

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток підтримки діяльності магазину з продажу спортивних товарів»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-82-0 Квітка Євген Сергійович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
з оцінкою**

_____ «__» _____ 2022 р.

Науковий керівник

(підпис)

к.т.н., доц., Парфененко Ю. В.

(науковий ступінь, вчене звання, прізвище та ініціали)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2022

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

Зав. кафедри ІТ

_____ В. В. Шендрик
«05» жовтня 2021 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ

Квітка Євген Сергійович

1 Тема роботи Web-додаток підтримки діяльності магазину з продажу спортивних товарів

керівник роботи Парфененко Юлія Вікторівна, к.т.н., доцент,

затверджені наказом по університету від «27» квітня 2022 р. №0301-IV

2 Строк подання студентом роботи «08» червня 2022 р.

3 Вхідні дані до роботи технічне завдання на розробку web-додатку діяльності магазину з продажу спортивних товарів

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) аналіз предметної області, проектування web-додатку, розробка web-додатку

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання: 01.10.2021

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	15.03.2022 – 27.03.2022	
2	Оформлення планування робіт	28.03.2022 – 05.04.2022	
3	Оформлення технічного завдання	06.04.2022 – 10.04.2022	
4	Реалізація web-додатку	11.04.2022 – 04.05.2022	
5	Тестування web-додатку	05.05.2022 – 15.05.2022	
6	Впровадження web-додатку в дію	16.05.2022 – 26.05.2022	
7	Оформлення пояснювальної записки	28.05.2022 – 09.06.2022	

Студент

(підпис)

Квітка Є. С.

Керівник роботи

(підпис)

к.т.н., доц. Парфененко Ю. В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток підтримки діяльності магазину з продажу спортивного товарів».

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку використаних джерел із 17 найменувань, трьох додатків. Загальний обсяг пояснювальної записки складає 76 сторінки, у тому числі 42 сторінки основного тексту, 2 сторінки списку використаних джерел, 32 сторінок додатків.

Кваліфікаційну роботу бакалавра присвячено розробці web-додатку підтримки діяльності магазину з продажу спортивного товарів.

У першому розділі досліджено предметну область, визначено переваги та недоліки та проведено огляд останніх досліджень і публікацій. Також було сформовано мету проекту та задачі кваліфікаційної роботи бакалавра.

У другому розділі було реалізовано проектування інформаційної системи за допомогою IDEF0 та IDEF1, спроектовано базу даних.

У третьому розділі описується програмна реалізація web-додатку, сформовано архітектуру web-додатку.

Результатом кваліфікаційної роботи бакалавра є розроблений web-додаток підтримки діяльності магазину з продажу спортивних товарів.

Ключові слова: web-додаток, JavaScript, React, PostgreSQL, JWT, спорт, створення інформаційних систем.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз програмних продуктів – аналогів	10
1.3 Постановка задачі.....	13
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	16
2.1 Структурно-функціональне моделювання в IDEF0	16
2.2 Моделювання використання web-додатку підтримки діяльності магазину з продажу спортивних товарів	18
2.3 Проектування бази даних.....	19
3 РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ДІЯЛЬНОСТІ МАГАЗИНУ З ПРОДАЖУ СПОРТИВНИХ ТОВАРІВ	21
3.1 Архітектура web-додатку	21
3.2 Програмна реалізація.....	22
3.3 Використання програмного додатку.....	29
ВИСНОВОК.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТОК А.....	45
ДОДАТОК Б	56
ДОДАТОК В.....	64

ВСТУП

На сьогодні можна сказати, що web-додатки мають важливу роль у повсякденному житті людини та для створення нових можливостей.

Web-додаток – це колекція загальнодоступних, поєднаних між собою однією темою web-сторінок, які мають однакове доменне ім'я.

Через велику кількість переваг в web-додатках зручніше робити покупки в Інтернеті, ніж ходити по звичайним торговим центрам.

Основними перевагами онлайн магазинів є:

- відсутність черг;
- не потрібно йти до торгових центрів за товарами;
- онлайн покупки можна робити в будь-якому місці, та в будь-який час;
- в інтернет магазинах можна зручніше порівнювати та досліджувати товари та їх ціну;
- вміст більш-розширеного асортименту товару ніж у торгових центрах.

З появою інтернет-магазинів з'явилася можливість продавати товари в різних містах і країнах, знявши всі географічні обмеження. Крім того, однією з головних переваг онлайн-покупок є доступність. Покупки в Інтернеті можуть заощадити час як покупців, так і продавців, зменшуючи кількість доступних телефонних дзвінків, перегляд технічних характеристик, годин роботи та іншу інформацію, яку легко знайти на сторінках компаній та продуктів [1].

На відміну від звичайних магазинів в торгових центрах, інтернет-магазини територіально необмежені, тим самим охоплення клієнтів збільшується як і в межах країни так і за кордоном.

Метою кваліфікаційної роботи бакалавра є розробка web-додатку підтримання діяльності магазину з продажу спортивних товарів «Kvitka sport shop».

Для виконання мети проекту потрібно виконати наступні задачі:

- визначення актуальності даної роботи, проведення аналізу аналогів та дослідження предметної області;
- створення технічного завдання;
- створення макетів та розробка web-додатку;
- розробити web-додаток з базою даних та провести його тестування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

Електронна комерція відноситься до процесу купівлі та продажу товарів і послуг в Інтернеті, хоча в останні роки, а тим більше під час пандемії, досвід покупок вийшов за межі цього. Наприклад, роздрібні продавці, які мають фізичне місцезнаходження, можуть включити електронну комерцію, дозволяючи клієнтам переглядати та купувати в Інтернеті та забирати в магазині.

Як і будь-яка цифрова технологія, електронна комерція продовжує швидко розвиватися. Починаючи з 1990-х років, електронна комерція почала орієнтуватися на споживчий ринок, і почали з'являтися інтернет-магазини. Тепер практично будь-який власник бізнесу або підприємець може використовувати платформу електронної комерції, щоб відкрити інтернет-магазин за лічені хвилини [2].

За статистикою, в Україні вже більше 25% всього населення веде здоровий спосіб життя, і ця цифра, за словами соціологів, неухильно зростає.

На фоні популярності ведення здорового образу життя не тільки зовсім нові види спорту, але і істотним чином розширюється асортимент спортивних товарів, які користуються великим спроможністю. Саме тому відкриття власного магазину спортивних товарів на сьогоднішній день знову набуло актуальність і хороші перспективи в бізнесі [3].

Технології для створення web-додатків існує велика кількість, проте вона завжди ділиться на дві частини:

– Front-end технології (розробка зі сторони клієнтської частини). Вони використовуються для розробки web-елементів, з які бачать користувачі та і з якими взаємодіють. Це включає кольори тексту, різні розміри та формати зображень, створення web-елементів та навігаційне меню [4].

– Back-end технології (розробка зі сторони серверної частини). Дані технології потрібні для розробки технічної сторони (зберігання та обробка даних та валідація, що все на інтерфейсі коректно працює). Наприклад, користувач авторизується до web-додатку та надає свої дані для авторизації, потім ця інформація збирається та відправляє на сторону серверу, щоб перевірити, чи ці облікові дані вірні. Після підтвердження облікових даних сервер відправить назад на клієнтську частину відповідь, яка міститиме інформацію, що дані вірні або навпаки.

Схематичне представлення, як працює серверна та клієнтська частина між собою зображено на рисунку 1.1

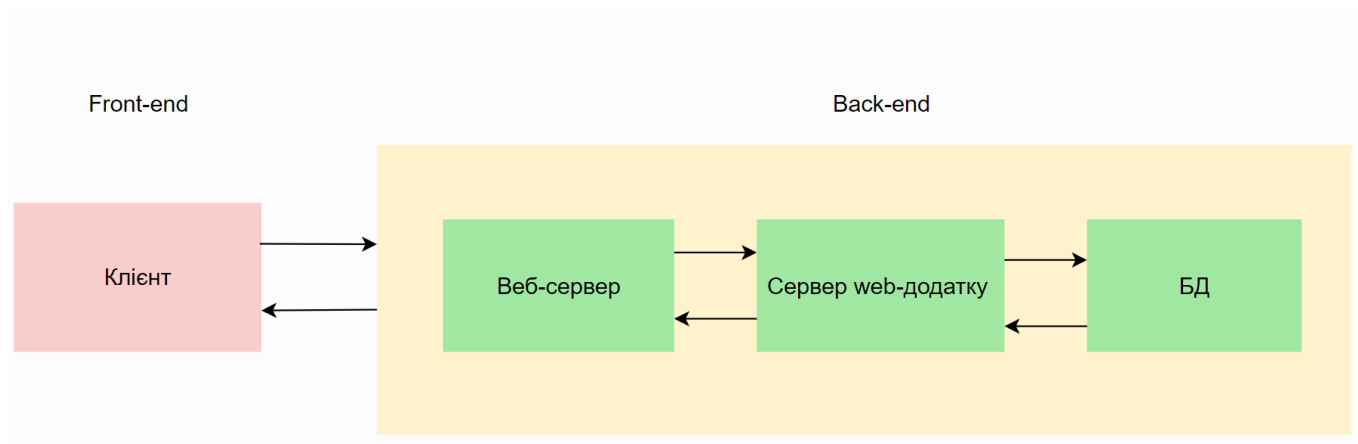


Рисунок 1.1 – Взаємодія роботи серверної та клієнтської частини

Інтернет-магазини можуть не мати роздрібних продавців чи гардеробних, але вони можуть рекомендувати товари на основі інтересів клієнтів, вибору стилю та переваг, використовуючи дані. Майбутнє онлайн-покупок залежить від фіксації моделей торгової поведінки. Це дозволяє бренду створювати стратегічне ціноутворення, рекламу та перехресне просування як на індивідуальній основі, так і на основі широко поширених тенденцій. Наприклад, бренд може показувати рекламу в соціальних мережах в години пік покупок, рекламувати товари, які часто купують разом, або показувати рекламу певного товару, який споживач раніше переглядав.

1.2 Аналіз програмних продуктів – аналогів

У наш час існує багато вже готових рішень інтернет магазинів. І тому велика кількість компаній та організацій замовляють або наймають розробників для розробки готового рішення або для підтримання його в тренді.

Серед основних інтернет магазинів, які спеціалізуються на спортивних товарах можна виділити наступні web-ресурси:

– «Decathlon» – французька компанія, яка має власне виробництво, та має спортивні товари для більше 70 видів спорту. Головна сторінка має світлий дизайн, шрифт та розмір є читабельним для користувачів. Всі web-елементи на сторінці виглядають логічно зрозумілі [5];

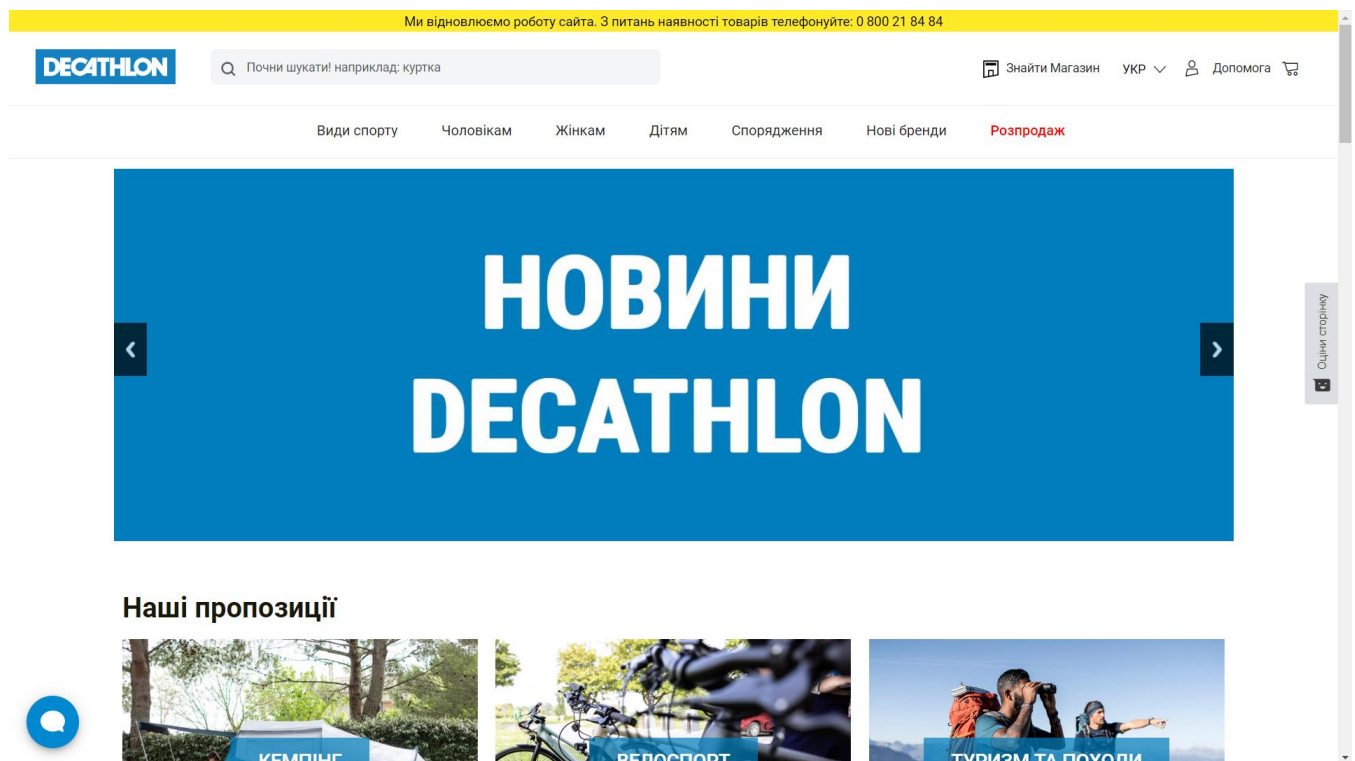


Рисунок 1.2– Головна сторінка web-додаток «Decathlon»

– «Adidas» – німецька компанія, яка розробляє спортивний одяг, взуття, та аксесуари. Найбільший виробник спортивних товарів у європейських країнах. На

головній сторінці можемо побачити досить стильний дизайн, в центрі є великий слайдер з рекламою спортивного товару. Навігація web-додатку інтуїтивно-зрозуміла [6];

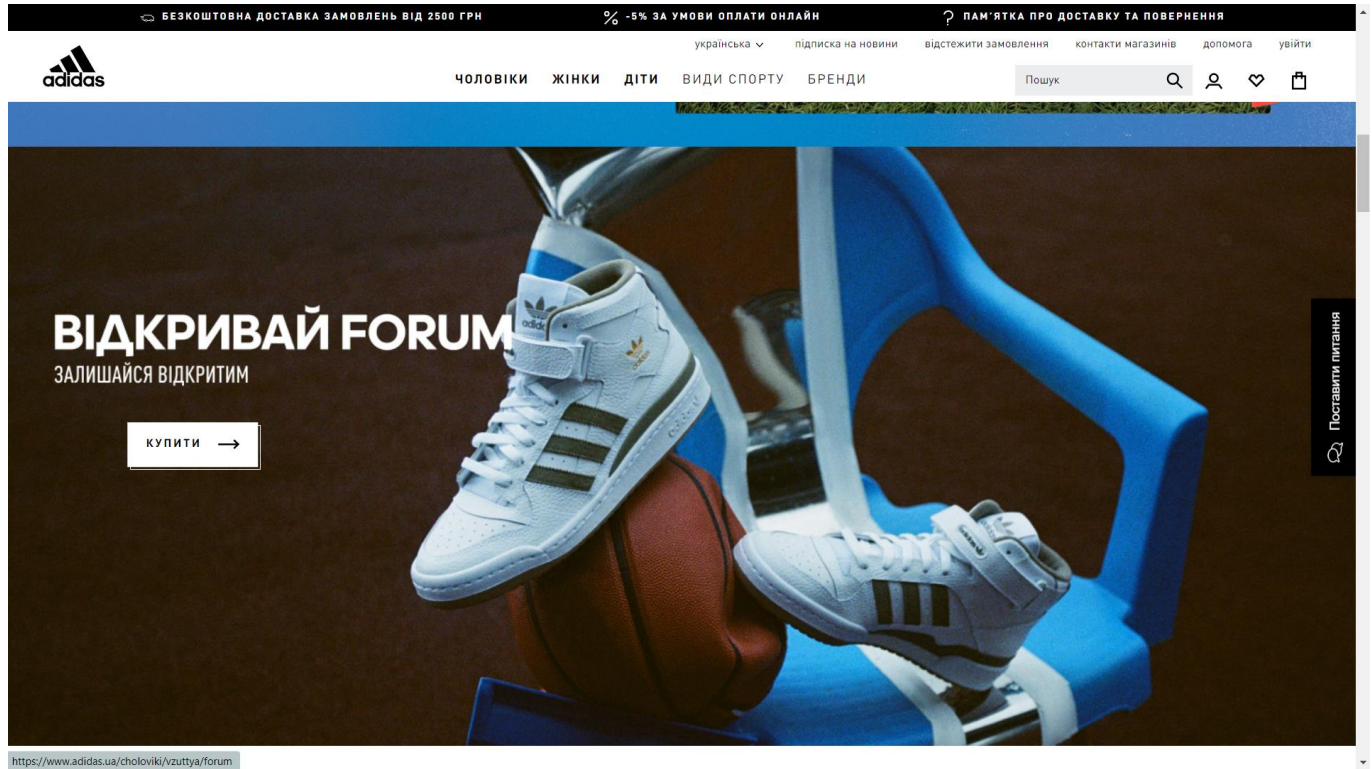


Рисунок 1.3 – Головна сторінка web-додатку «Adidas»

– «Rozetka» – українська продуктова компанія, яка має власний маркетплейс різних категорій, та самий найбільший онлайн-ритейлер в країні. На цьому продуктовому web-додатку можемо побачити величезну категорію спортивних товарів. Інтуїтивність на даному web-додатку не легка, потрібно знати як працювати с сортуванням по категоріям [7].

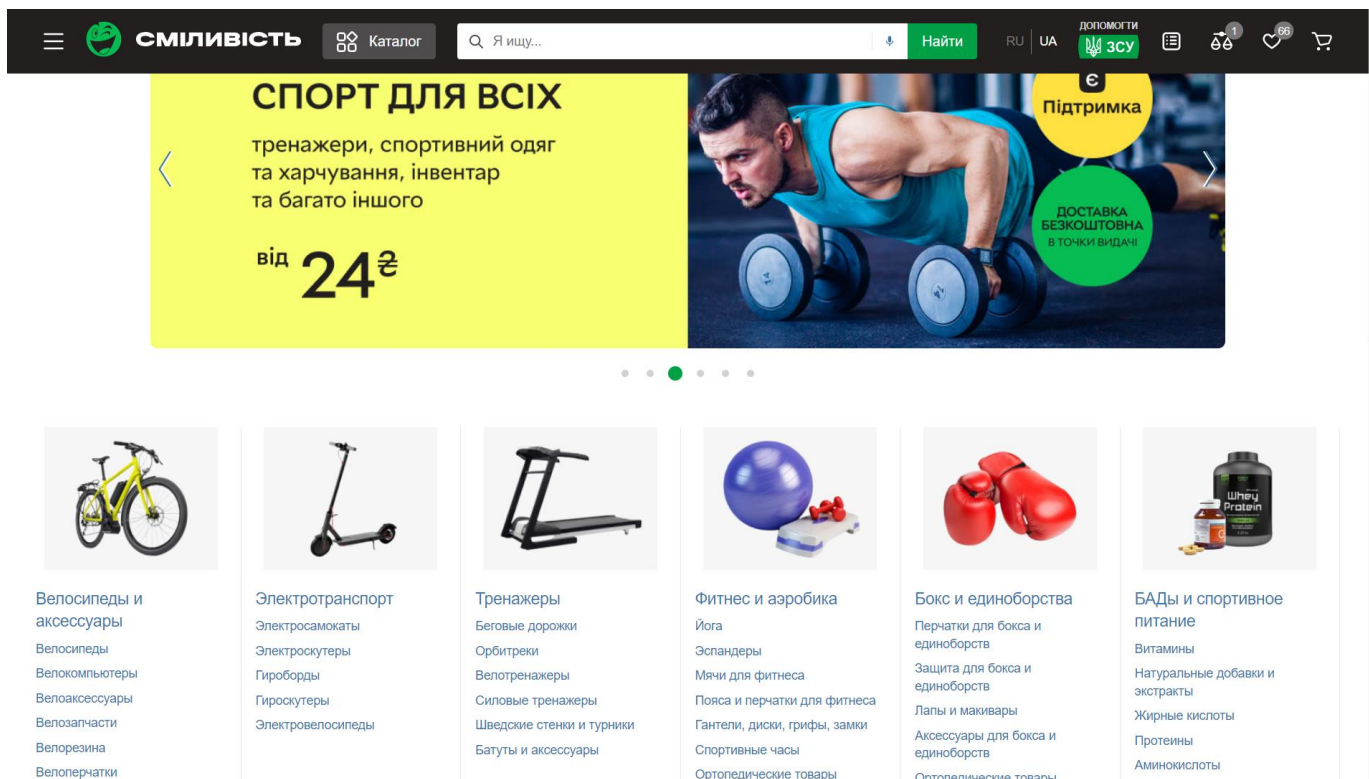


Рисунок 1.4 – Головна сторінка web-додатку «Rozetka»

Проаналізувавши та переглянувши аналоги, які показані вище, можна зробити висновок, що всі web-платформи мають схожу структуру та схожий функціонал. На кожному web-додатку є навігаційне меню, каталог товарів та характеристики про кожен товар. В кінці web-сторінок виведена інформація про саму web-платформу, та контакти для зворотнього зв'язку.

Загалом можна сказати, що онлайн-магазини з продажу спортивних товарів в майбутньому матиме великий потенціал розвитку для розширення ринку продаж [8].

Після проведення аналізу різноманітних web-додатків, була створена порівняльна таблиця 1.1.

Таблиця 1.1 – Порівняльна таблиця аналогів

Назва критерію	«Decathlon»	«Adidas»	«Rozetka»
Дизайн web-додатку	+	+	-
Зручність у навігації	-	+	-
Функція зворотнього зв'язку	+	+	-
Авторизація у web-додатку	+	+	+
Перевантаженість контентом	-	-	+
Адаптивність web-додатку	-	+	-

1.3 Постановка задачі

Основна мета проекту полягає у розробці web-додатку підтримки діяльності магазину з продажу спортивних товарів, призначенням якого є перегляд товарів, додавання до кошику, та оформлення замовлень.

Цільовою аудиторією є люди, які зацікавлені в купівлі спортивного товару. Основні вимоги дизайну до web-додатку: однаковий стиль до всіх сторінок web-додатку, інформація викладена на web-додаток має бути зрозуміла та лаконічно викладена, використання слайдеру з зображеннями на головній сторінці. Усі медіа-файли на web-додатку мають бути якісними та чіткими.

Види та розміри шрифтів мають бути лаконічними з дизайном та комфортними для перегляду користувачів. Навігаційні елементи повинні бути зручні та зрозумілі користувачу. Дизайн web-додатку повинен мати кросбраузерність в наступних браузерах: Google Chrome, Opera, Mozilla Firefox. Дизайн web-додатку має бути адаптивний до всіх пристроїв (ноутбуки, телефони).

Основні вимоги до функціоналу web-додатку: функція вибору за брендом та категорією спортивних товарів, додавання до кошику товарів, оформлення замовлення товарів.

Технічне завдання на розробку web-додатку описано в додатку А.

Для розробки проекту, було вибрано створення додатку з використанням таких технологій, як:

- JavaScript;
- CSS;
- React.js;
- Axios;
- PostgreSQL;
- Express;
- Node.js;
- Sequelize.

JavaScript використовують в багатьох аспектах розробки. Це сценарій мови, що використовується для створення та контролю динамічного вмісту web-додатку, тобто всього, що рухається, оновлюється чи іншим чином змінюється на екрані, не вимагаючи перезавантаження web-сторінки вручну [9].

CSS (Cascading Style Sheets) – це мова, яку розробники можуть використовувати для створення web-додатку. Мова стилів описує представлення певного web-додатку та його макет. Ця мова є основною технологією, яку web-розробники використовують для проектування та створення web-додатків.

React.js – javascript бібліотека для створення користувацьких інтерфейсів [10, 11].

Axios – бібліотека с відкритим вихідним кодом, що дозволяє робити запити HTTP на сервер.

PostgreSQL – об’єктно реляційна система управління базами даних (ORDBMS), що вільно розповсюджується, найбільш розвинена з відкритих СКБД і є реальною альтернативою комерційним БД [12].

Express – мінімалістичний та гнучкий фреймворк для програм Node.js, що надає широкий набір функцій для web-додатків. Також потрібний для створення надійного API.

Node.js – серверна платформа з відкритим вихідним кодом, для роботи з мовою JavaScript окремо від браузера (на сервері) [13].

Sequelize – зручний інструмент для організації взаємодії між платформою Node.js та реляційною базою даних без використання мови запитів SQL.

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Структурно-функціональне моделювання в IDEF0

IDEF0 означає визначення інтеграції для моделювання процесів — методологію створення загальнодоступних доменів, яка використовується для моделювання підприємств та їх процесів, щоб їх можна було зрозуміти та покращити. Це тип схеми.

IDEF0 використовують перед етапом імплементації, коли більш детально проектують процес роботи web-додатку.

Структурно-функціональне моделювання для web-додатку підтримки діяльності магазину з продажу спортивних товарів продемонстровано на рисунках 2.1 та 2.2.

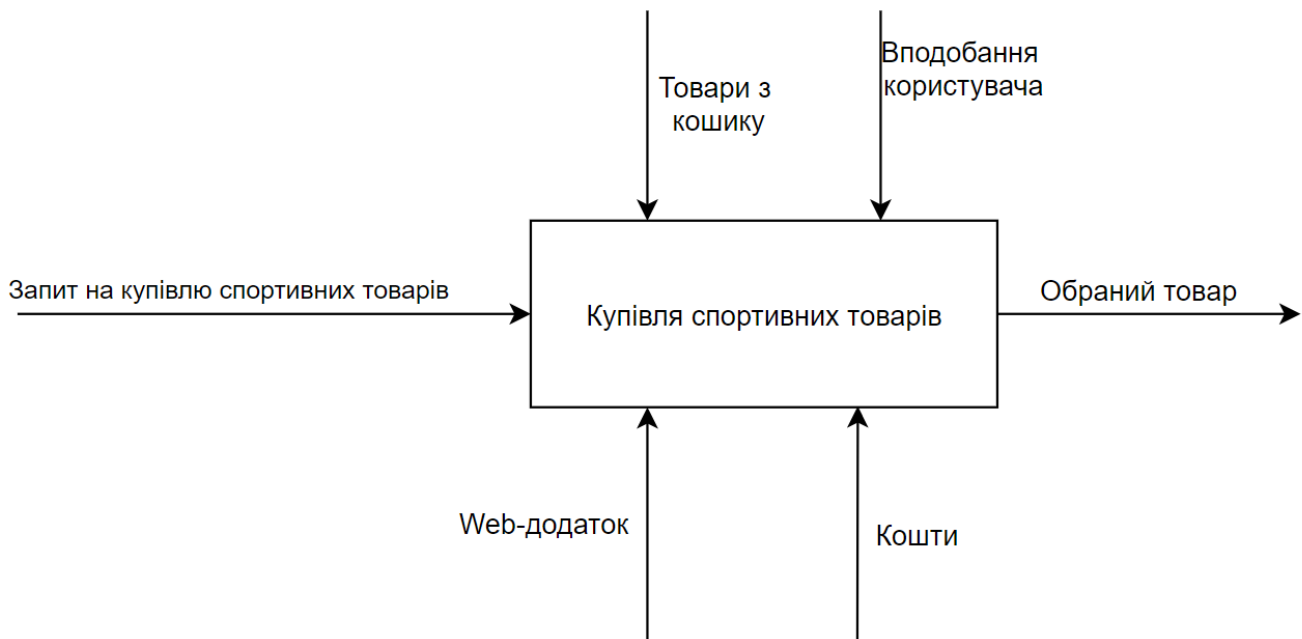


Рисунок 2.1 – Контекстна діаграма в нотації IDEF0

По середині схеми є головний блок, в якому вказується загальну операцію або головну дію.

По бокам прямокутника є стрілки (потоки) – це дані або конкретні об’єкти для підтримки опису роботи головного блоку.

В даній схемі містяться 4 потоки:

- Ліва стрілка: набір об’єктів або інших даних, яка потрібна для створення роботи виконання процесу.
- Стрілки зверху потрібні для задавання певних наборів правил або процесів для коректної роботи.
- Стрілки знизу використовуються для позначень спеціальних об’єктів за допомогою якою буде підтримуватись процес виконання роботи.
- Стрілка справа – отримання готового результату в результаті виконання певних попередніх процесів.

Декомпозиція першого рівня IDEF0 найбільше використовується для більш глибокого дослідження між інформаційними потоками взаємозв’язків для певного проекту[14].

Декомпозицію першого рівня використовують для вирішення завдань:

- Виявлення на ранньому етапі конфліктів, які виникли при недостатньому інформаційному зборі.
- Більш детально захоплюють зміст існуючих потоків під час виконання проекту.
- Потрібний для детального дослідження потоків, яким потрібно більше інформації для збільшення ефективності роботи.

Діаграма декомпозиції купівлі спортивних товарів з використанням web-додатку показана на рисунку 2.2.

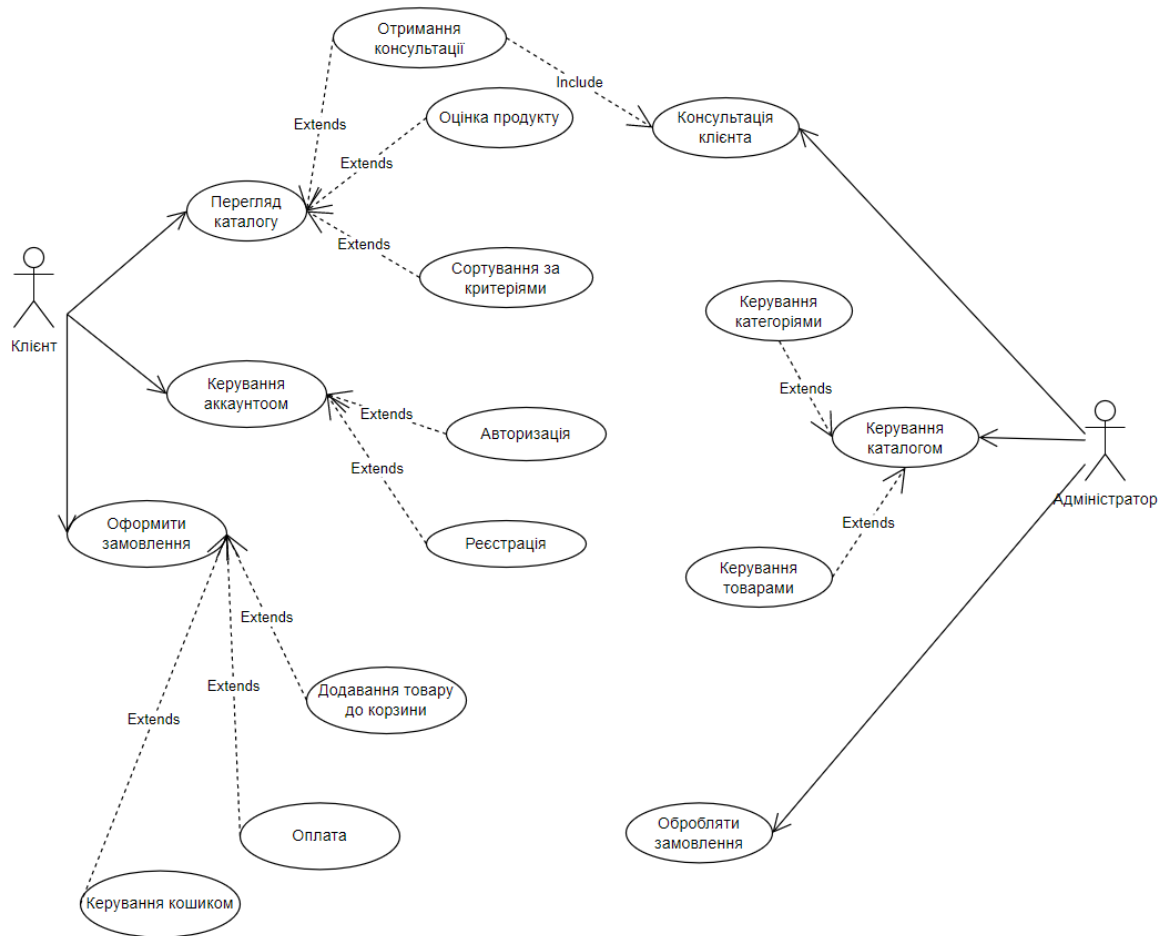


Рисунок 2.3 – Діаграма варіантів використання

2.3 Проектування бази даних

База даних є основною одиницею інформаційної системи. Вона включає сукупність даних із певної предметної області, якими потрібно керувати. Саме для цього використовують системи управління базами даних (СУБД)[16].

База даних інформаційної системи складатиметься з наступних таблиць:

Таблиця «baskets» - таблиця містить інформацію про індивідуальний кошик кожного користувача.

Таблиця «users» - таблиця, в якій зберігається інформація про користувачів веб-сайту, дату реєстрації та роль.

Таблиця «brand» - містить інформацію про назви брендів товарів.

Таблиця «ratings» - таблиця зберігає дані про оцінювання позицій товарів (один користувач може оцінити товар лише один раз).

Таблиця «type» - містить категорії позицій товарів.

Таблиця «product information» - таблиця містить повну інформацію про характеристику товарів

Таблиця «product» - таблиця містить дані про товар (ціна, рейтинг, зображення)

Таблиця «orders» - містить замовлення клієнта та його статус.

Структура бази даних зі зв'язками продемонстрована на рисунку 2.5

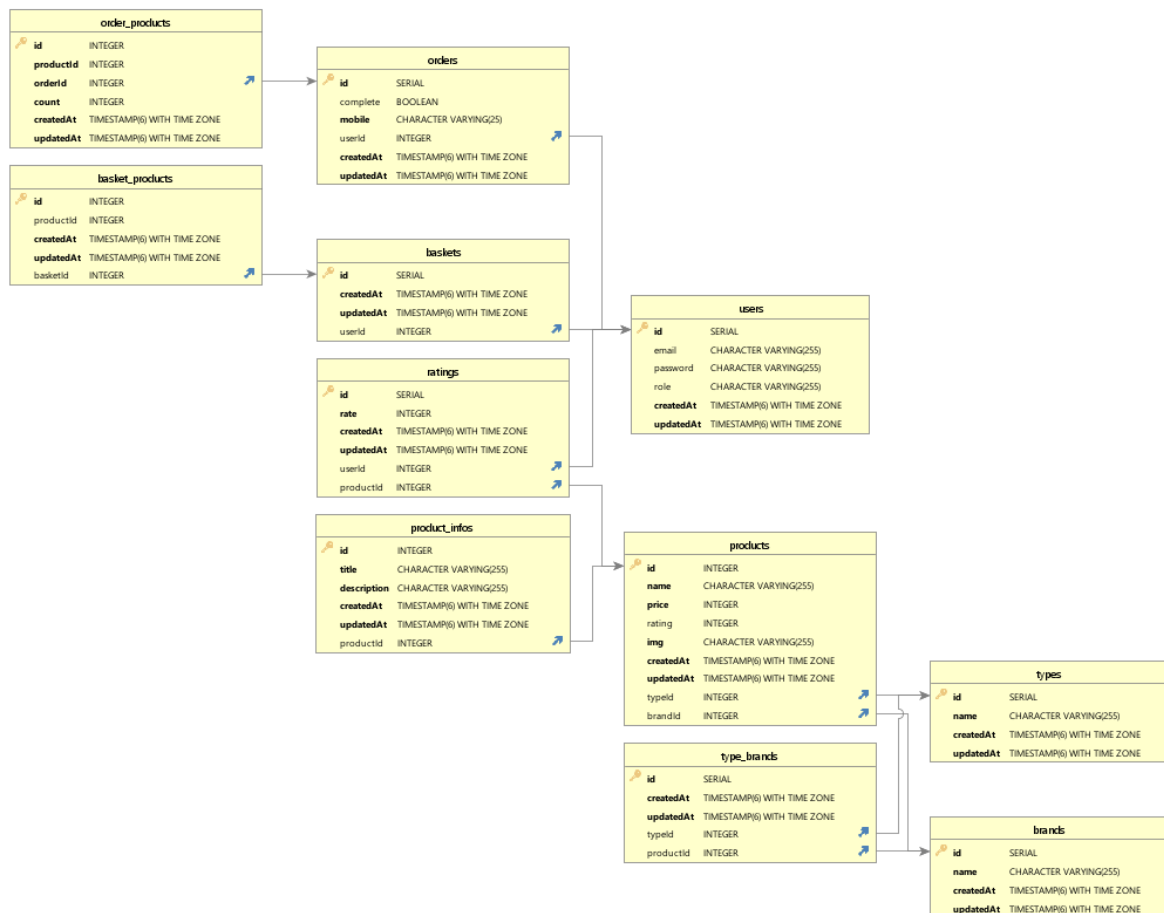


Рисунок 2.5 – Структура бази даних

3 РОЗРОБКА WEB-ДОДАТКУ ПІДТРИМКИ ДІЯЛЬНОСТІ МАГАЗИНУ З ПРОДАЖУ СПОРТИВНИХ ТОВАРІВ

3.1 Архітектура web-додатку

Архітектура web-додатку підтримки діяльності магазину з продажу спортивних товарів створена за сучасним та популярним принципом, в якій загальну структуру розділено на дві частини:

- Взаємодія з клієнтом (front-end) – в цій частині клієнту відображаються web-елементи та інформація з бази даних.
- Взаємодія з сервером (back-end) – в даній частині розробляється основний функціонал проекту (логіка роботи web-додатку, спілкування за допомогою API з клієнтською частиною, CRUD).

Візуальне відображення архітектури зображено на рисунку 5.3.

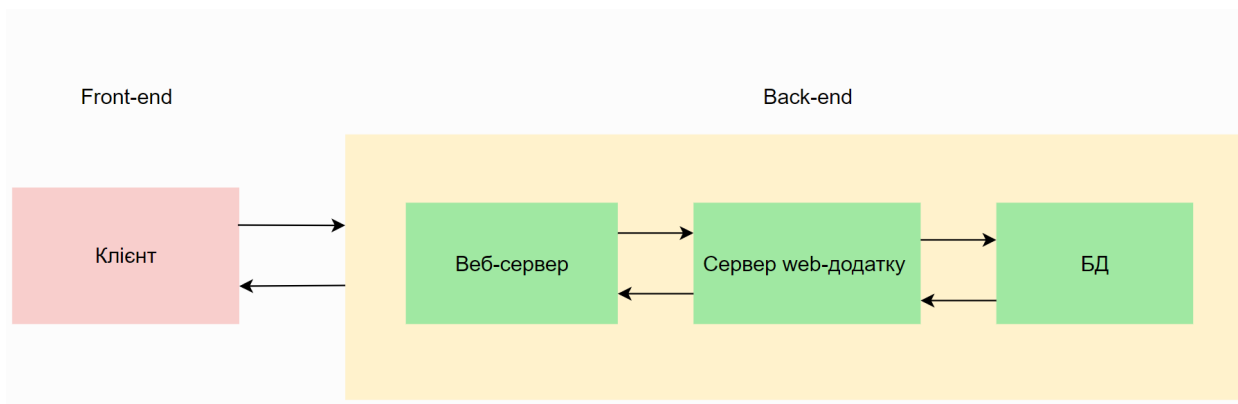


Рисунок 3.1 – Архітектура web-додатку

Клієнт, потрапивши в онлайн магазин, має можливість переглянути весь каталог товарів. Для того, щоб увійти, користувач переходить на сторінку авторизації або реєстрації. Наступним кроком, коли користувач натиснув на кнопку, на сервер відправились структуровані дані, потім дані проходять певні перевірки, та на основі

перевірок записуються до бази даних. В будь-якому випадку користувач повинен отримати відповідь.

Також при виконанні проекту потрібно притримуватись патерну MVC (Model View Controller) – за допомогою даного патерну можна розділити логіку програми, і це означає, що кожен модуль повинен бути незалежним та його легше тестувати, реалізовувати та підтримувати в подальшому (рис. 3.2)[17].

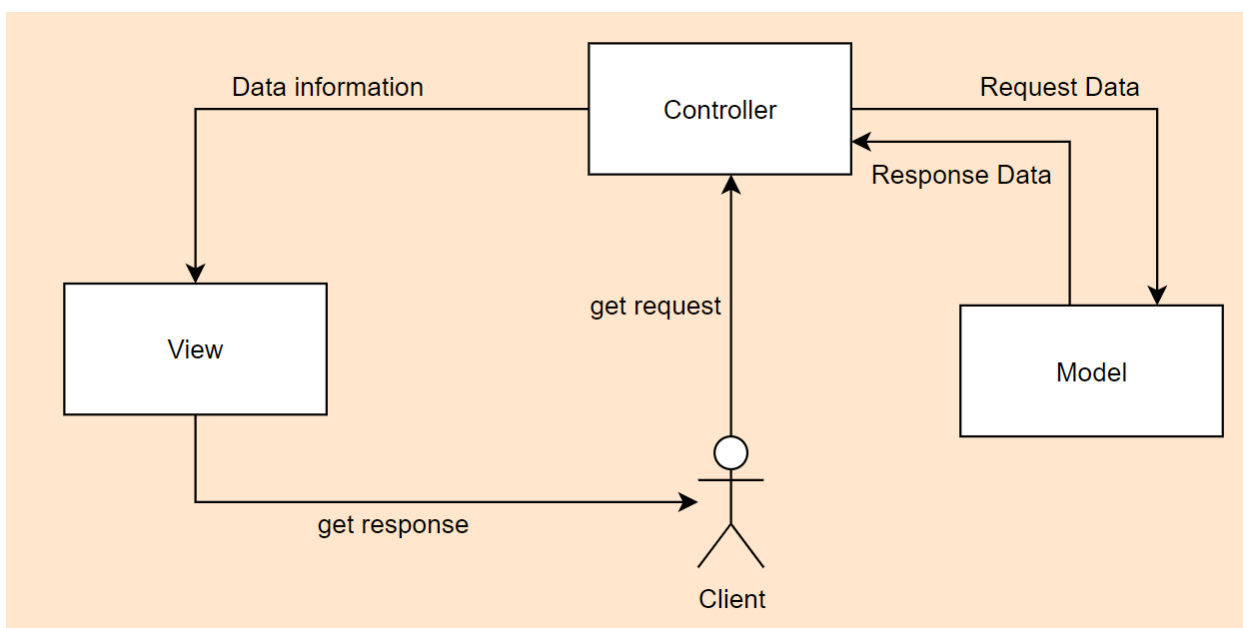


Рисунок 3.2 – Принцип роботи патерну MVC

3.2 Програмна реалізація

Структура web-додатку для підтримки діяльності магазину з продажу спортивних товарів має ієрархічний вигляд. З будь-якої сторінки користувачу надається можливість перейти на інші web-сторінки. Структура web-додатку зображена на рисунку 3.3.

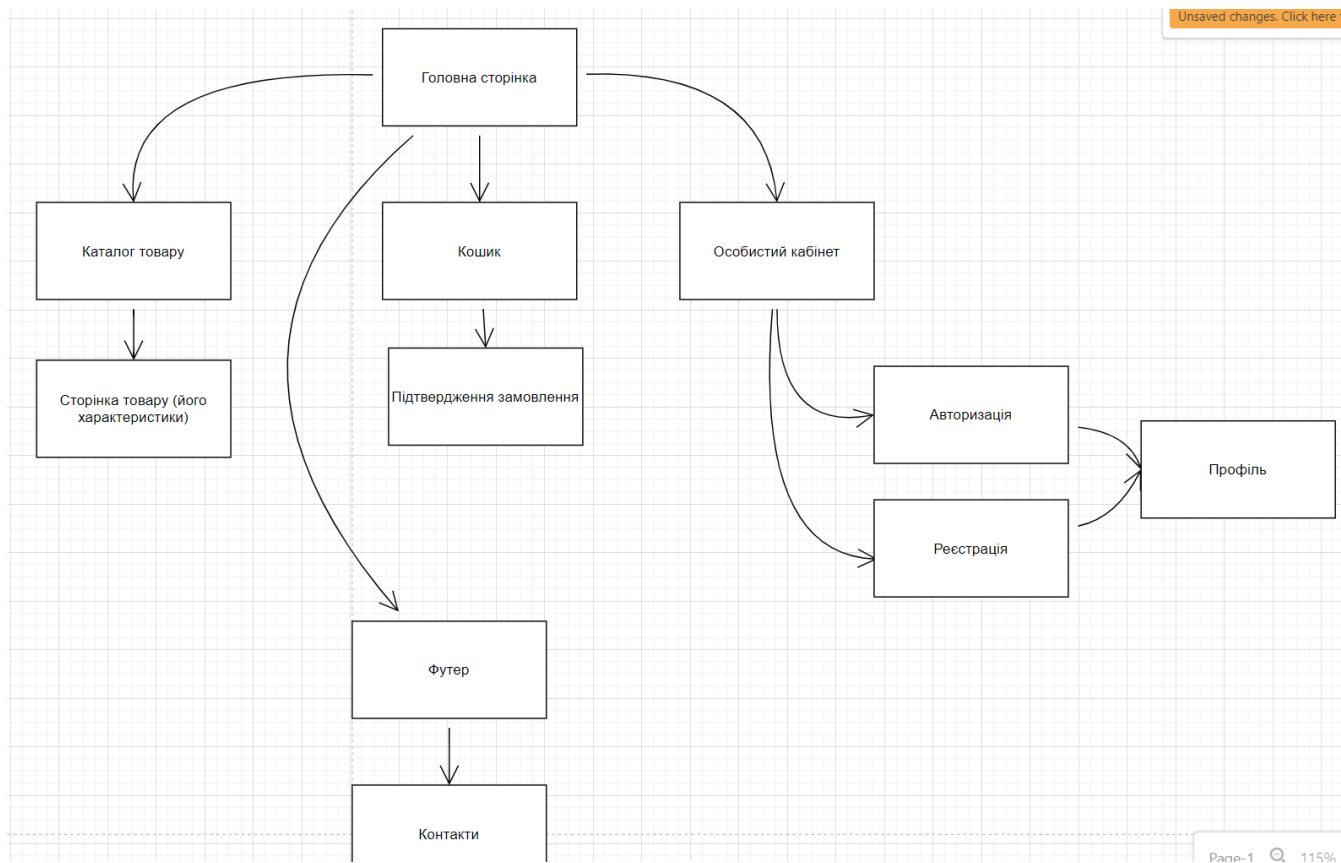


Рисунок 3.3 – Структура web-додатку для підтримки діяльності магазину

Наступним кроком після створення структури web-додатку є планування таблиць бази даних. Для даного проекту було прийнято рішення використовувати базу даних PostgreSQL. Для web-додатку з підтримки діяльності магазину з продажу спортивних товарів було створено таблиці, які зображено на рисунку 3.4.

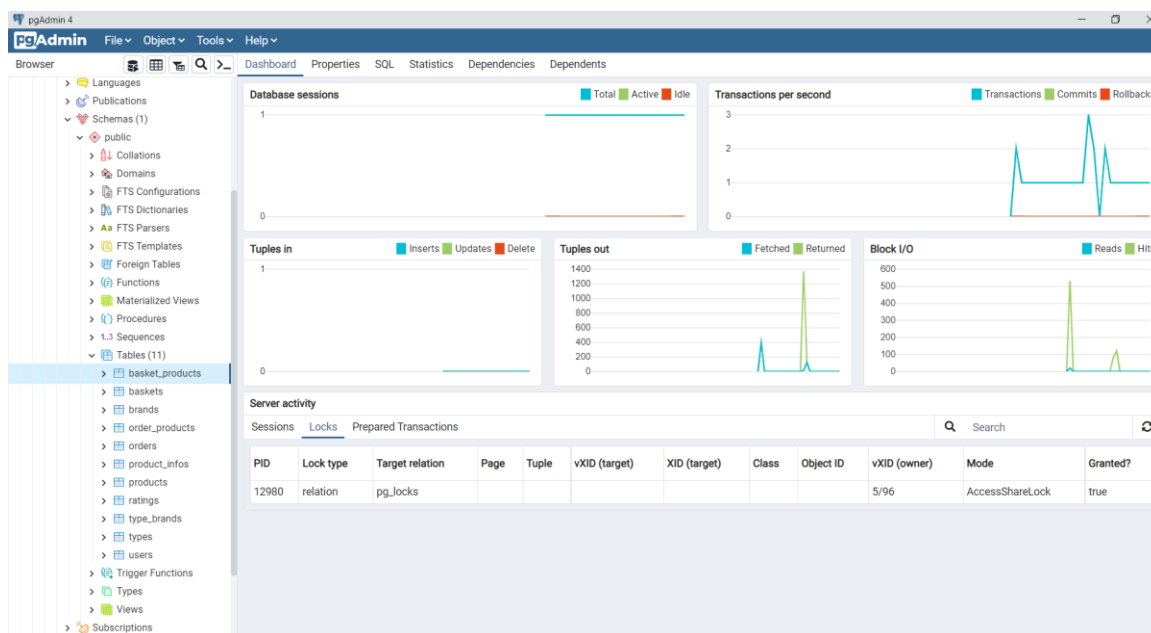


Рисунок 3.4 – Таблиці web-додатку

Web-додаток матиме однакову структуру та вигляд навігаційного меню (рис. 3.5) та підвалу сайту (рис 3.6), тому ці web-частини будуть зроблені в окремих файлах, на інших web-сторінках просто будуть імпортуватись, таким чином ми раціонально буде написано програмний код і його легше підтримувати та дороблювати.



Рисунок 3.5 – Навігаційне меню

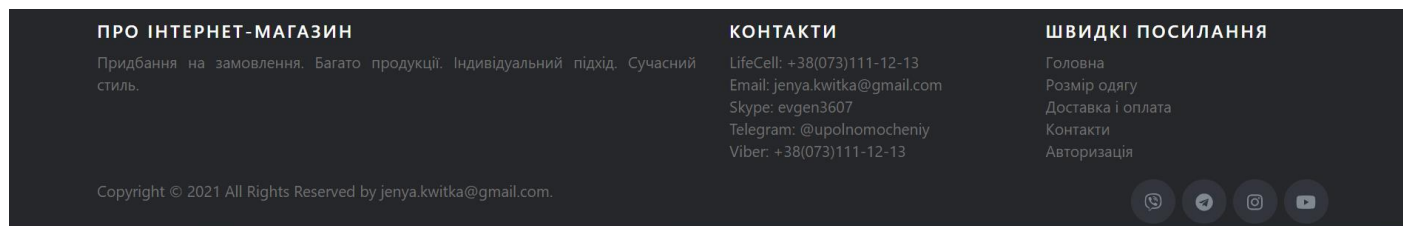


Рисунок 3.6 – Підвал сайту

Користувач перейшовши до каталогу, має можливість ознайомитись зі списком всіх товарів, які складаються з об'єктів у вигляді блоків (рис. 3.7).

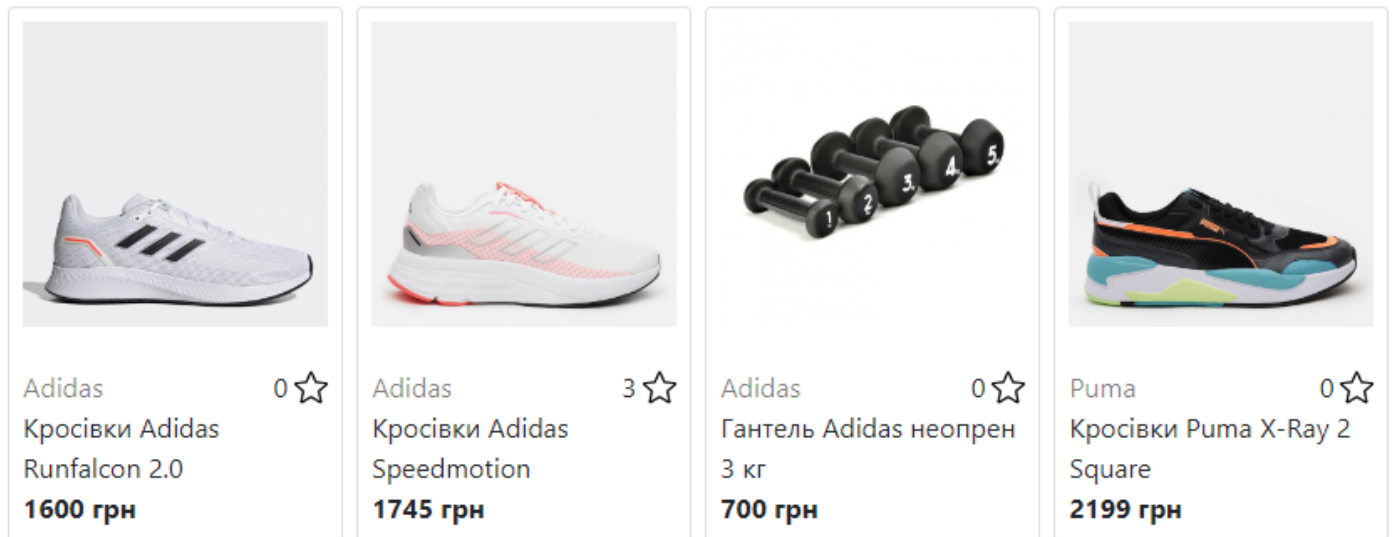


Рисунок 3.7 – Вигляд об'єктів каталогу у вигляді блоків

Програмний код виведення блоків наведений нижче:

```
<Card
  className="p-4"
  style={{ width: 300, cursor: "pointer" }}
  border={"Light"}
>
  <Image style={{ width: "100%" }} src={URL + product.id + product.img}/>
  <div className="d-flex justify-content-between align-items-center mt-4">
    <div className="text-black-50">{product && product.brand.name}</div>
    <div className="d-flex align-items-center">
      <Image className="ml-5" src={stars} style={{ width: "30px", height: "30px" }}/>
    </div>
  </div>
  <div>{product.name}</div>
  <div><b>{product.price} грн</b></div>
</Card>
```

Також користувач має можливість вибору категорій за власним вподобанням (рис. 3.8).

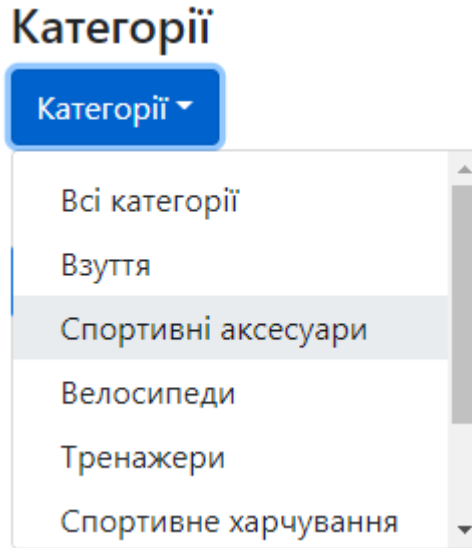


Рисунок 3.8 – Випадаюче меню з категоріями

Програмний код вибору категорій представлений нижче:

```
<DropDownItem  
  style={{ cursor: "pointer" }}  
  active={typeProduct.id === product.selectedType.id}  
  key={typeProduct.id}  
  onClick={() => product.setSelectedType(typeProduct)}  
>
```

Після пошуку певних товарів має можливість додати товар до кошика. Користувач має функції додавання та видалення товарів та посилання на перехід до товару. На рисунку 3.9 продемонстровано корзину з товаром.



Рисунок 3.9 – Корзина магазину

Програмний код додавання до корзини представлений нижче:

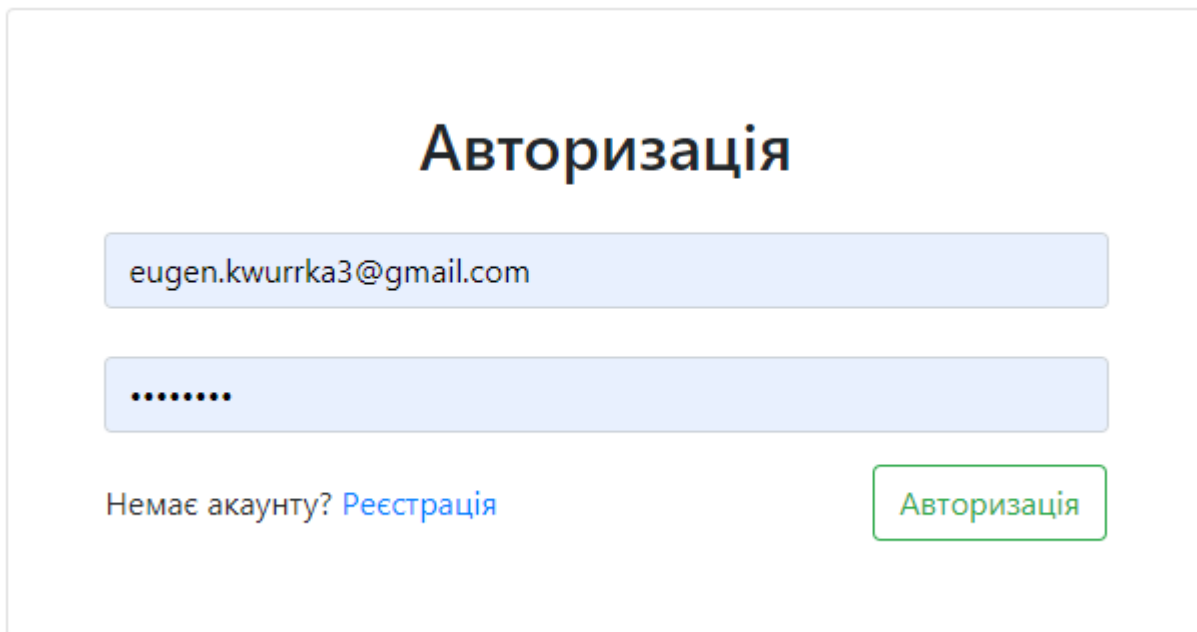
```
const addProduct = (product) => {
  if (user.isAuthenticated) {
    addProduct(product).then(() => basket.setBasket(product, true))
  } else {
    basket.setBasket(product);
  }
}
```

Наступним кроком було реалізовано підтвердження обраного замовлення. Замовлення можна підтвердити написавши номер телефону та натиснути на кнопку підтвердження (рис. 3.10).



Рисунок 3.10 – Підтвердження замовлення

Для того, щоб товар зберігався в кошику користувача, користувачу має авторизуватись або зареєструватись у web-додатку. Web-сторінка авторизації зображена на рисунку 3.11.



Авторизація

eugen.kwurrka3@gmail.com

.....

Немає акаунту? [Реєстрація](#)

Авторизація

Рисунок 3.11 – Сторінка авторизації

Програмний код авторизації наведений нижче:

```
checkUserCredential(request, response) {  
    userIsChecked = createToken(request.user.id, request.user.email);  
    return response.json({userIsChecked})  
}
```

Після того, як користувач підтвердив замовлення, адміністратор повинен переглянути замовлення, та відправити по пошті товар. Коли товар доставлений та оплачений, адміністратор завершує замовлення або видаляє. Сторінка замовлень продемонстровано на рисунку 3.12.

ID: 27

Телефон: 050308231199

Дата замовлення: Fri, 6/10/2022, 7:33 AM

Клієнт: Не зареєстрований

Статус замовлення: В процесі

Завершити

Видалити

ID: 23

Телефон: 380730812386

Дата замовлення: Tue, 6/7/2022, 2:57 AM

Дата виконання: Fri, 6/10/2022, 7:36 AM

Клієнт: Зареєстрований

Статус замовлення: Завершено

Відмінити

Видалити

Рисунок 3.12 – Web-сторінка замовлень

Програмний код виведення об'єкт замовлення наведено нижче:

```
<Modal.Body>
```

```
  ID:{id})
```

```
<br/><br/>
```

```
<ul>
```

```
  <li>Телефон: {user.mobile}</li>
```

```
  <li>Дата замовлення: {formatDate(user.order.createdAt)}</li>
```

```
  {complete ? `Дата виконання: ${formatDate(user.order.updatedAt)}` : false}
```

```
  <li>{userId ? 'Клієнт: Зареєстрований' : 'Клієнт: Не зареєстрований'}</li>
```

```
  <li>Статус замовлення: {complete ? 'Завершено' : 'В процесі'}</li>
```

```
</ul>
```

```
</Modal.Body>
```

3.3 Використання програмного додатку

Спочатку користувач потрапляє на головну сторінку (рис.3.13). З головної сторінки користувач може перейти до будь якої сторінки.

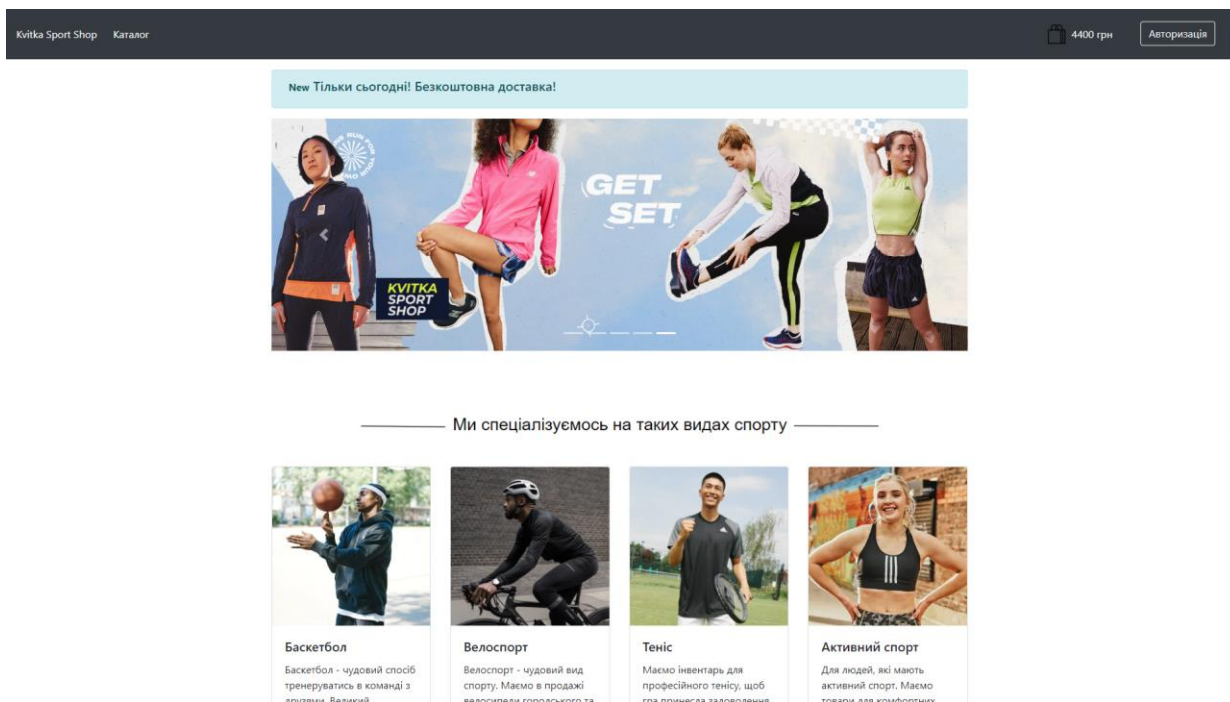


Рисунок 3.13 – Головна сторінка

Для користувача без авторизації буде доступний кошик (рис. 3.14), але товар не буде збережений, якщо користувач закриє web-додаток. Щоб обрані товари збереглися в кошику, користувачеві потрібно пройти авторизацію або зареєструватися, та перобрати товари.

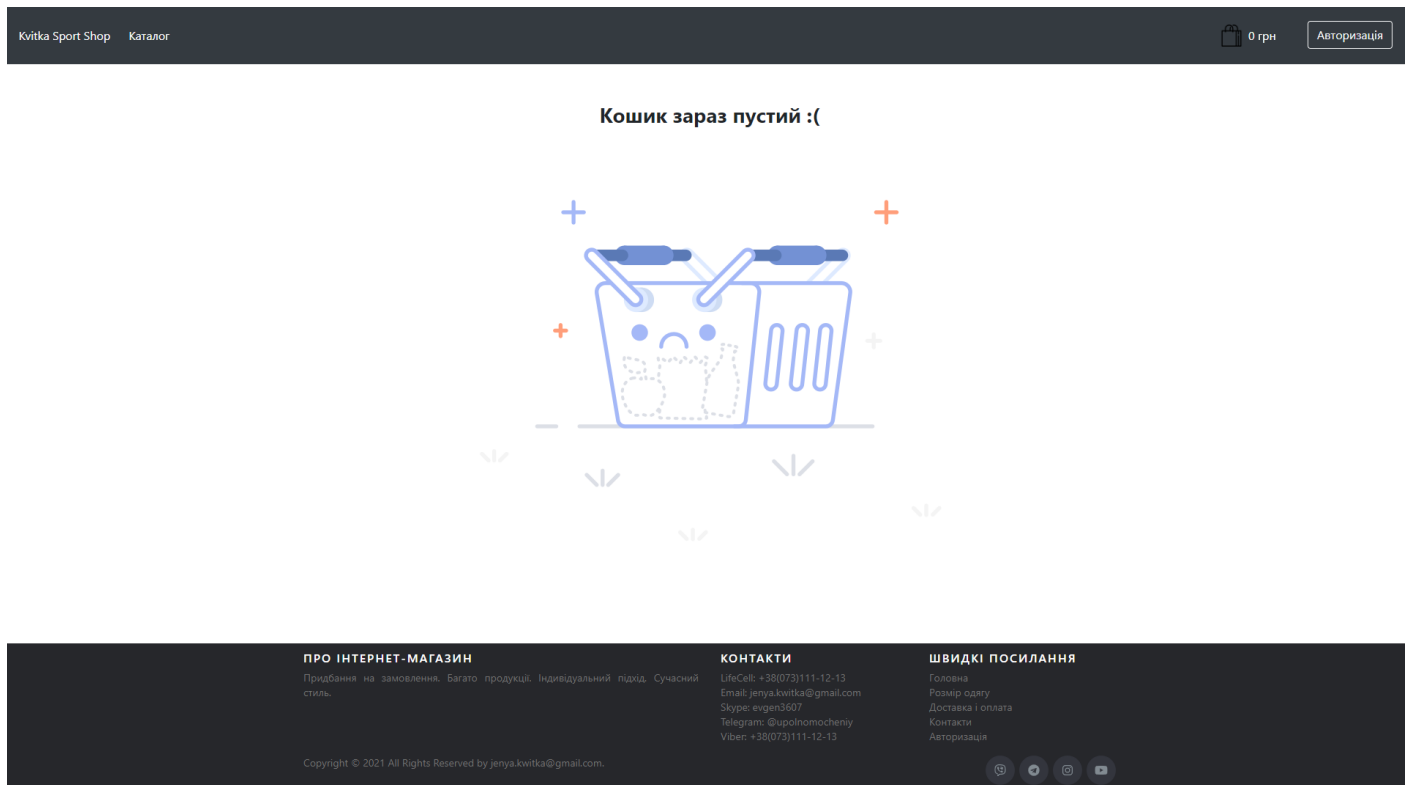


Рисунок 3.14 – Кошик без товарів

Користувачі клікнувши на посилання «Каталог», переходять в каталог товарів (рис. 3.15), користувачі одразу мають можливість переглянути ціну товару фото та рейтинг. Для більш зручного пошуку товарів було створено сортування за певними категоріями.

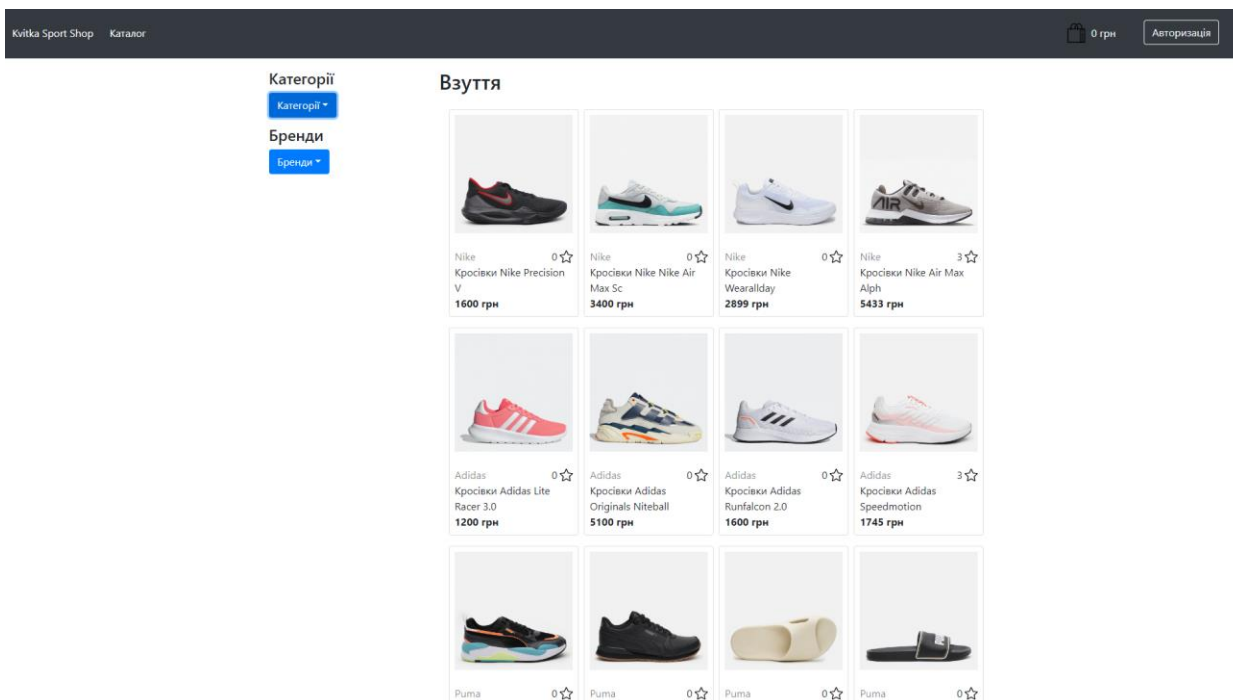


Рисунок 3.15 – Каталог товарів

Якщо натиснути на певний товар, користувача web-система перенаправить на сторінку товару (рис.3.16).

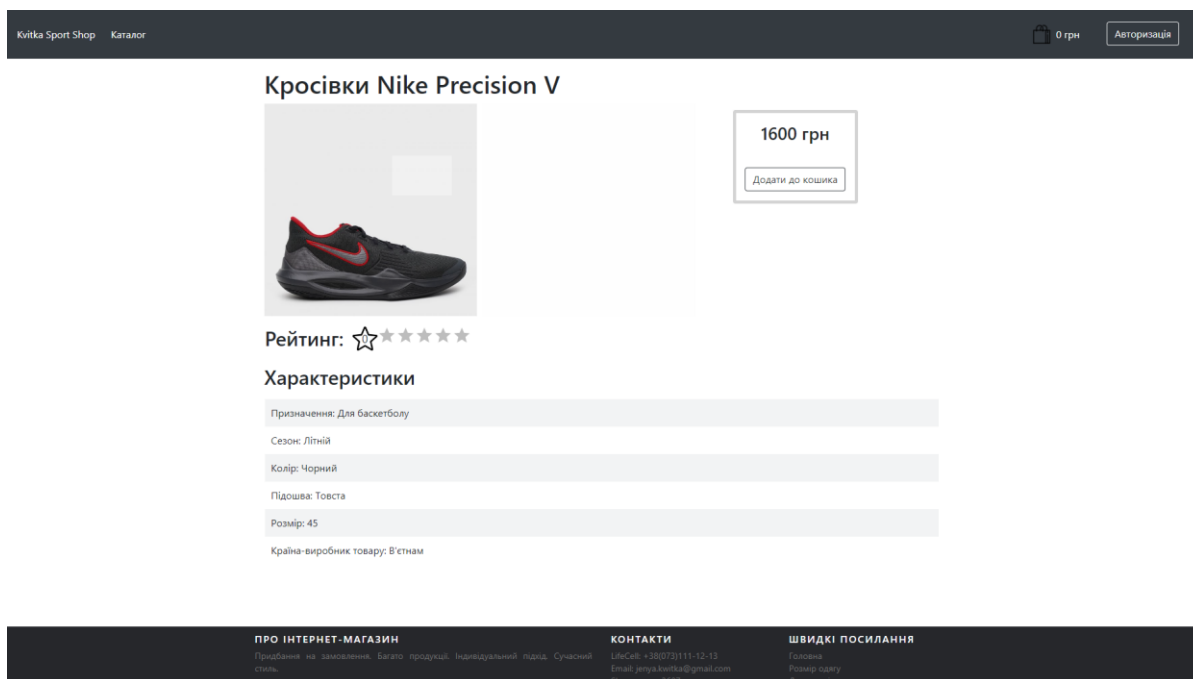


Рисунок 3.17 – Сама сторінка товару з характеристиками та ціною

Після пошуку потрібного товару, користувач має можливість додати товар до кошика (рис. 3.18).

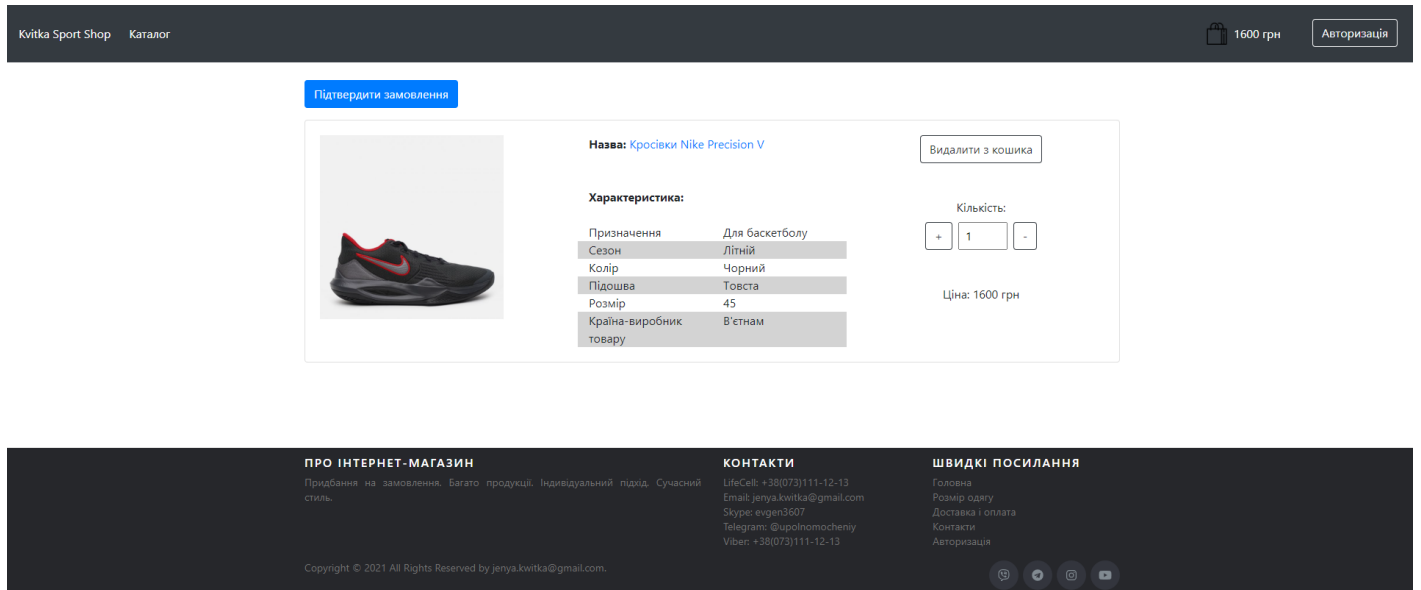


Рисунок 3.18 – Кошик з товаром

Після того, як користувач натиснув на кнопку «Підтвердити замовлення», веб-система перенаправить користувача на сторінку вводу мобільного номеру телефону (рис. 3.19) для подальшого контактування.

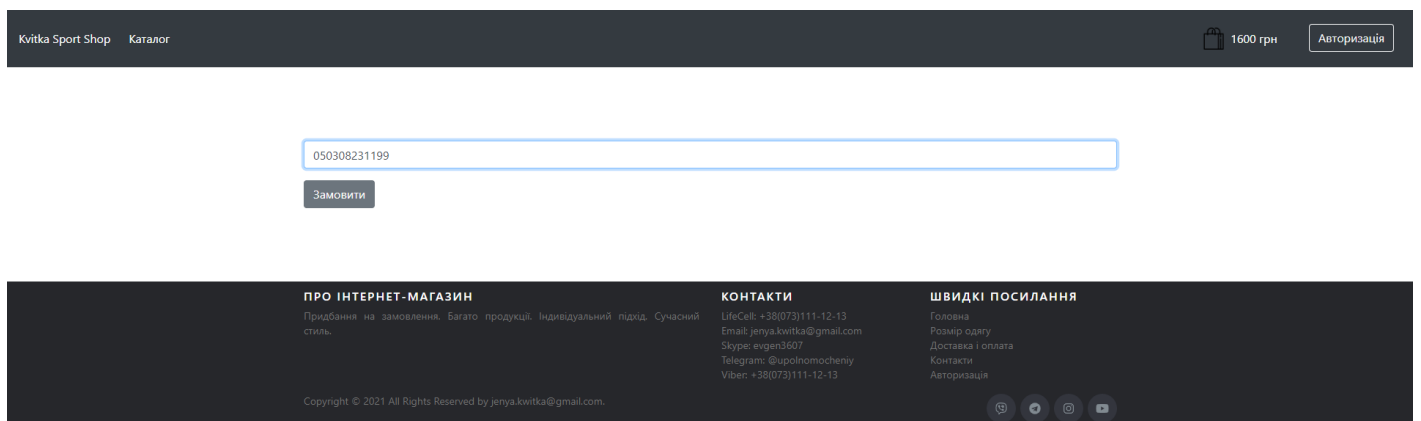
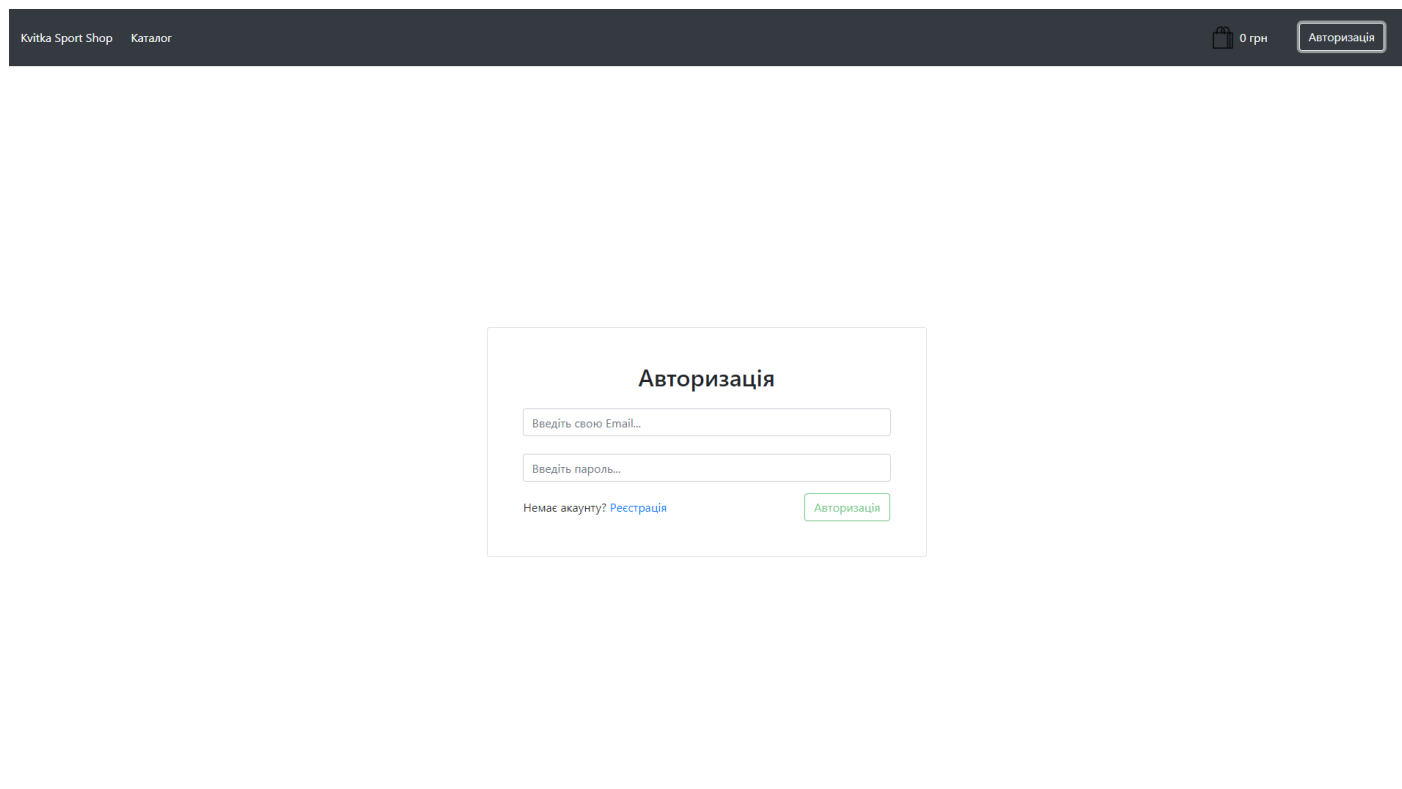


Рисунок 3.19 – Підтвердження замовлення

Також було розроблено можливість авторизації/реєстрації для користувачів (рис. 3.20). Web-додаток має всього дві ролі – це звичайний користувач та користувач з правами адміністратора.



The screenshot shows a web browser window with a dark header. On the left, it says "Kvitka Sport Shop" and "Каталог". On the right, there is a shopping cart icon with "0 грн" and a button labeled "Авторизація". The main content area is white and contains a centered box titled "Авторизація". Inside this box, there are two input fields: "Введіть свою Email..." and "Введіть пароль...". Below the second field, there is a link "Немає акаунту? Реєстрація" and a green button labeled "Авторизація".

Рисунок 3.20 – Авторизація

Адміністратор web-додатку має можливість через адміністративну панель додавати видаляти та змінювати товар та категорії товару (рис.3.21).

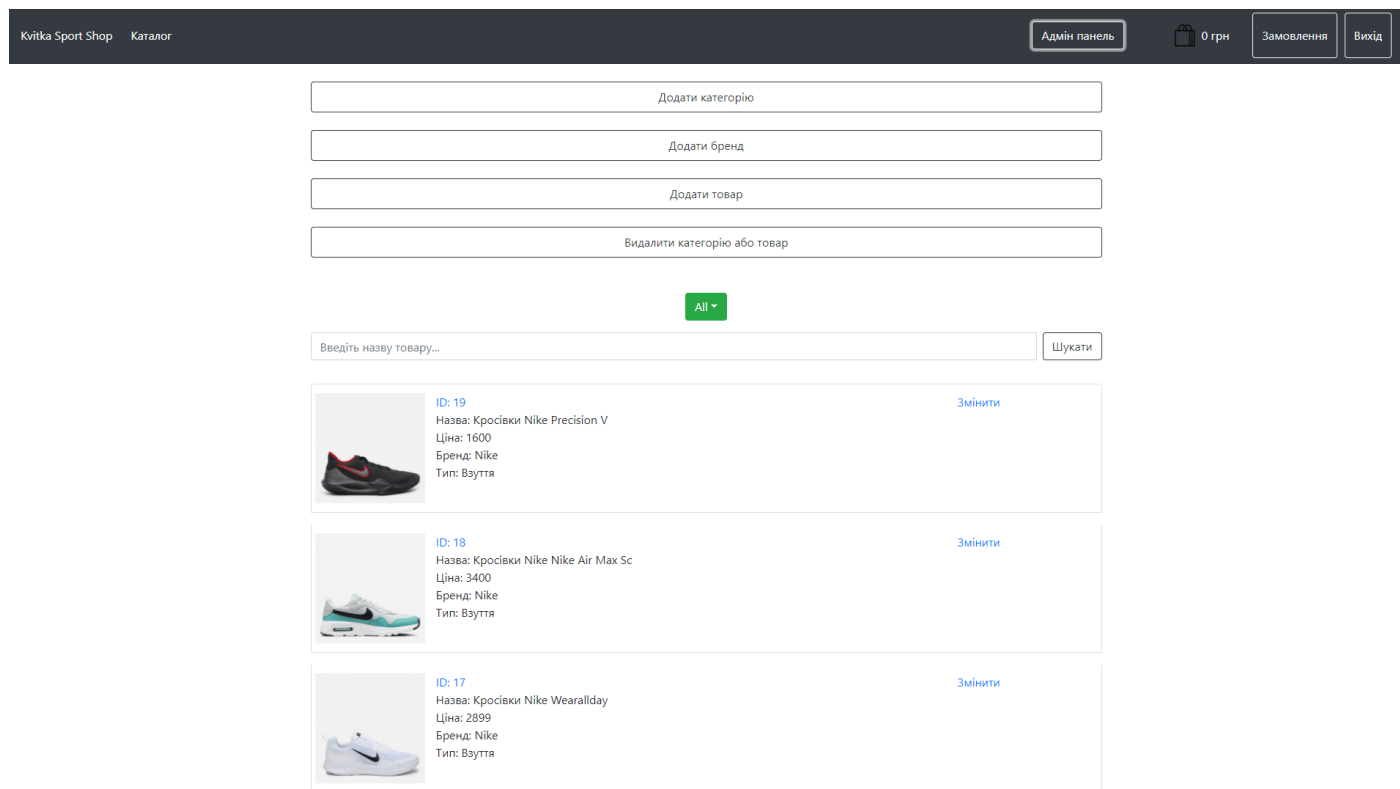


Рисунок 3.21 – Адміністративна панель

Якщо адміністратор вирішив додати новий товар, то необхідно натиснути на кнопку «Додати товар», та web-система відкриє модальне вікно (рис. 3.22), де користувач має змогу заповнити поля для створення нового товару.

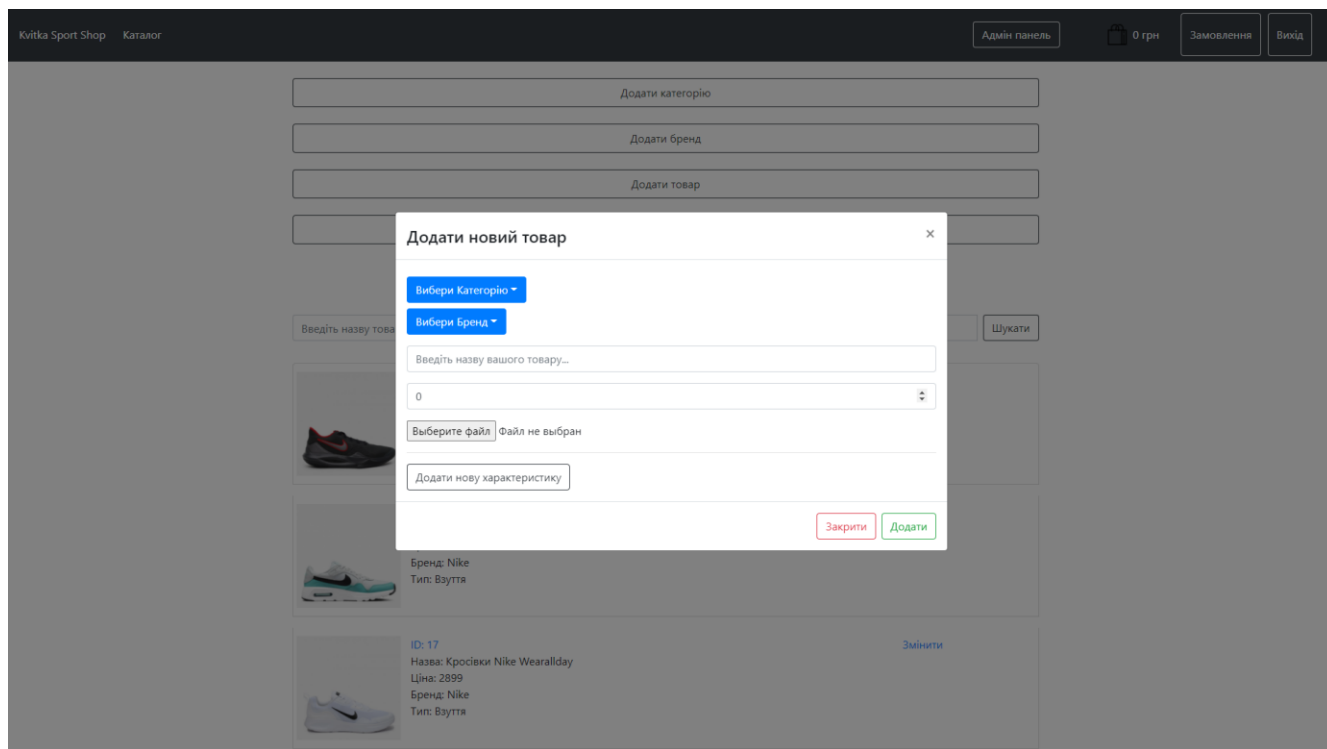


Рисунок 3.22 – Створення нового товару

На рисунку 3.23 зображено як виглядає сторінка для редагування товару.

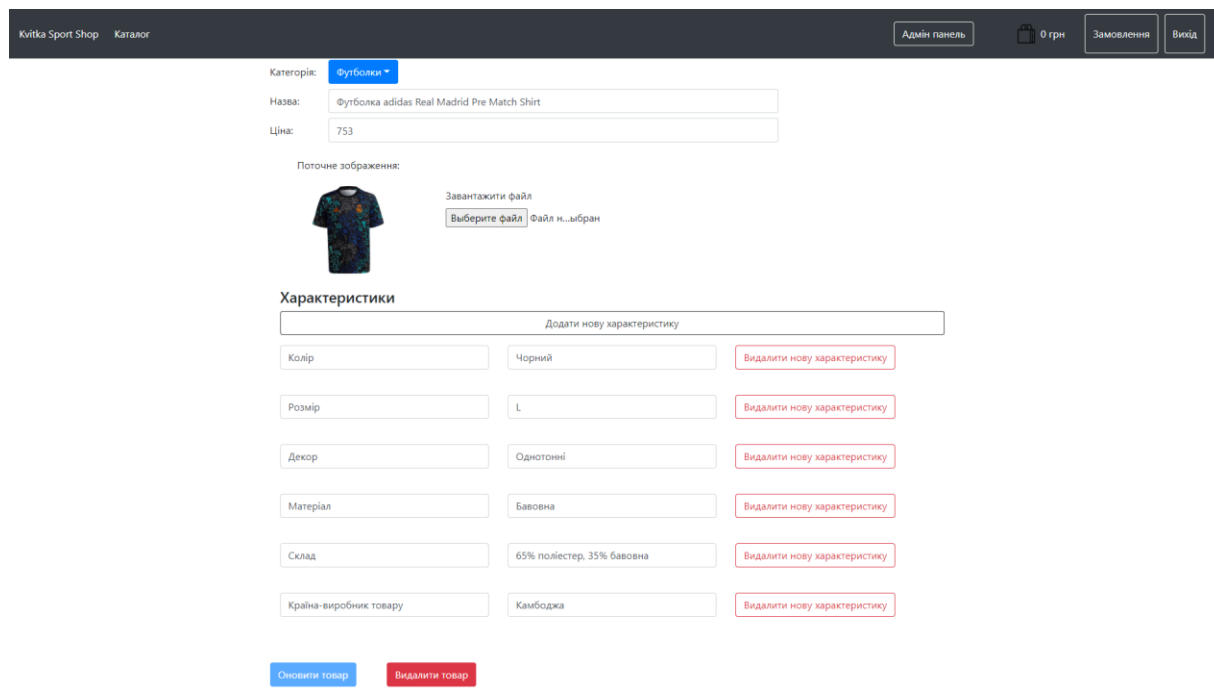


Рисунок 3.23 – Редагування товару

Після того, як користувач підтвердив власне замовлення, адміністратор може це побачити, натиснувши на кнопку «Замовлення» (рис. 3.24). Тут адміністратор може через номер телефону подзвонити клієнту та уточнити питання по доставці товару.

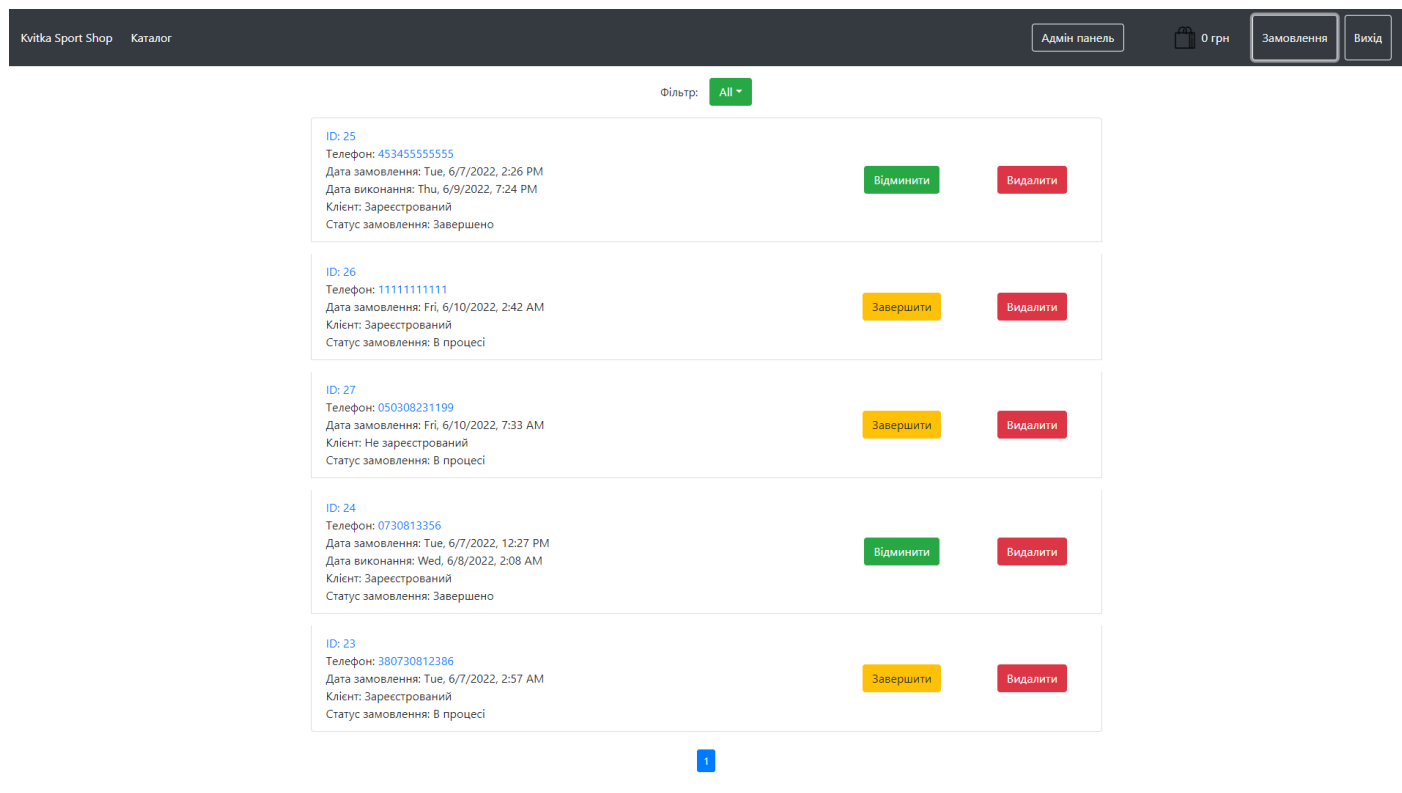



Рисунок 3.24 – Колекція замовлень


На рисунку 3.25 зображено товари, які входять в замовлення та кінцеву ціну замовлення.

Kvitka Sport Shop Каталог
Адмін панель
0 грн
Замовлення
Вихід

ID замовлення: 23
 Статус: В процесі
 Клієнт: 7
 Дата створення: вт, 07.06.2022 р., 02:57
 Телефон: [380730812386](tel:380730812386)



Бренд: Nike
 Категорія: Взуття
 Назва: Кросівки Nike Air Max Alph
 Ціна: 5433 грн
 Кількість: 1
 Ціна: 5433 грн



Бренд: Nike
 Категорія: Взуття
 Назва: Кросівки Nike Precision V
 Ціна: 1600 грн
 Кількість: 1
 Ціна: 1600 грн

Загальна ціна замовлення: 7033 грн

ПРО ІНТЕРНЕТ-МАГАЗИН
 Придбання на замовлення, багато продукції. Індивідуальний підхід. Сучасний стиль.

КОНТАКТИ
 LifeCall: +38073111-12-13
 Email: jenya.kvitka@gmail.com
 Skype: evgen3607
 Telegram: [@cipolnomocheny](https://t.me/cipolnomocheny)
 Viber: +38(073)111-12-13

ШВИДКІ ПОСИЛАННЯ
 Головна
 Розмір одягу
 Доставка і оплата
 Контакти
 Авторизація

Рисунок 3.25 – Вигляд замовлення

Якщо користувач вже отримав замовлення та забрав його з поштового відділення, адміністратор може перевести замовлення зі статусу «В процесі» на статус «Завершено» (рис. 3.26).

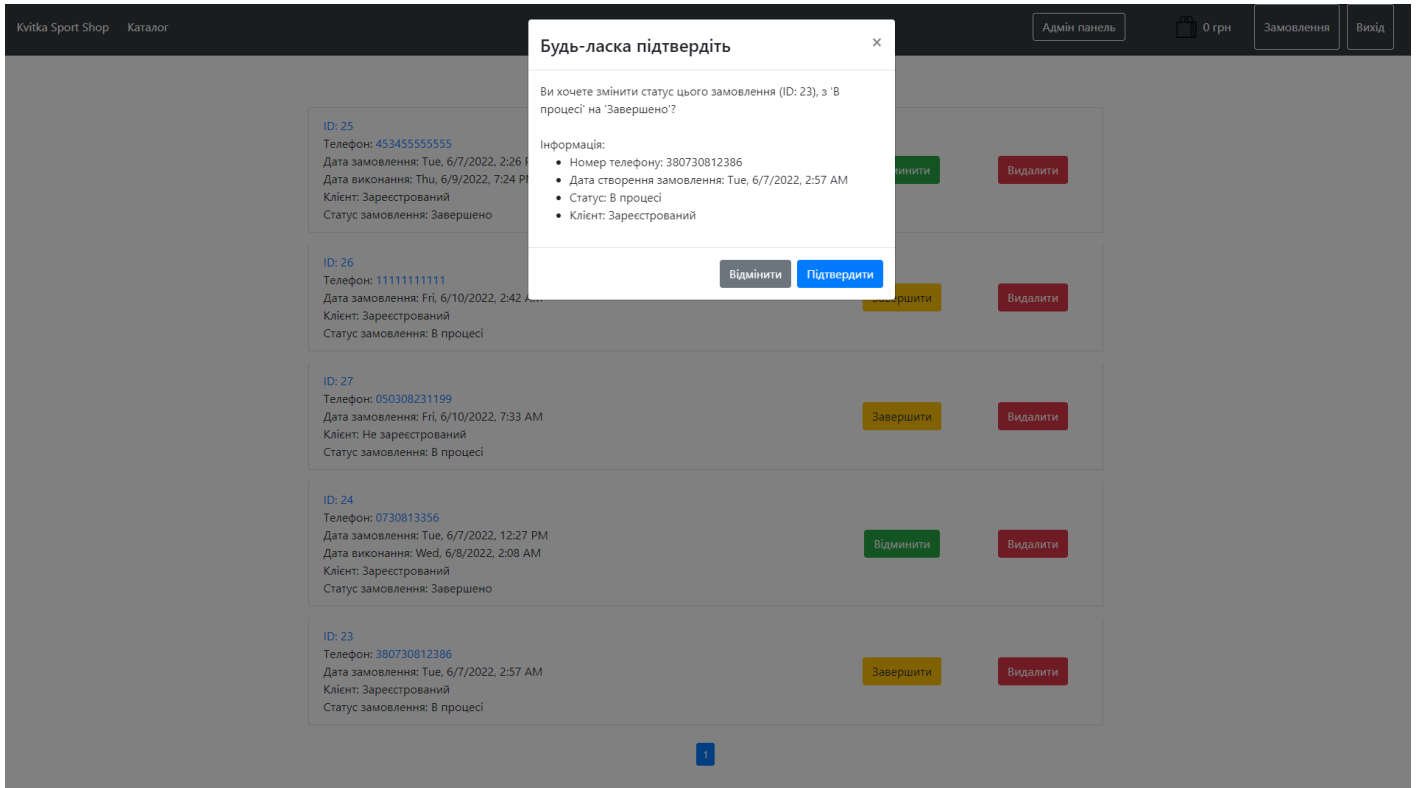


Рисунок 3.26 – Підтвердження завершення замовлення

На рисунку 3.27 зображено, як виглядає завершене замовлення.

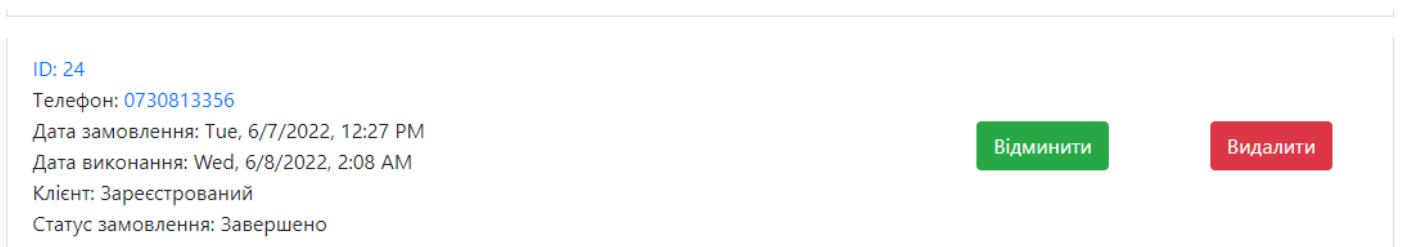


Рисунок 3.27 – Зміна статусу на «Завершено»

Для користувача виведено окрема сторінка для інформації стосовно доставки та оплати (рис. 3.28).

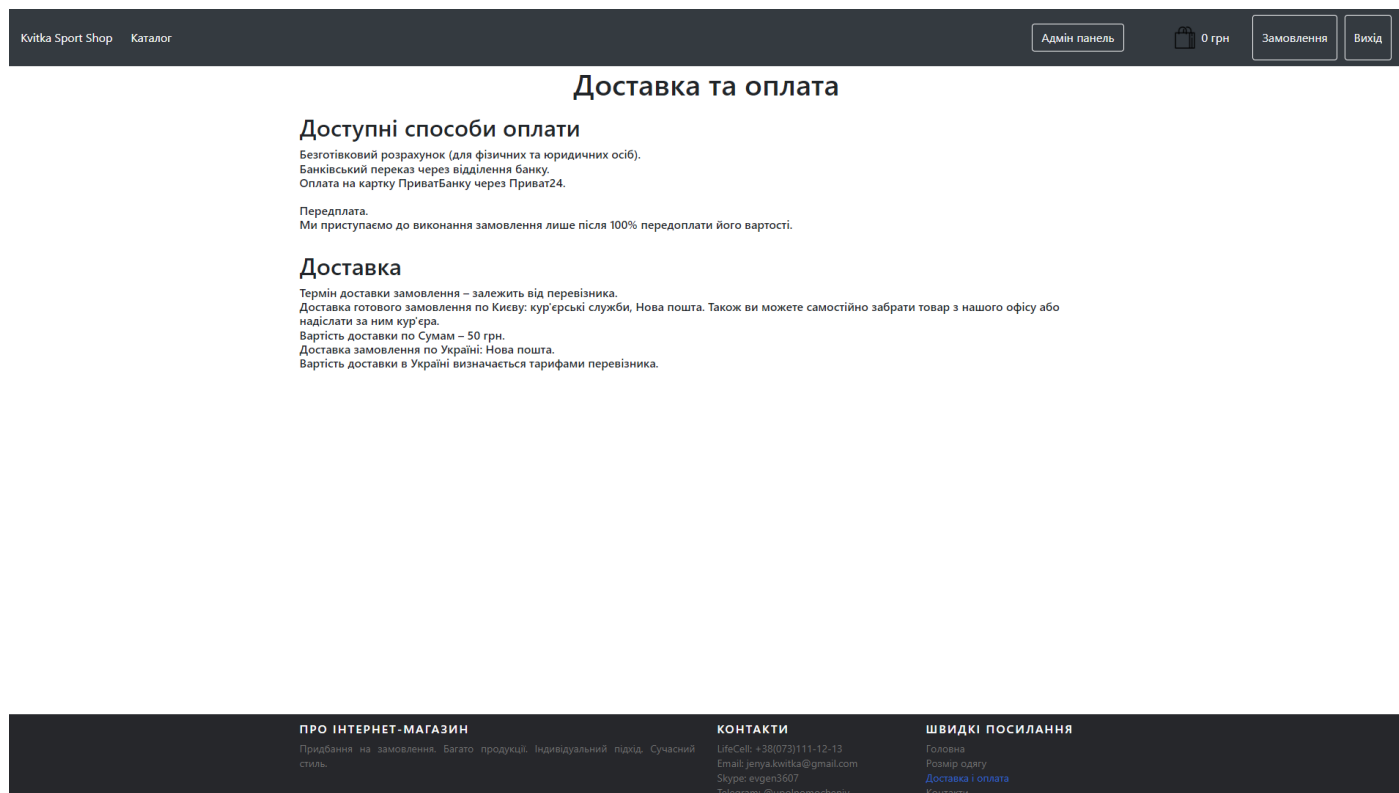


Рисунок 3.28 – Сторінка доставки

Також для реалізації web-системи було реалізовано адаптивну верстку для більш зручнішого користування через телефон. На рисунку 3.29 продемонстровано сторінку авторизації.

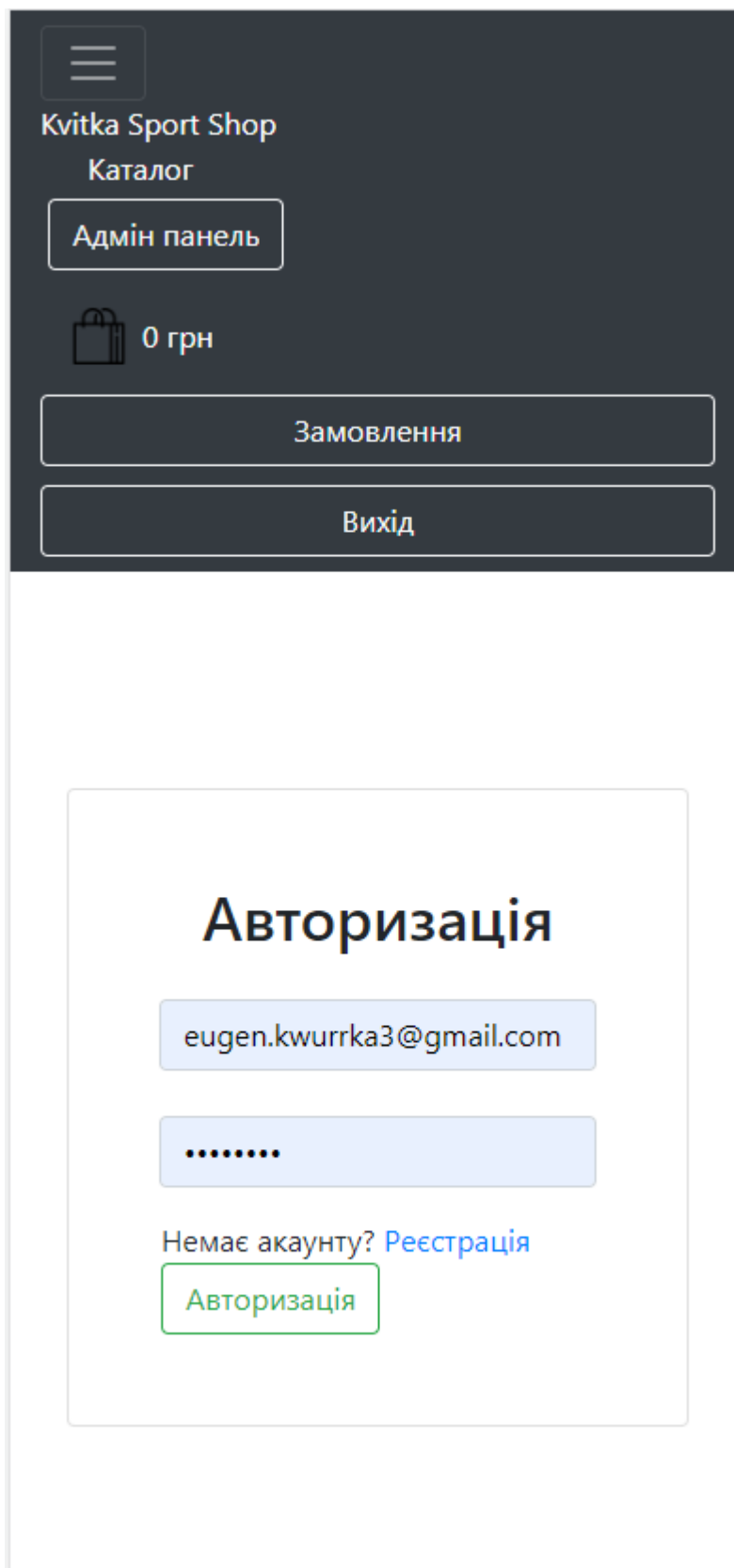


Рисунок 3.29 – Адаптивна верстка авторизації

ВИСНОВОК

Web-додаток для підтримки діяльності магазину орієнтований на зростання попиту на продукцію, а, отже, очікування на збільшення прибутку.

У ході створення проекту було досліджено предметну галузь та ринок аналогів. Визначаючи актуальність поставленої задачі було також проаналізовано вимоги до програмного продукту.

На етапі моделювання було вирішено наступні завдання:

- дослідження та аналіз предметної області проекту;
- постановка задачі та планування виконання проекту;
- розроблення технічного завдання створення програмного продукту та планування робіт;
- структурно-функціональне моделювання.

Дослідивши предметну галузь, було показано, що створення web-додатків для підтримки магазинів є невід'ємною частиною розвитку сучасного світу. Адже дані інтернет-ресурси пропонують користувачеві достатню кількість інформації, необхідні для прийняття рішення про покупку.

В результаті виконання кваліфікаційної роботи бакалавра було розроблено web-додаток для підтримки діяльності магазину з продажу спортивних товарів. Усі вимоги для розробки інформаційної системи відповідають технічному завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Які переваги та недоліки електронної комерції?. *Actualidad eCommerce*. URL: <https://www.actualidadecommerce.com/uk/які-переваги-та-недоліки-електронної-комерції/> (дата звернення: 21.05.2022).
2. CORE – Aggregating the world’s open access research papers. URL: <https://core.ac.uk/download/pdf/43283168.pdf> (дата звернення: 21.05.2022).
3. Навіщо бізнесу потрібен сайт? 8 важливих переваг, які ви отримуєте. *AG Marketing*. URL: <https://ag.marketing/blog/navishcho-biznesu-potriben-sayt/> (дата звернення: 19.05.2022).
4. Web-сайт: визначення й застосування. Статті про розробку та просування web-сайтів. URL: <http://www.webtec.com.ua/uk/articles/index/view/2011-05-05/web-site> (дата звернення: 24.05.2022).
5. Decathlon Ukraine - спорт для всіх: якість від новачка до профі. URL: <https://www.decathlon.ua/uk/> (дата звернення: 24.05.2022).
6. Adidas – офіційний інтернет-магазин в Україні. URL: <https://www.adidas.ua/> (дата звернення: 20.05.2022).
7. Спорт та захоплення - ROZETKA. URL: <https://rozetka.com.ua/sport-i-uvlecheniya/c4627893/> (дата звернення: 13.05.2022).
8. Спецпроект: топ-10 ритейлерів спорттоварів в Україні за кількістю магазинів. *Асоціація ритейлерів України – The profile association of retail market players*. URL: <https://rau.ua/novyni/top-10-ritejleriv-sporttovariv/> (дата звернення: 20.05.2022).
9. Npm docs. URL: <https://docs.npmjs.com/> (дата звернення: 23.05.2022).
10. React – JavaScript-бібліотека для створення пользовательських інтерфейсів. URL: <https://ru.reactjs.org/> (дата звернення: 23.05.2022).
11. GitHub - twbs/bootstrap: the most popular HTML, CSS, and javascript framework for developing responsive, mobile first projects on the web. *GitHub*. URL: <https://github.com/twbs/bootstrap> (дата звернення: 20.05.2022).

12. PostgreSQL: documentation. *PostgreSQL: The world's most advanced open source database*. URL: <https://www.postgresql.org/docs/> (дата звернення: 23.05.2022).
13. Node.js. URL: <https://nodejs.org/en/> (дата звернення: 24.05.2022).
14. Основы методологии IDEF0 - PDF Free Download. *Пользуйтесь удобными бесплатными инструментами для публикации, обмена и распространения любых документов!*. URL: <https://docplayer.com/26710804-Osnovy-metodologii-idef1-i-idef1x.html> (дата звернення: 08.06.2022).
15. Learning UML diagrams will make you a better developer. *Built In*. URL: <https://builtin.com/software-engineering-perspectives/uml-diagrams> (дата звернення: 08.06.2022).
16. ДДКБМТА - Основні поняття баз даних. *Google Sites: Sign-in*. URL: <https://sites.google.com/view/ddkbmta-info/лекції/системи-керування-базами-даних-microsoft-access/основні-поняття-баз-даних> (дата звернення: 08.06.2022).
17. Концепция MVC для чайников. *Видеоуроки по созданию сайта для бизнеса. Бесплатные видеоуроки*. URL: <https://ruseller.com/lessons.php?id=666> (дата звернення: 07.06.2022).

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Web-додаток для підтримки діяльності спортивного магазину»

ПОГОДЖЕНО:

Кандидат технічних наук

_____ Парфененко Ю.В.

Студент групи ІТ-82-0

_____ Квітка Є.С.

1. Призначення й мета створення web-додатку

1.1 Призначення web-додатку

Інформаційна система повинна собою представляти інтернет магазин спортивних товарів та мати назву «Kvitka-shop».

1.2 Мета створення web-додатку

Метою створення web-додатку полягає в тому, щоб залучити більше аудиторії, яка купує спортивні товари.

1.3 Цільова аудиторія

Цільовою аудиторією для даного проекту складається з:

- клієнтів, які бажають замовити певний товар;
- працівники, які підтримують діяльність web-додатку;
- інші зацікавлені користувачі в даному web-додатку.

2 Вимоги до web-додатку

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури й функціонування web-додатку

Інформаційна система повинна бути реалізована у вигляді web-додатку, розміщеного на хостингу разом з підключеною базою даних. Інформаційна система повинна складатися з розділів чітко розподіленим функціоналом.

2.1.2 Вимоги до персоналу

Персонал повинен мати загальні навички роботи з комп'ютером і має бути ознайомлений як працювати зі стандартними web-браузерами.

2.1.3 Вимоги до збереження інформації

У системі керування web-додатком повинно бути передбачено механізм збереження даних у базу даних, а також читання з неї. Безпека даних буде забезпечена за рахунок шифрування паролів, шляхом хешування інформації, за допомогою спеціальних інструментів.

2.1.4 Вимоги до розмежування доступу

Інформація, яка буде розташована на web-додатку, є в загальному доступі. Права доступу можна розділити на дві групи користувачів:

- адміністратор;
- користувач.

Адміністратору дозволяється редагувати інформаційне наповнення товарів, додавання категорій та назв брендів. Доступ до адміністративної частини повинен здійснюватися за допомогою адміністративної панелі з використанням унікального логіна та паролю.

Користувач має доступ тільки до загальнодоступної частини web-додатку. Доступ до адміністративної частини мають тільки адміністратор.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

Інформаційна система має складатись з наступних розділів:

- головна сторінка – виводиться основна інформація про організацію, плюси даного web-додатку, та контактна інформація в підвалі web-додатку;
- каталог – міститиме каталог товарів з елементами сортування по певним критеріям;
- сторінка товару – міститиме фото, оцінку, характеристики та ціну певного товару;

- кошик – сторінка з обраними товарами користувачем;
- сторінка контакту – містить контакти власників онлайн магазину;

2.2.2 Навігація

Користувацький інтерфейс web-додатку повинен забезпечувати наочне, інтуїтивно зрозуміле представлення структури розміщеної на ньому інформації, швидкий і логічний перехід розділів. Навігаційні елементи повинні забезпечувати однозначне розуміння користувачем їх змісту (назви web-елементів мають відповідати їх функціоналу та якнайкраще передавати суть її призначення). Система повинна відображати відповідну інформацію.

2.2.3 Наповнення web-додатку (контент)

Наповнення web-додатку буде зроблено за наданим замовником інформації відповідно до тематики інформаційної системи. Інші дані будуть автоматично заноситись програмним шляхом на основі бази даних на сервері.

У разі зміни наповнення розділів повинна здійснюватися за допомогою адміністративної панелі.

2.2.4 Дизайн та структура додатку

Стиль web-додатку можна описати як сучасний, діловий. У якості фонового кольору рекомендується використовувати світлі кольори. Він повинен бути інтуїтивно зрозумілий. Оформлення не повинне перешкоджати інформативності. Приблизне розташування елементів на головній сторінці зображено на рисунку А.1.

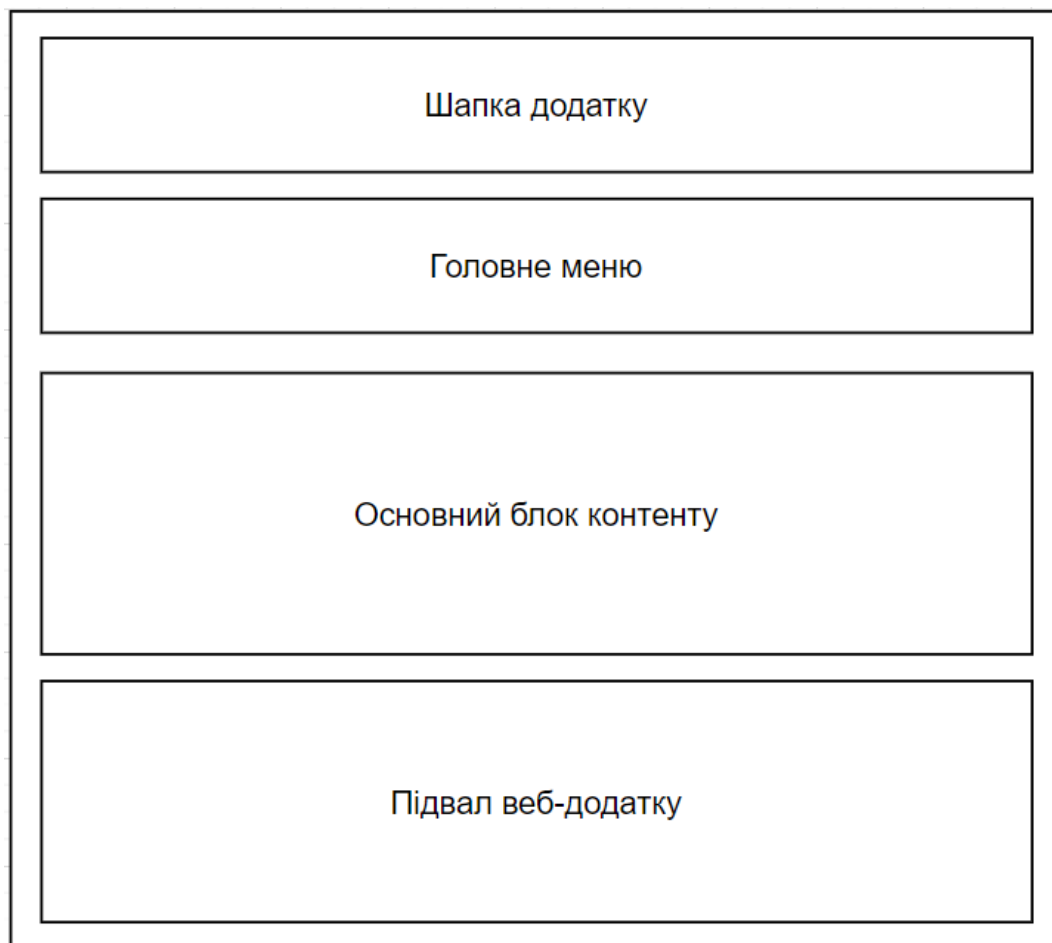


Рисунок А.1 – Схема головної сторінки

2.2.5 Система навігації (карта web-додатку)

Карта web-додатку зображена на рисунку А.2.

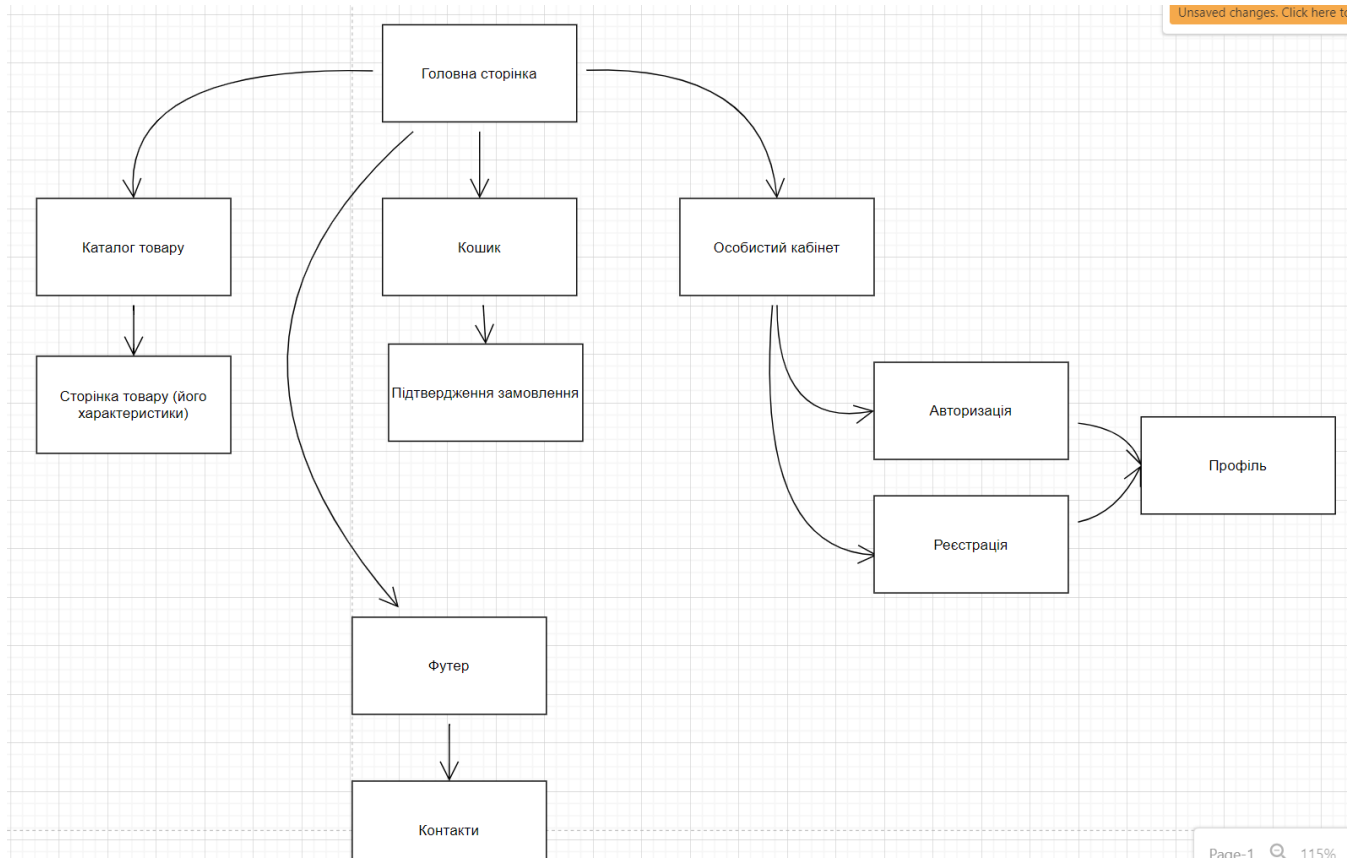


Рисунок А.2 – Карта web-додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.1.

Таблиця А.1 – Потреби користувача

ID	Потреби користувача	Джерело
UN-01	Перегляд товарів спортивного магазину	Клієнт
UN-02	Перегляд характеристики товарів	Клієнт
UN-03	Оцінювання товару	Клієнт
UN-04	Додавання до кошику товарів	Клієнт
UN-05	Замовлення товару	Клієнт
UN-06	Редагування даних	Адміністратор
UN-07	Додавання даних	Адміністратор
UN-08	Перегляд інформації про web-додаток	Клієнт
UN-09	Перегляд контактної інформації	Клієнт
UN-10	Розрахунок загальної вартості замовлення	Клієнт

2.3.2 Функціональні вимоги

На головній сторінці будуть знаходитись наступні дані:

- навігаційне меню;
- блок з перевагами онлайн магазину;
- підвал web-додатку.

На сторінці каталогу з товарами:

- перегляд товарів;
- вибір параметрів для пошуку.

На сторінці товару:

- перегляд фото товару;
- список характеристик товару;
- кнопка с додаванням до кошику;
- кнопка з оцінюванням товару.

На сторінці з правами адміністратора:

- додавання товару;
- редагування товару;
- додавання брендів та категорій для пошуку.

На сторінці контактів:

- перегляд контактних даних.

2.3.3 Системні вимоги

В цьому розділі визначено, розподілено та вказано розробником на системні вимоги. Перелік вимог наведений в таблиці А.2.

Таблиця А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Розподілення товарів по певним категоріям	S	Надає можливість передивлятись товари по певним категоріям
SR-02	Вхід/Регістрація користувачів	M	Надає розмежування доступу
SR-03	Каталог товарів	M	Перегляд всіх доступних товарів
SR-04	База даних замовлень	M	Надає список всіх замовлень клієнтів
SR-05	База даних з спортивними товарами	M	Збереження всіх позицій товарів
SR-06	Адміністративна панель	M	Надає можливість обробляти та змінювати дані
SR-07	База даних з користувачами	M	Відповідає за розмежування ролей

Продовження таблиці А.2 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-08	Оцінювання товарів користувачами	С	Надає можливість оцінити товар
SR-09	Кошик товарів	М	Зберігання товару до кошика, замовлення товару

Умовні позначення в таблиці А.2:

- Must have (M) – вимоги, для обов’язкової реалізації в системі.
- Should have (S) – вимоги, які мають бути виконані, але з меншою пріоритетністю.
- Could have (C) - необов’язкові вимоги, які не є основними вимогами на проекті.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до інформаційного забезпечення

Реалізація web-додатку відбуватиметься з використанням:

- JavaScript;
- CSS;
- React.js;
- Axios;
- PostgreSQL;
- Express;
- Node.js;
- Sequelize.

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додаток має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- web-браузер: Mozilla Firefox 3.5 і вище, Opera 9.5 і вище, Google Chrome 2 і вище;
- включена підтримка JavaScript, cookies.

3 Склад і зміст робіт зі створення web-додатку

Основний склад та зміст робіт зі створення web-додатку наведено в таблиці А.3.

Таблиця А.3 – Етапи створення web-додатку

№	Назва етапу створення web-додатку	Тривалість
1	Проведення збору аналізу аналогів	2 дні
2	Написання технічного завдання	3 дні
3	Створення першого макету	2 дні
4	Написання front-end частини	3 дні
5	Написання back-end частини	3 дні
6	Налаштування роботи клієнтської та серверної частини	3 дні
7	Наповнення web-додатку контентом	1 день
8	Тестування web-додатку в цілому	2 дні
9	Фінальна підготовка web-додатку	1 день
	Загальна тривалість проекту	20 днів

4 Вимоги до складу й змісту робіт із введення web-додатку в експлуатацію

Для створення умов функціонування, при яких гарантується відповідність створюваного web-додатку вимогам сьогодення ТЗ і можливість його ефективної роботи, в організації замовника повинен бути проведений певний комплекс заходів.

Для переносу web-додатку на хостинг необхідно, щоб параметри хостингу відповідали вимогам, зазначеним у ТЗ. На хостинг переноситься програма (web-додаток), розроблений шаблон дизайну й структура й наповнення бази даних з подальшою їх доробкою.

ДОДАТОК Б

Планування робіт

Сьогодні web-додатки відіграють важливу роль у розширеному бізнесі компанії.

Web-додаток – це сукупність загальнодоступних, взаємопов'язаних web-сторінок з одним доменним ім'ям. Web-додатки можуть створювати окремі особи, групи, компанії чи організації для різних цілей.

Через значну кількість переваг більшість людей кажуть, що сьогодні краще робити покупки в Інтернеті, ніж у звичайних магазинах.

Головними перевагами онлайн магазинів являються:

- Зручність – покупки можна робити в будь-якому місці.
- Порівняння цін – інтернет магазин дозволяє набагато швидше відстежувати ціни та якості товару.
- Відсутність черг.
- Відсутність потреби йти до магазину за товаром.

З появою інтернет-магазинів з'явилася можливість продавати товари в різних містах і країнах, знявши всі географічні обмеження.

Покупки в Інтернеті можуть заощадити час як покупців, так і продавців, зменшуючи кількість доступних телефонних дзвінків, перегляд технічних характеристик, годин роботи та іншу інформацію, яку легко знайти на сторінках компаній та продуктів.

Деталізація мети проекту методом SMART

Ціль SMART використовується для встановлення цілей. SMART – це абревіатура, що означає Specific, Measurable, Achievable, Relevant, Time bound. Визначення цих критеріїв допомагає гарантувати, що всі цілі будуть досягнуті протягом певного періоду часу. Такий підхід включає загальні відомості та припущення, встановлює чітку

тимчасову шкалу та спрощує відстеження прогресу та виявлення пропущених етапів. Результати деталізації методом SMART розміщені у таблиці Б.1.

Таблиця Б.1 – Деталізація мети проекту методом SMART

Назва	Визначення
Specific (конкретна)	Створити web-додаток підтримки діяльності магазину з продажу товарів для спорту.
Measurable (вимірювана)	Створити web-додаток в короткий термін з використанням мінімальної кількості ресурсів.
Achievable (досяжна, узгоджена)	Для даної реалізації задачі необхідні знання з Node.js, React.js, npm пакетів, таких мов програмування, як: JavaScript, та досвід роботи з реляційними базами даних PostgreSQL.
Relevant (реалістична)	Web-додаток дозволить автоматизувати роботи магазинів, чим вплине на комфортність перегляду товарів та порівнювання технічних характеристик та замовлення товарів з будь-якої точки країни, а отже збільшення кількості клієнтів.
Time-framed (обмежена в часі)	Мета проекту має часові обмеження. Дати виконання проекту погоджені між замовником та виконавцем.

2 Опис проекту на фазі розробки

2.1 Планування змісту структури проекту

Work breakdown structure (WBS) – це візуальна, ієрархічна та орієнтована на результат проекту. Дана діаграма дозволяє розбити обсяг свого проекту та візуалізувати всі завдання, необхідні для завершення проекту.

Дана аббревіатура розшифровується як:

– Work (Робота).

– Breakdown (Декомпозиція) – поділ частин, для розбиття складних процесів на більш прості.

– Structure (Структура) – фіксована впорядкована безліч об’єктів та відносин між ними, класифікація чогось за заданою основою.

Усі етапи роботи над проектом викладено в діаграмі декомпозиції робіт, що робить її важливим інструментом планування проекту. Кінцевий результат проекту, а також пов’язані з ним завдання та робочі пакети знаходяться у верхній частині WBS, а рівні WBS, розташовані нижче, поділяють обсяг проекту, щоб вказати завдання, результати та робочі пакети, які необхідні для завершення проекту. Тобто декомпозиція виконується до етапу, поки вони не стануть простими.

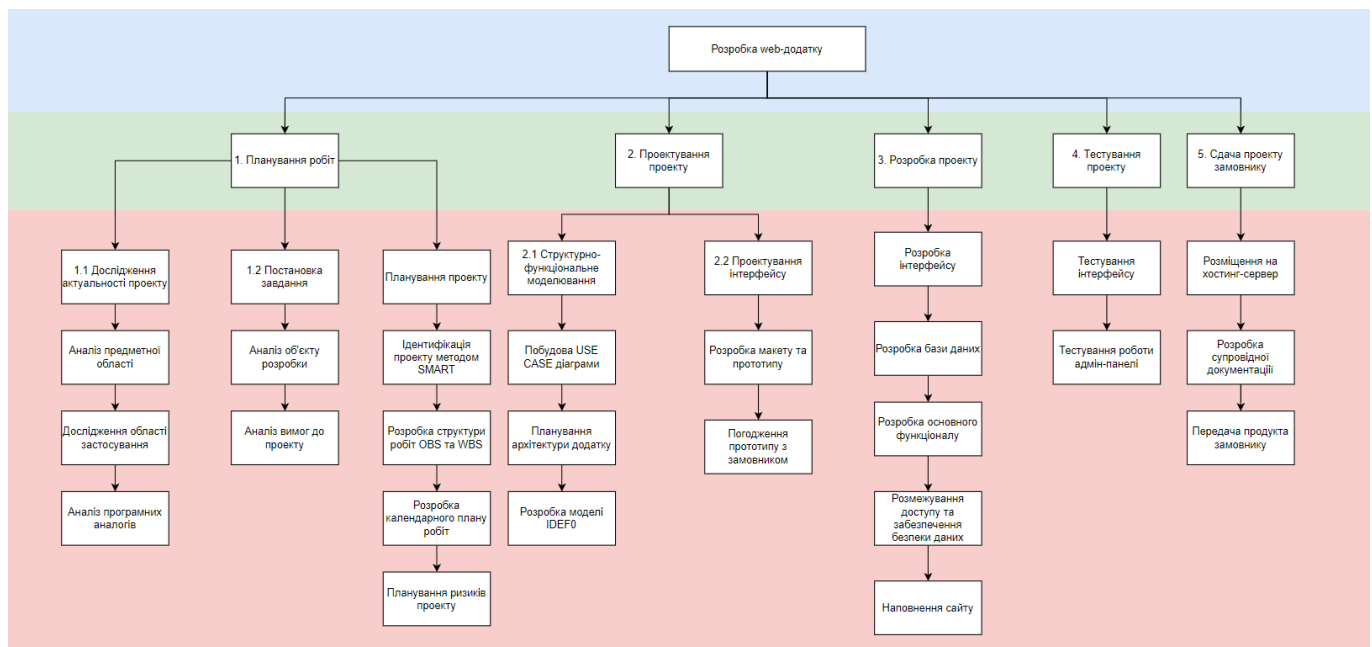


Рисунок Б.1 – Фрагмент WBS – структура робіт проекту

2.2 Планування структури організації

Наступним етапом в створенні додатку є розробка організаційної структури проекту або OBS. Дана модель використовується для підвищення якості управління та

взаємодії у проекті шляхом визначення та візуалізації процесів взаємодії як між внутрішніми, так і із зовнішніми учасниками проекту.

На рисунку Б.2 представлено фрагмент організаційної структури.

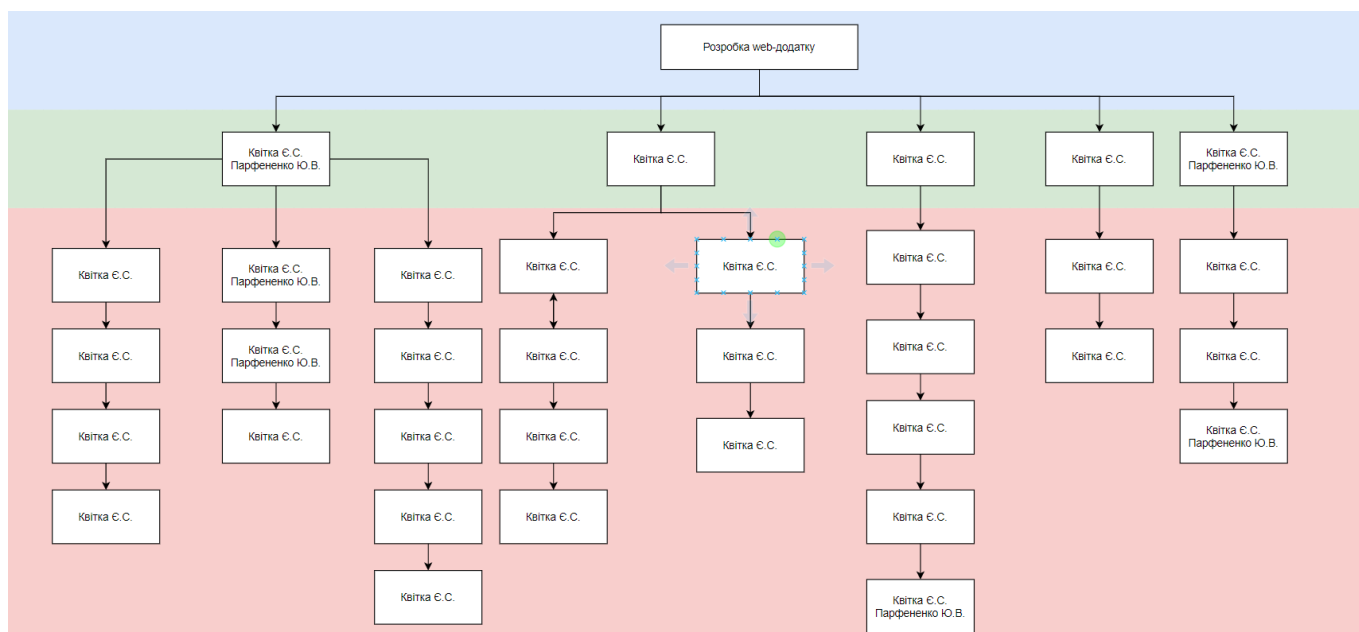


Рисунок Б.2 – Фрагмент OBS – структура робіт проекту

2.3 Побудова таблиці виконавців проекту

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Квітка Є.С.	Виконання front-end та back-end розробки проекту
Проектувальник	Квітка Є.С.	Виконує проектування та розробку бази даних
Тестувальник	Квітка Є.С.	Виконує розробку тест плану, тест кейсів та функціональне тестування
Керівник проекту	Парфененко Ю.В.	Формує завдання на розробку проекту, затверджує ТЗ
Менеджер проекту	Квітка Є.С.	Відповідальний за своєчасне виконання проекту та розподіл ресурсів проекту.

2.4 Побудова календарного графіку

Діаграма Ганта (Gantt Chart) – це стовпчаста діаграма, яка забезпечує візуальне представлення завдань проекту, запланованих із часом. Діаграма Ганта використовується для планування проекту: це корисний спосіб показати, які роботи планується виконати в певні дні. Gantt Chart допомагає всім членам команди переглядати дати початку, закінчення та основні етапи графіку проекту на одній простій стовпчастій діаграмі з накопиченням.

Календарний графік проекту представлено на рисунках.

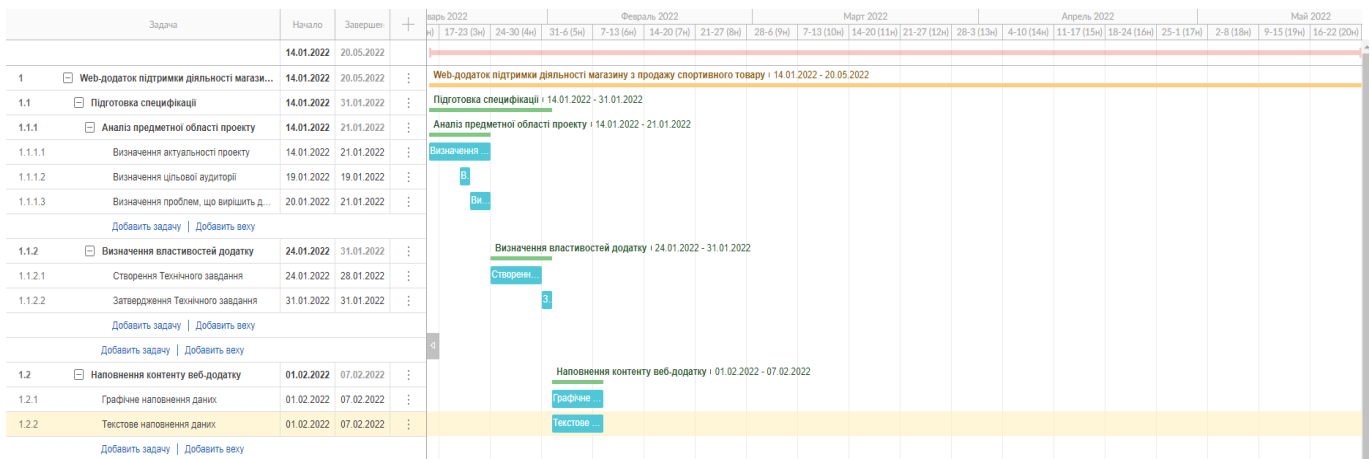


Рисунок Б.3 – Частина діаграми Ганта

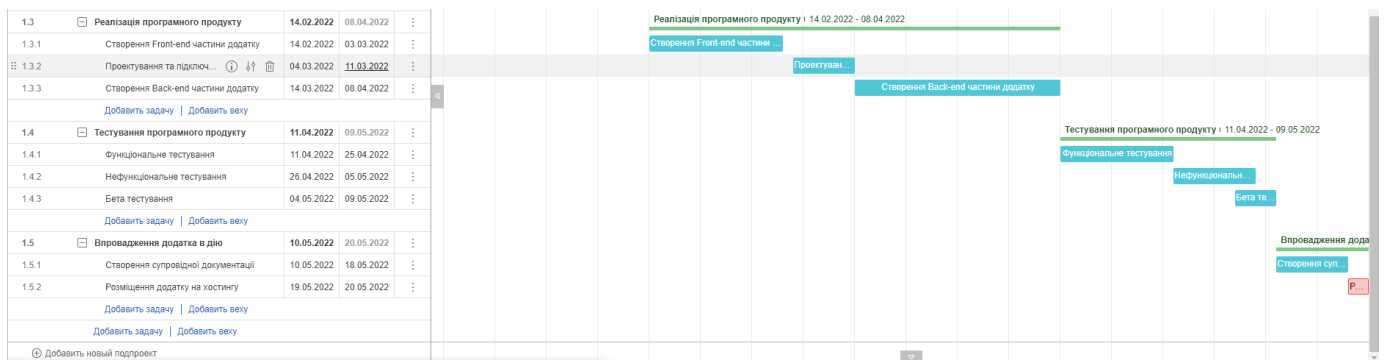


Рисунок Б.4 – Друга частина діаграми Ганта

2.4 Управління ризиками проекту

Аналіз та управління ризиками – це ключова практика управління проектами, що забезпечує мінімальну кількість несподіваних проблем під час реалізації проекту.

Враховуючи ступень важливості ризику, кожен ризик має свій пріоритет та буде виправлений в залежності від пріоритету. У таблиці зображено шкалу для класифікації ризиків за впливом на проект.

Таблиця Б.3 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять (номера)
1	Прийнятні	$1 \leq R \leq 2$	R6
2	Виправдані	$3 \leq R \leq 4$	R1, R4, R7, R9
3	Недопустимі	$6 \leq R \leq 9$	R2, R3, R5, R8

У таблиці Б.4 проведений аналіз ризиків.

Таблиця Б.4 – Ризики проекту

Ризик	Назва ризику	Опис ризику
1	Невиконання роботи в поставлені терміни	Недотримання термінів завершення робіт
2	Зміна вимог	Зміна об'єму та терміну роботи замовником
3	Низька якість вхідного матеріалу	Матеріал для наповнення web-додатку має низьку якість
4	Низький рівень знань працівника	Розробникам web-додатку не вистачає знань на певному етапі розробки
5	Підтримка проекту	Замовник не зможе самостійно підтримувати web-додаток

У таблиці Б.5 проведена класифікація ризиків.

Таблиця Б.5 – Класифікація ризиків

ІД ризику	Ризик	Ймовірність виникнення	Вплив ризику	Ранг ризику
R1	Поява альтернативного продукту	Висока	Середній	3
R2	Втрата актуальності продукту	Низька	Високий	9
R3	Не розрахування часових ресурсів	Середня	Високий	9
R4	Зміна вимог замовником	Низька	Середній	4
R5	Низька якість вхідного матеріалу	Низька	Високий	8
R6	Низька кваліфікація розробника	Середня	Середній	4
R7	Нездатність персоналу самостійно підтримувати продукт	Середня	Низький	2
R8	Вибір неефективної технології розробки	Середня	Середній	4
R9	Помилка в логіці роботи при створенні програмного продукту	Висока	Високий	9
R10	Неповне проведення тестування роботи програмного продукту	Середня	Середній	5

Після класифікації та оцінки ризиків було складено план реагування на ризики, який зображено у таблиці Б.6.

Таблиця Б.6 – План реагування на ризики

ІД ризику	Ризик	Реакція на ризик
R1	Поява альтернативного продукту	Створення унікальної стратегії створення програмного продукту
R2	Втрата актуальності продукту	Створення унікальної стратегії створення програмного продукту
R3	Невиконання роботи в поставлені терміни	Аналіз причини затримки процесів, узгодження з замовником додаткових годин
R4	Зміна вимог замовником	Документування зміни вимог та зміна часу розробки програмного продукту
R5	Низька якість вхідного матеріалу	Разом з замовником обдумати всі проблемні кроки або розповісти, як можна реалізувати завдання
R6	Нездатність персоналу самостійно підтримувати продукт	Залучення кваліфікованих спеціалістів
R7	Вибір неефективної технології розробки	Задokumentувати неефективні технології та провести аналіз технологій розробки
R8	Помилка в логіці роботи при створенні програмного продукту	Звернення до документації, аналіз проблеми, внесення до документації додаткового часу
R9	Неповне проведення тестування роботи програмного продукту	Залучення додаткового спеціаліста, звернутись до документації

ДОДАТОК В

ЛІСТИНГ ПРОГРАМНОГО КОДУ ОСНОВНИХ МОДУЛІВ WEB-ДОДАТКУ

MainPage.js

```
import React from "react";
import 'bootstrap/dist/css/bootstrap.min.css'
import Nav from "react-bootstrap/Nav";
const MainPage = () => {
  return (
    <Container>
      <div className='my-3'>
        <Alert variant={"info"}>
          <h5><Badge bg="warning">New</Badge>Тільки сьогодні! Безкоштовна
доставка!</h5>
        </Alert>
      </div>
      <Carousel>
        <Carousel.Item>
          
          <Carousel.Caption>
            <h3></h3>
          </Carousel.Caption>
        </Carousel.Item>
        <Carousel.Item>
          
          <Carousel.Caption>
            <h3></h3>
          </Carousel.Caption>
        </Carousel.Item>
        <Carousel.Item>
          
          <Carousel.Caption>
            <h3></h3>
          </Carousel.Caption>
        </Carousel.Item>
      </Carousel>
    </Container>
  );
};
```



```

        src="/slider/reebok.jpg"
        alt="Third slide"
    />
    <Carousel.Caption>
        <h3></h3>
    </Carousel.Caption>
</Carousel.Item>
<Carousel.Item>
    
    <Carousel.Caption>
        <h3></h3>
    </Carousel.Caption>
</Carousel.Item>
</Carousel>
<div className='blog my-5' style={{pointerEvents: "none"}}>
    <center></center>
    <Row className='mb-4'>
        <Col md='3'>
            <Card>
                <Card.Img
                    variant="top"
                    src="/modalBlock/basketball.webp"
                />
                <Card.Body>
                    <Card.Title>Баскетбол</Card.Title>
                    <Card.Text>
                        Баскетбол - чудовий спосіб тренуватись в команді
з друзями.
                    </Card.Text>
                    <Card.Text>
                        Великий асортимент товарів
                    </Card.Text>
                </Card.Body>
            </Card>
        </Col>
        <Col md='3'>
            <Card>
                <Card.Img
                    variant="top"
                    src="/modalBlock/cycling.webp"
                />
                <Card.Body>
                    <Card.Title>Велоспорт</Card.Title>
                    <Card.Text>
                        Велоспорт - чудовий вид спорту. Маємо в продажі
велосипеди
                    </Card.Text>
                    <Card.Text>
                        городського та горного типу та аксесуари.
                    </Card.Text>

```

```

        </Card.Body>
      </Card>
    </Col><Col md='3'>
      <Card>
        <Card.Img
          variant="top"
          src="/modalBlock/tennis.webp"
        />
        <Card.Body>
          <Card.Title>Теніс</Card.Title>
          <Card.Text>
            Маємо інвентарь для професійного тенісу, щоб
            гра принесла задоволення та високі результати.
          </Card.Text>
        </Card.Body>
      </Card>
    </Col><Col md='3'>
      <Card>
        <Card.Img
          variant="top"
          src="/modalBlock/training.webp"
        />
        <Card.Body>
          <Card.Title>Активний спорт</Card.Title>
          <Card.Text>
            Для людей, які мають активний спорт. Маємо товари для
            комфортних тренувань та обладнання.
          </Card.Text>
        </Card.Body>
      </Card>
    </Col>
  </Row>
</div>
<div className='info'>
  <center></center>
  <Tab.Container id="left-tabs-example" defaultActiveKey="first">
    <Row>
      <Col sm={3}>
        <Nav variant="pills" className="flex-column">
          <Nav.Item>
            <Nav.Link eventKey="first">Асортимент</Nav.Link>
          </Nav.Item>
          <Nav.Item>
            <Nav.Link eventKey="second">Доставка</Nav.Link>
          </Nav.Item>
          <Nav.Item>
            <Nav.Link eventKey="third">Вигода</Nav.Link>
          </Nav.Item>
          <Nav.Item>
            <Nav.Link eventKey="fourth">Аналітика</Nav.Link>
          </Nav.Item>
          <Nav.Item>

```

```

        <Nav.Link eventKey="fifth">Робота 24/7</Nav.Link>
    </Nav.Item>
</Nav>
</Col>
<Col sm={9}>
    <Tab.Content>
        <Tab.Pane eventKey="first">
            В онлайн магазині Kvikta sport shop маємо великий
асортимент товарів який постійно оновлюється.
            На кожен товар ми надаємо гарантію три місяці.
Магазин Kvikta sport shop співпрацює з великою кількістю
            перевірених постачальників, та рекомендуємо
покупати товари лише від перевірених компаній. Перед тим,
            як викласти в магазин товар, наша команда особиста
перевіряє якість продукції.
        </Tab.Pane>
        <Tab.Pane eventKey="second">
            Маємо великий вибір доставки. В основному Kvitka
sport shop відправляє товари Новою поштою та Укрпоштою.
            Також маємо послуги кур'єра за додаткову плату або
якщо сума чеку більше 15 тисяч гривень. Рекомендуємо
            перед оплатою товару на поштовому відділені
спочатку перевірити товар.
        </Tab.Pane>
        <Tab.Pane eventKey="third">
            Вигода купляти саме в нашому магазині полягає у
тому, що ми надаємо клієнту великий
            асортимент товару та доставляємо за допомогою
поштових служб в максимально короткі терміни.
            Для всіх товарів маємо великий термін гарантії, та
можливість заміни товару, якщо не підійшов розмір.
        </Tab.Pane>
        <Tab.Pane eventKey="fourth">
            Аналітика грає одну з найголовніших ролей
життєдіяльності нашого онлайн магазину.
            Наша команда активно проводить аналітику в
залежності від попиту продукції. Ми стараємось,
            щоб клієнт був макимально задоволений від нашого
сервісу.
        </Tab.Pane>
        <Tab.Pane eventKey="fifth">
            Завдяки тому що це онлайн магазин. Ми можемо
працювати 24/7. Обробка замовлень відбувається
            кожний день з 8 годин ранку до 22 вечора без перерв
та вихідних.
        </Tab.Pane>
    </Tab.Content>
</Col>
</Row>
</Tab.Container>
</div>
</Container>
);
};
export default MainPage;

```

Store.js

```

import ListOfProduct from "../components/ListOfProduct";
import {observer} from "mobx-react-lite";
import {Context} from "../index";
import {effortBrands, effortProduct, effortTypes} from "../http/productAPI";
import Pages from "../components/Pages";
const store= observer(() => {
  const {products} = useContext(Context);
  useEffect(() => {
    effortTypes().then(inf=> product.setTypes(inf));
    effortProduct(null, null, 1, 16).then(inf => {
      products.setProducts(inf.rows);
      products.setTotalCount(inf.count);
    });
  }, []);
  useEffect(
    () => {
      if (product.selectedType === "all") {
        effortProduct(null, products.selectedBrand.id, products.page,
16).then(inf => {
          products.setProducts(inf.rows);
          products.setTotalCount(inf.count);
        });
      } else {
        effortProduct(products.selectedType.id, products.selectedBrand.id,
products.page, 16).then(inf => {
          products.setProducts(inf.rows);
          products.setTotalCount(inf.count);
        });
      }
    }
  ),
);
return (
  <Container>
    <Row className="pt-3">
      <Col md={3}>
        <TypeBar/>
      </Col>
      <Col md={9}>
        <BrandBar/>
        <ProductList/>
        <Pages/>
      </Col>
    </Row>
  </Container>
);
});
export default store;

```

footer.js

```

import './Footer.css';
import React from 'react';
import {observer} from "mobx-react-lite";
import {FontAwesomeIcon} from '@fortawesome/react-fontawesome'
import {faInstagram, faYoutube, faTg, faViber} from "@fortawesome/free-brands-svg-
import {NavLink} from "react-router-dom";

```

```

const Footer = observer(() => {
  return (
    <footer className="site-footer">
      <div className="container">
        <div className="row">
          <div className="col-sm-12 col-md-6">
            <h6>Про інтернет-магазин</h6>
            <p className="text-justify">
              Придбання на замовлення.
              Багато продукції.
              Індивідуальний підхід.
              Сучасний стиль.</p>
          </div>
          <div className="col-xs-6 col-md-3">
            <h6>Контакти</h6>
            <ul className="footer-links">
              <li>LifeCell: <a href="/">+38 (073) 111-12-13</a></li>
              <li>Email: <a href="mailto:jenya.kwitka@gmail.com">jenya.kwitka@gmail.com</a></li>
              <li>Skype: <a href="skype:evgen3607?add">evgen3607</a></li>
              <li>Telegram: <a href="https://t.me/upolnomocheniyy">@upolnomocheniyy</a></li>
              <li>Viber: <a href="viber://chat?number=%2B3845435345">+38 (073) 111-12-13</a></li>
            </ul>
          </div>
          <div className="col-xs-6 col-md-3">
            <h6>Швидкі посилання</h6>
            <ul className="footer-links">
              <li><NavLink to={STORE_ROUTE}>Головна</NavLink></li>
              <li><NavLink to={SIZES}>Розмір одягу</NavLink></li>
              <li><NavLink to={DELIVERY}>Доставка і оплата</NavLink></li>
              <li><NavLink to={CONTACTS}>Контакти</NavLink></li>
              <li><NavLink to={AUTH_ROUTE}>Авторизація</NavLink></li>
            </ul>
          </div>
        </div>
      </div>
      <div className="container">
        <div className="row">
          <div className="col-md-8 col-sm-6 col-xs-12">
            <p className="copyright-text">Copyright &copy; 2021 All Rights
              Reserved by
              <a href="jenya.kwitka@gmail.com">
                jenya.kwitka@gmail.com</a>.
            </p>
          </div>
          <div className="col-md-4 col-sm-6 col-xs-12">
            <ul className="social-icons">
              <li><a className="facebook" href="/"><FontAwesomeIcon
                icon={faViber}/></a></li>
              <li><a className="twitter" href="/"><FontAwesomeIcon
                icon={faTg}/></a></li>
              <li><a className="dribbble" href="/"><FontAwesomeIcon
                icon={faInstagram}/></a></li>
              <li><a className="linkedin" href="/"><FontAwesomeIcon
                icon={faYoutube}/></a></li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  );
});

```

```

        </div>
      </div>
    </div>
  </footer>
)
});
export default Footbar;

```

Ordering.js

```

import React, {useContext, useEffect, useState} from 'react';
const OrderingProduct = () => {
  const {basket, user} = useContext(Context);
  const [number, setNumber] = useState(null);
  const history = useHistory();
  const [numberDirty, setNumberDitry] = useState(false)
  const [numberMiss, setNumberMiss] = useState("Некоректний номер телефону")
  const [formValid, setFormValid] = useState(false)
  const numberHandler = (e) => {
    setNumber(e.target.value)
    const regex = /^[+]?[(]?[0-9]{3}[)]?[-\s\.]?[0-9]{3}[-\s\.]?[0-9]{4,6}$/
    if (!regex.test(String(e.target.value).toLowerCase())) {
      setNumberMiss('Не корректний номер телефону')
    } else {
      setNumberMiss("")
    }
  }
  useEffect(() => {
    if(numberMiss) {
      setFormWeighty(false)
    } else {
      setFormWeighty(true)
    }
  }, [numberMiss])
  const blurHandler = (e) => {
    switch (e.target.className) {
      case 'number form-control' :
        setNumberDitry(true)
    }
  }
  const buy = () => {
    let bill = {
      mobile: number,
      basket: baskets.Basket
    }
    if(user.isAuthorization) {
      bill.authorization = true;
    }
    sendOrder(bill).then(inf => {
      console.log(inf );
      basket.setRemoveAllProduct();
      history.push(STORE_ROUTE);
    });
  }
  return (
    <>
      <Form style={{marginTop: 100}}>

```

```

        {(numberDirty && numberMiss) && <div style={{color:
'red'}}>{numberMiss}</div>}
        <Form.Control
          className="number"
          placeholder="Введіть номер телефону..."
          value={number}
          onChange={e => numberHandler(e)}
          onBlur={e => blurHandler(e)}
          maxLength={13}
        />
      </Form>
      <Row className="mt-3">
        <Col xs={12}>
          <Button variant="secondary" onClick={buy}
disabled={!formValid}>Замовити</Button>
        </Col>
      </Row>
    </>
  );
};
export default OrderingProduct;

```

FirstOrder.js

```

import React, {useEffect, useState} from 'react';
import {Col, Container, Image, Row, Spinner} from "react-bootstrap";
import {useParameters} from "react-router-dom";
import {getFirstOrderProducts} from "../http/API";
const FirstOrder = () => {
  const [load, setLoad] = useState(true);
  const [order, setOrder] = useState([]);
  const {id} = useParams();
  let sumPrice = Number(0);
  useEffect(() => {
    getOneOrderProducts(id).then(inf => {
      setOrder(inf);
      setLoad(false);
    })
  }, []);
  if (load) {
    return <Spinner animation="grow"/>
  }
  //Format date (createdAt)
  const formatDate = (propsDate) => {
    const date = new Date(Date.parse(propsDate));
    const options = {
      weekday: "short",
      hour: 'numeric',
      minute: 'numeric',
      year: 'numeric',
      month: 'numeric',
      day: 'numeric',
      timezone: 'UTC'
    };
    return date.toLocaleString("UA", options);
  }
  return (
    <div>

```

```

    <Container className="">
      ID замовлення: {id} <br/>
      Статус: {order?.description.complete ? "Завершено" : "В процесі"} <br/>
      Клієнт: {order?.description.userId ? order.description.userId : "Клієнт
не зареєстрований"} <br/>
      Дата створення: {formatDate(order?.description.createdAt)} <br/>
      {order?.description.complete ?
formatDate(order.description.complete.updatedAt) : false}
      <a href={`tel:${order?.description.mobile}`}>Телефон:
{order?.description.mobile}</a>
      <br/>
      {order?.products.map(({count, description}, i) => {
        sumPrice += description.price;
      })}
      {order?.products.map(({count, description}, i) => {
        return (
          <Row key={i} className="p-3">
            <Col xs={10}>
              <div style={{marginLeft:"100px"}} >
                Бренд: {description.brand.name}<br/>
                Категорія: {description.type.name}<br/>
                Назва: {description.name}<br/>
                Ціна: {description.price} грн<br/>
                Кількість: {count}<br/>
                Ціна: {count * description.price} грн
              </div>
            </Col>
          </Row>
        )
      })}
      <h5><b>Загальна ціна замовлення: {sumPrice} грн</b></h5>
    </Container>
  </div>
);
};
export default FirstOrder;

```

reomoveBrandOrType.js

```

import React, {useEffect, useState} from 'react';
import {Button, Dropdown, Modal} from "react-bootstrap";
useEffect(() => {
  effortTypes().then(inf => setTypes(inf));
  effortBrands().then(inf => setBrands(inf));
}, []);
const Remove() => {
  if(brand === "Бренд") {
    if(selectBrand.name !== "Бренд не обраний") {
      removeBrand(selectBrand.id).then(inf => {
        showSucsgFunc(inf);
        onHide();
        setSelectBrand({name: "Бренд не обраний"});
      });
    } else {
      setMsgErr("Виберіть бренд");
      setShowMsgError(true);
    }
  }
}

```



```

} else {
  if(selectType.name !== "Категорія не обрана") {
    removeType(selectType.id).then(inf => {
      showSucsgFunc(inf);
      onHide();
      setSelectType({name: "Категорія не обрана"});
    });
  } else {
    setMsgErr("Виберіть категорію");
    setShowMsgError(true);
  }
});
<Modal
  show={show}
  onHide={onHide}
  size="lg"
  centered>
  <Modal.Header closeButton>
    <Modal.Title>
      Видалить бренд або категорію
    </Modal.Title>
  </Modal.Header>
  <Modal.Body>
    {showMsgErr &&
      <>
        <p style={{color: "red", textAlign: "center"}}>{msgErr}</p>
      </>
      Виберіть категорію:
      <Dropdown className="mb-3" style={{margin: "0 auto"}}>
        <Dropdown.Toggle variant="success" id="dropdown-basic">
          {brandOrType}
        </Dropdown.Toggle>
        <Dropdown.Menu>
          {brandOrType === "Бренд" ? <Dropdown.Item
disabled>Бренд</Dropdown.Item> : <Dropdown.Item onClick={() =>
setBrandOrType("Бренд")}>Бренд</Dropdown.Item>}
          {brandOrType === "Категорія" ? <Dropdown.Item
disabled>Категорія</Dropdown.Item> : <Dropdown.Item onClick={() =>
setBrandOrType("Категорія")}>Категорія</Dropdown.Item>}
        </Dropdown.Menu>
      </Dropdown>
      <Dropdown className="mb-3" style={{margin: "0 auto"}}>
        <Dropdown.Toggle variant="успіх" id="dropdown-basic">
          {brandOrType === "Бренд" ? selectBrand.name : selectType.name}
        </Dropdown.Toggle>
        <Dropdown.Menu>
          </Dropdown.Menu>
        </Dropdown>
    </Modal.Body>
    <Modal.Footer>
      <Button variant="outline-danger" onClick={onHide}>Закрити</Button>
      <Button variant="outline-success" onClick={Remove}>Видалити</Button>
    </Modal.Footer>
  </Modal>
);
};
export default RemoveBrandOrType;

```