$data['product']['product_discount']?>%</span>
    <?php endif;?>
    <?php if ($data['product']['product_new'] != 0):?>
        <span class="badge text-bg-light mb-4">NEW</span>
    <?php endif;?>
    <span class="badge text-bg-light mb-4"> <?=$data['product']['product_discount'] != 0 ? 'Краща ціна' :
'1000 бонусів за 1000 гривень'?></span>
    <span class="badge text-bg-light mb-4"><?=round($data['product']['product_price'] / 10) . " грн за
клубною картою"?></span>
<br>
    <form method="post" action="https://fitwear/basket/add/<?=$data['product']['product_id']?>">
        <?php
        $sizes = explode(' ', $data['product']['product_sizes']);
        foreach ($sizes as $size):
            ?>
            <input type="radio" class="btn-check" name="size" id="size<?=explode('-', $size)[0]?>"

```

					ПЗ.КР.Б-121-24-ПЗ

9.97 8.586 15.974 5.768 6.005-2.82 8.587-9.971 5.768-15.975A12.013 12.013 0 0027 11.027L27
9.98l1.127.057c.818.933 1.52 1.995 2.072 3.172 3.17 6.754.266 14.8-6.488 17.97-6.754 3.171-14.8.266-17.97-
6.488-3.17-6.754-.266-14.8 6.488-17.97a13.451 13.451 0 018.086-1.076c.859.152 1.7.386 2.511.699a.719.719
0 01.385.972.788.788 0 01-1 .398 12.08 12.08 0 00-2.157-.592 11.951 11.951 0 00-7.187.957z"></path><path
fill-rule="evenodd" clip-rule="evenodd" d="M25.688 8.828c0-.008.006-.013.013-.012l5.183.864a.75.75 0 11-
.247 1.48l-3.45-.576v3.427a.75.75 0 11-1.5 0v8.828zM14.743 14.011h-1.111a1.5 1.5 0 00-1.496 1.386l-.728
9.468a1.5 1.5 0 001.496 1.615h10.192a1.5 1.5 0 001.496-1.615l-.728-9.468a1.5 1.5 0 00-1.496-1.386h-1.111c-
.07-1.03-.293-1.955-.707-2.68-.537-.938-1.4-1.538-2.55-1.538s-2.014.6-2.55 1.538c-.414.725-.636 1.65-.707
2.68zm1.504 0h3.506c-.067-.834-.248-1.484-.506-1.936-.307-.538-.71-.782-1.247-.782-.538 0-.94.244-
1.247.782-.258.452-.439 1.102-.506 1.937zm-2.615 1.5l-.728 9.47h10.192l-.728-9.47h-8.736zm5.306
3.563a.75.75 0 010 1.5h-1.875a.75.75 0 010-1.5h1.875z"></path></svg>Гарантія
повернення
</p>
<p><svg viewBox="0 0 36 37" width="36px" height="36px" class="ZQ0ciz FaQMg4 "><path fill-
rule="evenodd" clip-rule="evenodd" d="M17.25 24.027c5.385 0 9.75-4.365 9.75-9.75 0-5.384-4.365-9.75-9.75-
9.75S7.5 8.893 7.5 14.277c0 5.385 4.365 9.75 9.75 9.75zm0 1.5c6.213 0 11.25-5.036 11.25-11.25 0-6.213-
5.037-11.25-11.25-11.25S6 8.064 6 14.277c0 6.214 5.037 11.25 11.25 11.25z"></path><path fill-
rule="evenodd" clip-rule="evenodd" d="M16.445 8.96a.9.9 0 011.61 0l1.201 2.404 3.062.612a.9.9 0 01.363
1.603l-2.353 1.765.59 2.95a.9.9 0 01-1.35.946l-2.318-1.412-2.318 1.412a.9.9 0 01-1.35-.945l.59-2.951-2.354-
1.765a.9.9 0 01.364-1.603l3.061-.612 1.202-2.404zm.805 1.744l-.797 1.593a.9.9 0 01-.628.48l-2.033.407 1.582
1.187a.9.9 0 01.343.896l-.396 1.98 1.46-.89a.9.9 0 01.937 0l1.46.89-.395-1.98a.9.9 0 01.342-.896l1.583-1.187-
2.033-.407a.9.9 0 01-.629-.48l-.796-1.593zM6.902 30.435l2.971-8.164 1.41.513-2.545 6.991 2.958-.999a.9.9 0
011.057.385l1.623 2.666 2.545-6.99 1.41.512-2.972 8.164a.9.9 0 01-1.614.16l-2.024-3.322-3.685 1.245a.9.9 0
01-1.134-1.161zM24.253 22.27l2.971 8.164a.9.9 0 01-1.134 1.16l-3.685-1.244-2.014 3.307a.9.9 0 01-1.627-
.198l-1.255-3.992a.75.75 0 011.432-.45l.857 2.73 1.575-2.587a.9.9 0 011.057-.385l2.958 1-2.545-6.992 1.41-
.513z"></path></svg>Оригінальні товари
</p>
<!-- Button trigger modal -->
<p data-bs-toggle="modal" data-bs-target="#exampleModal"><svg viewBox="0 0 24 24" width="36px"
height="36px" class="w02XiG RxRUPG"><path d="M12 23.6c-.3 0-.6-.3-.6-.6v-1c0-.3-.6-.6-.6s.6.6.6v1c0 .3-
.3-.6-.6zm8-3c-.2 0-.3-.1-.4-.2l-1-1c-.2-.2-.2-.6 0-.8.2-.2.6-.2.8 0l1 1c.2.2.6 0 .8-.1.1-.2-.4.2zm-16 0c-.2 0-.3-
.1-.4-.2-.2-.2-.6 0-.8l1-1c-.2-.2.6-.2.8 0.2.2.6 0 .8l1 1c.1.1-.2-.2-.4.2zm4.6-2c-.2 0-.5-.1-.6-.2-.4-.3-.6-.7-.5-
1.2l.5-3.3-2.3-2.4c-.3-.3-.4-.8-.3-1.2.2-.4.6-.7 1-.7l3.2-.5 1.4-3c.1-.2.2-.4.4-.5.4-.2.9-.2 1.2 0.2.1.3.3.4.5l1.4 3
3.2.5c.4.1.8.4.9.8.1.4 0 .9-.3 1.2l16 13.9l.5 3.3c.1.5-.1.9-.5 1.2-.4.3-.8.3-1.2.1l12 16.9l-2.8 1.6c-.2.1-.4.1-
.6.1zM12 6.7l-1.5 3.2c-.1.2-.3.3-.5.3l-3.5.5 2.6 2.5c.1.1.2.3.2.5l-.6 3.6 3-1.7c.2-.1.4-.1.6 0l3.1 1.7-.7-3.6c0-.2 0-
.4.2-.5l2.5-2.6-3.4-.4c-.2 0-.4-.2-.5-.3l12 6.7zm-2 2.9zm1.9-3zm11.1 6h-1c-.3 0-.6-.3-.6-.6s.3-.6.6-.6h1c.3 0
.6.3.6.6s-.3-.6-.6zm-21 0h1c-.3 0-.6-.3-.6-.6s.3-.6.6-.6h1c.3 0 .6.3.6.6s-.3-.6-.6zm17-7c-.2 0-.3-.1-.4-.2-.2-.2-.2-
.6 0-.8l1-1c-.2-.2.6-.2.8 0s.2.6 0 .8l1 1c-.1.1-.2.2-.4.2zm-14 0c-.2 0-.3-.1-.4-.2l-1-1c-.2-.2-.2-.6 0-.8s.6-.2.8 0l1
1c.2.2.6 0 .8-.1.1-.2.2-.4.2zm7-3c-.3 0-.6-.3-.6-.6v1c0-.3-.6-.6-.6s.6.6.6v1c0 .3-.6-.6.6z"></path></svg>
Опис та характеристики
</p>
<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel"
aria-hidden="true">
<div class="modal-dialog modal-xl">
<div class="modal-content">
<div class="modal-header">
<h1 class="modal-title fs-6" id="exampleModalLabel"><?=

```

$хdata['product']['product_category_type_ua']. ' '. $хdata['product']['product_brand']. ' '.
$хdata['product']['product_name']?></h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
    </div>
    <div class="modal-body">
        <div class="row">
            <div class="col">
                <span class="fs-5">Характеристики товару</span><br>
                <ul class="list-group list-group-flush">
                    <li class="list-group-item">Стать : <?=$хdata['product']['product_gender']?></li>
                    <li class="list-group-item">Артикул : <?=$хdata['product']['product_articul']?></li>
                    <li class="list-group-item">Знижки : <?=$хdata['product']['product_discount'] != 0 ?
'Краща ціна, ' : '1000 бонусів за 1000 гривень, '?>Знижка за Клубною картою, Безкоштовна доставка</li>
                    <li class="list-group-item">Сезон : <?=$хdata['product']['product_season']?></li>
                    <li class="list-group-item">Матеріал : <?=$хdata['product']['product_content']?></li>
                    <li class="list-group-item">Виробник : <?=$хdata['product']['product_maker']?></li>
                    <li class="list-group-item"><?=$хdata['product']['product_avaliability'] != 0 ? 'Товар в
наявності' : 'Немає в наявності'?></li>
                    <li class="list-group-item">Категорія :
<?=$хdata['product']['product_category_type_ua']?></li>
                    <li class="list-group-item">Бренд : <?=$хdata['product']['product_brand']?></li>
                    <?php if (!empty($хdata['collection'])):?>
                    <li class="list-group-item">Колекція : <?=$хdata['collection']['collection_name']?></li>
                    <?php endif; ?>
                    <li class="list-group-item">Колір : <?=$хdata['product']['product_color']?></li>
                </ul>
            </div>
            <div class="col">
                <?php if ($хdata['product']['product_description'] != null):?>
                <span class="fs-5">Опис товару</span><br>
                <span class="fw-light text-break"><?=$хdata['product']['product_description'] ?>
</span><br>
                <?php endif;?>
                <?php if (!empty($хdata['collection'])):?>
                <span class="fs-5">Технології</span><br>
                <span class="fw-light text-break"><?=$хdata['collection']['collection_description'] ?>
</span>
                <?php endif;?>
            </div>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>

```

```

        <ul class="list-group list-group-flush">
            <li class="list-group-item">Склад : <?=$data['product']['product_content']?></li>
            <li class="list-group-item">Сезон : <?=$data['product']['product_season']?></li>
            <li class="list-group-item">Артикул : <?=$data['product']['product_articul']?></li>
        </ul>
    </div>
</div>
<div class="row">
    <hr class="mb-4 mt-2">
    <h5>Вам може сподобатись</h5>
    <?php foreach ($data['rec_products'] as $Item): ?>
        <div class="col-3">
            <div class="card border-white" style="width: 18rem;">
                <a class="link-underline link-underline-opacity-0"
href="https://fitwear/product/index/<?=$Item['product_link']?>">
                    <div class="w-100" style="position: absolute">
                        <div style="display: inline-block;position: absolute; left: 10px; top: 10px">
                            <?php if ($Item['product_new'] != 0):?>
                                <div class="badge bg-light text-dark p-1">NEW</div><br>
                            <?php endif;?>
                            <?php if ($Item['product_discount'] != 0):?>
                                <div class="badge bg-danger text-light p-1"><?=$Item['product_discount']?>%</div>
                            <?php endif;?>
                        </div>
                        <div class="text-dark" style="display: inline-block;position: absolute; right: 10px; top: 10px">
                            <?php session_start();
                            $is_liked = false;
                            if (isset($_SESSION['user'])) {
                                foreach (explode(',', substr($_SESSION['user']['user_liked_products'], 0, -1)) as $item) {
                                    if ($item == $Item['product_id'])
                                        $is_liked = true;
                                }
                            }
                            ?>
                            <a style="z-index: 10" href="https://fitwear/product/like/<?=$Item['product_id']?>"
onclick="function () {return false;}"
                                class="link-dark link-offset-2 link-underline link-underline-opacity-0">
                                    <?php if (!$is_liked):?>
                                        <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" fill="currentColor"
class="bi bi-heart" viewBox="0 0 16 16">
                                            <path d="m8 2.748-.717-.737C5.6.281 2.514.878 1.4 3.053c-.523 1.023-.641 2.5.314
4.385.92 1.815 2.834 3.989 6.286 6.357 3.452-2.368 5.365-4.542 6.286-6.357.955-1.886.838-3.362.314-
4.385C13.486 8.78 10.428 8.717 2.01z"m8 15C8 14.373 2.625 14 0 14C0 14 0 8 8 8z"/>
                                        </svg>
                                    <?php else:?>

```


```

                <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24" fill="currentColor"
class="bi bi-heart-fill" viewBox="0 0 16 16">
                    <path fill-rule="evenodd" d="M8 1.314C12.438 3.248 23.534 4.735 8 15-7.534 4.736
3.562-3.248 8 1.314"/>
                </svg>
            <?php endif; ?>
        </a>
        <?=$Item['product_likes']?>
    </div>
</div>

</a>
<div class="card-body">
    <h6 class="card-title">
        <a class="link-underline link-underline-opacity-0 text-dark fw-normal"
href="product/index/<?=$Item['product_id']?>">
            <?=$Item['product_category_type_ua'] . ' ' . $Item['product_brand'] . ' ' .
$Item['product_name'] ?>
        </a>
    </h6>
    <?php
    $sizes = explode(' ', $Item['product_sizes']);
    foreach ($sizes as $size):
        ?>
        <span class="card-text border border-secondary" style="padding: 1px 3px; margin-bottom:
5px"><?=$Item['product_sizes']?></span>
    <?php endforeach;?>
    <p class="card-text d-flex">
        <?php if ($Item['product_discount'] != 0):?>
            <span class="fw-bolder"><?=$Item['product_price'] * (100 -
$Item['product_discount']/100) . ' грн' ?></span>&nbsp;
            <span class="text-muted text-decoration-line-through"><?=$Item['product_price'] . ' грн'
?></span>
        <?php else :?>
            <span class="fw-bolder"><?=$Item['product_price'] . ' грн' ?></span>
        <?php endif;?>
    </p>
</div>
</div>
</div>
<?php endforeach;?>
</div>
<div class="row">

```

```

<div class="col-md-6 offset-md-3">
  <div class="d-flex justify-content-center">
    <form method="post" class="bg-light border p-3" action="https://fitwear/product/add_comment/<?=$data['product']['product_id'] ?>">
      <h5>Залиште відгук</h5>
      <div class="row mb-3">
        <div class="col-sm-10">
          <input type="email" class="form-control" name="user_email" id="user_email"
placeholder="Електронна пошта*" value="<?=$SESSION['user'] ? $SESSION['user']['user_email'] :
"?>">
        </div>
      </div>
      <div class="row mb-3">
        <div class="col-sm-10">
          <input type="text" class="form-control" name="user_name" id="user_name"
placeholder="Ім'я*" value="<?=$SESSION['user'] ? $SESSION['user']['user_middlename'] : '?'>">
        </div>
      </div>
      <div class="row mb-3">
        <h6>Ваша оцінка</h6>

        <div class="col-sm-10 d-flex justify-content-center">
          <div class="form-check form-check-inline">
            <input type="radio" class="form-check-input star-ratio" name="rate" id="rate1" value="1">
            <label class="form-check-label" for="rate1">
              1
            </label>
          </div>
          <div class="form-check form-check-inline">
            <input type="radio" class="form-check-input star-ratio" name="rate" id="rate2" value="2">
            <label class="form-check-label" for="rate2">
              2
            </label>
          </div>
          <div class="form-check form-check-inline">
            <input type="radio" class="form-check-input star-ratio" name="rate" id="rate3" value="3">
            <label class="form-check-label" for="rate3">
              3
            </label>
          </div>
          <div class="form-check form-check-inline">
            <input type="radio" class="form-check-input star-ratio" name="rate" id="rate4" value="4">
            <label class="form-check-label" for="rate4">
              4
            </label>
          </div>
        </div>
      </div>
    </div>
  </div>

```


```

        <div class="form-check form-check-inline">
            <input type="radio" class="form-check-input star-ratio" name="rate" id="rate5" value="5">
            <label class="form-check-label" for="rate5">
                5
            </label>
        </div>
    </div>
</div>

<div class="row mb-3">
    <div>
        <textarea class="form-control" id="exampleFormControlTextarea1" name="description"
placeholder="Ваші враження від товару" rows="3"></textarea>
    </div>
</div>
<button type="submit" class="btn btn-primary">Надіслати відгук</button>
<br>
<span style="font-size: 10px" class="text-muted">Відгук залишиться на сайті після модерації.
Дякую за співпрацю</span>
    </form>
</div>
</div>
</div>

```