

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 спорттоварів»

Виконав студент групи ПЗк-20-1
ЯНКО Володимир Олегович

Керівник роботи:
ЧИЖМОТРЯ Олена Геннадіївна

Рецензент:
ВЛАСЕНКО Олег Васильович

Житомир – 2023

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

«ЗАТВЕРДЖУЮ»

В.о. зав. кафедри інженерії
програмного забезпечення

Андрій МОРОЗОВ

«14» лютого 2023 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: ЯНКО Володимир Олегович

Керівник роботи: ЧИЖМОТРЕ Олена Геннадіївна

Тема роботи: «Інтернет-магазин спорттоварів»

затверджена Наказом закладу вищої освіти від «14» лютого 2023 р., №70с

Вихідні дані для роботи: розробка, впровадження та функціонування інтернет-магазину спорттоварів.

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря О.Г.	16.02.2023р.	16.02.2023р.
2	Чижмотря О.Г.	16.02.2023р.	16.02.2023р.
3	Чижмотря О.Г.	16.02.2023р.	16.02.2023р.

Календарний план

№	Назва етапів випускної кваліфікаційної роботи	Термін виконання роботи	Примітка
1	Постановка задачі	16 лютого 2023 – 20 лютого 2023	Виконано
2	Аналіз існуючих розробок	21 лютого 2023 – 24 лютого 2023	Виконано
3	Визначення вимог до додатку	27 лютого 2023 – 8 березня 2023	Виконано
4	Обирання технологій для розробки веб-проекту	9 березня 2023 – 11 березня 2023	Виконано
5	Проектування та реалізація алгоритмів системи	13 березня 2023 – 17 березня 2023	Виконано
6	Розробка архітектури веб-додатка	18 березня 2023 – 24 березня 2023	Виконано
7	Розробка об'єктно-орієнтованої моделі	26 березня 2023 – 2 квітня 2023	Виконано
8	Написання програмного коду	4 квітня 2023 – 9 травня 2023	Виконано
9	Тестування роботи програмного продукту	11 травня 2023 – 17 травня 2023	Виконано
10	Оформлення пояснювальної записки	19 травня 2023 – 27 травня 2023	Виконано
11	Попередні захист	5 травня 2023	Виконано

Реферат

Пояснювальна записка до кваліфікаційної роботи освітнього ступеня «бакалавр» на тему: «Інтернет-магазин спорттоварів» складається з вступу, трьох розділів, висновків, джерел інформації та додатків.

Пояснювальна записка до кваліфікаційної роботи містить 76 сторінок, 46 ілюстрацій, 19 таблиць, додаток та 15 літературних джерел.

Метою розробки кваліфікаційної розробки є автоматизація та організація продажу товарів спортивного призначення через мережу «Інтернет».

У першому розділі буде проаналізовано напрямки використання технологій, аналоги програмного продукту та обґрунтовано вибір інструментальних засобів та описано вимоги до програмного продукту.

У другому розділі буде описано попереднє проектування системи, розроблено схему бази даних, та продемонстровано безпосереднє проектування програмного продукту.

У третьому розділі буде продемонстровано візуальний інтерфейс програмного продукту та буде проведено тестування системи.

Висновок міститиме підсумки до кожного розділу та до кваліфікаційної роботи загалом.

У додатках представлений акт впровадження та лістинг розробленого програмного продукту.

КЛЮЧОВІ СЛОВА: ОНЛАЙН МАГАЗИН, АДМІН ПАНЕЛЬ, ВЕБ ДОДАТОК

					ІПЗ.КР.Б – 121 – 23 – ПЗ			
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>	Інтернет-магазин спорттоварів	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Розроб.</i>		<i>В. О. Янко</i>						
<i>Керівник</i>		<i>О.Г. Чижмотря</i>					4	76
<i>Рецензент</i>		<i>О. В. Власенко</i>				<i>ФІКТ Гр. ІПЗк-20-1</i>		
<i>Зав. каф.</i>		<i>А.В. Морозов</i>						

ABSTRACT

The explanatory note to the qualification work of the bachelor's degree on the topic: "Online sporting goods store" consists of an introduction, three chapters, conclusions, sources of information and appendices.

The explanatory note to the qualification work contains 76 pages, 46 illustrations, 19 tables, an appendix and 15 literary sources.

The purpose of the qualification work is to automate and organize the sale of sports goods via the Internet.

The first section will analyze the areas of technology use, analogues of the software product, justify the choice of tools and describe the requirements for the software product.

The second section will describe the preliminary design of the system, develop a database schema, and demonstrate the direct design of the software product.

In the third section, the visual interface of the software product will be demonstrated and the system will be tested.

The conclusion will contain summaries for each chapter and for the qualification work as a whole.

The appendices contain the implementation act and the listing of the developed software product.

KEYWORDS: ONLINE STORE, ADMIN PANEL, WEB APPLICATION

					ІІЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Зміст

Перелік умовних позначень	7
Вступ.....	8
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	11
1.1 Постановка задачі.	11
1.2 Аналіз аналогів програмного продукту	12
1.3 Визначення ключових термінів	16
1.4 Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення	17
Висновки до розділу 1	22
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ	23
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи	23
2.2 Розробка бази даних системи.....	27
2.3 Проектування та реалізація алгоритмів роботи системи	34
2.4 Реалізація програмного продукту	35
Висновки до розділу 2	39
РОЗДІЛ 3. ПОРЯДОК РОБОТИ З ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ТА ІНТЕРФЕЙС	40
3.1 Порядок встановлення та налаштування параметрів системи	40
3.2 Структура інтерфейсу.....	40
3.3 Тестування роботи програмного комплексу	55
Висновки до розділу 3	56
Висновки	57
Список використаних джерел	58
Додатки.....	60

Перелік умовних позначень

CSS – Cascading Style Sheets (каскадні таблиці стилів).

HTML – HyperText Markup Language (стандартизована мова розмітки документів в мережі).

HTTP – HyperText Transfer Porotocol (протокол передачі гіпертекстових документів).

URL – Uniform Resource Locator (уніфікований локатор ресурсів).

PHP – Hypertext Processor (це мова сценаріїв загального призначення та інтерпретатор, який є у вільному доступі та широко використовується для веб-розробки).

БД – база даних.

JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування

MVC – Model-view-controller (це шаблон проектування програмного забезпечення, який зазвичай використовується для розробки користувальницьких інтерфейсів, який розділяє відповідну програмну логіку на три взаємопов'язані елементи. Це робиться для того, щоб відокремити внутрішнє представлення інформації від того, як інформація подається та приймається)

Laravel – безкоштовний, з відкритим кодом PHP-фреймворк і призначений для розробки вебдодатків відповідно до шаблону model–view–controller (MVC)

					ІПЗ.КР.Б – 121 – 23 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Вступ

Об'єкт та предмет дослідження.

У сучасному цифровому світі, де доступ до Інтернету став загальною нормою, споживачі віддають перевагу зручності та широкому вибору товарів, які можуть придбати в онлайн-режимі. Інтернет-магазини виконують роль віртуальних вітрин, що дозволяє клієнтам замовляти товари з будь-якого місця та в будь-який час. Це відкриває безліч можливостей для бізнесу, забезпечуючи глобальну присутність і підвищуючи конкурентоспроможність підприємства.

Обґрунтування актуальності обраної теми, її практичне значення.

Сучасна споживацька поведінка все більше зміщується в напрямку електронної комерції, більшості потенційним клієнтам комфортніше не йти до магазину кожного разу, а в будь-який момент часу та з будь-якого місця придбати товар який його цікавить і спортивні товари не є винятком. Отже, розробка та впровадження інтернет-магазину спорттоварів має велике практичне значення, особливо звертаючи увагу на зростання популярності онлайн-покупок, можливість виходу на світовий ринок, зниження витрат внаслідок діджиталізації, покращення персоналізації та аналітики.

Мета і завдання роботи.

Метою роботи є розробка, реалізація та впровадження функціонального, естетичного та користувачем-орієнтованого веб-інтерфейсу інтернет-магазину, що задовольнятиме потреби та очікування клієнтів, сприятиме збільшенню продажів спортивних товарів та забезпечить успішну комерційну діяльність компанії.

Загальна мета полягає у створенні цілісного та ефективного онлайн-простору, де клієнти зможуть з легкістю переглядати, вибирати та придбавати спортивні товари, а компанія зможе пропонувати якісні продукти, збільшувати клієнтську базу та збут, та підтримувати довгострокові стосунки з клієнтами.

					ІПЗ.КР.Б – 121 – 23 – ІПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

Завдання роботи.

Головним завданням кваліфікаційної роботи є створення інтернет застосунку або веб-додатку, що буде застосовуватись будь-яким магазином спорттоварів для оформлення замовлень.

Також необхідно проаналізувати ринок та конкурентне середовище для з'ясування потенційних можливостей та визначення основних конкурентних переваг, спроектувати та розробити функціональний веб-інтерфейс, який забезпечуватиме зручне переглядання та вибір спортивних товарів, зручне оформлення замовлення, підтримку різних платіжних систем та забезпечення безпеки даних клієнтів.

Методи аналізу та інформаційна база дослідження.

Методи аналізу ефективності використання інтернет-магазину спорттоварів включають:

- Аналіз вимог клієнтів, щоб з'ясувати їх очікування, потреби та преференції щодо функціональності, дизайну та інших аспектів інтернет-магазину спорттоварів.
- Дослідження та аналіз існуючих Інтернет-магазинів, які спеціалізуються на спорттоварах, для виявлення їхніх сильних та слабких сторінок, а також для здобуття ідей та визначення можливих конкурентних переваг.
- Огляд технічних рішень та платформ, що використовуються для розробки інтернет-магазинів. Для реалізації максимально зручного програмного продукту потрібно вивчити технологічні можливості, які доступні для розробки на впровадження такого рішення.
- Аналіз трендів та вивчення актуальних трендів у сфері електронної комерції та спортивних товарів, а також аналіз ринкових досліджень, що стосуються споживчих преференцій, покупного поведінки та перспектив розвитку ринку спорттоварів.

Щодо інформаційної бази дослідження, можна використовувати наступні джерела:

- Наукові статті, журнали та книги з електронної комерції, дизайну інтерфейсу, маркетингу та інших відповідних тем.
- Офіційні документи, стандарти та рекомендації щодо розробки Інтернет-магазинів та електронної комерції.
- Дослідження та звіти ринкових аналітичних компаній, що присвячені електронній комерції та ринку спорттоварів.
- Інформація з відкритих джерел, таких як спеціалізовані веб-сайти, форуми, блоги та соціальні медіа, де експерти та користувачі обговорюють та діляться своїми досвідами щодо розробки та експлуатації Інтернет-магазинів.

Практичне значення отриманих результатів.

Розроблений програмний продукт використовується для реалізації діджиталізації процесу продажу спорттоварів:

- Забезпечення зручного доступу до каталогу товарів;
- Комфортність та привабливість для клієнта при виборі способу покупки спорттоварів;
- Спрощення процесу каталогом;
- Зручність та ефективність.

					ІПЗ.КР.Б – 121 – 23 – ІПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1. Постановка задачі

Опис проблеми: З огляду на постійне зростання популярності онлайн-торгівлі та зміну споживацьких звичок, існує великий попит на інтернет-магазини спорттоварів. Це створює потребу в розробці зручної та функціональної платформи, яка забезпечує клієнтам доступ до широкого асортименту спортивного обладнання, одягу та аксесуарів з можливістю зручного оформлення та оплати замовлень.

Метою кваліфікаційної роботи є розробка функціонального інтернет-магазину спорттоварів з можливістю перегляду та додавання товарів до кошика, оформлення замовлень та оплати через інтернет. Для досягнення цієї мети необхідно виконати наступні завдання:

- Розробити дизайн та структуру сайту, використовуючи сучасні технології веб-розробки. Дизайн сайту має бути зрозумілим та зручним для користувачів, з врахуванням особливостей спортивних товарів.
- Розробити функціональні модулі для роботи з товарами, кошиком, замовленнями та платежами. Модуль для роботи з товарами повинен містити інформацію про кожен товар, включаючи назву, опис, ціну, фото тощо. Модуль для кошика повинен дозволяти додавати товари до кошика, видаляти їх та редагувати кількість товарів. Модуль для замовлень повинен дозволяти користувачам оформляти замовлення та зберігати історію замовлень. Модуль для платежів повинен дозволяти користувачам безпечно оплачувати замовлення через інтернет.
- Використовуючи засоби програмування та бази даних, реалізувати функціональні модулі та забезпечити взаємодію між ними. Необхідно забезпечити швидкий доступ до бази даних, що містить інформацію про товари, замовлення та інші дані.

- Дослідити питання захисту сайту від зловмисників та забезпечення його безпеки для клієнтів та адміністраторів.

1.2. Аналіз аналогів програмного продукту

Зараз на ринку досить багато інтернет-магазинів спортивних товарів. Найбільш відомі з них – Sport Life, Decathlon, Nike, Adidas, Puma та інші. Усі вони пропонують широкий асортимент спортивного спорядження, від одягу та взуття до спортивних аксесуарів та приладдя.

Adidas.com є одним з провідних інтернет-магазинів спорттоварів у світі. Сайт Adidas.com пропонує широкий асортимент спортивного взуття, одягу та аксесуарів для різних видів спорту.

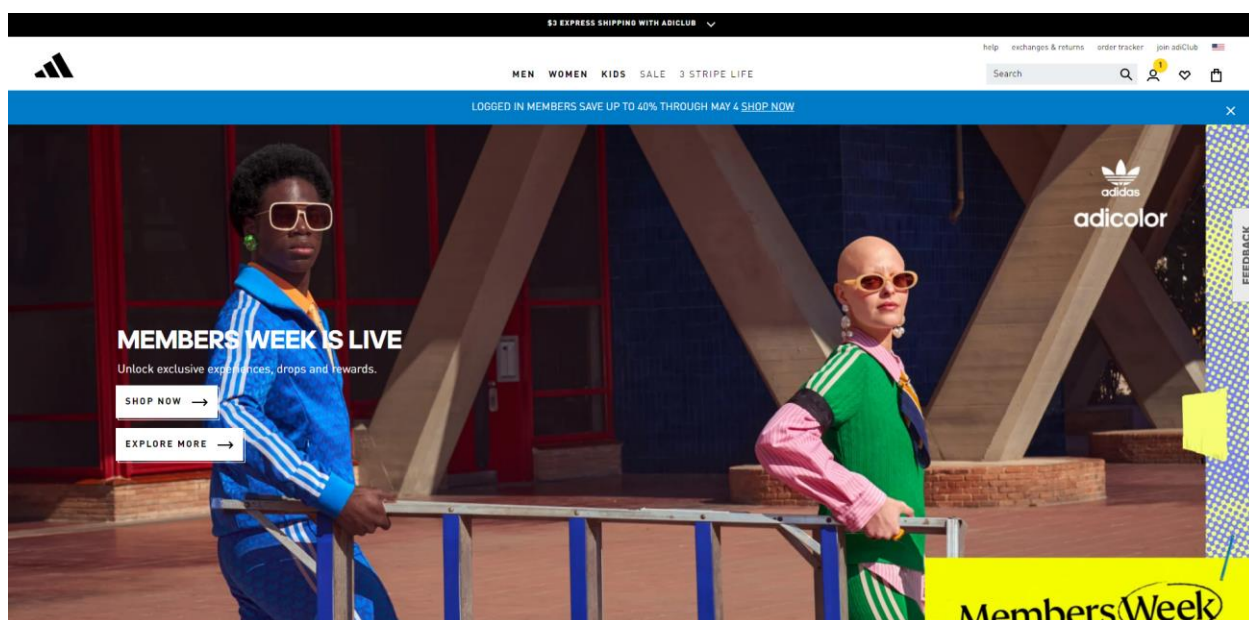


Рис. 1.1 – Головна сторінка сайту Adidas

Нижче наведено аналіз деяких аспектів цього інтернет-магазину:

Плюси:

- Широкий асортимент продукції: Adidas.com пропонує великий вибір спортивного взуття, одягу та аксесуарів для різних видів спорту, що дозволяє задовольнити потреби різних груп споживачів.

					ІПЗ.КР.Б – 121 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

- Зручний процес оплати: Сайт пропонує різні способи оплати, такі як кредитні картки, PayPal тощо.

СТАТУС ЗАМОВЛЕННЯ



Кросівки OZWEEGO
КОЛІР: Cloud White / Cloud White / Core Black
РОЗМІР: 7.5 UK / К-сть: 1
4 999 грн.

ПОТРІБНА ДОПОМОГА?

[Доставка](#)

[Повернення](#)

[Контакти](#)

МЕТОДИ ОПЛАТИ

VISA



Рис. 1.4 – Способи оплати на сайті Adidas

Мінуси:

- Висока ціна: Продукція Adidas має високу ціну, що може здатися недоступною для деяких груп споживачів.
- Обмежений вибір розмірів: Деякі розміри товарів можуть бути обмежені або недоступні.
- Довгий час доставки: Може знадобитися довгий час для доставки товарів у деякі регіони.

Узагальнюючи, Adidas.com є популярним інтернет-магазином спорттоварів, який пропонує широкий вибір якісної продукції. Однак, висока ціна та обмежений вибір розмірів можуть стати проблемою для деяких споживачів.

					ІПЗ.КР.Б – 121 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

На даний момент український інтернет-магазин спорттоварів sportlife.ua є одним з провідних на українському ринку спортивних товарів. Особливості інтернет-магазину sportlife.ua:

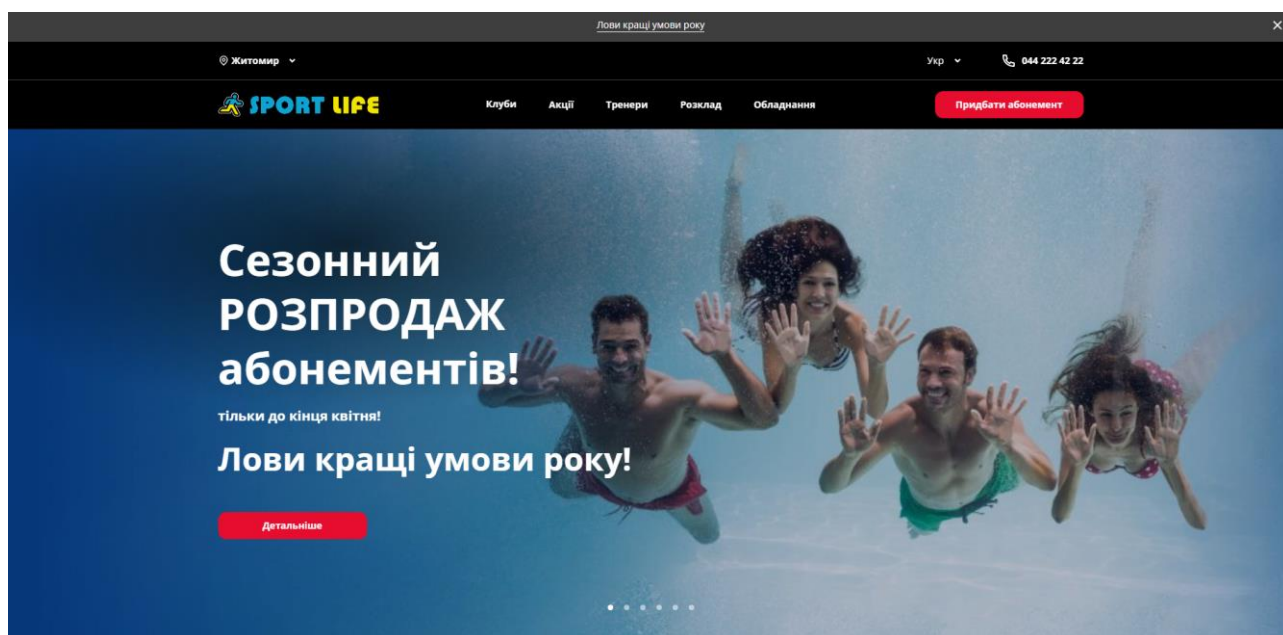


Рис. 1.5 – Головна сторінка магазину Sport Life

- Асортимент товарів: на сайті sportlife.ua представлено велике розмаїття товарів для спорту та активного відпочинку. В асортименті є спортивне взуття, одяг, аксесуари, обладнання для фітнесу, бодібілдингу, бігу, тенісу та інших видів спорту.

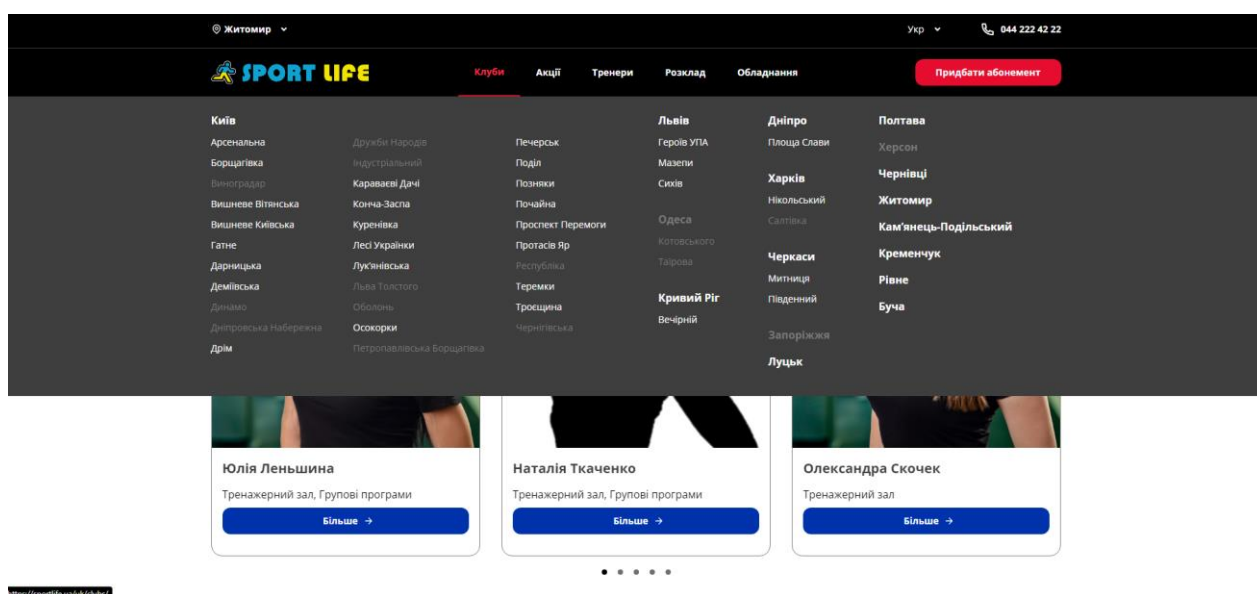


Рис. 1.6 – Асортимент товару Sport Life

- **Ціни:** ціни на сайті sportlife.ua конкурентні, зазвичай знаходяться в середньому ціновому діапазоні порівняно з іншими українськими інтернет-магазинами спорттоварів.

- **Дизайн та навігація сайту:** сайт sportlife.ua має сучасний та зручний дизайн, що дозволяє легко знайти необхідний товар та швидко здійснити покупку. Навігація по сайту досить інтуїтивна.

- **Доставка та оплата:** sportlife.ua пропонує декілька способів доставки замовлення (кур'єрська служба, самовивіз з магазинів, поштова доставка) та різні способи оплати (готівкою, безготівковий розрахунок, онлайн-оплата).

- **Програма лояльності:** sportlife.ua має свою програму лояльності, що дозволяє клієнтам збирати бонуси за кожну покупку та обмінювати їх на знижки на майбутні покупки.

- **Комунікація з клієнтами:** на сайті sportlife.ua є можливість звернення до служби підтримки клієнтів через телефон, електронну пошту та онлайн-чат.

1.3. Визначення ключових термінів

Інтернет-магазин - електронний комерційний сайт, який дозволяє користувачам придбати товари чи послуги через Інтернет.

Спорттовари - предмети одягу, взуття, спортивне обладнання, аксесуари, засоби здоров'я та інші товари, які використовуються для занять спортом.

Кошик - частина інтернет-магазину, де користувач може зберігати товари перед оформленням замовлення.

Оформлення замовлення - процес заповнення необхідних даних покупця для придбання товарів в інтернет-магазині.

Оплата через Інтернет - процес сплати за товари чи послуги в інтернет-магазині з використанням електронних платіжних систем.

Адміністратор - особа, яка керує та забезпечує роботу інтернет-магазину, додає товари до каталогу, відстежує замовлення та забезпечує зв'язок з клієнтами.

					ІПЗ.КР.Б – 121 – 23 – ІПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Landing page - веб-сторінка, створена з метою отримання максимальної конверсії (наприклад, реєстрації, продажу тощо).

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для розробки інтернет-магазину спорттоварів було обрано наступні інструменти:

- **HTML (HyperText Markup Language):** HTML є основною мовою розмітки для створення структури веб-сторінок. Використання HTML дозволяє створити зручний та легкозрозумілий код для розміщення контенту, зображень, посилань і форм на сторінці [5].

Плюси мови HTML:

- Простота в освоєнні: HTML має простий синтаксис та легкі правила розмітки. Це робить його доступним для початківців у веб-розробці.
- Універсальність: HTML є стандартом для веб-розробки та підтримується всіма сучасними веб-браузерами. Він дозволяє створювати веб-сторінки, що працюють на різних платформах та пристроях [7].
- Інтеграція з іншими технологіями: HTML підтримує інтеграцію з CSS для стилізації сторінок та з JavaScript для реалізації динамічної поведінки.

- **CSS (Cascading Style Sheets):** CSS використовується для задання вигляду та стилів веб-сторінок. За допомогою CSS можна налаштувати колір, шрифти, розташування елементів і створити привабливий дизайн для інтернет-магазину спорттоварів [8].

Плюси мови CSS:

- Гнучкість та контроль: CSS надає широкий спектр можливостей для налаштування вигляду сторінки, включаючи кольори, шрифти, розташування, розміри, анімацію та інші ефекти. Ви можете точно контролювати вигляд і розташування елементів на сторінці [13].

– Розділення стилів від структури: CSS дозволяє відокремити стиль та оформлення веб-сторінки від її структури. Це дозволяє легко змінювати вигляд сторінки без впливу на її зміст, а також полегшує підтримку та оновлення веб-сайту [4].

– Можливість створювати адаптивні та респонсивні веб-сторінки, які коректно відображатимуться на різних пристроях та розмірах екранів. Це особливо важливо для магазину спорттоварів, оскільки користувачі можуть переглядати його на різних пристроях.

- PHP (Hypertext Preprocessor): PHP є мовою програмування на стороні сервера, яка дозволяє взаємодіяти з базою даних, обробляти форми, генерувати динамічний контент і керувати логікою програми. Використання PHP дозволить створити функціональність інтернет-магазину, таку як реєстрація користувачів, авторизація, додавання товарів до кошика, обробка замовлень і т. д. [12]

Плюси мови PHP:

– Широке поширення: PHP є однією з найпоширеніших мов програмування для розробки веб-додатків. Велика кількість хостинг-провайдерів та серверів підтримують PHP, що робить його доступним для розгортання та використання на більшості веб-сайтів [15].

– Легкість в освоєнні: PHP має простий синтаксис, схожий на синтаксис мови C, що полегшує вивчення і розробку. Вона також має велику спільноту розробників, яка надає документацію, підручники та підтримку.

– Розширюваність: PHP має велику кількість розширень і бібліотек, що дозволяють розробникам легко розширювати функціональність своїх додатків. Існують готові рішення та фреймворки, які прискорюють розробку та забезпечують готові модулі для часто використовуваних завдань [14].

- JavaScript: JavaScript є мовою програмування на стороні клієнта, яка додає інтерактивність до веб-сторінок. Використання JavaScript дозволить створити анімацію, валідацію форм, перехід між сторінками без

перезавантаження та інші динамічні ефекти для поліпшення користувацького досвіду [3].

Плюси мови JavaScript

- Клієнтська взаємодія: JavaScript дозволяє реалізувати взаємодію з користувачем на веб-сторінці. Він забезпечує динамічну поведінку, можливість валідації форм, анімацію, обробку подій та багато іншого. JavaScript дозволяє створювати багатофункціональні та інтерактивні веб-додатки.
- Широке застосування: JavaScript може використовуватись як на клієнтській стороні, так і на сервері. Це дозволяє розробникам створювати повноцінні веб-додатки, які взаємодіють з сервером без перезавантаження сторінки (за допомогою AJAX технології) [2].
- Велика спільнота та екосистема: JavaScript має одну з найбільших спільнот розробників у світі програмування. Це означає, що є безліч ресурсів, бібліотек, фреймворків та інструментів, які полегшують розробку і допомагають розвивати якісні веб-додатки [10].
- Незалежність від платформи: JavaScript може виконуватися на будь-якому сучасному веб-браузері. Це дозволяє створювати кросс-платформені веб-додатки, які працюють на різних пристроях і операційних системах.
- MySQL: MySQL є реляційною базою даних, яка забезпечує зберігання та керування даними для інтернет-магазину. Використання MySQL дозволить створити базу даних для зберігання інформації про товари, замовлення, користувачів та інші дані, необхідні для роботи магазину [6].

Плюси системи MySQL:

- Простота використання: MySQL має простий синтаксис та інтуїтивно зрозумілий інтерфейс. Це дозволяє розробникам легко вивчити та використовувати мову для створення, збереження, оновлення та вибору даних [9].

- Швидкодія: MySQL є дуже швидкодіючою СУБД, що забезпечує ефективну обробку та виконання запитів до бази даних. Вона може ефективно працювати з великим обсягом даних та високою навантаженістю.
- Масштабованість: MySQL може легко масштабуватися від невеликих проєктів до великих підприємств. Вона підтримує розподілені бази даних та реплікацію, що дозволяє розробникам створювати високодоступні та надійні системи.
- Велика спільнота та підтримка: MySQL має широку спільноту розробників, що надає підтримку, документацію, підручники та інші ресурси. Це дозволяє швидко розв'язувати проблеми та отримувати поради від експертів [\[11\]](#).
- Laravel: Фреймворк Laravel є одним з найпопулярніших інструментів для розробки веб-додатків на основі мови програмування PHP:
 - Зручна робота з базою даних: Laravel надає елегантну ORM (Object-Relational Mapping) для роботи з базою даних. Це спрощує взаємодію з базою даних і дозволяє легко виконувати операції створення, читання, оновлення та видалення даних.
 - Зручна робота з базою даних: Laravel надає елегантну ORM (Object-Relational Mapping) для роботи з базою даних. Це спрощує взаємодію з базою даних і дозволяє легко виконувати операції створення, читання, оновлення та видалення даних.
 - Масштабованість: Laravel забезпечує гнучкість та масштабованість для розвитку веб-додатків. Він підтримує розширення функціональності за допомогою додаткових пакетів та може працювати з великими проєктами, забезпечуючи зручну організацію та структуру коду.
 - Безпека: Laravel має вбудовані механізми для забезпечення безпеки веб-додатків. Він надає можливості для захисту від атак XSS (міжсайтовий скриптинг), CSRF (міжсайтова подія) та інших потенційних загроз безпеці.

При виборі інструментальних засобів та вимог до апаратного забезпечення для програмного продукту "Інтернет-магазин спорттоварів" необхідно враховувати наступні фактори:

- **Функціональні вимоги.** Забезпечити необхідний функціонал для користувачів, такий як пошук товарів за назвою або за спеціальними фільтрами, додавання товарів до кошика, оформлення замовлення та оплата товарів.

- **Технічні вимоги.** При розробці необхідно врахувати технічні вимоги системи такі як продуктивність, зручність та безпека [\[1\]](#). Для коректної роботи сайту необхідні:

- Веб-сервер: Для запуску інтернет-магазину потрібен веб-сервер, який підтримує виконання PHP-скриптів і взаємодію з базою даних MySQL.
- База даних: Для зберігання даних про товари, замовлення та користувачів потрібна база даних MySQL. Необхідно мати сервер, на якому буде встановлена MySQL, а також достатньо місця для зберігання даних.
- Ресурси сервера: Залежно від очікуваної кількості відвідувачів та обсягу даних, може знадобитися достатньо потужний сервер з достатньою кількістю оперативної пам'яті і процесорної потужності для забезпечення швидкої та надійної роботи інтернет-магазину.
- Мережеве з'єднання: Для доступу до інтернет-магазину потрібне стабільне та швидке Інтернет-з'єднання. Це забезпечить користувачам зручний доступ до магазину і швидке завантаження сторінок.

Враховуючи переваги вказаних вище технологій вони були обрані мною для розробки інтернет-магазину спорттоварів.

Для написання та заповнення веб-сторінки контентом буде використовуватись мова HTML.

За допомогою CSS буде оформлено та визначено стилі веб-сторінки.

Реалізація динамічних ефектів та інтерактивності на сторінці буде реалізовано за допомогою JavaScript.

З допомогою мови програмування PHP буде реалізовано Бекенд або серверну частину.

Бази даних та зв'язки між ними будуть створені та реалізовані за допомогою системи управління базами даних – MySQL.

Висновки до розділу 1

У ході виконання першого розділу було проаналізовано аналоги програмного продукту та зроблено порівняння функціональних можливостей. Було створено та проаналізовано постановку задачі, визначено ключові терміни системи та обґрунтовано вибір інструментальних засобів системи та вимоги до апаратного забезпечення.

					ІПЗ.КР.Б – 121 – 23 – ІПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

При розробці програмних продуктів одним з важливих аспектів є правильне розуміння функціональних можливостей різних об'єктів системи. Одним із найкращих інструментів для їх відображення є діаграми UML.

Для такого відображення найкраще підходять діаграми варіантів використання UML.

UML-діаграма варіанту використання використовується для визначення основних функцій системи, визначення учасників і графічного відображення зв'язків між ними. Це дозволяє розробникам краще зрозуміти системні вимоги та взаємодію між її компонентами за допомогою графічних зображень.

UML діаграма варіантів використання складається з декількох елементів. Основні елементи UML-діаграми варіантів використання:

- **Актори.** Зазвичай акторами називають суб'єкти, наприклад користувачі, системи або сервіси, які взаємодіють з системою.
- **Варіанти використання.** Простими словами це функціонал актора. Вони описують можливості акторів або сценарії їх використання.
- **Взаємодія.** Лінії асоціації між акторами та варіантами використання, що відображають взаємодію між ними. Існує декілька видів взаємодій.

Типи взаємодій або відношення між акторами та варіантами використання:

- **Асоціація (association)** – це конкретна роль, яку грає актор при взаємодії з варіантом використання;
- **Розширення (extend)** – визначає зв'язок екземплярів одного варіантів використання з більш загальним;
- **Узагальнення (generalization)** – вказує що варіант використання А може бути узагальненим до варіанту використання В; А – є спеціалізацією або нащадком, В- прашур;

- **Включення (include)** – вказує що деяка поведінка варіанту використання являється складовою поведінки іншого варіанту використання.

Позначення компонентів та відношень діаграми варіантів використання можна побачити на рис. 2.1.

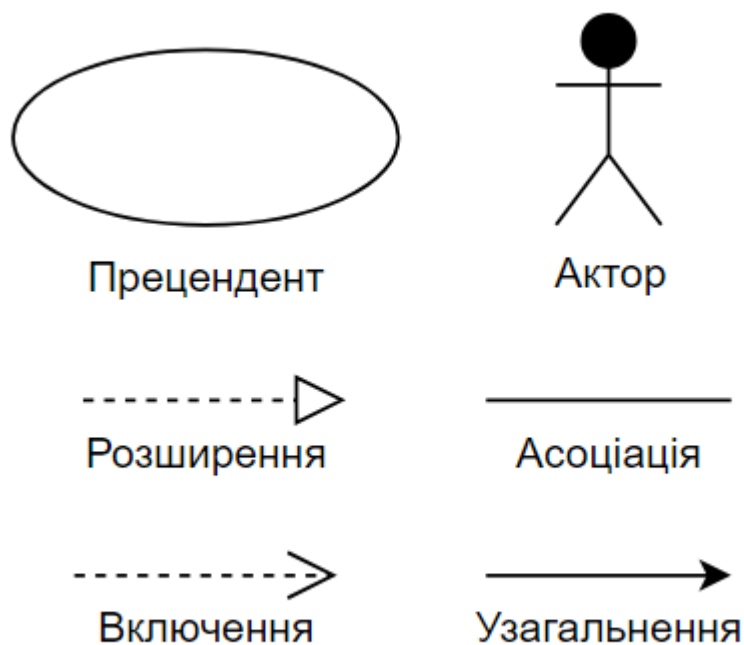


Рис. 2.1 – Компоненти та відношення діаграми використання

Згідно поставленого завдання було побудовано діаграму використання клієнта і відображено результат на рис. 2.2.

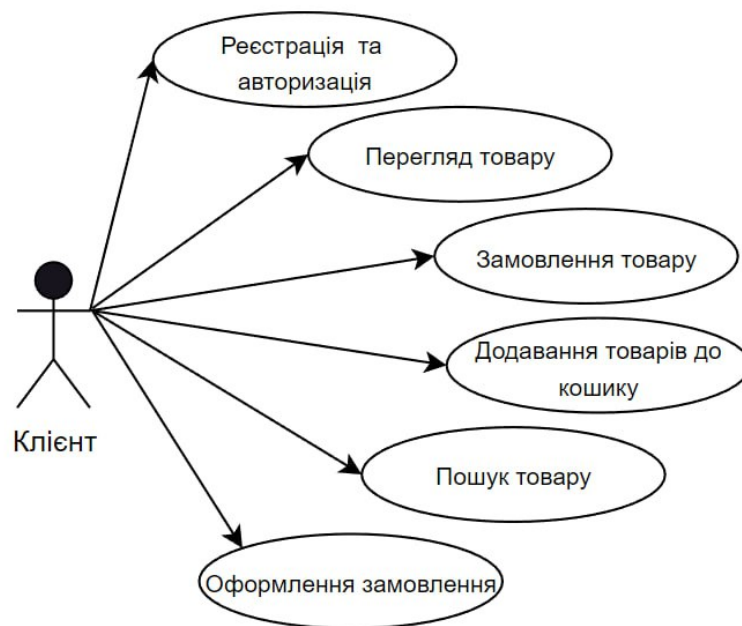


Рис. 2.2 – Діаграма варіантів використання для клієнта

Зображена діаграма представляє основні функції клієнта:

- Реєстрація та авторизація;
- Перегляд товару;
- Замовлення товару;
- Додавання товарів до кошику;
- Пошук товару;
- Оформлення замовлення.

Другим актором інтернет-магазину спорттоварів виступає адміністратор. Адміністратор відповідає за управління та заповнення категорій, підкатегорій, керування товаром, а також параметрами. Відповідна діаграма варіантів використання зображена на рис. 2.3.

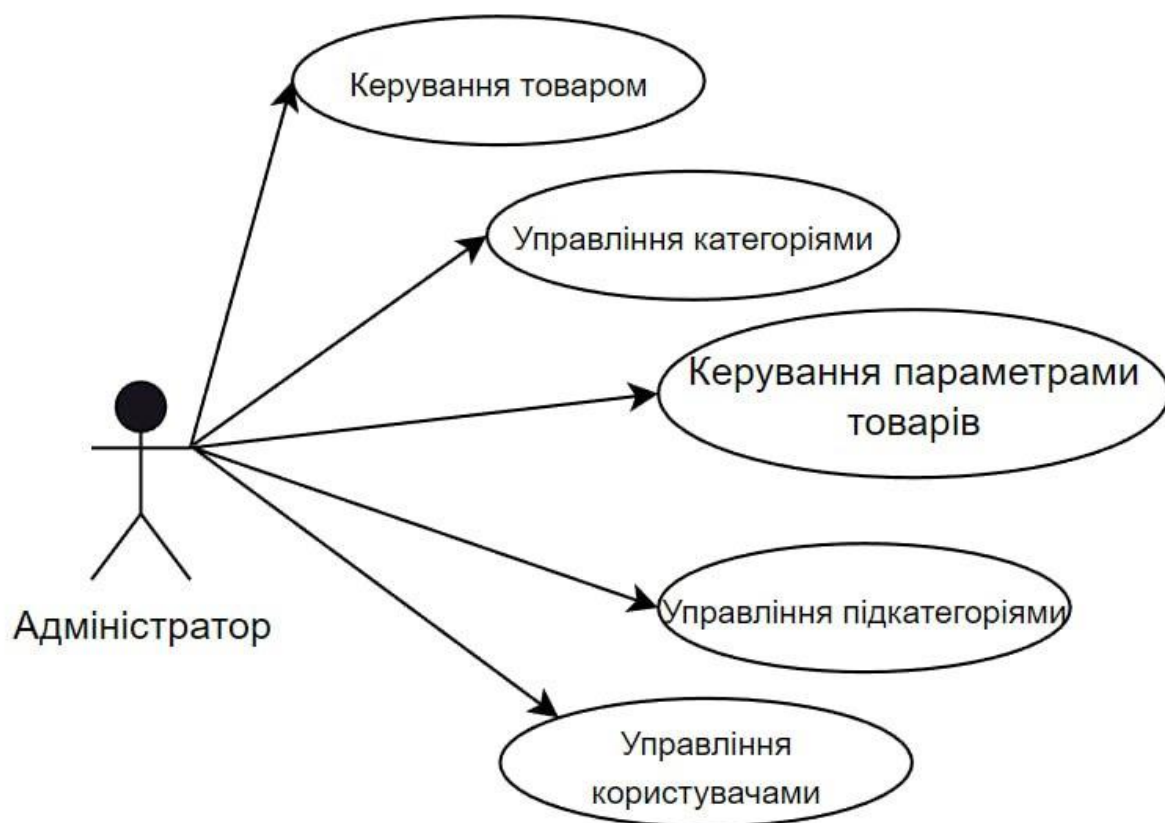


Рис. 2.3 – Діаграма варіантів використання адміністратора
Зображена діаграма відображає основні функції адміністратора:

- Керування товаром;
- Управління категоріями;
- Керування параметрами товарів;
- Управління підкатегоріями;
- Управління користувачами;

2.2. Розробка бази даних системи

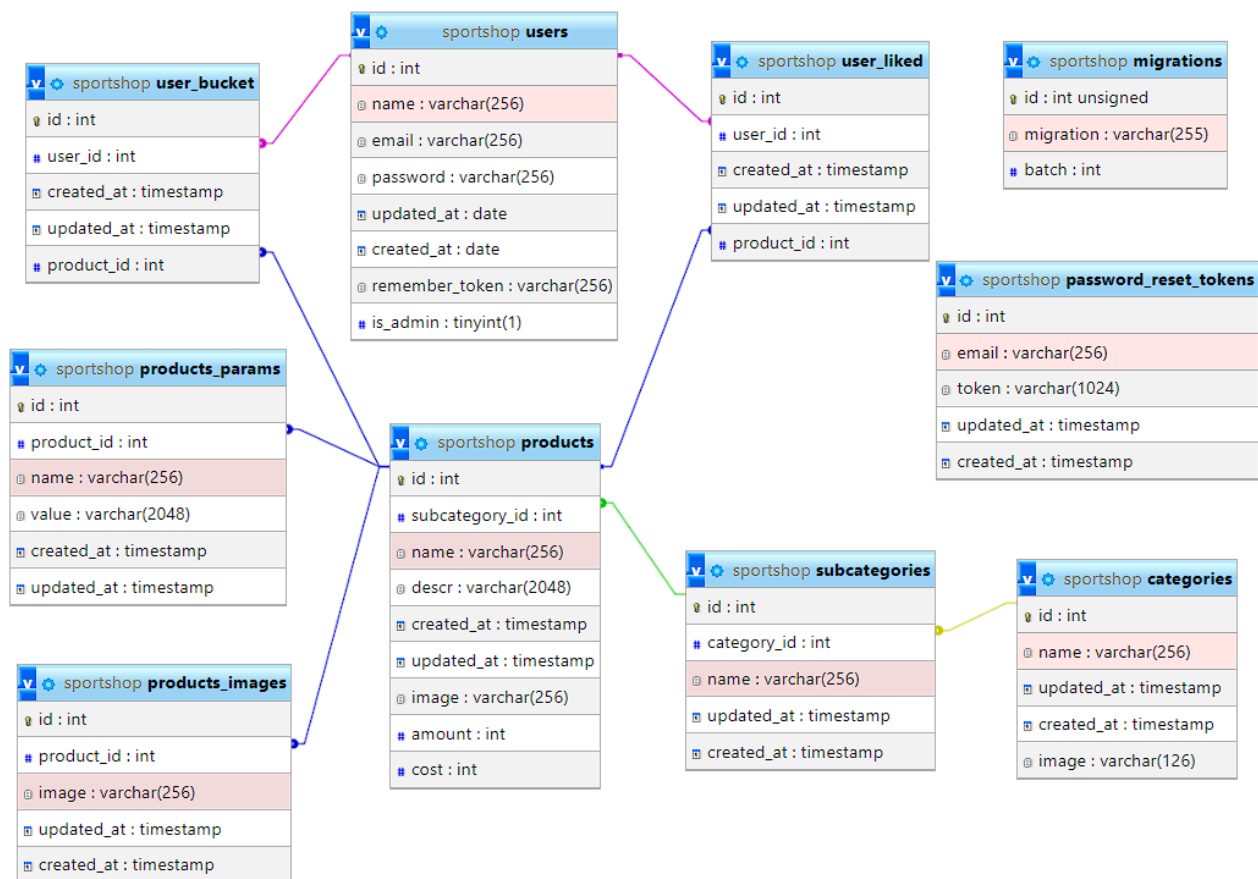


Рис. 2.4 – Схема бази даних

Головною таблицею в представлений базі даних являється таблиця «Продукти» (products). Ця таблиця зберігає інформацію про кожен товар у системі, також вона містить у собі інформацію про те, до якої категорії та підкатегорії належить конкретний товар, параметри, фото, кількість наявного товару і ціну.

Таблиця 2.1

Структура таблиці бази даних «Продукти»

Назва	Тип даних	Опис поля
id	Int	Ключове поле, ідентифікатор товару
subcategory_id	int	Ідентифікатор підкатегорії

name	varchar(256)	Назва продукту
descr	varchar(2048)	Опис продукту
created_at	timestamp	Часова мітка створення
updated_at	timestamp	Часова мітка оновлення
image	varchar(256)	Фото продукту
amount	int	Час останнього входу
cost	int	ПІБ мати

В таблиці «Категорії» (categories) зберігається інформація про категорії товарів. Ця таблиця являє собою усі категорії товарів та пов'язана з таблицею «Підкатегорії» (subcategories). Кожен наявний на сайті продукт повинен відноситись до якоїсь одної підкатегорії та категорії. Поля таблиці «Категорії» представлено в таблиці 2.2.

Таблиця 2.2

Структура таблиці бази даних «Категорії»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор категорії
name	varchar(256)	Назва категорії
updated_at	timestamp	Часова мітка оновлення
created_at	timestamp	Часова мітка створення
image	varchar(126)	Фото категорії

Сполучна таблиця «Підкатегорії» (subcategories) зберігає інформацію про всі підкатегорії інтернет-магазину. Кожна підкатегорія відноситься тільки до однієї категорії, так як у однієї категорії може бути багато підкатегорій. Детальний опис полів представлено в таблиці 2.3.

Таблиця 2.3

Структура таблиці бази даних «Підкатегорії»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор підкатегорії
category_id	int	Ідентифікатор категорії
name	varchar(256)	Назва підкатегорії
updated_at	timestamp	Часова мітка оновлення
created_at	timestamp	Часова мітка створення

Допоміжна таблиця «Фото продукту» (products_images) зберігає в собі фото для кожного продукту. Ця таблиця була створена для того, щоб мати можливість створити більше ніж одне фото для одного продукту. У продукту може бути декілька фото, але одне фото відноситься тільки до одного продукту. Опис полів представлено у таблиці 2.4.

Таблиця 2.4

Структура таблиці бази даних «Фото продукту»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор фото продукту
product_id	int	Ідентифікатор продукту
image: varchar(256)	int	Код групи
updated_at	timestamp	Часова мітка оновлення
created_at	timestamp	Часова мітка створення

Допоміжна таблиця «Параметри продукту» (products_params) зберігає інформацію про усі параметри продукту. Ця таблиця була створена для того, щоб

мати змогу створити більше ніж один параметр для кожного продукту. Один параметр може відноситись тільки до одного продукту, в той час коли у одного продукту може бути декілька параметрів. Полі таблиці «Параметри продукту» представлено в таблиці 2.5.

Таблиця 2.5

Структура таблиці бази даних «Параметри продукту»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор фото продукту
product_id	int	Ідентифікатор продукту
name	varchar(256)	Сума боргу
value	varchar(2048)	
created_at	timestamp	Часова мітка створення
updated_at	timestamp	Часова мітка оновлення

Таблиця «Вподобайки» (user_liked) зберігає інформацію про те, який конкретний продукт був вподобаний конкретним користувачем. Якщо звернути увагу на те, як «Вподобайки» виглядають в інтернет-магазині спорттоварів складається враження, наче в одній вподобайці може бути багато товарів, а один товар може відноситись до багатьох вподобайок. Насправді кожен допис зі списку вподобаних товарів на стороні клієнта це окремий об'єкт вказаної вище таблиці, який за допомоги ідентифікаторів відображається для клієнта як вподобаний. Тобто одна вподобайка може відноситись тільки до одного товару, а один товар може відноситись до багатьох вподобайок. Детальний опис полів представлено в таблиці 2.6.

Таблиця 2.6

Структура таблиці бази даних «Вподобайки»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор вподобайки
user_id	int	Ідентифікатор юзера
created_at	timestamp	Часова мітка створення
updated_at	timestamp	Часова мітка оновлення
product_id	int	Ідентифікатор продукту

Таблиця «Корзина» (user_bucket) було створено за таким самим принципом як і таблиця «Вподобайки». Тобто кожен допис зі списку доданих до корзини товарів на стороні клієнта це окремий об'єкт даної таблиці, який за допомоги ідентифікаторів відображається для клієнта як «доданий до корзини». Один об'єкт корзини відноситься тільки до одного продукту, один продукт може відноситись до багатьох об'єктів корзини. Опис полів представлено в таблиці 2.7.

Таблиця 2.7

Структура таблиці бази даних «Корзина»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор вподобайки
user_id	int	Ідентифікатор юзера
created_at	timestamp	Часова мітка створення
updated_at	timestamp	Часова мітка оновлення
product_id	int	Ідентифікатор продукту

Таблиця «Користувачі» (users) зберігає інформацію про кожного користувача сайту. Вона містить такі поля як імейл, пароль та назву – текстове поле в яке юзер може внести своє ім'я або нік, який буде відображатися на сайті, а також ця таблиця визначає чи являється конкретний користувач адміністратором чи ні. Детальний опис полів відображено в таблиці 2.8.

Таблиця 2.8

Структура таблиці бази даних «Користувачі»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор користувача
name	varchar(256)	Ім'я\нікнейм користувача
email	varchar(256)	Імейл користувача
password	varchar(256)	Пароль користувача
updated_at	date	Дата оновлення користувача
created_at	date	Дата створення користувача
remember_token	varchar(256)	Поле токену
is_admin	tinyint(1)	Перевірка на адміністратора

Також в базі було створено дві допоміжні таблиці які є стандартними для фреймворку Laravel. Перша таблиця – «Міграції» (migrations). Вона використовується для того, щоб змігувати базу даних при внесенні будь-яких змін до бази даних. Поля представлені в таблиці 2.9.

Таблиця 2.9

Структура таблиці бази даних «Міграції»

Назва	Тип даних	Опис поля
id	int unsigned	Ключове поле, ідентифікатор міграції
migrations	varchar(255)	Назва міграції
batch	int	Значення batch

Наступна таблиця, яка була створена за допомогою фреймворку Laravel називається «Маркери скидання паролю» (password_reset_tokens). Ця таблиця використовується у разі необхідності скидання паролю для юзера. Опис полів продемонстровано у таблиці 2.10.

Таблиця 2.10

Структура таблиці бази даних «Маркери скидання паролю»

Назва	Тип даних	Опис поля
id	int	Ключове поле, ідентифікатор маркеру скидання паролю
semestr	varchar(128)	Пошта користувача
token	Varchar(1024)	Назва токєну
updated_at	timestamp	Часова мітка оновлення
created_at	timestamp	Часова мітка створення

2.3. Проектування та реалізація алгоритмів роботи системи

Проектування та реалізація алгоритмів роботи системи Інтернет-магазину спорттоварів є ключовим етапом дипломної роботи. Основна мета полягає в розробці ефективних та надійних алгоритмів, які забезпечать коректну та зручну роботу магазину для користувачів. Нижче наведено основні етапи проектування та реалізації алгоритмів системи Інтернет-магазину спорттоварів:

- **Аналіз вимог:** Спочатку проводиться детальний аналіз вимог до системи Інтернет-магазину спорттоварів. Визначаються функціональні вимоги, такі як реєстрація користувачів, перегляд та пошук товарів, додавання до кошика, оформлення замовлення тощо. Також враховуються нефункціональні вимоги, наприклад, швидкість завантаження сторінок, безпека даних та інші аспекти, які впливають на користувацький досвід.
- **Проектування структури бази даних:** Розробляється структура бази даних, яка буде використовуватися для зберігання інформації про користувачів, товари, замовлення тощо. Визначаються зв'язки між таблицями та необхідні поля для забезпечення ефективного доступу до даних.
- **Вибір технологій:** Враховуючи вимоги та обсяг проекту, визначаються оптимальні технології та інструменти для реалізації Інтернет-магазину спорттоварів. Це можуть бути мови програмування (наприклад, PHP, Java, Python), фреймворки (наприклад, Laravel, Spring, Django) та інші компоненти, необхідні для побудови системи.
- **Розробка алгоритмів:** На основі аналізу вимог та обраної технології проводиться розробка алгоритмів роботи системи. Це включає алгоритми для обробки та валідації даних, керування сесіями користувачів, обробки замовлень, взаємодії з базою даних та інші функціональні аспекти.
- **Тестування та оптимізація:** Розроблені алгоритми піддаються тестуванню для перевірки їхньої працездатності та відповідності вимогам. Під час тестування виявляються можливі помилки та недоліки, які потребують

виправлення. Також проводиться оптимізація алгоритмів для покращення швидкодії та ефективності системи.

2.4. Реалізація програмного продукту

Одною із головних реалізацій програмного продукту являється описані веб-маршрути, які описані у файлі web.php. У цьому файлі описано маршрути до кожного представлення, яке було реалізовано під час розробки інтернет-магазину спорттоварів. Першу частину реалізації веб-маршрутів зображено на рис. 2.5, другу частину реалізації зображено на рис. 2.6, третю частину реалізацію зображено на рис. 2.7.

```
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\AdminController;
use App\Http\Controllers\HomeController;
/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "web" middleware group. Make something great!
|
*/
Auth::routes();

Route::group(['namespace' => 'App\Http\Controllers'], function () {
    Route::get('/', 'HomeController@index')->name('index');
    Route::get('/home', 'HomeController@index')->name('home');
    Route::get('/categories', 'HomeController@categories')->name('categories');
    Route::get('/top', 'HomeController@top')->name('top');
    Route::get('/liked', 'HomeController@liked')->name('liked');
    Route::get('/inbucket', 'HomeController@inbucket')->name('inbucket');
    Route::get('/special', 'SubcategoryController@special')->name('special');
    Route::post('/like', 'HomeController@like')->name('like');
    Route::post('/bucket', 'HomeController@bucket')->name('bucket');
});
```

Рис. 2.5 – Перша частина реалізації веб-маршрутів

```
Route::group(['middleware' => 'auth', 'prefix' => 'admin', 'namespace' => 'App\Http\Controllers'], function () {
    Route::get('/', 'AdminController@index')->name('admin');
    Route::get('/categories', 'CategoryController@index')->name('admin/categories');
    Route::get('/categories/create', 'CategoryController@create')->name('admin/categories/create');
    Route::get('/categories/edit', 'CategoryController@edit')->name('admin/categories/edit');
    Route::post('/categories/store', 'CategoryController@store')->name('admin/categories/store');
    Route::post('/categories/edit', 'CategoryController@edit')->name('admin/categories/edit');
    Route::post('/categories/update', 'CategoryController@update')->name('admin/categories/update');
    Route::post('/categories/delete', 'CategoryController@delete')->name('admin/categories/delete');

    Route::get('/subcategories', 'SubcategoryController@index')->name('admin/subcategories');
    Route::get('/subcategories/create', 'SubcategoryController@create')->name('admin/subcategories/create');
    Route::post('/subcategories/create', 'SubcategoryController@create')->name('admin/subcategories/create');
    Route::post('/subcategories/edit', 'SubcategoryController@edit')->name('admin/subcategories/edit');
    Route::post('/subcategories/store', 'SubcategoryController@store')->name('admin/subcategories/store');
    Route::post('/subcategories/update', 'SubcategoryController@update')->name('admin/subcategories/update');
    Route::post('/subcategories/delete', 'SubcategoryController@delete')->name('admin/subcategories/delete');
});

Route::group(['middleware' => 'auth', 'prefix' => 'admin/products', 'namespace' => 'App\Http\Controllers'], function () {
    Route::get('/', 'ProductController@index')->name('admin/products');
    Route::get('/create', 'ProductController@create')->name('admin/products/create');
    Route::post('/insert', 'ProductController@insert')->name('admin/products/insert');
    Route::post('/edit', 'ProductController@edit')->name('admin/products/edit');
    Route::post('/delete', 'ProductController@delete')->name('admin/products/delete');
    Route::post('/update', 'ProductController@update')->name('admin/products/update');
});
```

Рис. 2.6 – Друга частина реалізації веб-маршрутів

					ІПЗ.КР.Б – 121 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

```

Route::group(['middleware' => 'auth', 'prefix' => 'admin/product_params', 'namespace' => 'App\Http\Controllers'], function () {
    Route::post('/create', 'ProductParamController@create')->name('admin/product_params/create');
    Route::post('/insert', 'ProductParamController@insert')->name('admin/product_params/insert');
    Route::post('/delete', 'ProductParamController@delete')->name('admin/product_params/delete');
    Route::post('/update', 'ProductParamController@update')->name('admin/product_params/update');
    Route::post('/add', 'ProductParamController@add')->name('admin/product_params/add');
});

Route::group(['middleware' => 'auth', 'prefix' => 'categories', 'namespace' => 'App\Http\Controllers'], function () {
    Route::post('/subcategories', 'CategoryController@subcategories')->name('categories/subcategories/create');
    Route::post('/subcategories/products', 'SubcategoryController@products')->name('categories/subcategories/products');
    Route::post('/subcategories/products/product', 'SubcategoryController@product')->name('categories/subcategories/product');
});

```

Рис. 2.7 – Третя частина реалізації веб-маршрутів

Одною із головних функцій інтернет-магазину спорттоварів є створення елементів, наприклад продуктів, категорій та підкатегорій. Для прикладу, реалізація додавання продукту зображена на рис. 2.8. Реалізація додавання інших елементів, наприклад категорії, підкатегорії, параметрів та фото продукту було реалізовано за схожим принципом.

```

public function insert(Request $request) {
    $product = new Product();
    $product->name = $request->input('name');
    $product->descr = $request->input('descr');
    $product->subcategory_id = $request->input('subcategory');
    $product->amount = $request->input('amount');
    $product->cost = $request->input('cost');

    if($request->hasFile('image') && $product->name != null) {
        $file = $request->file('image');
        $extension = $file->getClientOriginalExtension();
        $filename = time().'.'.$extension;
        $file->move('src/products/', $filename);
        $product->image = $filename;
        $product->save();
    }
}

```

Рис. 2.8 – Реалізація створення продукту

Наступною важливою функцією, яка відноситься до більшості об'єктів, наприклад продуктів, категорій та підкатегорій являється функція редагування. Для прикладу, реалізація редагування категорії зображена на рис. 2.9. Реалізація редагування інших елементів, наприклад продукту, підкатегорії, параметрів та фото продукту було реалізовано за схожим принципом.

```

public function update(Request $request) {
    $category = Category::find($request->id);
    $category->name = $request->input('name');
    $subcategories = Subcategory::get()->where('category_id', $request->id);

    if($request->hasFile('image')) {
        $file = $request->file('image');
        $extension = $file->getClientOriginalExtension();
        $filename = time().'.'.$extension;
        $file->move('src/categories/', $filename);
        $imageName = $category->image;
        File::delete('src/products/' . $imageName);
        $category->image = $filename;
    }

    Category::where('id', $request->id)->update(['name' => $category->name, 'image' => $category->image]);
    return view('admin.categories.edit', ['category' => $category, 'subcategories' => $subcategories]);
}

```

Рис. 2.9 – Реалізація редагування категорії

Ще одною важливою функцією, яка відноситься до більшості об'єктів, наприклад продуктів, категорій та підкатегорій являється функція видалення. Для прикладу, реалізація видалення підкатегорії зображена на рис. 2.10. Реалізація видалення інших елементів, наприклад продукту, категорії, параметрів та фото продукту було реалізовано за схожим принципом.

```

public function delete(Request $request) {
    Subcategory::where('id', $request->subcategory_id)->delete();

    $category = Category::find($request->category_id);
    $subcategories = Subcategory::get()->where('category_id', $request->category_id);
    return view('admin.categories.edit')->with('category', $category)->with('subcategories', $subcategories);
}

```

Рис. 2.10 – Реалізація видалення підкатегорії

Одні із головних інструментів, які будуть використовувати користувачі інтернет-магазину це «Вподобайка» та «Корзина». За допомогою інструменту «Вподобайка» користувач має змогу додати будь-який товар до списку вподобаних товарів, що полегшить пошук та запам'ятовування товарів, які сподобались користувачеві. Реалізація функції позначення товару користувачем як «Вподобане» відображено на рис. 2.11. Позначення користувачем продукту як «Доданий до корзини» було реалізовано за схожим принципом.

```

public function like(Request $request) {
    if (auth()->user() == null) {
        return redirect('login');
    }
    $user_id = auth()->user()->id;
    $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();
    $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();
    $bucket_sum = DB::table('user_bucket')
        ->join('products', 'user_bucket.product_id', '=', 'products.id')
        ->where('user_id', $user_id)->sum('cost');

    $product = Product::where('id', $request->id)->first();
    $user_id = auth()->user()->id;
    DB::table('user_liked')->insert([
        'user_id' => $user_id,
        'product_id' => $product->id
    ]);
    return redirect('liked')
        ->with('bucket_count', $bucket_count)
        ->with('liked_count', $liked_count)
        ->with('bucket_sum', $bucket_sum);
}

```

Рис. 2.11 – Реалізація позначення товару як «Вподобане»

Відображення списку товарів, позначених як «Вподобане» було реалізовано за допомогою функції, яка відображена на рис. 2.12. Відображення списку товарів, котрі були додані до корзини було реалізовано за схожим принципом.

```

public function liked() {
    if (auth()->user() == null) {
        return redirect('login');
    }
    $user_id = auth()->user()->id;
    $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();
    $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();
    $bucket_sum = DB::table('user_bucket')
        ->join('products', 'user_bucket.product_id', '=', 'products.id')
        ->where('user_id', $user_id)->sum('cost');

    $products = DB::table('products')
        ->join('subcategories', 'products.subcategory_id', '=', 'subcategories.id')
        ->join('categories', 'subcategories.category_id', '=', 'categories.id')
        ->join('user_liked', 'products.id', '=', 'user_liked.product_id')
        ->select('user_liked.id as liked_id', 'products.*', 'subcategories.name as subcategory_name', 'categories.name as category_name')
        ->where('user_id', $user_id)
        ->get();

    return view('liked')->with('liked', $products)
        ->with('bucket_count', $bucket_count)
        ->with('liked_count', $liked_count)
        ->with('bucket_sum', $bucket_sum);
}

```

Рис. 2.12 – Реалізація функції для відображення списку вподобаних товарів

Видалення товару зі списку товарів, які позначені як у «Корзині» було реалізовано за допомогою функції, яка зображена на рис. 2.13. Функцію

видалення товару зі списку «Вподобане» було реалізовано за схожим принципом.

```

public function deletebucket(Request $request) {
    UserBucket::where('id', $request->id)->delete();

    $user_id = auth()->user()->id;
    $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();
    $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();
    $bucket_sum = DB::table('user_bucket')
        ->join('products', 'user_bucket.product_id', '=', 'products.id')
        ->where('user_id', $user_id)->sum('cost');

    return redirect('inbucket')
        ->with('bucket_count', $bucket_count)
        ->with('liked_count', $liked_count)
        ->with('bucket_sum', $bucket_sum);
}

```

Рис. 2.13 – Реалізація функції для відображення списку вподобаних товарів

Висновки до розділу 2

В даному пункті було визначено варіанти використання та об'єктно-орієнтовану структуру системи. Було розроблено схему бази даних спроектовано та реалізовано алгоритми роботи системи. А також було детально описано реалізація програмного продукту.

РОЗДІЛ 3. ПОРЯДОК РОБОТИ З ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ТА ІНТЕРФЕЙС

3.1. Порядок встановлення та налаштування параметрів системи

Для користування веб-додатком обов'язково потрібна наявність підключення до мережі.

Мінімальні вимоги до комп'ютера клієнта для коректної роботи сайту інтернет-магазину спорттоварів:

- Операційна система: Версія Windows не нижче Windows 7.
- Процесор: Мінімум двоядерний процесор з тактовою частотою 1.6 ГГц або еквівалентний.
- Оперативна пам'ять: Мінімум 2 ГБ оперативної пам'яті.
- Браузер: Остання версія популярного веб-браузера, такого як Google Chrome, Mozilla Firefox, Microsoft Edge або Safari.
- Інтернет-з'єднання: Стабільне та швидке інтернет-з'єднання для завантаження сторінок та отримання даних з сервера.

Мінімальні вимоги до комп'ютера (сервера) для коректної роботи сайту інтернет-магазину спорттоварів:

- Процесор: Мінімум чотирьохядерний процесор з тактовою частотою 2.0 ГГц або більше.
- Оперативна пам'ять: Мінімум 4 ГБ оперативної пам'яті (RAM).
- Жорсткий диск: Достатньо місця на жорсткому диску для збереження файлів сайту, бази даних та інших ресурсів.
- Мережеве з'єднання: Стабільне та швидке підключення до Інтернету з високою швидкістю передачі даних.

3.2. Структура інтерфейсу

При завантаженні сайту користувач потрапляє на головну сторінку інтернет-магазину спорттоварів. Користувач бачить хедер сайту з основними елементами навігації, основні блоки де відображено кнопку «Перейти», декілька категорій товарів які в даний момент є наявними в інтернет-магазині та

					ІПЗ.КР.Б – 121 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

відображені у випадковому порядку, а також футер сайту. Головна сторінка інтернет-магазину зображена на рис. 3.1.

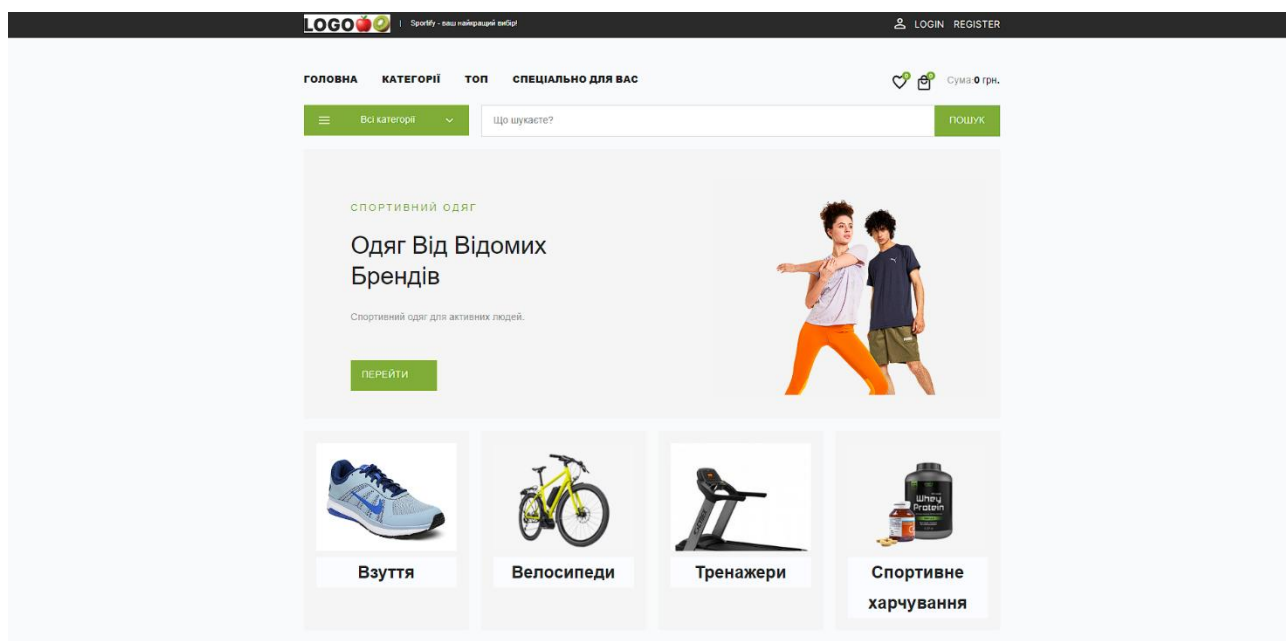


Рис. 3.1 – Головна сторінка інтернет-магазину спорттоварів

В хедері інтернет-магазину було створено 4 розділи – «Головна», «Категорії», «Топ», «Спеціально для вас». Також хедер містить логотип інтернет-магазину, назву та девіз інтернет-магазину, інформацію про те, який саме користувач в даний момент авторизований на сайті, та, якщо ніхто не авторизований – кнопки «Логін» (LOGIN) та «Реєстрація» (REGISTER), спрощене навігаційне меню у вигляді кнопки «Всі категорії», пошук, «Вподобані» та «Корзина», а також сума вартості усіх товарів які були додані до корзини. Хедер представлений на рис. 3.2.

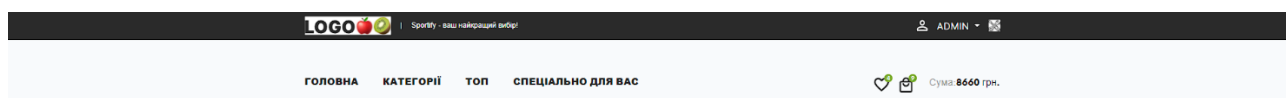


Рис. 3.2 – Хедер інтернет-магазину спорттоварів

У футері сайту відображено інформацію про інтернет магазин, а саме логотип магазину, адресу, корисні посилання, та посилання на соціальні мережі. Футер представлено на рис. 3.3.

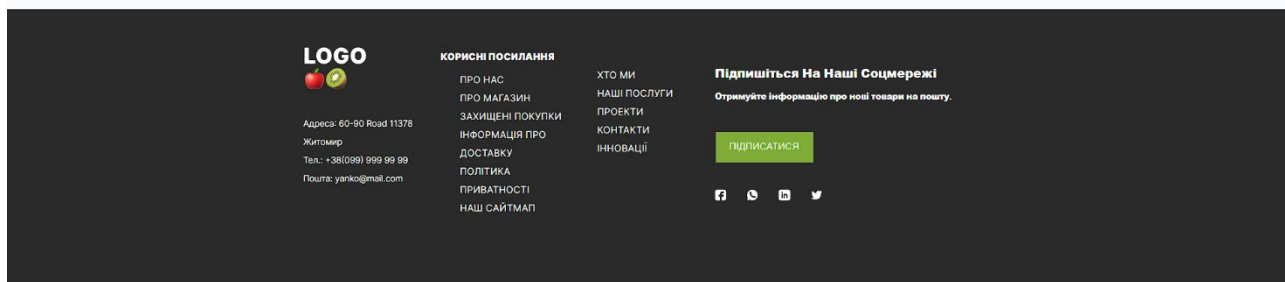


Рис. 3.3 – Футер інтернет-магазину спорттоварів

Якщо натиснути на кнопку «Всі категорії», яка знаходиться у лівій частині зверху екрану для користувача відобразиться спрощений вигляд структури категорій та підкатегорій. Результат роботи кнопки «Всі категорії» відображено на рис. 3.4.

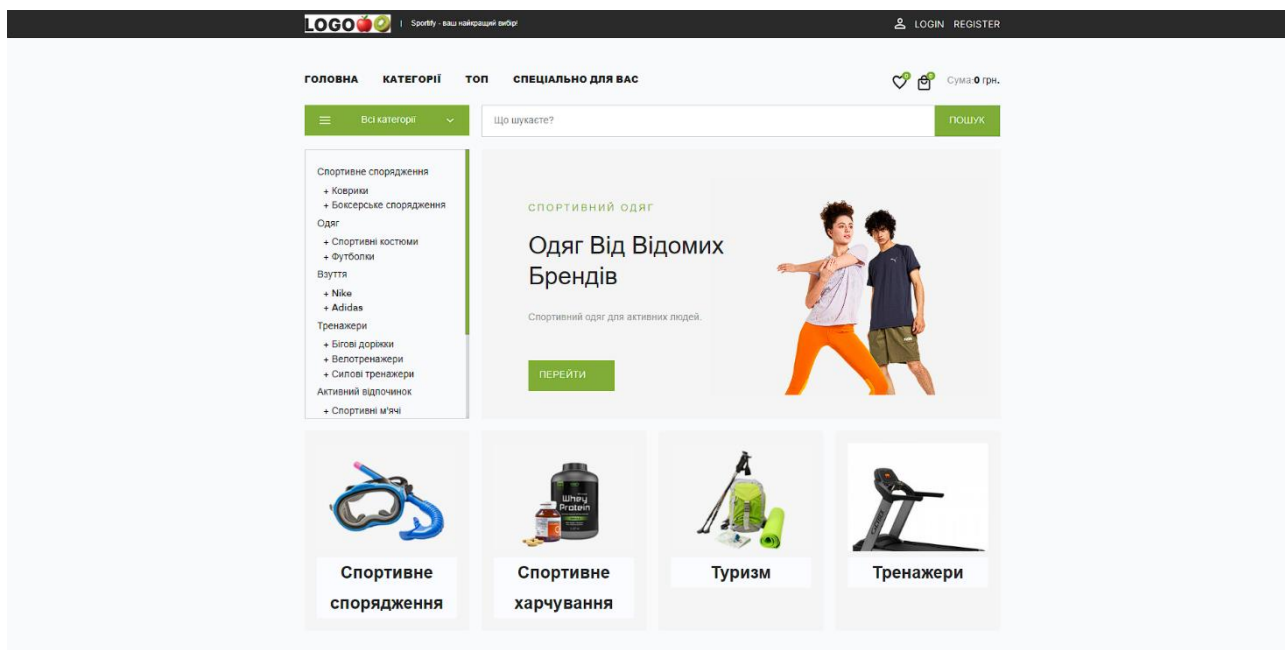


Рис. 3.4 – Результат роботи кнопки «Всі категорії» на головній сторінці сайту

При натисканні на будь-який пункт навігації, наприклад категорію «Спортивне спорядження» або «Одяг» користувач потрапляє на сторінку обраної категорії де відображаються список підкатегорій, які відносяться до обраного пункту навігаційного меню. Результат роботи натискання на пункт навігаційного меню «Спортивне спорядження» відображено на рис. 3.5.

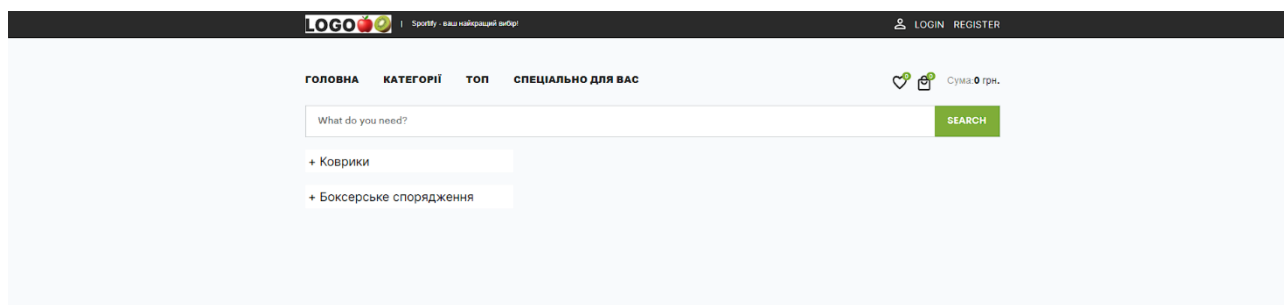


Рис. 3.5 – Результат натискання на пункт навігаційного меню «Спортивне спорядження»

Якщо користувач натисне на підкатегорію «Боксерське спорядження» то він потрапить на сторінку де будуть відображені усі товари які відносяться до обраної ним підкатегорії. Результат натискання на підкатегорію зображено на рис. 3.6.

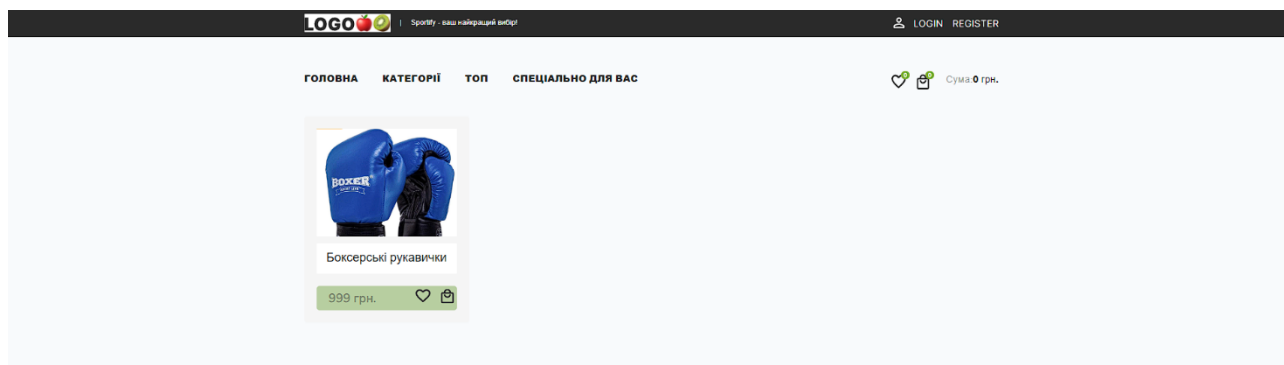


Рис. 3.6 – Результат натискання на підкатегорію «Боксерське спорядження»

За схожим принципом працює перехід до будь-якої підкатегорії на пряму за допомоги навігаційного меню кнопки «Всі категорії». Якщо користувач натисне на підкатегорію «Спортивні костюми» то він потрапить на сторінку обраної підкатегорії, де будуть відображені усі товари які відносяться до відповідної підкатегорії. Результат роботи кнопки «Спортивні костюми» навігаційного меню відображено на рис. 3.7.

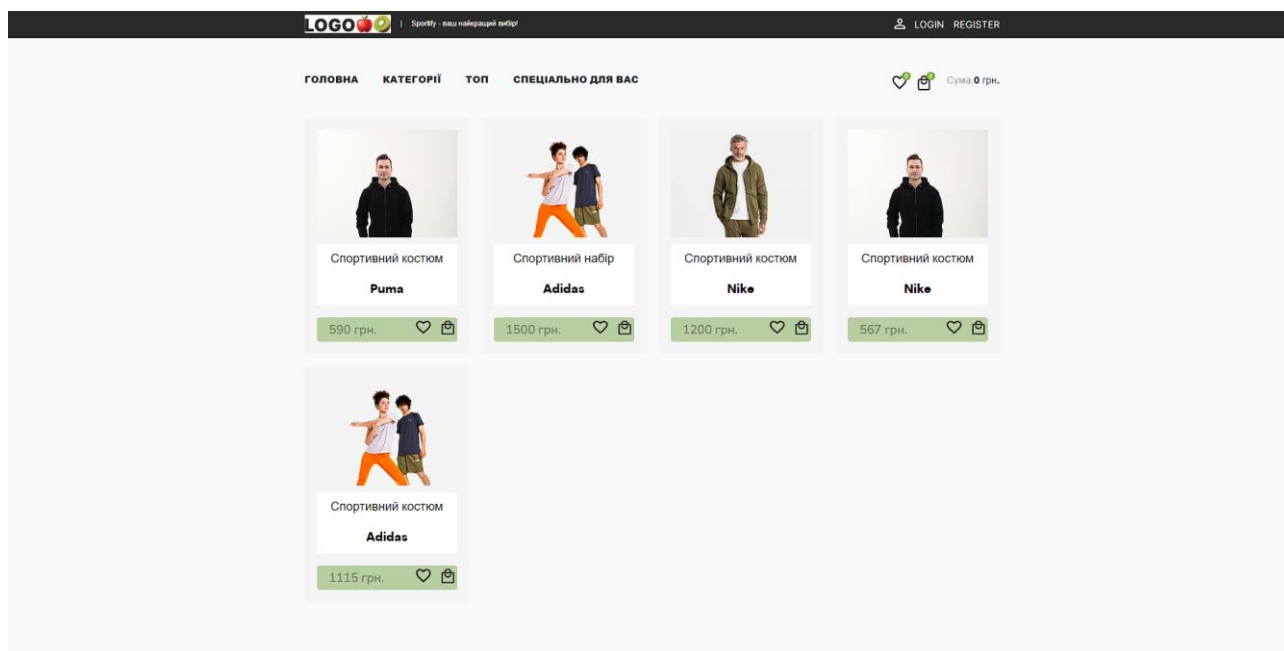


Рис. 3.7 – Результат роботи кнопки навігаційного меню «Спортивні костюми»

Далі користувач може обрати будь-який товар з підкатегорії, яку було обрано раніше. Якщо користувач натисне на товар, то він потрапить на сторінку де користувач матиме змогу побачити усі фото та параметри, які відносяться до обраного товару, його назву, ціну, а також користувач матиме можливість додати товар до списку «Вподобані» або «Корзина». Приклад розкритого товару зображено на рис. 3.8.

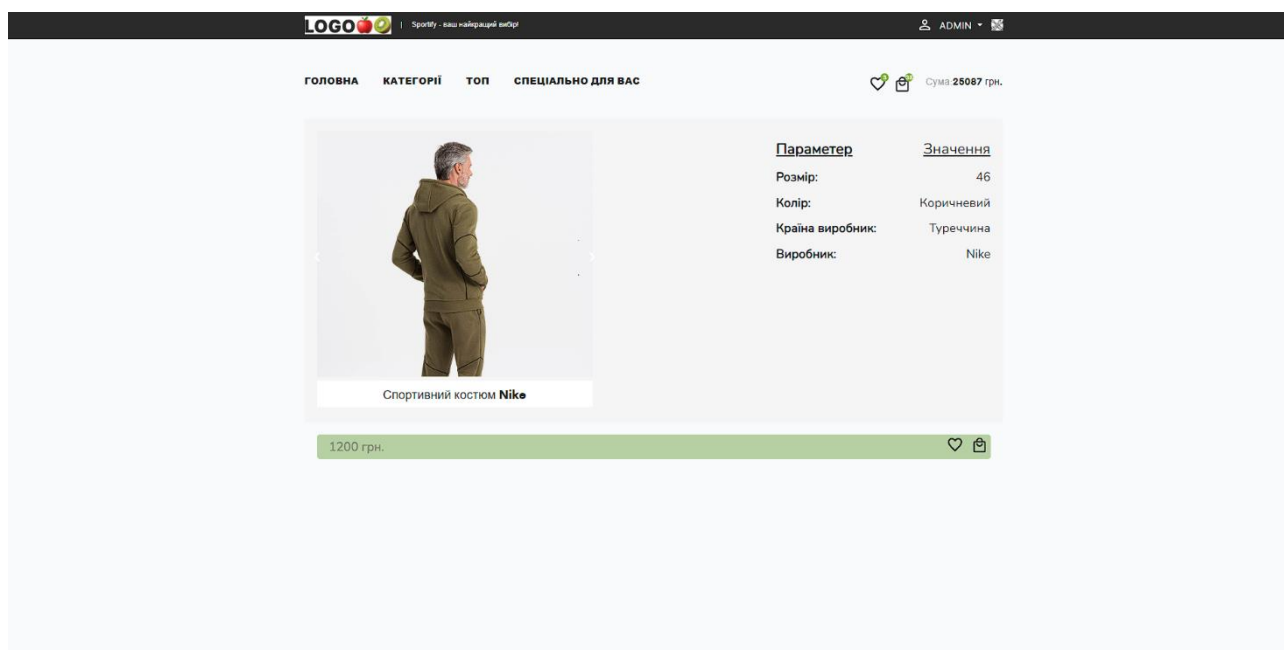


Рис. 3.8 – Приклад розкритого товару

Будь-який товар авторизований користувач може додати в «Вподобане» або в «Корзину». Для того щоб це зробити, авторизований користувач повинен обрати необхідний для нього товар та розкрити його. Після цього, навпроти вартості товару будуть зображені кнопки «Вподобайка» та «Корзина», за допомогою яких авторизований користувач може додати обраний товар в необхідне для себе місце. Якщо користувач натискає на кнопку «Вподобайка», то товар автоматично додається до списку «Вподобані» та відкриває для користувача цей список. Список «Вподобайки» створений для спрощення процесу вибору товарів для замовлення, а також полегшує процес запам'ятовування товарів, які сподобались користувачеві і замість того, щоб щоразу шукати товар, котрий сподобався, через пошук або навігацію сайту користувач завжди може зайти до списку вподобаних товарів та вирішити чи придбати вподобаний товар, чи ні, що значною мірою економить час для кожного користувача інтернет-магазину. Також за допомогою цього списку можна легко реалізувати статистику, задля розуміння який товар найбільше подобається користувачам. Приклад списку вподобаних товарів відображено на рис. 3.9.

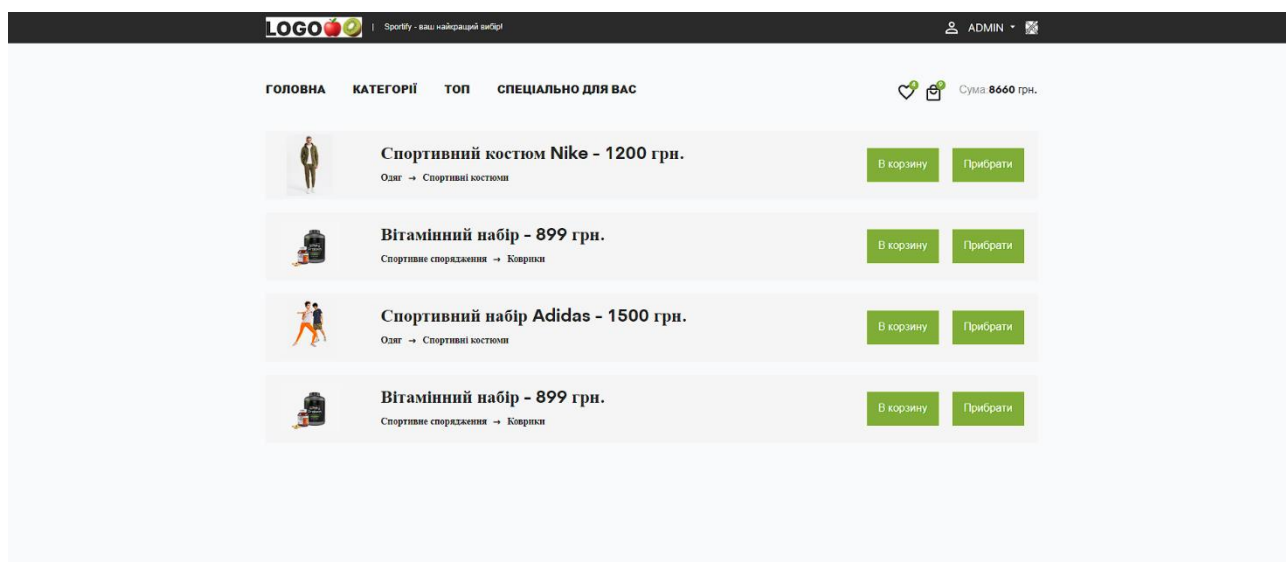


Рис. 3.9 – Приклад списку вподобаних товарів «Вподобане»

Якщо користувач натискає на кнопку «Корзина» то товар автоматично потрапляє до «Корзини». Цей список було створено задля того щоб полегшити процес придбання товару користувачем, адже в корзині користувач має змогу

					ІПЗ.КР.Б – 121 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

побачити всі товари які були додані до цього списку, суму їх загальної вартості, а також користувач має можливість придбати усі товари з корзини за допомогою однієї кнопки. Приклад списку товарів у корзині зображено на рис. 3.10.

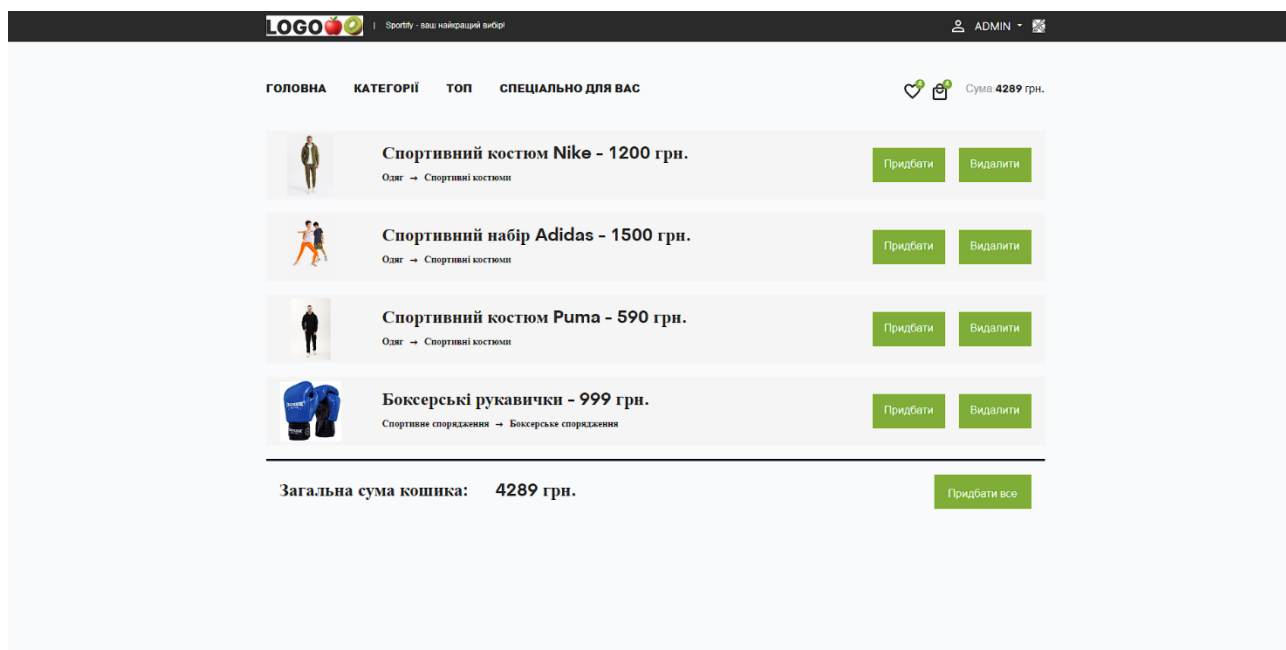


Рис. 3.10 – Приклад списку товарів доданих до «Корзини»

В інтернет-магазині спорттоварів реалізовано реєстрацію та авторизацію користувачів. Якщо користувач не авторизований на сайті, то він не зможе додати товар до списку вподобаних або до корзини, так як для цього обов'язково потрібен ідентифікатор користувача. Якщо користувач натисне на кнопку «Реєстрація» (REGISTER), котра знаходиться у хедері сайту, то він потрапить на форму реєстрації. Форму реєстрації зображено на рис. 3.11.

Рис. 3.11 – Форма реєстрації користувача

Якщо користувач натисне на кнопку «Логін» (LOGIN), яка знаходиться у хедері сайту, або натисне на кнопку «додати до вподобайок» або «додати до кошика» при перегляді будь-якого товару, то користувач потрапить на форму логіна. Тільки після авторизації користувач може додати будь-який товар до списку вподобаних або до кошика. Форму логіна представлено на рис. 3.12

Рис. 3.12 – Форма логіна користувача

В програмі прописані пошта та пароль які є адміністраторськими. Тобто якщо на сайт авторизується користувач і він використовує пошту та пароль, котрі вказані в програмі як «адміністративні пошта та пароль» то це означає що авторизувався адміністратор. Адміністратор може так само як і звичайний користувач користуватися усіма функціями інтернет-магазину, але також у нього є доступ до допоміжного меню.

Щоб потрапити до сторінки управління магазином, адміністратор повинен натиснути на значок «Інструменти», який знаходиться правіше від ім'я авторизованого користувача в хедері. Значок «Інструменти» відображено на рис. 3.13

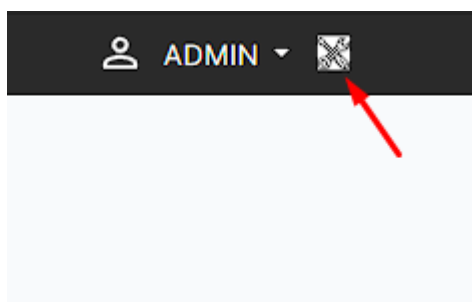


Рис. 3.13 – Значок «Інструменти» в профілі адміністратора

На сторінці управління магазином відображено два пункти меню – «Категорії» та «Продукти». Відображення сторінки управління магазином зображено на рис. 3.14.

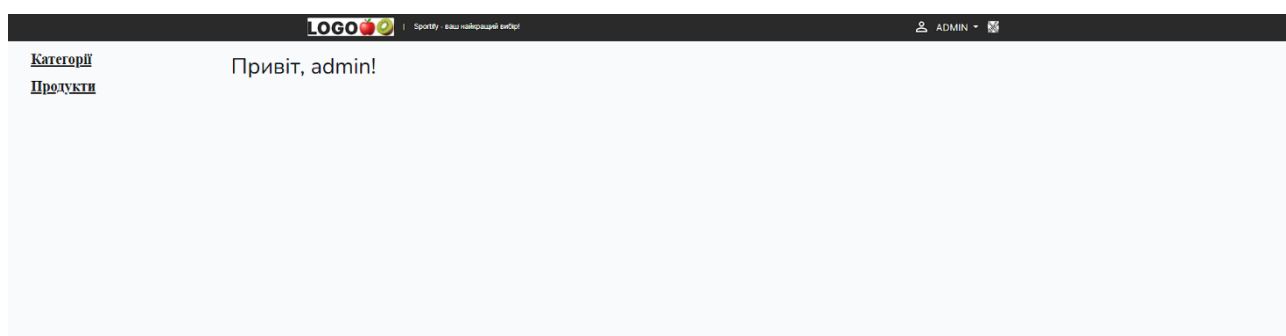


Рис. 3.14 – Сторінка адміністратора управління інтернет-магазином

Якщо натиснути на кнопку «Категорії» на сторінці управління інтернет-магазином то адміністратор потрапить на сторінку де відображені усі категорії інтернет-магазину спорттоварів. Адміністратор має змогу створити нову категорію, редагувати або видаляти вже існуючі категорії. Результат роботи кнопки «Категорії» на сторінці управління інтернет-магазином зображено на рис. 3.15.

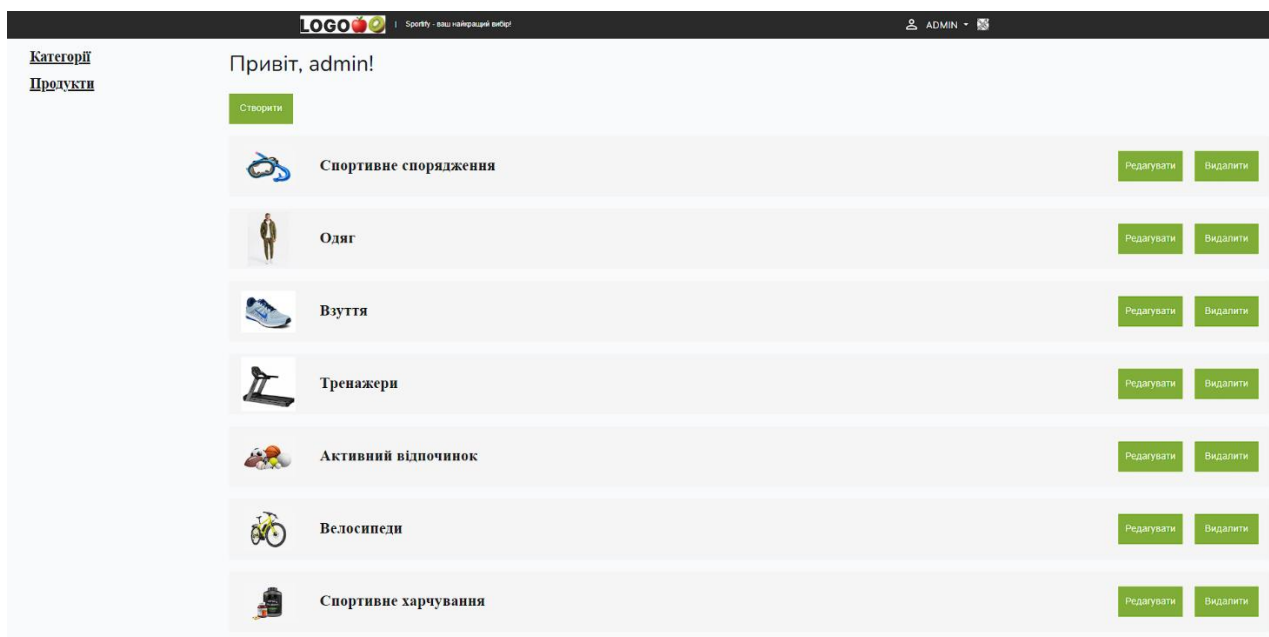


Рис. 3.15 – Результат роботи кнопки «Категорії» на сторінці управління інтернет-магазином

Якщо адміністратор натисне на кнопку «Створити» на сторінці управління категоріями, то він потрапить на форму де він має змогу повернутися назад, або створити нову категорію ввівши назву категорії в текстове поле «Назва категорії», обравши зображення категорії та натиснувши кнопку «Створити». Форма створення нової категорії відображена на рис. 3.16.

Рис. 3.16 – Форма створення нової категорії

Якщо адміністратор, знаходячись на сторінці управління категоріями натисне кнопку «Редагувати» навпроти будь-якої категорії то він потрапить до форми редагування категорії, де він може змінити назву або фото категорії, а також створити та видалити підкатегорії, які відносяться до даної категорії. Форма редагування категорії відображена на рис. 3.17

Рис. 3.17 – Форма редагування категорії

При натисканні кнопки «Створити» під розділом «Налаштування підкатегорій» адміністратор потрапляє на форму створення підкатегорій де, для створення підкатегорії, необхідно лише ввести назву підкатегорії. Форма створення підкатегорії відображена на рис. 3.18.

Рис. 3.18 – Форма створення підкатегорії

При натисканні кнопки «Редагувати» навпроти будь-якої з підкатегорій адміністратор потрапляє на форму редагування підкатегорій де, для редагування підкатегорії необхідно лише змінити назву підкатегорії. Форма редагування підкатегорії відображена на рис. 3.19.

Рис. 3.19 – Форма редагування підкатегорії

Якщо на сторінці управління інтернет-магазином адміністратор натискає на кнопку «Продукти», то на сторінці відобразиться список усіх продуктів які в даний момент наявні на сайті. Результат роботи кнопки «Продукти» відображено на рис. 3.20.

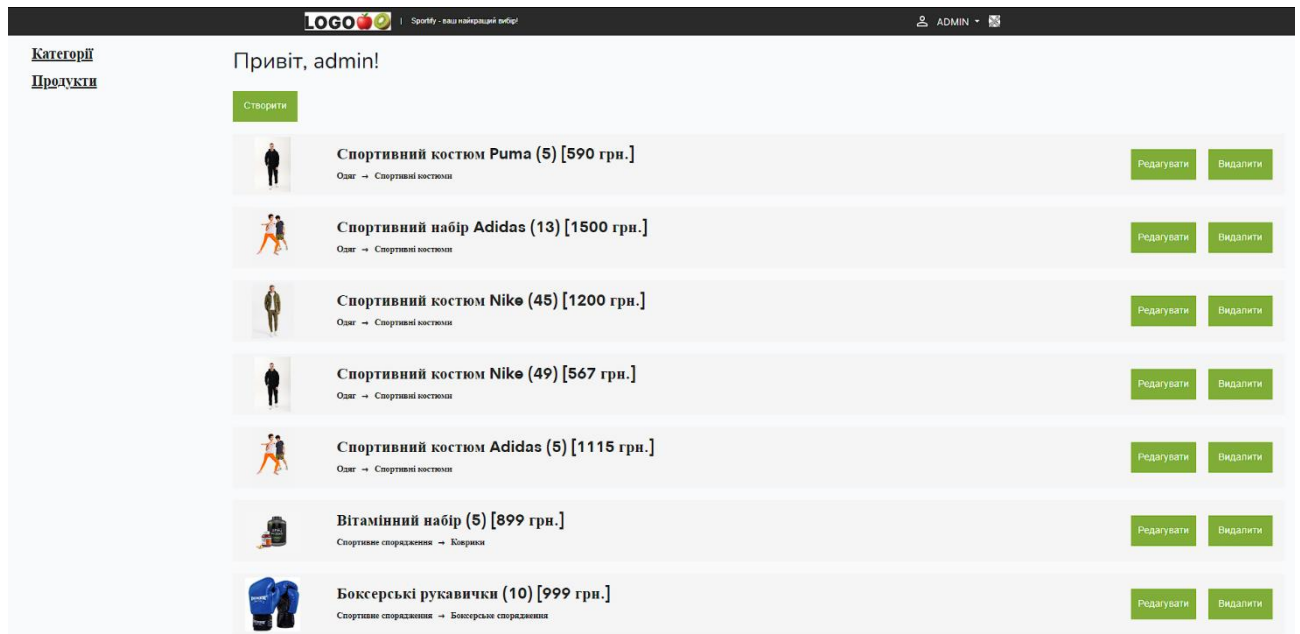


Рис. 3.20 – Результат роботи кнопки «Продукти» на сторінці управління інтернет-магазином

При натисканні кнопки «Створити» для адміністратора відображається форма для створення товару. Форму створення товару відображено на рис. 3.21.

Рис. 3.21 – Форма створення товару

При натисканні кнопки «Редагувати» навпроти будь-якого товару на сторінці управління товарами для адміністратора відображається форма для редагування товару. Першу частину форми редагування товару відображено на рис. 3.22, другу частину відображено на рис. 3.23.

Рис. 3.22 – Перша частина форми редагування товару

Рис. 3.23 – Друга частина форми редагування товару

Також з форми редагування товару адміністратор за допомоги кнопки «Створити» під розділом «Параметри» може створити параметри до обраного

продукту, а за допомогою кнопки «Оновити» навпроти кожного товару адміністратор може оновити кожен параметр для обраного продукту.

Також на сторінках управління адміністратор має змогу видалити будь-які записи наявні в інтернет-магазині за допомогою кнопки «Видалити».

На сайті інтернет-магазину спорттоварів було створено додаткові інструменти навігації. Якщо користувач знаходиться на будь-якій сторінці інтернет-магазину, окрім форм логіну, реєстрації, або сторінки адміністратора то для користувача доступні допоміжні інструменти навігації в інтернет-магазині. При натисканні на кнопку «Головна», яка знаходиться у футері сайту, користувач потрапить на сторінку «Головна». Результат роботи кнопки «Головна» зображено на рис. 3.24.

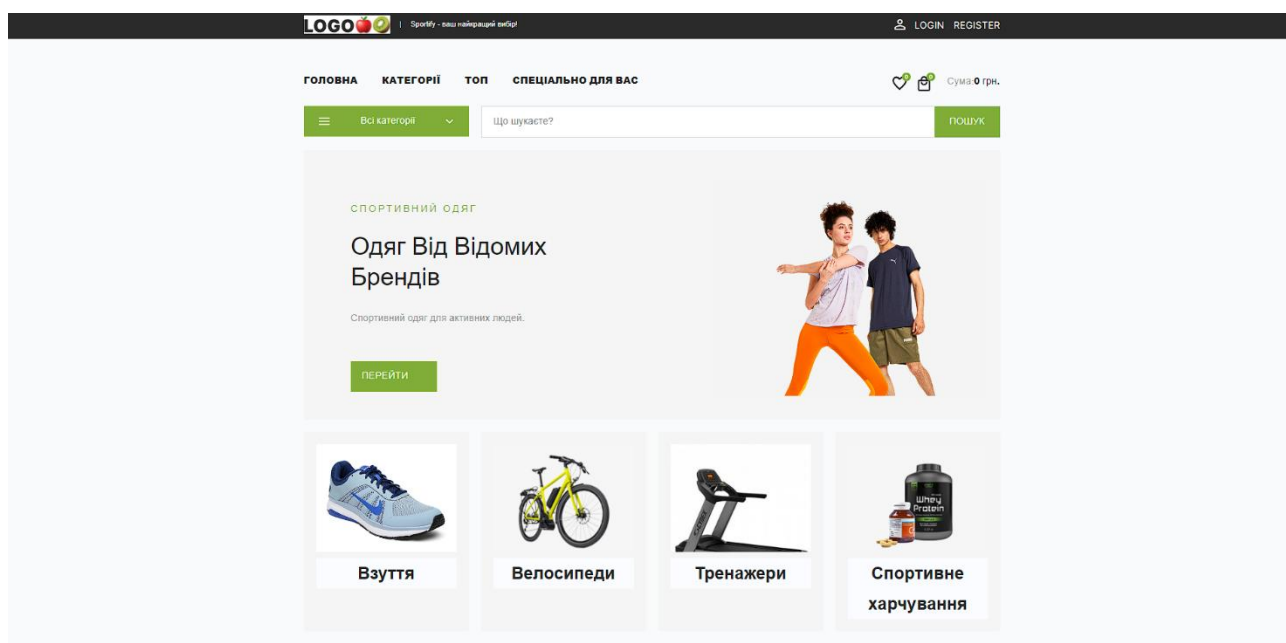


Рис. 3.24 – Результат роботи кнопки «Головна» в хедері сайту.

Якщо користувач знаходиться на будь-якій сторінці сайту і натисне на кнопку «Категорії» в хедері інтернет-магазину то для користувача відобразиться сторінка з усіма категоріями інтернет-магазину. Результат роботи кнопки «Категорії» відображено на рис. 3.25.

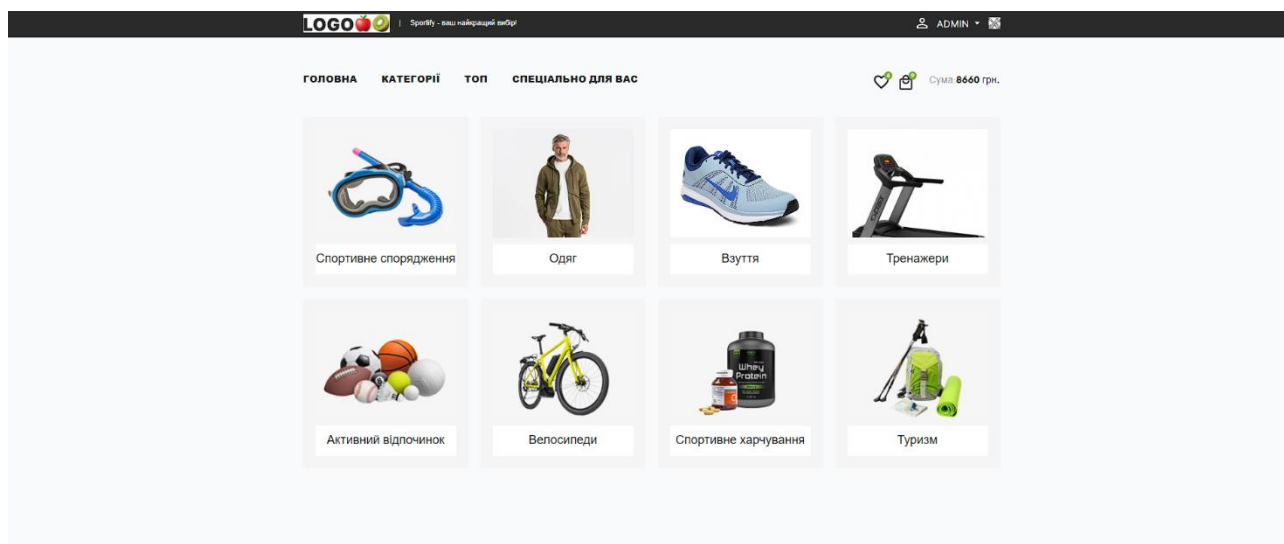


Рис. 3.25 – Результат роботи кнопки «Категорії» в хедері сайту.

Якщо користувач, знаходячись на будь-якій сторінці інтернет-магазину натисне на кнопку «Топ», яка знаходиться у хедері сайту, то користувач потрапить на сторінку де буде відображено 32 випадкові категорії, які в даний момент є в інтернет-магазині. Результат роботи кнопки «Топ» зображено на рис. 3.26.

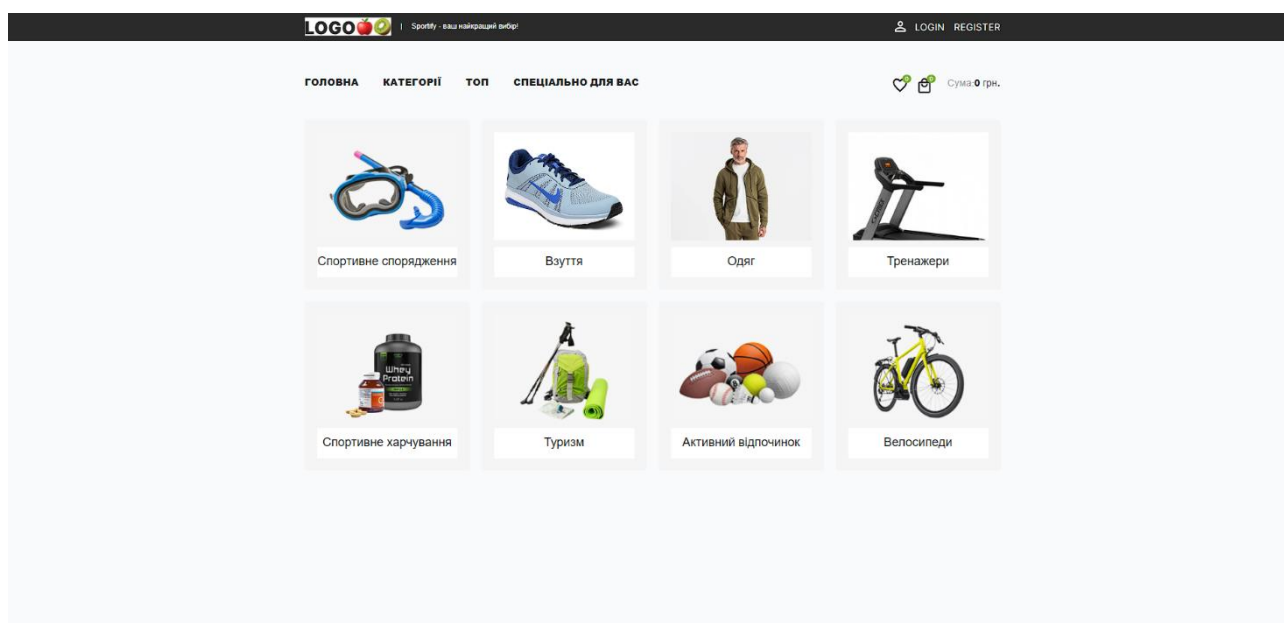


Рис. 3.26 – Результат роботи кнопки «Топ» в хедері сайту.

Якщо користувач, знаходячись на будь-якій сторінці інтернет-магазину натисне на кнопку «Спеціально для вас», яка знаходиться у хедері сайту, то користувач потрапить на сторінку де буде відображено 32 товари у випадковому

порядку, які в даний момент є на складі. Результат роботи кнопки «Спеціально для вас» зображено на рис. 3.27.

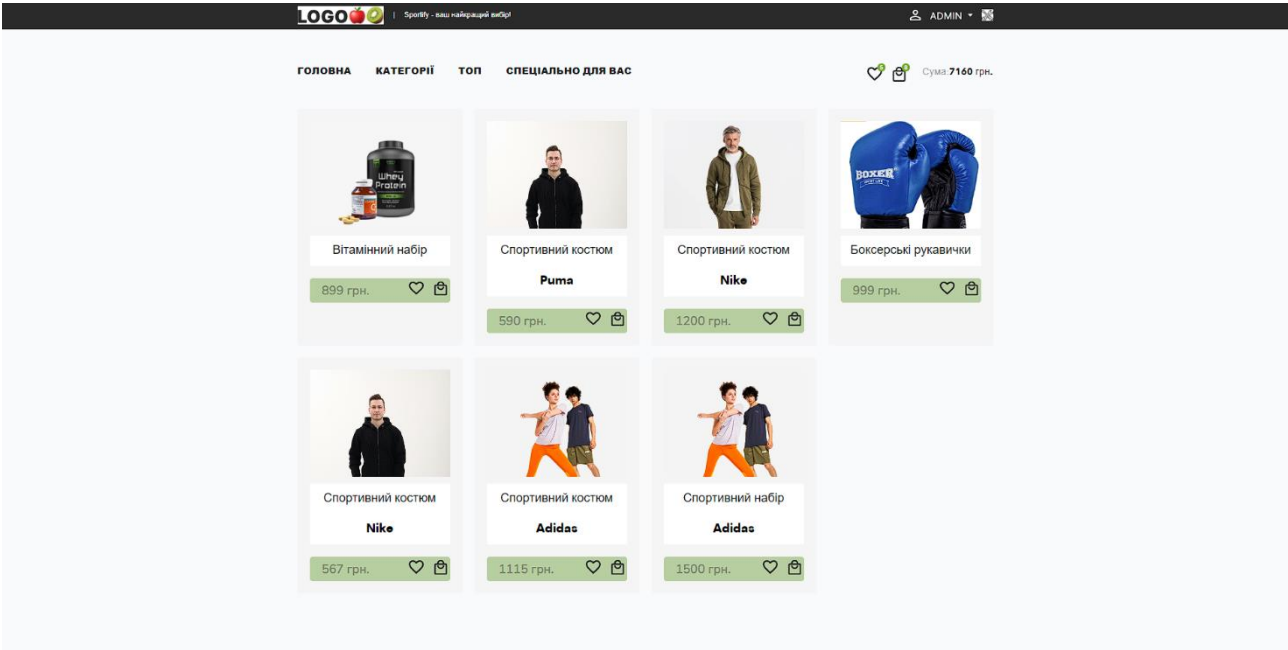


Рис. 3.27 – Результат роботи кнопки «Спеціально для вас» в хедері сайту

3.3. Тестування роботи програмного комплексу

В таблиці 3.1 відображено тестування форми логіну.

Таблиця 3.1

Тестування форми логіну

Номер тесту	Очікуваний результат	Фактичний результат	Пройдено/не пройдено
1	Пусті поля	Пусті поля не зчитуються	Пройдено
2	Користувач вводить невірні дані	Вивід повідомлення користувачу про невірні дані	Пройдено

В таблиці 3.2 наведені тестування форми створення продукту.

Тестування форми створення продукту

Номер тесту	Очікуваний результат	Фактичний результат	Пройдено/не пройдено
1	У поля, куди можливо ввести тільки числові значення неможливо ввести символи	Якщо намагатися ввести символи в поля, наприклад «Ціна», «Кількість» - символи не вводяться	Пройдено
2	Створення товару з усіма пустими полями неможливе	Якщо не ввести жодного значення в жодне поле то форма не пропускає створення продукту, так як спрацьовує виключення	Пройдено
3	Створення товару з декількома пустими полями неможливе	Якщо декілька або одне поле при створенні продукту залишити пустим то форма не пропускає створення товару	Пройдено

Було проведено різноманітні тестування веб-додатку, включаючи ролі звичайного користувача та адміністратора. Під час тестування функціоналу у ньому не було виявлено недоліків чи критичних помилок у роботі.

Висновки до розділу 3

У ході виконання розділу 3 було встановлено порядок встановлення та налаштування параметрів системи, було описано структуру інтерфейсу та проведено тестування роботи програмного комплексу.

Висновки

У кваліфікаційній роботі освітнього ступеня «бакалавр» за спеціальністю 121 «Інженерія програмного забезпечення» (освітня програма «Інженерія програмного забезпечення») на тему: Інтернет-магазин спорттоварів проведено аналіз існуючих Інтернет-магазинів, які спеціалізуються на спорттоварах.

У ході виконання розділу 1 було створено та проаналізовано постановку задачі. Було проаналізовано аналоги програмного продукту та зроблено порівняння функціональних можливостей. Було визначено ключові терміни системи. Та обґрунтовано вибір інструментальних засобів системи та вимоги до апаратного забезпечення.

У ході виконання розділу 3 було визначено варіанти використання та об'єктно-орієнтовану структуру системи. Було розроблено схему бази даних спроектовано та реалізовано алгоритми роботи системи. А також було детально описано реалізація програмного продукту.

У ході виконання розділу 3 було встановлено порядок встановлення та налаштування параметрів системи, було описано структуру інтерфейсу та проведено тестування роботи програмного комплексу.

Висновки до кожного розділу підкреслюють важливість проведених аналізів, проектування та тестування для успішної реалізації інтернет-магазину спорттоварів.

Отже, розроблений веб-додаток є надзвичайно корисним та актуальним у наш час.

Список використаних джерел

1. Dave Thomas, Andy Hunt, The Pragmatic Programmer: Your Journey to Mastery, Addison-Wesley Professional, 2019, 352 с.
2. David Flanagan, JavaScript: The Definitive Guide, O'Reilly Media, 2020, 706 с.
3. Douglas Crockford, JavaScript: The Good Parts, O'Reilly Media, 2008, 176 с.
4. Eric A. Meyer, CSS: The Definitive Guide, O'Reilly Media, 2017, 1090 с.
5. Eric Freeman, Elisabeth Robson, Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages, O'Reilly Media, 2012, 768 с.
6. Hugh E. Williams, Saied M.M. Tahaghoghi, Learning MySQL, O'Reilly Media, 2006, 562 с.
7. Jennifer Niederst Robbins, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics, O'Reilly Media, 2018, 810 с.
8. Jon Duckett, HTML and CSS: Design and Build Websites, John Wiley & Sons, 2011, 490 с.
9. Julie C. Meloni, Sams Teach Yourself PHP, MySQL and Apache All in One, Sams Publishing, 2012, 672 с.
10. Larry Ullman, PHP and MySQL for Dynamic Web Sites, Peachpit Press, 2017, 704 с.
11. Luke Welling, Laura Thomson, PHP and MySQL Web Development, Addison-Wesley Professional, 2016, 704 с.
12. Matt Zandstra, PHP Objects, Patterns, and Practice, Apress, 2013, 552 с.
13. Robin Nixon, Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, O'Reilly Media, 2014, 812 с.

14. Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites, O'Reilly Media, 2018, 832 с.
15. Steve McConnell, Code Complete: A Practical Handbook of Software Construction, Microsoft Press, 2019, 960 с.

					ІПЗ.КР.Б – 121 – 23 – ІПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

Додатки

					ІПЗ.КР.Б – 121 – 23 – ІЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Лістинг коду

```
<?php

namespace App\Http\Controllers;

use App\Models\Category;
use App\Models\Subcategory;
use App\Models\UserLiked;
use App\Models\Product;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        //$this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
}
```

					ІПЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function index()
{
    $bucket_count = 0;

    $liked_count = 0;

    $bucket_sum = 0;

    if(auth()->user() != null) {

        $user_id = auth()->user()->id;

        $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

        $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

        $bucket_sum = DB::table('user_bucket')

            ->join('products', 'user_bucket.product_id', '=', 'products.id')

            ->where('user_id', $user_id)->sum('cost');

    }

    $categories = Category::with('categories')->get();

    $randomCategories = Category::inRandomOrder()->limit(4)->get();

    return view('homepage', ['categories' => $categories, 'random_categories' => $randomCategories])

        ->with('bucket_count', $bucket_count)

        ->with('liked_count', $liked_count)

        ->with('bucket_sum', $bucket_sum);

}

public function top()
{
    $bucket_count = 0;

    $liked_count = 0;

    $bucket_sum = 0;

    if(auth()->user() != null) {

```

					ІІЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$user_id = auth()->user()->id;

$bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

$liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

$bucket_sum = DB::table('user_bucket')

->join('products', 'user_bucket.product_id', '=', 'products.id')

->where('user_id', $user_id)->sum('cost');

}

```

```

$randomCategories = Category::inRandomOrder()->limit(32)->get();

return view('categories', ['categories' => $randomCategories])

->with('bucket_count', $bucket_count)

->with('liked_count', $liked_count)

->with('bucket_sum', $bucket_sum);

}

```

public function categories()

```

{

    $bucket_count = 0;

    $liked_count = 0;

    $bucket_sum = 0;

    if(auth()->user() != null) {

        $user_id = auth()->user()->id;

        $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

        $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

        $bucket_sum = DB::table('user_bucket')

        ->join('products', 'user_bucket.product_id', '=', 'products.id')

        ->where('user_id', $user_id)->sum('cost');

    };

}

```

```

$categories = Category::with('categories')->get();

```

					ІІЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

return view('categories', ['categories' => $categories])

->with('bucket_count', $bucket_count)

->with('liked_count', $liked_count)

->with('bucket_sum', $bucket_sum);
}

public function deleteliked(Request $request) {

    UserLiked::where('id', $request->id)->delete();

    $user_id = auth()->user()->id;

    $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

    $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

    $bucket_sum = DB::table('user_bucket')

        ->join('products', 'user_bucket.product_id', '=', 'products.id')

        ->where('user_id', $user_id)->sum('cost');

    $products = DB::table('products')

        ->join('subcategories', 'products.subcategory_id', '=', 'subcategories.id')

        ->join('categories', 'subcategories.category_id', '=', 'categories.id')

        ->join('user_liked', 'products.id', '=', 'user_liked.product_id')

        ->select('user_liked.id as liked_id', 'products.*', 'subcategories.name as subcategory_name', 'categories.name as
category_name')

        ->where('user_id', $user_id)

        ->get();

    $categories = Category::with('categories')->get();

    return view('liked')->with('liked', $products)

        ->with('bucket_count', $bucket_count)

        ->with('liked_count', $liked_count)

        ->with('bucket_sum', $bucket_sum);
}

```

					ІІЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function like(Request $request) {

    if (auth()->user() == null) {

        return redirect('login');

    }

    $user_id = auth()->user()->id;

    $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

    $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

    $bucket_sum = DB::table('user_bucket')

        ->join('products', 'user_bucket.product_id', '=', 'products.id')

        ->where('user_id', $user_id)->sum('cost');


    $product = Product::where('id', $request->id)->first();

    $user_id = auth()->user()->id;

    DB::table('user_liked')->insert([

        'user_id' => $user_id,

        'product_id' => $product->id

    ]);

    return redirect('liked')

        ->with('bucket_count', $bucket_count)

        ->with('liked_count', $liked_count)

        ->with('bucket_sum', $bucket_sum);

}


public function bucket(Request $request) {

    if (auth()->user() == null) {

        return redirect('login');

    }

    $user_id = auth()->user()->id;

    $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

    $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

```

					ІІЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$bucket_sum = DB::table('user_bucket')

->join('products', 'user_bucket.product_id', '=', 'products.id')

->where('user_id', $user_id)->sum('cost');

$product = Product::where('id', $request->product_id)->first();

$user_id = auth()->user()->id;

DB::table('user_bucket')->insert([

    'user_id' => $user_id,

    'product_id' => $product->id

]);

Product::where('id', $product->id)->update(['amount' => $product->amount - 1]);

//delete only if we're deleting from liked section

if ($request->isLiked != null) {

    $recordId = DB::table('user_liked')->where(['user_id' => $user_id, 'product_id' => $product->id])->orderBy('id')-
>first();

    UserLiked::where('id', $recordId->id)->delete();

}

return redirect('inbucket')

->with('bucket_count', $bucket_count)

->with('liked_count', $liked_count)

->with('bucket_sum', $bucket_sum);

}

public function liked() {

    if (auth()->user() == null) {

        return redirect('login');

    }

```

```

$user_id = auth()->user()->id;

$bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

$liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

$bucket_sum = DB::table('user_bucket')

->join('products', 'user_bucket.product_id', '=', 'products.id')

->where('user_id', $user_id)->sum('cost');


$products = DB::table('products')

->join('subcategories', 'products.subcategory_id', '=', 'subcategories.id')

->join('categories', 'subcategories.category_id', '=', 'categories.id')

->join('user_liked', 'products.id', '=', 'user_liked.product_id')

->select('user_liked.id as liked_id', 'products.*', 'subcategories.name as subcategory_name', 'categories.name as
category_name')

->where('user_id', $user_id)

->get();

return view('liked')->with('liked', $products)

->with('bucket_count', $bucket_count)

->with('liked_count', $liked_count)

->with('bucket_sum', $bucket_sum);
}

public function inbucket() {

    if (auth()->user() == null) {

        return redirect('login');

    }

    $user_id = auth()->user()->id;

    $bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

    $liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

```

```

$bucket_sum = DB::table('user_bucket')

->join('products', 'user_bucket.product_id', '=', 'products.id')

->where('user_id', $user_id)->sum('cost');

$products = DB::table('products')

->join('subcategories', 'products.subcategory_id', '=', 'subcategories.id')

->join('categories', 'subcategories.category_id', '=', 'categories.id')

->join('user_bucket', 'products.id', '=', 'user_bucket.product_id')

->select('user_bucket.id as bucket_id', 'products.*', 'subcategories.name as subcategory_name', 'categories.name as
category_name')

->where('user_id', $user_id)

->get();

return view('bucket')->with('bucket', $products)

->with('bucket_count', $bucket_count)

->with('liked_count', $liked_count)

->with('bucket_sum', $bucket_sum);

}

public function search() {

$user_id = auth()->user()->id;

$bucket_count = DB::table('user_bucket')->where('user_id', $user_id)->count();

$liked_count = DB::table('user_liked')->where('user_id', $user_id)->count();

$bucket_sum = DB::table('user_bucket')

->join('products', 'user_bucket.product_id', '=', 'products.id')

->where('user_id', $user_id)->sum('cost');

$products = Product::inRandomOrder()->limit(32)->get();

return view('products')->with('products', $products)

->with('bucket_count', $bucket_count)

->with('liked_count', $liked_count)

```

```

        ->with('bucket_sum', $bucket_sum);

    }

}

<!doctype html>

<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">

<head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- CSRF Token -->

    <meta name="csrf-token" content="{{ csrf_token() }}">

    <!-- Fonts -->

    <link rel="dns-prefetch" href="//fonts.bunny.net">

    <link href="https://fonts.bunny.net/css?family=Nunito" rel="stylesheet">

    {{-- <link rel="stylesheet" href="{{ URL::asset('css/main.css') }}">--}}

    {{-- <link rel="stylesheet" href="{{ URL::asset('css/fonts.css') }}">--}}

    {{-- <link rel="stylesheet" href="{{ URL::asset('css/admin.css') }}">--}}

    {{-- <link rel="stylesheet" href="{{ URL::asset('css/app.css') }}">--}}

    <script src="{{ URL::asset('js/app.js') }}" defer></script>

    <!-- Scripts and Styles -->

    @vite(['resources/sass/app.scss', 'resources/js/app.js'])

    @vite(['public/css/main.css', 'public/css/fonts.css'])

    @vite(['public/css/admin.css'])

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Sportify</title>

```

```

</head>

<body>

  <div id="app">

    <header class="hole_container">

      <div class="main_container vertical_center">

        <div class="header_wrapper display_flex">

          <div class="header_contact display_flex">

            <div class="header_contact_element display_flex vertical_center">

              <div class="header_nav_logo">

                <div class="logo_container vertical_center">

                  <a href="{{ route('home') }}"></a>

                </div>

              </div>

            </div>

            <div class="header_contact_element display_flex header_text vertical_center"></div>

            <div class="header_contact_element display_flex header_text vertical_center">Sportify - ваш найкращий
вибір!</div>

          </div>

          <div class="header_right display_flex">

            <div class="header_right_login display_flex vertical_center">

              <div class="header_text display_flex">

                @guest

                @if (Route::has('login'))

                  <li class="nav-item">

                    <a class="nav-link" href="{{ route('login') }}">{{ __('Login') }}</a>

                  </li>

                @endif

                @if (Route::has('register'))

```

```

<li class="nav-item">

    <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>

</li>

@endif

@else

<li class="nav-item dropdown">

    <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-bs-
toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>

        {{ Auth::user()->name }}

    </a>

    <div class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdown">

        <a class="dropdown-item" href="{{ route('logout') }}"

            onclick="event.preventDefault();

                document.getElementById('logout-form').submit();"

            {{ __('Logout') }}

        </a>

        <form id="logout-form" action="{{ route('logout') }}" method="POST" class="d-none">

            @csrf

        </form>

    </div>

</li>

@if(Auth::user()->is_admin == 1)

    <div class="to-admin vertical_center">

        <a class="nav-link" href="{{ route('admin') }}"></a>

    </div>

@endif

@endguest

</div>

```

```

        </div>

    </div>

</div>

</div>

</header>

@yield('subheader')

<main id="main_container">

    @yield('content')

</main>

@yield('footer')

</div>

</body>

</html>

<?php

```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Support\Facades\DB;
use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    use HasFactory;

    protected $fillable = [
        'name', 'image', 'descr'
    ];

    public static function products() {

        return DB::table('products')
    }
}

```

					ІПЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		

```

->join('subcategories', 'products.subcategory_id', '=', 'subcategories.id')

->join('categories', 'subcategories.category_id', '=', 'categories.id')

->select('user_liked.id as liked_id', 'products.*', 'subcategories.name as subcategory_name', 'categories.name as
category_name')->get();

}

```

```

public static function getProducts() {

return DB::table('products')

->join('subcategories', 'products.subcategory_id', '=', 'subcategories.id')

->join('categories', 'subcategories.category_id', '=', 'categories.id')

->select('products.*', 'subcategories.name as subcategory_name', 'categories.name as category_name')->get();

}

```

```

public static function subcategories() {

return DB::table('subcategories')

->join('categories', 'subcategories.category_id', '=', 'categories.id')

->select('subcategories.id', 'subcategories.name as subcategory_name', 'categories.name as category_name')

->orderBy('categories.name')->orderBy('subcategories.name')

->get();

}

}

```

```
@extends('layouts.app')
```

```
@extends('layouts.subheader')
```

```
@extends('layouts.footer')
```

```
@section('content')
```

```
<div class="main_container">
```

```
<div class="search_container display_flex">
```

```
<div class="category_combobox_container display_flex" onclick="expandCategoryMenu()">
```

					ІІЗ.КР.Б – 121 – 23 – ІІЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

<div class="category_element">

</div>

<div class="category_element">Бєи катєропїї</div>

<div class="category_element">

</div>

</div>

<div class="searchbox_container">

    <form class="searchbox_container display_flex" action="search/product" method="post"> @csrf

        <input class="searchbox_input" type="text" placeholder="Що шукаєтє?" alt="" name="search_pattern"
required>

        <button class="searchbox_button">Пошук</button>

    </form>

</div>

</div>

<div class="main_container">

    <div class="main_category_block display_flex">

        <div class="category_list_container">

            @foreach($categories as $key => $category)

                <div class="category_list_element">

                    <form action="{{URL('categories/subcategories')}}" method="post"> @csrf

                        <input type="text" name="id" value="{{ $category->id }}" hidden/>

                        <button class="categories_button categories_button_list">{{ $category->name }}</button>

                    </form>

                </div>

                @foreach($category->categories as $key => $subcategory)

                    <div class="category_list_element category_list_subelement">

                        <form action="{{URL('categories/subcategories/products')}}" method="post"> @csrf

                            <input type="text" name="id" value="{{ $subcategory->id }}" hidden/>

```

```

        <button class="categories_button categories_button_list"> + {{ $subcategory->name }}</button>

    </form>

</div>

@endforeach

@endforeach

</div>

<div class="main_category_container display_flex">

    <div class="main_category_info">

        <div class="category_info_name">СПОРТИВНИЙ ОДЯГ</div>

        <div class="category_info_header">Одяг Від Відомих Брендів</div>

        <div class="category_info_description">Спортивний одяг для активних людей.</div>

        <a href="{{ route('categories') }}"><div class="button_to_category">ПЕРЕЙТИ</div></a>

    </div>

    <div class="main_category_image_wrapper">

        <div class="main_category_image">

        </div>

    </div>

</div>

</div>

</div>

<div class="main_container">

    <div class="categoryes_small_container display_flex">

        @foreach($random_categories as $key => $random_category)

            <div class="categoryes_small_element">

                <div class="categoryes_small_element_img">

                </div>

                <form action="{{ URL('categories/subcategories') }}" method="post" @csrf>

                    <input type="text" name="id" value="{{ $random_category->id }}" hidden/>

                    <button class="categoryes_small_element_name btn_homepage">{{ $random_category->name }}</button>

```

```
</form>

</div>

@endforeach

</div>

</div>

@vite(['public/js/main_page.js'])

<script src="js/main_page.js"></script>

@endsection
```