

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
**Факультет електроніки та інформаційних технологій**  
**Кафедра інформаційних технологій**

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Світлана ВАЩЕНКО

\_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**

**на здобуття освітнього ступеня бакалавр**

зі спеціальності 122 «Комп'ютерні науки»,

освітньо-професійної програми «Інформаційні технології проектування»

на тему: Вебдодаток підтримки діяльності магазину спортивних товарів

Здобувача групи ІТ-03-1 Попок Владислав Іванович  
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

\_\_\_\_\_ (підпис)

Владислав ПОПОК

(Ім'я та ПРІЗВИЩЕ здобувача)

Керівник д.т.н., професор Євгеній ЛАВРОВ  
(посада, науковий ступінь, вчене звання, ім'я та ПРІЗВИЩЕ) (підпис)

**Суми – 2024**

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

**ЗАТВЕРДЖУЮ**

В. о. зав. кафедри ІТ

\_\_\_\_\_ Світлана ВАЩЕНКО  
«\_\_» \_\_\_\_\_ 2024 р.

## **З А В Д А Н Н Я**

### **НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

Попок Владислав Іванович

**1 Тема роботи** Вебдодаток підтримки діяльності магазину спортивних товарів

**керівник роботи** Лавров Євгеній Анатолійович, д.т.н. , професор

затверджені наказом по університету від «07» травня 2024 р. №0482-VI

**2 Строк подання студентом роботи** «26» травня 2024 р.

**3 Вхідні дані до роботи** Технічне завдання на розробку Вебдодатка підтримки діяльності магазину спортивних товарів

**4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)** Аналіз предметної області, проектування веб-додатку, практична реалізація веб-додатку.

**5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** Актуальність, постановка задачі, аналіз аналогів веб-додатку, таблиця порівняння аналогів, функціональні вимоги, структура інформаційної системи (контекстна діаграма, діаграма декомпозиції, діаграма використання), модель бази даних, технології розробки, структура проекту, реалізація проекту та інтерфейсу, висновки . Слайдів 22

## 6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7.Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Оформлення плану роботи	04.04.2024 - 10.04.2024	
2	Оформлення технічного завдання	11.04.2024 - 16.04.2024	
3	Аналіз предметної області	17.04.2024 – 20.04.2024	
4	Проектування веб-додатку	21.04.2024 – 30.04.2024	
5	Розробка веб-додатку	01.05.2024 – 15.05.2024	
6	Завантаження веб-додатку на хостинг	16.05.2024 – 19.05.2024	
7	Оформлення пояснювальної записки.	20.05.2024 – 24.05.2024	

Студент

\_\_\_\_\_

(підпис)

Владислав ПОПОК

Керівник роботи

\_\_\_\_\_

(підпис)

д.т.н. , професор Євгеній ЛАВРОВ

## РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Вебдодаток підтримки діяльності магазину спортивних товарів».

Пояснювальна записка складається зі вступу, 3 розділів, висновків, списку використаних джерел із 17 найменувань, додатків. Загальний обсяг роботи – 66 сторінок , у тому числі 30 сторінок основного тексту , 3 сторінка списку використаних джерел, 33 сторінки додатків.

Кваліфікаційну роботу бакалавра присвячено розробці вебдодатку підтримки діяльності магазину спортивних товарів.

До першого розділу входить: огляд останніх досліджень і публікації, аналіз аналогів програмних продуктів. Також в даному розділі було поставлено мету та задачі дослідження.

У другому розділі було виконано проектування інформаційної системи, у результаті чого розроблено діаграми IDEF та UML-схеми. Було проведено детальний аналіз вимог користувачів, що допомогло визначити основні функціональні можливості системи. Також було спроектовано базу даних. Було розроблено структуру проекту для демонстрації роботи веб-додатка.

Результатом виконання даної роботи є створений та опублікований веб-додаток.

Ключові слова: React.js, Node.js, PostgreSQL, вебдодаток

# ЗМІСТ

ВСТУП .....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Огляд останніх досліджень і публікацій.....	7
1.2 Аналіз аналогів програмних продуктів.....	8
1.3 Мета та задачі дослідження.....	11
2 ПРОЕКТУВАННЯ ВЕБДОДАТКУ.....	12
2.1 Проектування інформаційної системи.....	12
2.2 Проектування бази даних.....	14
2.3 Архітектура додатку.....	20
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ.....	22
3.1 Програмна реалізація.....	22
3.2 Використання розроблюваного продукту.....	22
ВИСНОВКИ .....	30
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	31
ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ.....	34
ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ.....	41
ДОДАТОК В. ЛІСТИНГ ПРОГРАМНОГО КОДУ.....	47

## ВСТУП

У сучасному світі, де використання сучасних інформаційних технологій проникає в усі сфери бізнесу, веб-додатки стають ключовим елементом успішної комерційної діяльності. Магазины спортивних товарів, зокрема, можуть значно виграти від впровадження спеціалізованих веб-додатків. Ці інструменти не лише дозволяють розширити клієнтську базу, але й надають зручний спосіб для здійснення покупок та організації доставки. Вони сприяють персоналізації пропозицій і підвищують рівень взаємодії з клієнтами.

Веб-додаток також відкриває можливості для проведення ефективних маркетингових кампаній, збору даних для аналітики. Всі ці переваги підкреслюють важливість і актуальність розробки вебдодатку для підтримки діяльності магазину спортивних товарів.

Спортивні товари сьогодні користуються великим попитом серед різних категорій споживачів, від професійних спортсменів до людей, що ведуть активний спосіб життя. Проте, висока конкуренція серед виробників і продавців спортивних товарів вимагає від компаній впровадження інноваційних рішень для забезпечення конкурентних переваг. Веб-додаток може стати критичним фактором успіху, допомагаючи залучати нових клієнтів, підвищувати рівень продажів.

Незважаючи на численні дослідження у сфері електронної торгівлі та онлайн-магазинів спортивних товарів, створення ефективного веб-додатку для підтримки діяльності магазину спортивних товарів залишається актуальною і важливою задачею.

Мета – розробка вебдодатку для підтримки діяльності магазину спортивних товарів, що допоможе покращити бізнес-процеси та залучити більше клієнтів.

Для досягнення цієї мети, необхідно вирішити наступні завдання:

- провести аналіз предметної області та існуючих веб-додатків.
- розробити структуру та функціонал веб-додатку;
- реалізувати веб-додаток і забезпечити його стабільну роботу;

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Огляд останніх досліджень і публікацій

На сьогодні використання веб-технологій відбувається майже у кожній галузі людського життя – починаючи від особистих вподобань і до господарської діяльності. Після настання інформаційної ери виникли такі зміни: люди спілкуються між собою (находячись друг від друга на відстані), проводять обмін інформацією, навчаються та працюють. [1]

Виникнення веб технологій створило новий напрямок в збільшенню прибутку для підприємців, тому, що використання даної технології дає можливість, продавати товар не залежно від місця розташування магазину. Використовуючи веб технології збільшиться зв'язок між підприємцями та клієнтами.

На сьогоднішній день коли швидко росте прогрес у графічному та текстовому оформленні вже майже не можливо нічим здивувати. Для того, щоб на ринку розробки веб-сторінок відбувалась конкуренція, для цього відвідувачам надають різні можливості: авторизація, пошукова система, підтримка та багато іншого.

В результаті виконання даної роботи магазин з продажу спортивних товарів отримає веб-ресурс, завдяки якому він стане більш конкурентним в даній галузі.

Всі методи розробки сайтів можна умовно розділити на 3 основні частини: клієнтська частина, адміністративна панель та серверна частина.

Клієнтська частина - це фронтенд додаток, який відповідає за відображення інтерфейсу користувачу. Вона написана з використанням бібліотеки React і містить всі необхідні компоненти для відображення каталогу товарів, корзини покупок, форм замовлення та іншого функціоналу.

Адміністративна панель - це інтерфейс, призначений для адміністраторів магазину. Вона також написана з використанням React і містить компоненти для управління товарами, замовленнями, користувачами та іншими аспектами магазину.

Серверна частина - це бекенд додаток, який відповідає за обробку запитів від клієнтської та адміністративної частин. Вона написана на Node.js 18 і використовує PostgreSQL як базу даних. Для комунікації між клієнтською та серверною частиною використовується GraphQL.

Node.js є безкоштовним серверним середовищем з відкритим вихідним кодом, яке дозволяє виконувати JavaScript-код на сервері. Воно побудоване на двигуні JavaScript V8, розробленому компанією Google для використання в браузері Google Chrome. Node.js дозволяє виконувати JavaScript-код поза браузером, що розширює можливості мови JavaScript на серверному боці.[2]

PostgreSQL це вільна і відкрита система керування базами даних (СУБД) з відкритим вихідним кодом. Є однією з найпопулярніших СУБД, тому її використовують тисячі організацій по всьому світу. [3]

Фреймворк React.js — це фреймворк і бібліотека JavaScript з відкритим вихідним кодом, розроблені та підтримувані Facebook і Instagram. Він використовується для швидкого та ефективного створення інтерактивних користувацьких інтерфейсів і вебдодатків із застосуванням значно меншої кількості коду, ніж під час використання звичайного JavaScript.[4]

GraphQL - мова запитів з відкритим вихідним кодом, розроблена компанією Facebook. Вона створювалася як ефективніша альтернатива REST для розробки і використання програмних інтерфейсів застосунків.[5]

## **1.2 Аналіз аналогів програмних продуктів**

Спочатку над уже існуючи інформаційні системи проведемо аналіз та порівняємо їх між собою, для визначення переваг та недоліків.

Для проведення аналізу аналогів було взято: SPORTMARKET, NEWT, FITNSS MARKET .



Перший сайт, магазин SPORTMARKET. Виконаний у чорних, жовтих та білих кольорах, інформація поділена по категоріях, присутня система пошуку.

Недоліком є велика кількість інформації в верхній частині сайту.[6]

Сторінки магазину зображена на рисунку поданий нижче (рис.1.1)

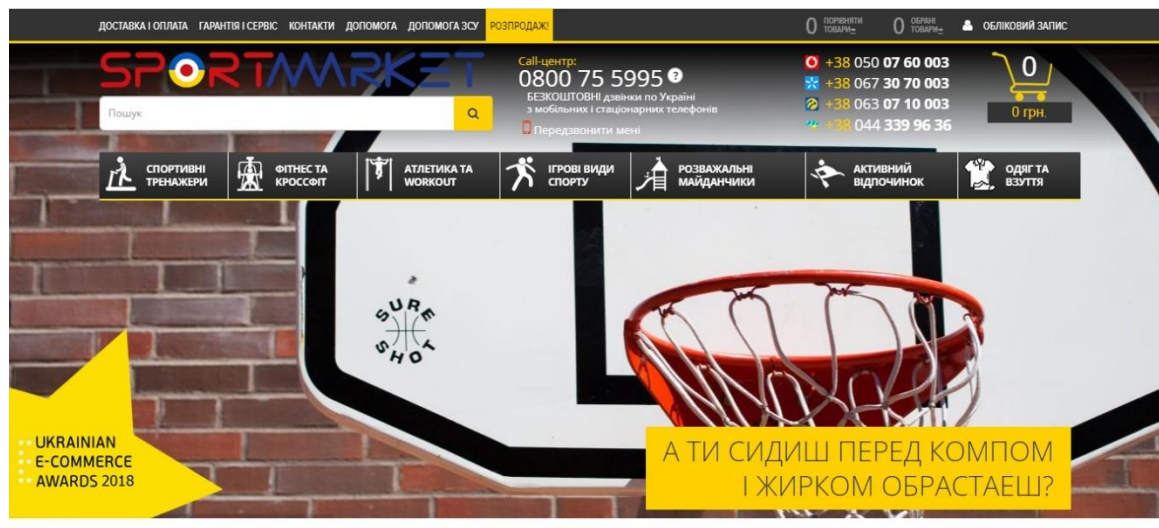


Рисунок 1.1 – Дизайн магазину SPORTMARKET

Другий сайт, магазин NEWT . Виконаний у мінімалістичних тонах, інформація поділена на категорії, присутня система пошуку. Недоліком є не упорядковане найменування цін, одна ціна пишеться « від 1437» , а на іншому товарі пишеться « від 1,625».[7]

Сторінки магазину зображена на рисунку поданий нижче (рис.1.2)

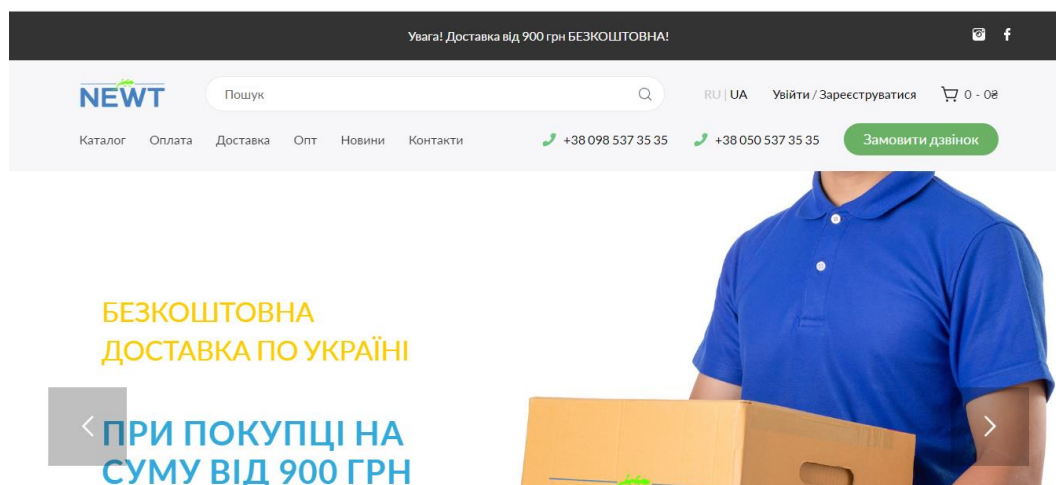


Рисунок 1.2 – Дизайн магазину NEWT

Третій сайт, магазин FITNSS MARKET. Серед всіх проаналізованих сайтів, цей сайт по функціоналу та по розташуванню об'єктів являється більш реалізований. [8]

Сторінки магазину зображена на рисунку поданий нижче (рис.1.3)

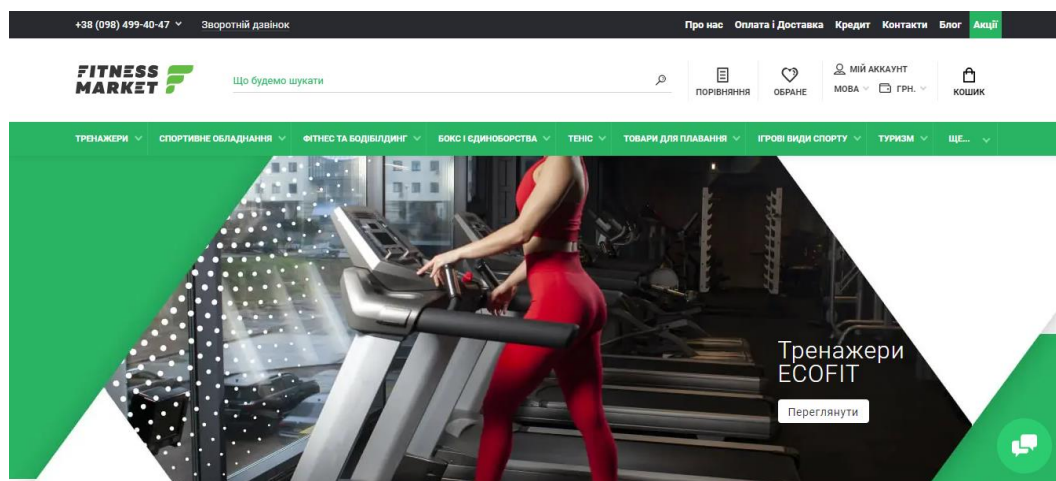


Рисунок 1.3 – Дизайн магазину FITNSS MARKET

Порівняння аналогів магазинів спортивних товарів приведено нижче в таблиці 1.1.

Таблиця 1.1 – Порівняння аналогів

	Сайт PORTMARKET	Сайт NEWT	Сайт FITNSS MARKET
Система пошуку	+	+	+
Адаптивність	+	+	+
Дизайн	-	+-	+
Авторизація / реєстрація	+	+	+
Оцінка	3	3,5	4

Після проведення аналізу вже існуючих ресурсів, були виявлені як позитивні, так і негативні їх якості. Цей аналіз дозволив краще зрозуміти, які характеристики є бажаними, а яких недоліків необхідно уникнути у розроблюваному проекті.

Особливу увагу необхідно приділити питанням ергономічності[9-12], юзабіліті [13 ] та надійності людино-машинної взаємодії [14-16], та, що буде доброю базою забезпечення ефективності розробки[17] .

### **1.3 Мета та задачі дослідження**

Метою даної роботи є розробка веб-додатку підтримки діяльності магазину спортивних товарів. Дана інформаційна система повинна мати такий функціонал:

- Інформація про категорії товарів;
- Інформація про новинки сезону;
- Зв'язок з адміністратором;
- Розмежування рівня доступу;
- Можливість авторизуватися реєструватися;
- Можливість замовлення товарів.

Для досягнення запланованої мети потрібно виконати такі задачі:

- Провести аналіз аналогів;
- Провести розробку дизайну для веб-додатку;
- Провести реалізацію функціоналу веб-додатку.

Для реалізації клієнтської частини було обрано React, а для розробки розмітки було обрано SCSS.

Повне технічне завдання наведено нижче у додатку А.

## 2 ПРОЕКТУВАННЯ ВЕБДОДАТКУ

### 2.1 Проектування інформаційної системи

Для моделювання функціональної частини було використано методологію IDF0 - це визначення інтеграції для моделювання процесів, методологія загальнодоступного домену, яка використовується для моделювання бізнесу та їхніх процесів, щоб їх можна було зрозуміти та вдосконалити.

Функціональна модель для веб-додатку підтримки діяльності магазину спортивних товарів продемонстрована на рисунку 2.1.1.



Рисунок 2.1.1 – Діаграма IDF0

Для створення концептуальної моделі було використано такі дані :

- вхідні данні: дані користувача , параметри замовлення;

- механізми: прикладне програмне забезпечення, веб додаток , технічне забезпечення, користувач;
- управління: права доступу , список товарів;
- вихідні данні: оформлене замовлення

Діаграма декомпозиції для функціональної моделі веб-додатку підтримки діяльності магазину спортивних товарів продемонстрована на рисунку 2.1.2

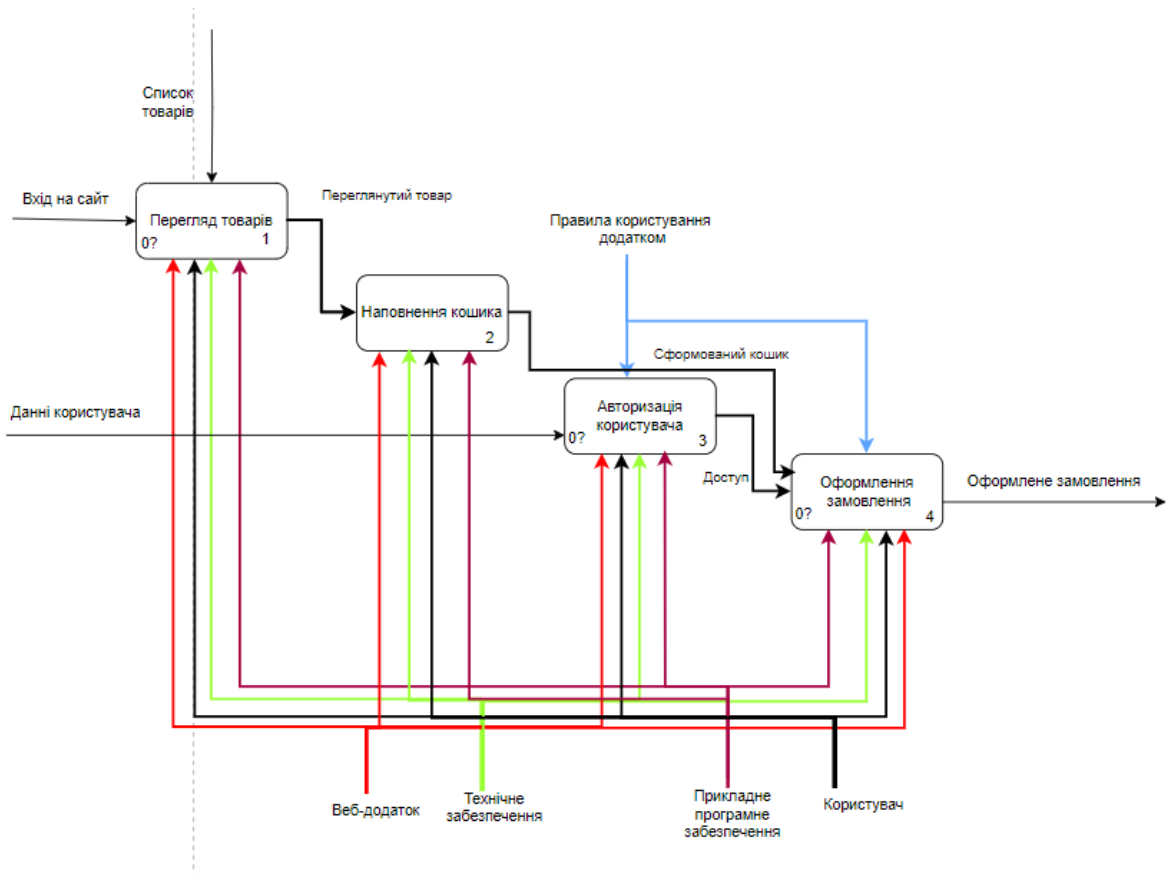


Рисунок 2.1.2 – Діаграма декомпозиції

Діаграма декомпозиції з таких основних блоків: перегляд товарів, наповнення кошика, авторизація користувача , оформлення замовлення.

Діаграма варіантів використання (use case diagram, діаграма сценаріїв) - це вихідна концептуальна модель системи в процесі її проектування і розробки, на

якій зображуються відношення між зовнішніми агентами і варіантами використання.

Діаграма варіантів використання представлена на рисунку 2.1.3.

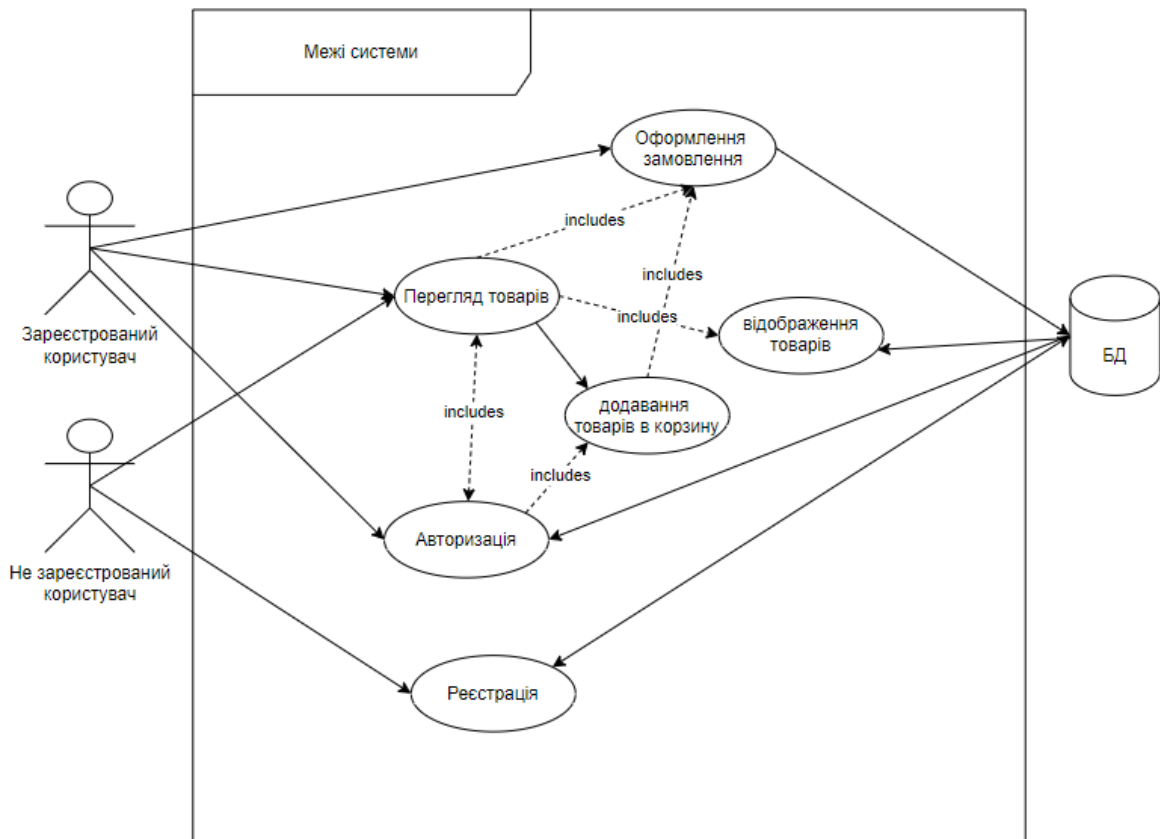


Рисунок 2.1.3 – Діаграма варіантів використання

У даній діаграмі використання було задіяно 3 сутності:

- Не зареєстрований користувач;
- Зареєстрований користувач;
- База даних (БД) .

## 2.2 Проектування бази даних

Реляційна база даних – це структурована організована у вигляді таблиць. У кожній таблиці дані представлені у вигляді рядків та стовбців, де рядки відображають окремі запити, а стовбці визначають типи даних. Діаграма бази

даних веб додатку підтримки діяльності магазину спортивних товарів наведена на рисунку 2.2.1.

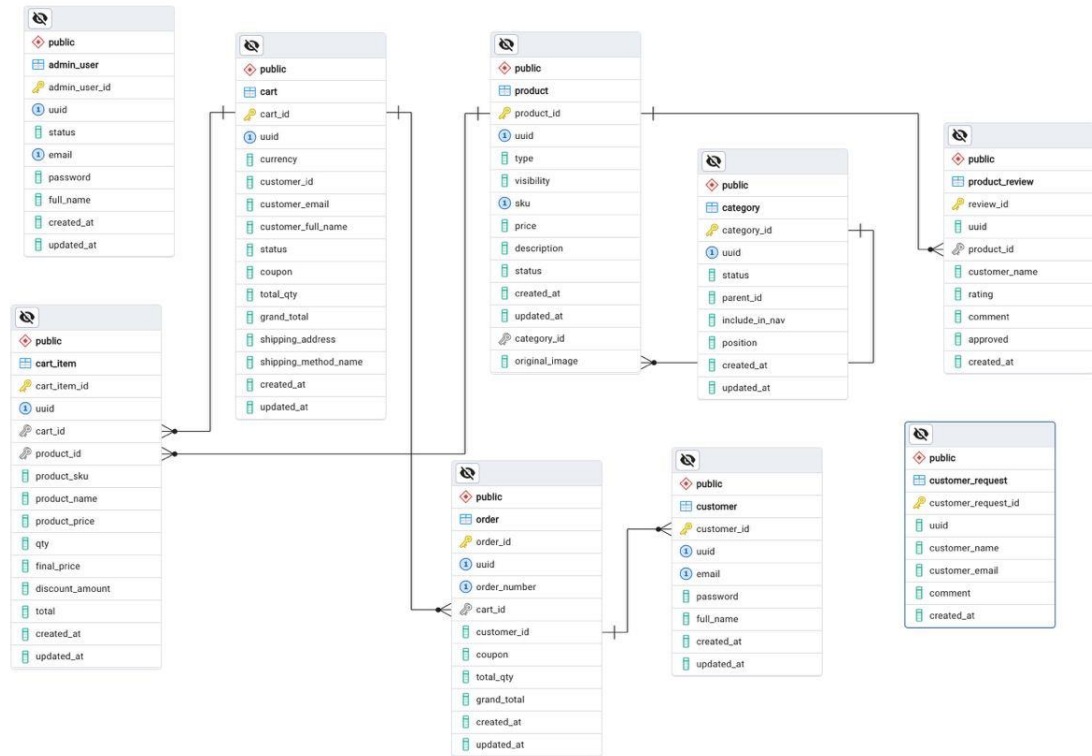


Рисунок 2.2.1 – Діаграма бази даних

Призначення кожного сховища в базі даних:

admin\_user - користувач в адмін панелі.

cart - корзина для оформлення замовлення, може бути створена навіть не для за реєстрованого юзера.

cart\_item - товари в корзині.

order - замовлення, має посилання на корзину з якої було зроблене.

customer - користувач сайту(клієнт).

customer\_request - заявка з форми зконтакуйте з нами.

product - товар на сайті.

category - категорії товарів на сайті.

product\_review - відгуки юзерів про товар.

Опис атрибутів для кожного сховища наведено в таблицях 2.2.1-2.2.7

Таблиця 2.2.1 – Опис атрибутів сховища «admin\_user»

Атрибут	Опис
admin_user_id	Унікальний ідентифікатор адміністратора.
uuid	Унікальний ідентифікатор для глобальної унікальності.
status	Статус адміністратора
email	Електронна пошта адміністратора.
password	Пароль адміністратора
full_name	Повне ім'я адміністратора
created_at	Дата та час створення запису
updated_at	Дата та час останнього оновлення запису

Таблиця 2.2.3 – Опис атрибутів сховища «cart»

Атрибут	Опис
cart_id	Унікальний ідентифікатор кошика
uuid	Унікальний ідентифікатор для глобальної унікальності
currency	Валюта кошика
customer_id	Посилання на користувача сайту
customer_email	Електронна пошта користувача
customer_full_name	Повне ім'я користувача
status	Статус кошика
coupon	Купон, застосований до кошика.
total_qty	Загальна кількість товарів у кошику
grand_total	Загальна вартість кошика
shipping_address	Адреса доставки.
shipping_method_name	Назва методу доставки
created_at	Дата та час створення кошика
updated_at	Дата та час останнього оновлення кошика



Таблиця 2.2.4 – Опис атрибутів сховища «cart\_item»

Атрибут	Опис
cart_item_id	Унікальний ідентифікатор товару в кошику
cart_id	Посилання на кошик
uuid	Унікальний ідентифікатор для глобальної унікальності
product_id	Посилання на товар
product_sku	SKU товару
product_price	Ціна товару
qty	Кількість товару
final_price	Кінцева ціна товару з урахуванням знижок
discount_amount	Сума знижки
total	Загальна вартість товарів у кошику
created_at	Дата та час додавання товару до кошика
updated_at	Дата та час останнього оновлення товару в кошику

Таблиця 2.2.5 – Опис атрибутів сховища «category»

Атрибут	Опис
category_id	Унікальний ідентифікатор категорії
uuid	Унікальний ідентифікатор для глобальної унікальності
status	Статус категорії
parent_id	Ідентифікатор батьківської категорії
include_in_nav	Чи включена категорія в навігацію.
position	Позиція категорії в списку.
created_at	Дата та час створення категорії.
updated_at	Дата та час останнього оновлення категорії.

Таблиця 2.2.6 – Опис атрибутів сховища «customer\_request»

Атрибут	Опис
customer_request_id	Унікальний ідентифікатор заявки.
uuid	Унікальний ідентифікатор для глобальної унікальності
customer_name	Ім'я відправника заявки.
customer_email	Електронна пошта відправника заявки.
comment	Повідомлення, надіслане користувачем.
created_at	Дата та час створення заявки.

Таблиця 2.2.7 – Опис атрибутів сховища «order»

Атрибут	Опис
order_id	Унікальний ідентифікатор товару в кошику
uuid	Унікальний ідентифікатор для глобальної унікальності.
order_number	Номер замовлення.
cart_id	Посилання на кошик, з якого було створено замовлення.
customer_id	Посилання на користувача сайту.
customer_email	Електронна пошта користувача.
customer_full_name	Повне ім'я користувача
status	Статус замовлення.
coupon	Купон, застосований до замовлення.
total_qty	Загальна кількість товарів у замовленні
grand_total	Загальна вартість замовлення
payment_method_name	Назва методу оплати.
created_at	Дата та час створення замовлення
updated_at	Дата та час останнього оновлення замовлення.

Таблиця 2.2.8 – Опис атрибутів сховища «product»

Атрибут	Опис
product_id	Унікальний ідентифікатор товару.
uuid	Унікальний ідентифікатор для глобальної унікальності.
type	Тип товару.
visibility	Видимість товару на сайті
price	Ціна товару.
weight	Вага товару
status	Статус товару
created_at	Дата та час створення товару.
updated_at	Дата та час останнього оновлення товару
category_id	Посилання на категорію, до якої належить товар.

Таблиця 2.2.9 – Опис атрибутів сховища «product\_review»

Атрибут	Опис
review_id	Унікальний ідентифікатор відгуку.
product_id	Посилання на товар
customer_name	Ім'я користувача, що залишив відгук.
rating	Рейтинг, наданий користувачем
comment	Коментар користувача.
approved	Чи затверджений відгук
created_at	Дата та час створення відгуку.

Таблиця 2.2.10 – Опис атрибутів сховища «customer»

Атрибут	Опис
coupon_id	Унікальний ідентифікатор купону
uuid	Унікальний ідентифікатор для глобальної унікальності.
code	Код купону
status	Статус купону

## Продовження таблиці 2.2.10

description	Опис купону
discount_amount	Сума знижки
free_shipping	Чи надає купон безкоштовну доставку
start_date	Дата початку дії купону
end_date	Дата закінчення дії купону
created_at	Дата та час створення купону
updated_at	Дата та час останнього оновлення купону

### 2.3 Архітектура додатку

Архітектура проекту складається : клієнтської частини , адміністративної панелі та серверної частини

Клієнтська частина – відповідає за відображення інтерфейсу користувача.

Адміністративна панель – відповідає за адміністрування магазину.

Серверна частина - відповідає за обробку запитів від клієнтської та адміністративної частин.

Архітектура веб-додатку зображена на рисунку 2.3.1

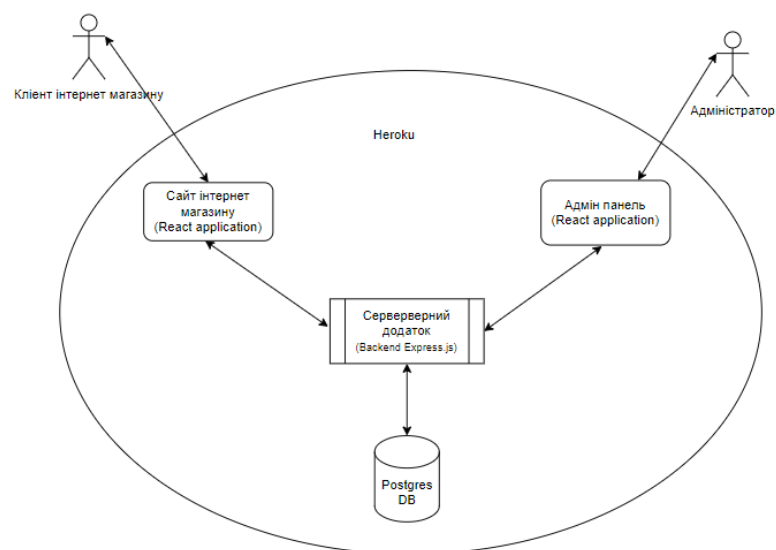


Рисунок 2.3.1 – Архітектура веб-додатку

Веб-додаток виконаний у стилі чистий дизайн, основні характеристики такого стилю:

Простота і лаконічність – мінімальна кількість на сторінці;

Багато простору – велика кількість білого простору;

Обмежена кольорова палітра – Використання невеликої кількості кольорів;

Чітка типографіка – використання простих та читабельних шрифтів;

Мінімум графічних елементів – відсутність зайвих декоративних елементів.

## 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ

### 3.1 Програмна реалізація

Для управління структурою базой даних в проєкті використовується міграції які дозволяють додавати видаляти та змінювати таблиці безпечно та контрольовано  
 рисунок 3.1.1

```
const { execute } = require('@evershop/postgres-query-builder');

// eslint-disable-next-line no-multi-assign
module.exports = exports = async (connection) : Promise<void> => {
  await execute(
    connection,
    `CREATE TABLE "customer_request" (
      "customer_request_id" INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) PRIMARY KEY,
      "uuid" UUID NOT NULL DEFAULT gen_random_uuid (),
      "customer_name" varchar NOT NULL,
      "customer_email" varchar NOT NULL,
      "comment" varchar DEFAULT NULL,
      "created_at" TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
    )
  `);
};
```

Рисунок 3.1.1 – Використання міграції

Для додавання нового адміністратора використовується наступна команда:

```
npm run user:create -- --email "user email" --password "user password" --name "user name"
```

Для зміни пароля адміністратора використовується наступна команда:

```
npm run user:changePassword -- --email "user email" --password "new password"
```

Лістинг коду продемонстровано нижче в ДОДАТКУ В.

### 3.2 Використання розроблюваного продукту

Після входу на сайт ми потрапляємо на головну сторінку рисунок 3.2.1 на якій можливо побачити 4 основні блоки :

- шапка сайту;
- інформаційна частина;
- категорії товарів;
- новинки сезону.

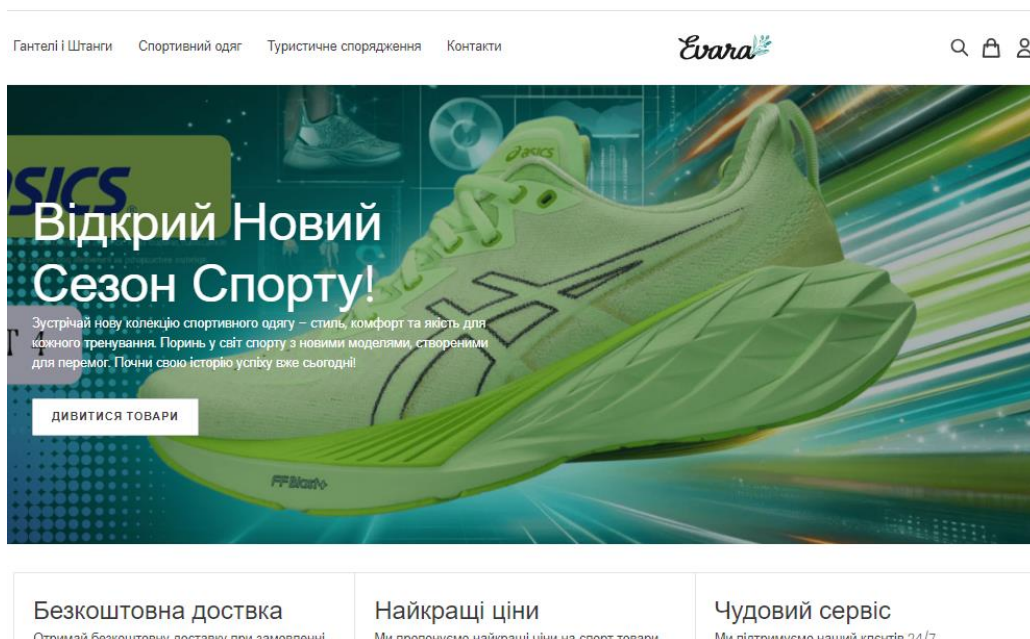


Рисунок 3.2.1 – Головна сторінка

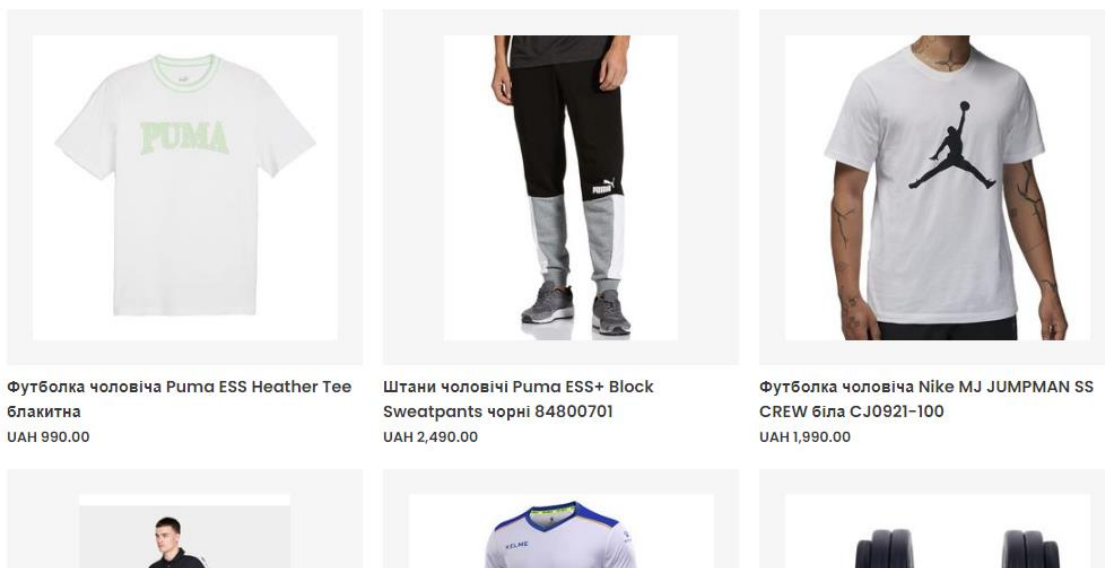
В шапці головної сторінки, як і в інших сторінках заходиться панель навігації рисунок 3.2.2. На даній панелі розміщено: категорії всіх товарів, контакти, логотип веб-додатку, пошук товарів , корзина та профіль клієнта.



Рисунок 3.2.2 – Навігаційна панель

В розділі «Новинка сезону» рисунок 3.2.3 знаходяться колекція нових товарів.

## НОВИНКИ СЕЗОНУ



Футболка чоловіча Puma ESS Heather Tee  
блакитна  
UAH 990.00

Штани чоловічі Puma ESS+ Block  
Sweatpants чорні 84800701  
UAH 2,490.00

Футболка чоловіча Nike MJ JUMPMAN SS  
CREW біла CJ0921-100  
UAH 1,990.00

Рисунок 3.2.3 – Новинка сезону

При виборі товару відкривається сторінка з інформацією про обраний товар рисунок 3.2.4 , також можна побачити в якій категорії розташований товар.

[Home](#) / [Спортивний одяг](#) / Футболка чоловіча Puma ESS Heather Tee блакитна

Рисунок 3.2.4 – Перегляд одного товару

При додаванні товару в кошик відображається форма рисунок 3.2.5 , для сповіщення про успішне додавання товару до вашої корзини.



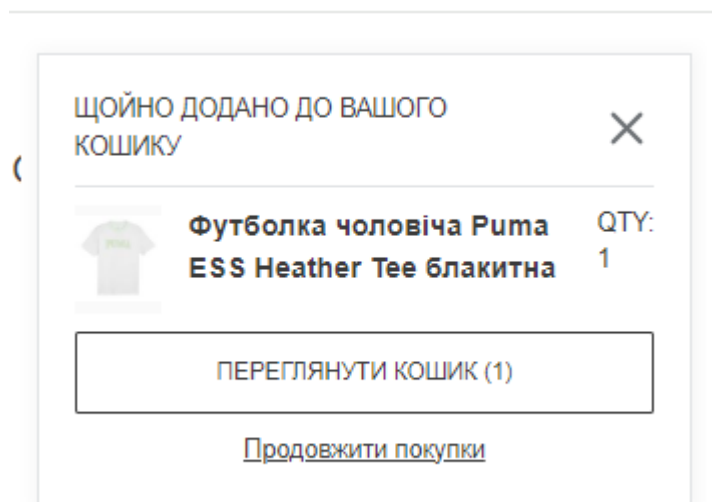


Рисунок 3.2.5 – Товар додано в корзину

Кошик товарів рисунок 3.2.6 містить товари які були додані до кошику.

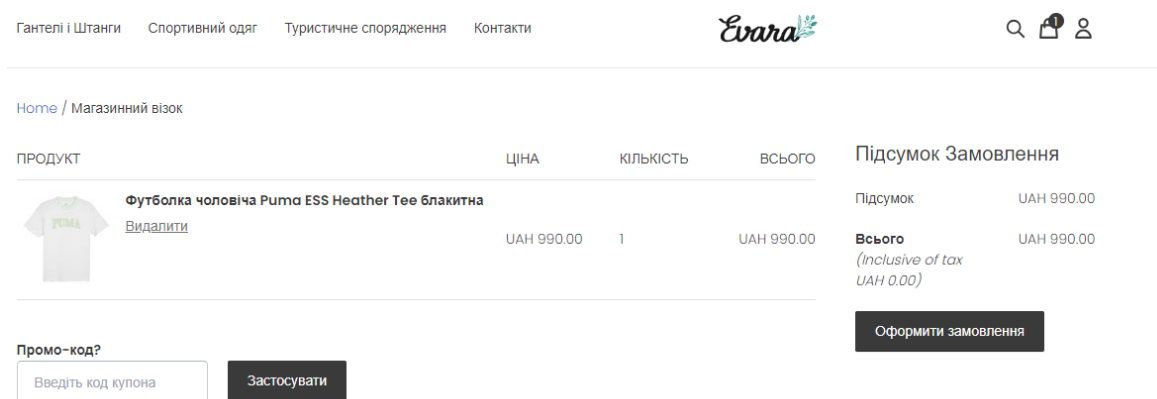
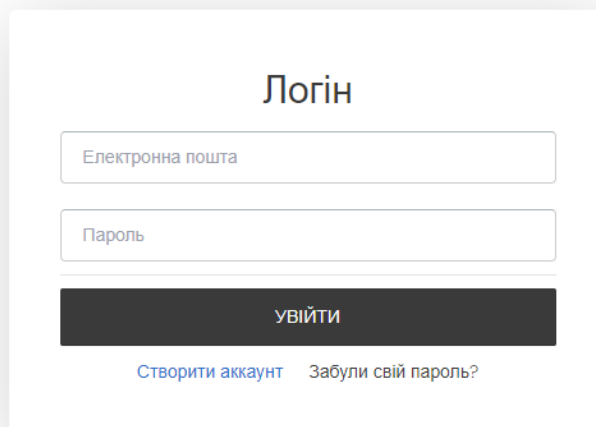


Рисунок 3.2.6 - Корзина

Для оформлення замовлення клієнту необхідно авторизуватися , сторінка авторизації рисунок 3.2.7.

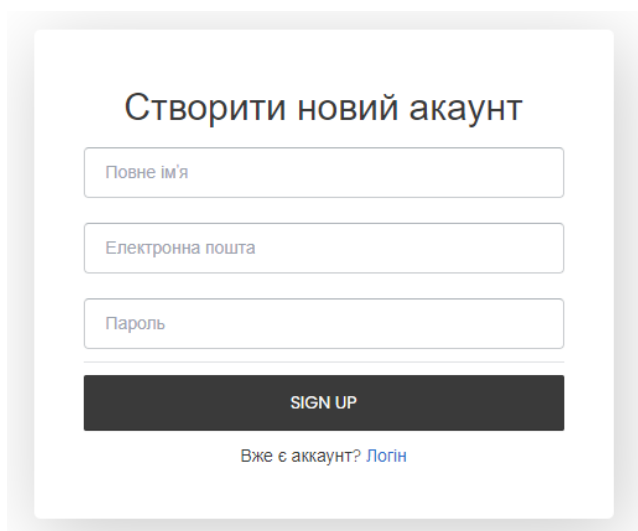


The screenshot shows a login form with the following elements:

- Title: Логін
- Input field: Електронна пошта
- Input field: Пароль
- Button: УВІЙТИ
- Links: [Створити акаунт](#) and [Забули свій пароль?](#)

Рисунок 3.2.7 – Сторінка авторизації

Якщо у клієнт не зареєстрований , він може відкрити сторінку Створити акаунт рисунок 3.2.8 де він зможе зареєструватися .



The screenshot shows a registration form with the following elements:

- Title: Створити новий акаунт
- Input field: Повне ім'я
- Input field: Електронна пошта
- Input field: Пароль
- Button: SIGN UP
- Link: [Вже є акаунт? Логін](#)

Рисунок 3.2.8 – Сторінка реєстрації

Після того як клієнт авторизувався у нього відкривається вікно оформлення замовлення рисунок 3.2.9

Контактна інформація > Доставка > Payment

Контакт vpopok802@gmail.com [Зміна](#)

**Адреса доставки**

Повне ім'я  Телефон

Адреса

Місто

Країна

Поштовий індекс

**Спосіб доставки**

Sorry, there is no available method for your address


	<b>Футболка чоловіча Puma ESS Heather</b> Тее блакитна	UAH 792.00
<b>Підсумок</b>	1 елементи	UAH 990.00
<b>Знижка</b>	NEWCUSTOMER20	UAH 198.00
<b>Всього</b> <i>(Inclusive of tax UAH 0.00)</i>		<b>UAH 792.00</b>

Рисунок 3.2.9 – Оформлення замовлення

Після того як користувач ввів всі данні у нього проявляється додатковий рядок де відображається спосіб доставки рисунок 3.2.10.

**Адреса доставки**

Повне ім'я  Телефон

Адреса

Місто

Країна

Область  Поштовий індекс

**Спосіб доставки**

Нова пошта - UAH 70.00

[Продовжити до оплати](#)

<b>Підсумок</b>	1 елементи	UAH 990.00
<b>Знижка</b>	NEWCUSTOMER20	UAH 198.00
<b>Всього</b> <i>(Inclusive of tax UAH 0.00)</i>		<b>UAH 792.00</b>

Рисунок 3.2.10 – Спосіб доставки

Після того як клієнт вибере продовжити до оплати у нього відкриється сторінка «Платіжний адрес» рисунок 3.2.11.

The screenshot shows the checkout page for Evara. The navigation bar includes links for 'Гантелі і Штани', 'Спортивний одяг', 'Туристичне спорядження', and 'Контакти'. The Evara logo is on the right, along with search, cart, and user icons. The breadcrumb trail is 'Home / Checkout'. The current step is 'Payment', with previous steps being 'Контактна інформація' and 'Доставка'. The contact information section shows an email 'vropok802@gmail.com' and a shipping address 'Перемоги, Обмачів, Ukraine'. The payment address section has a checked box indicating it matches the shipping address. The payment method 'CASH ON DELIVERY' is selected. The order summary on the right lists the items and their prices.

Контактна інформація		
Контакт	vropok802@gmail.com	<a href="#">Зміна</a>
Відправити до	Перемоги, Обмачів, Ukraine	<a href="#">Зміна</a>

**Платіжна адреса**

Моя платіжна адреса збігається з адресою доставки

**Спосіб оплати**


CASH ON DELIVERY

**Зробити замовлення**

Сума		
Підсумок	1 елементи	UAH 990.00
Доставка	Нова пошта	UAH 70.00
Знижка	NEWCUSTOMER20	UAH 198.00
<b>Всього</b>		<b>UAH 862.00</b>
	(Inclusive of tax UAH 0.00)	

Рисунок 3.2.11 – Платіжна адреса

Після того як клієнт нажав «Зробити замовлення» на екрані відобразився чек рисунок 3.2.12

 Порядок #10035  
Дякую Vladislav!

<b>Контактна інформація</b> Vladislav vpopok802@gmail.com	<b>Спосіб оплати</b> Оплата при отриманні
<b>Адреса доставки</b> Vladislav Popok Перемоги 16503, Батурин Chernihivska oblast, Ukraine 0635284780	<b>Платіжна адреса</b> Vladislav Popok Перемоги 16503, Батурин Chernihivska oblast, Ukraine 0635284780

ПРОДОВЖИТИ ПОКУПКИ

Рисунок 3.2.12 – Чек

## ВИСНОВКИ

В результаті виконання даної роботи було розроблено вебдодаток підтримки діяльності магазину спортивних товарів.

Під час проведення аналізу аналогів було визначено основні які повинні бути реалізовані в подальшій реалізації проекту.

Під час проектування та модулювання проекту, для кращого розуміння системи інформації та дій під час реалізації проекту було розроблено діаграму IDF0, діаграму декомпозиції, діаграму використання. Для кращого розуміння реалізації бази даних було розроблено логічну модель бази даних.

На етапи розробки було реалізовано веб-додаток та виставлено його на хостинг. Реалізовано веб-додаток за допомогою React та Node.js в якості хостингу було використано Heroku.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Сучасність web-технології. URL: <https://angular.tutorial.in.ua/what-are-nodejs-and-npm/> (дата звернення: 06.04.2024).
2. Node.js. URL: <https://angular.tutorial.in.ua/what-are-nodejs-and-npm/> (дата звернення: 07.04.2024).
3. PostgreSQL. URL: <https://foxminded.ua/postgresql-shcho-tse/> (дата звернення: 14.04.2024).
4. React. URL: <https://dan-it.com.ua/uk/blog/chto-takoe-react-js-i-dlja-chego-on-nuzhen/> (дата звернення: 16.04.2024).
5. GraphQL. URL: <https://devzone.org.ua/post/graphql-dlia-rochatkivtsiv> (дата звернення: 19.04.2024).
6. SPORTMARKET. URL: <https://sportmarket.ua> (date of access: 02.03.2024).
7. NEWT. URL: <https://newt.com.ua/> (дата звернення: 10.03.2024).
8. FITNESS MARKET. URL: <https://fitnessmarket.ua/> (дата звернення: 07.03.2024).
9. E. Lavrov, N. Pasko and V. Borovyk, "Management for the Operators Activity in the Polyergatic System. Method of Functions Distribution on the Basis of the Reliability Model of System States," 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2018, pp. 1-6, doi: 10.1109/INFOCOMMST.2018.8632102.
10. E. Lavrov et al., "Human Factors in Automated Systems. Methodology for the Computer System "International Center for the Support of Ergonomic Design and Research"," 2023 IEEE 5th International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 2023, pp. 1-5, doi: 10.1109/AICT61584.2023.10452692.
11. E. Lavrov, O. Siryk, B. Rybalov and D. Velykodnyi, "The Method of Adaptive Control of the Process of Human-Machine Interaction to Ensure the Reliability

and Efficiency of Automated Control Systems," 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021, pp. 403-407, doi: 10.1109/PICST54195.2021.9772126.

12. E. Lavrov, O. Siryk, Y. Chybiriak, L. Danilova, V. Nahorni and S. Vakal, "A Model for the Organization of Adaptive Dialogue Interaction "Man-Computer" Taking into Account the Requirements of Reliability and Efficiency," 2021 IEEE 4th International Conference on Advanced Information and Communication Technologies (AICT), Lviv, Ukraine, 2021, pp. 31-35, doi: 10.1109/AICT52120.2021.9628939.

13. V. Kompaniets, A. Lyz and A. Kazanskaya, "An Empirical Study of Goal Setting in UX/UI-design," 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT), Tashkent, Uzbekistan, 2020, pp. 1-5, doi: 10.1109/AICT50176.2020.9368570.

14. E. A. Lavrov, N. B. Pasko and V. E. Snytyuk, "Information Technology for Distribution of Functions between Operators as a Means of Improving the Reliability of Polyergatic Systems," 2018 Third International Conference on Human Factors in Complex Technical Systems and Environments (ERGO)s and Environments (ERGO), 2018, pp. 71-76, doi: 10.1109/ERGO.2018.8443832.

15. E. Lavrov, G. Kozhevnykov, N. Pasko, V. Gonchar and V. Mukoseev, "Improvement for Ergonomic Quality of Man-Machine Interaction in Automated Systems based on the Optimization Model," 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2018, pp. 735-740, doi: 10.1109/INFOCOMMST.2018.8632074.

16. E. A. Lavrov, A. A. Volosiuk, N. B. Pasko, V. P. Gonchar and G. K. Kozhevnikov, "Computer Simulation of Discrete Human-Machine Interaction for Providing Reliability and Cybersecurity of Critical Systems," 2018 Third International Conference on Human Factors in Complex Technical Systems and Environments (ERGO)s and Environments (ERGO), 2018, pp. 67-70, doi: 10.1109/ERGO.2018.8443846.

17. Lavrov, E., Pasko, N. (2018). Development of Models for Computer Systems of Processing Information and Control for Tasks of Ergonomic Improvements.



In: Damaševičius, R., Vasiljeviėnė, G. (eds) Information and Software Technologies. ICIST 2018. Communications in Computer and Information Science, vol 920. Springer, Cham. [https://doi.org/10.1007/978-3-319-99972-2\\_8](https://doi.org/10.1007/978-3-319-99972-2_8)

## **ДОДАТОК А. ТЕХНІЧНЕ ЗАВДАННЯ**

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ  
СУМЛЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

### **Технічне завдання на створення програмного продукту**

«Вебдодаток підтримки діяльності магазину спортивних товарів»

## **1. Призначення та метод створення web-додатку**

### **1.1 Призначення web-додатку**

Дана інформаційна система збільшення конкурентної спроможності в сфері продажу спортивних товарів.

### **1.2 Мета створення web-додатку**

Метою інформаційної системи є збільшення кількості клієнтів спортивних товарів.

### **1.3 Цільова аудиторія**

До цільової аудиторії відносяться такі групи:

1. Працівники
2. Відвідувачі
3. Клієнти

## **2. Вимоги до web-додатку**

### **2.1 Вимоги до web-додатку в цілому**

#### **2.1.1 Вимоги до структури й функціоналу web-додатку**

Даний інформаційний ресурс повинен бути виконаний у вигляді web-додатку

#### **2.1.2 Вимоги до персоналу**

Від персоналу вимагається базові навички роботи з web-додатком та базові навички для роботи з персональним комп'ютером .

#### **2.1.3 Вимоги до збереження інформації**

Уся інформація яка надана у інформаційній системі повинна зберігатися в базі даних.

### **2.1.4 Вимоги до розмежування доступу**

Користувачів сайту є 2 категорії :

Адміністратор – може редагувати інформацію на сайті.

Користувач – переглядає інформацію та створює замовлення.

## **2.2 Структура інформаційної системи**

### **2.2.1 Загальна інформація про структуру web-додатку**

Структурою інформаційної системи є набір сторінок, які зв'язані між собою.

Головна – містить інформацію про категорії та новинки сезону.

Гантелі і Штанги – категорія товарів

Спортивний одяг – категорія товарів

Туристичне спорядження – категорія товарів

Контакти – зв'язок з адміністратором

Логотип – перехід на головну сторінку

Лупа – відповідає за пошук товарів

Корзина – перехід на сторінку вибраних товарів

Авторизація – переходить на сторінку з якої можна авторизуватися та реєструватися.

### **2.2.2 Навігація**

Для переходу між сторінками створено шапку, де будуть продемонстровані всі сторінки. Воно потрібне для швидкого переходу між сторінками.

### **2.2.3 Наповнення інформаційної системи (контент)**

Заповнення та керування інформаційною системою повинно відбуватися через панель адміністратора.

## 2.2.4 Дизайн та структура додатку

Інформаційна система повинна бути виконаною у сучасному стилі. Демонстрація розташування елементів на головній сторінці, продемонстровано нижче на рисунку А.1



Рисунок А.1 – Схема головної сторінки

## 2.3 Вимоги до функціонування системи

### 2.3.1 Потреби користувача

Потреби користувача продемонстровані в таблиці А.1

Таблиця А.1 – Потреби користувача.

<b>ID</b>	<b>Потреби користувача</b>	<b>Джерело</b>
UN-01	Перегляд творів	Клієнт
UN-02	Можливість замовлення товарів	Клієнт
UN-03	Система пошуку товарів	Клієнт
UN-04	Система сортування товарів	Клієнт
UN-05	Панель адміністратора	Адміністратор
UN-06	Перегляд новинок	Клієнт

### 2.3.2 Функціональні вимоги

Відповідно до потреб користувача було визначено функціональні вимоги:

- Система замовлення товарів.
- Можливість пошуку товарів.
- Можливість сортувати товари.
- Можливість редагувати інформацію.
- Можливість додавати інформацію

### 2.3.3 Системні вимоги

Перелік всіх системних вимог продемонстровано в табл.А.2.

Таблиця А.2 – Системні вимоги

<b>ID</b>	<b>Системні вимоги</b>	<b>Пріоритет</b>	<b>Опис</b>
SR-01	Каталог	Обов'язково	Створює асортимент товарів
SR-02	Новинка сезону	Бажано	Демонстрація нових сезонних товарів

SR-03	Система пошуку	Обов'язково	Допомагає знайти потрібний товар
SR-03	Система сортування	Обов'язково	Допомагає знайти потрібну категорію товарів
SR-04	Панель адміністратора	Обов'язково	Надає можливість збільшити асортимент товарів

## **2.4 Вимоги до видів забезпечення**

### **2.4.1 Вимоги до інформаційного забезпечення**

Реалізація інформаційної системи відбувається за допомогою:

- React
- SCSS

### **2.4.2 Вимоги до лінгвістичного забезпечення**

Інформаційна система повинна бути виконана українською мовою.

### **2.4.3 Вимоги до програмного забезпечення**

Для використання інформаційної системи програмне забезпечення повинно відповідати таким вимогам:

Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 3.5 і вище, або Opera 9.5 і вище, або Safari 3.2.1 і вище, або Chrome 2 і вище.

### 3 Склад і зміст робіт зі створення інформаційної системи

Детальний опис етапів зі створенням інформаційної системи продемонстровано в таблиці А.3.

Таблиця А.3 – етапи створення інформаційної системи

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Постановка задач	1 день
2	Аналіз аналогів	1 день
3	Розробка технічного завдання	3 дні
4	Верстка дизайну	4 дня
5	Розробка каталогу	4 дня
6	Перевірка роботи	1 день
7	Завершення роботи	1 день
	Загальна тривалість робіт	19 днів

### 4 Вимоги до складу й змісту робіт із введення інформаційної системи в експлуатацію

Для того, щоб можна було вводити інформаційну систему в експлуатацію потрібно придбати доменне ім'я та місце на хостингу. Для коректної роботи інформаційної системи потрібно, дотримуватися всіх вимог зазначених в ТЗ.



## ДОДАТОК Б. ПЛАНУВАННЯ РОБІТ

**Деталізація мети проекту методом SMART.** Продуктом дипломної роботи є веб додаток підтримки діяльності магазину спортивних товарів.

Результати деталізації методом SMART продемонстровані нижче в табл.Б.1.

Таблиця Б.1 – Деталізація методом SMART

Specific (конкретна)	Створити інформаційну систему підтримки діяльності магазину спортивних товарів.
Measurable (вимірювана)	Результатом роботи проекту є оцінка замовника.
Achievable (досяжна)	Реалізація системи здійснюється в середовищі Visual Studio Code.
Relevant (реалістична)	Для реалізації є всі необхідні програмні та технічні засоби. Розробник має достатню кваліфікацію для виконання завдання.
Time-framed (обмежена у часі)	Виконання роботи повинно бути виконано в термін, якій був узгоджений. Проект має бути завершений відповідно до календарного плану.

**Планування WBS.** WBS згрупована ієрархічна діаграма, яка використовується для групування елементів проекту у вигляді пакету робіт. Діаграма декомпозиції продемонстрована на рис.Б.1.

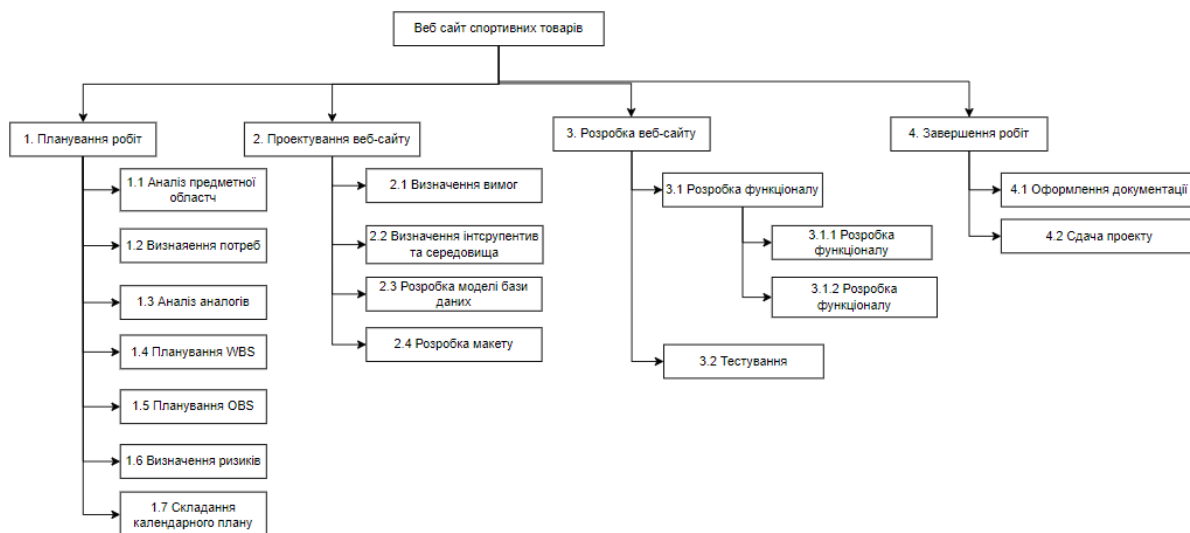


Рисунок Б.1 – WBS. Структура робіт проекту

**Планування OBS.** OBS ця структура використовується для організації виконавців WBS. OBS діаграма продемонстрована на рис.Б.2. Список виконавців, що мають роль в проекті продемонстровано в табл. Б.2.

Таблиця Б.2 – Виконавці проекту

Роль	Ім'я	Проектна роль
Розробник	Попок В.І.	Займається розробкою інформаційної системи
Проектувальник	Попок В.І.	Розробляє базу даних та дизайн
Тестувальник	Попок В.І.	Відповідає за тестування інформаційної системи
Консультант проекту	Лавров Є.А.	Формує завдання на розробку проекту.
Менеджер проекту	Попок В.І.	Відповідає за дотримання термінів розробки. Виконує збір та аналіз даних.

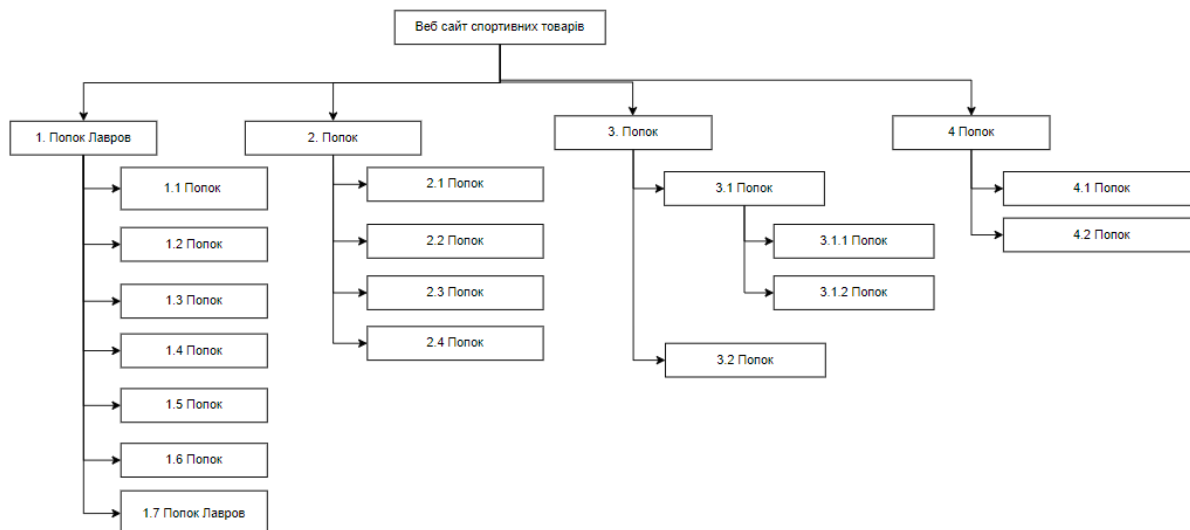


Рисунок Б.2 – OBS

**Діаграма Ганта.** Розроблено календарний план для виконання дипломного проекту. Діаграма Ганта є найпоширенішим форматом графіку у великій кількості галузях. Даною діаграмою користуються менеджери проектів для того щоб візуалізувати терміни виконання та етапи розробки проекту. Діаграма Ганта продемонстрована на рис Б.3 – Б.4.

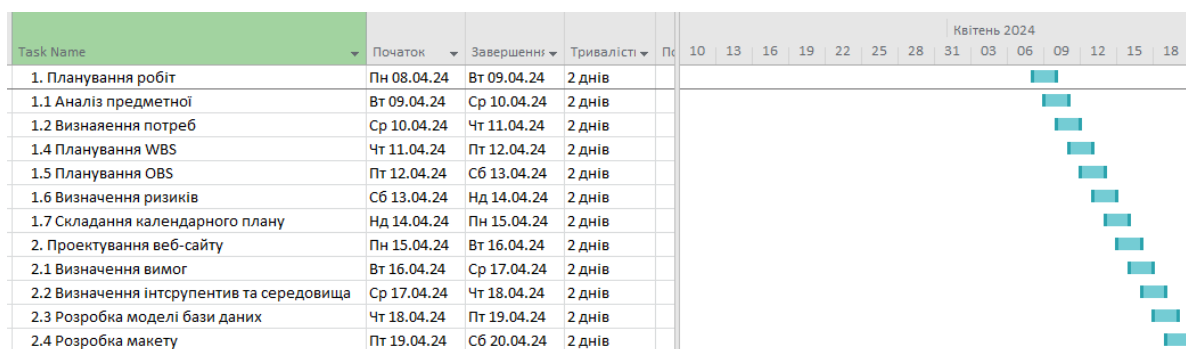


Рисунок Б.3 – Діаграма Ганта

3. Розробка веб-сайту	Сб 20.04.24	Нд 21.04.24	2 днів	
3.1 Розробка функціоналу	Нд 21.04.24	Пн 22.04.24	2 днів	
3.1.1 Розробка функціоналу	Пн 22.04.24	Вт 23.04.24	2 днів	
3.1.2 Розробка функціоналу	Вт 23.04.24	Ср 24.04.24	2 днів	
3.2 Тестування	Ср 24.04.24	Чт 25.04.24	2 днів	
4. Завершення робіт	Чт 25.04.24	Пт 26.04.24	2 днів	
4.1 Оформлення документації	Пт 26.04.24	Нд 28.04.24	2 днів	
4.2 Сдача проекту	Нд 28.04.24	Вт 30.04.24	3 днів	

Рисунок Б.4 – Продовження діаграми Ганта

**Аналіз ризиків.** Розробляємо кількісну та якісну оцінку ризиків. Під час проведення якісної оцінки ризиків визначаються ризики, що потребують швидкого реагування. Завдяки такій оцінці ризиків можна визначити рівень пріоритетності ризику і спосіб реагування. Виконання кількісної оцінки ризиків потрібна для повної ідентифікації ризиків та рівня впливу на проект. Класифікація ризиків за ймовірністю їх виникнення та величині втрат продемонстровано у табл. Б.3.

Розробляємо план реагування на ризики. Виконується для зниження негативного впливу на проект. Проаналізувати ефективність розробки реагування, визначити можливість виникнення наслідків впливу негативний або позитивний. Проводиться оцінка ризиків за показниками, які продемонстровані в табл. Б.3. На основі оцінки ризиків створюємо матрицю ймовірності їх виникнення та впливу, яка зображена на рис. Б.5.

Таблиця Б.3. – Шкала оцінювання ймовірності ризику.

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

3	RS_3	RS_3	RS_1, RS_2
2	RS_6		RS_4
1		RS_8	RS_5
	1	2	3

Рисунок Б.5 – Матриця ймовірності виникнення та впливу ризиків

червоний колір – недопустимі ризики;

жовтий колір – виправданні ризики;

зелений колір – прийнятні ризики.

Рівень оцінювання ризиків продемонстровано в таблиці Б.4.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$1 \leq R \leq 3$	6,8
2	Виправдані	$4 \leq R \leq 6$	5,7
3	Недопустимі	$7 \leq R \leq 9$	1,2,3,4

Таблиця Б.5 – Оцінка ймовірності виникнення, величини витрат та індексу ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Збій роботи технічного забезпечення	Низька	Високий	9	Додавання системи резервного копіювання даних	Попередження	Обновити технічне забезпечення
RS_2	Відкритий	Збій роботи програмного забезпечення	Низька	Високий	9	Додавання системи резервного копіювання даних	Попередження	Своєчасна обнова необхідного програмного забезпечення
RS_3	Відкритий	Внесення змін від замовника	Низька	Високий	7	Детально обговорити розробки проекту	Пом'якшення	Перерахунок вартості роботи
RS_4	Відкритий	Помилка в розробці	Середній	Високий	8	Тісно співпрацювати з замовником	Пом'якшення	Проводити регулярно перевірку результату
RS_5	Відкритий	Мала потужність технічного забезпечення	Середній	Середній	6	Детально обговорити задачу з врахування потужності обладнання	Пом'якшення	Замінити на більш потужне технічне забезпечення
RS_6	Відкритий	Відсутність резервного копіювання	Низька	Середній	3	Налаштувати систему резервного копіювання даних	Попередження	Робити копію на кожному етапі розробки
RS_7	Відкритий	Відсутній моніторинг проекту	Середній	Середній	4	Здійснення контролю головних етапів розробки	Перенос	Надання результатів всіх етапів
RS_8	Відкритий	Верстка не того проекту	Низька	Низька	2	Проведення контролю етапів розробки	Пом'якшення	Тісна співпраця з розробником

## ДОДАТОК В. ЛІСТИНГ ПРОГРАМНОГО КОДУ

### Програмний код

Повний код знаходиться за посиланням <https://github.com/vladislav-porok/VLAD.git>

#### Logo.jsx

```
import React from "react";

function Logo() {
  return (
    <div>
      <a href="/">
        
      </a>
    </div>
  );
}

export default Logo;

export const layout = {
  areaId: "header",
  sortOrder: 5,
};
```

#### Layout.jsx

```
import React from "react";
import Area from "@components/common/Area";
import LoadingBar from "@components/common/LoadingBar";
import "../css/global.scss";
import "./Layout.scss";
import "./tailwind.scss";

export default function Layout() {
  return (
    <>
      <LoadingBar />
      <div className="header">
        <div className="page-width flex justify-between">
          <Area
```

```

        id="header"
        noOuter
        coreComponents={[
          {
            component: { default: Area },
            props: {
              id: "icon-wrapper",
              className: "icon-wrapper flex justify-between
space-x-1",
            },
            sortOrder: 20,
          },
        ]}
      />
    </div>
  </div>
  <main className="content">
    <Area id="content" className="" noOuter />
  </main>
  <div className="footer">
    <div className="page-width grid grid-cols-1 md:grid-cols-2
gap-2 justify-between">
      <div>
        <div className="card-icons flex justify-center space-
x-1 md:justify-start">
          <div>
            <svg
              xmlns="http://www.w3.org/2000/svg"
              width="38"
              height="24"
              aria-labelledby="pi-visa"
              viewBox="0 0 38 24">
              <path
                d="M35 0H3C1.3 0 0 1.3 0 3v18c0 1.7 1.4 3 3
3h32c1.7 0 3-1.3 3-3V3c0-1.7-1.4-3-3-3z"
                opacity="0.07"
              />
              <path
                fill="#fff"
                d="M35 1c1.1 0 2 .9 2 2v18c0 1.1-.9 2-2 2H3c-
1.1 0-2-.9-2-2V3c0-1.1-.9-2 2-2h32"
              />
              <path
                fill="#142688"
                d="M28.3 10.1H28c-.4 1-.7 1.5-1 3h1.9c-.3-1.5-
.3-2.2-.6-3zm2.9 5.9h-1.7c-.1 0-.1 0-.2-.11-.2-.9-.1-.2h-2.4c-.1
0-.2 0-.2.21-.3.9c0 .1-.1.1-.1.1h-2.11.2-.5L27 8.7c0-.5.3-.7.8-

```



```
.7h1.5c.1 0 .2 0 .2.2l1.4 6.5c.1.4.2.7.2 1.1.1.1.1.1.1.2zm-13.4-
.3l.4-1.8c.1 0 .2.1.2.1.7.3 1.4.5 2.1.4.2 0 .5-.1.7-.2.5-.2.5-
.7.1-1.1-.2-.2-.5-.3-.8-.5-.4-.2-.8-.4-1.1-.7-1.2-1-.8-2.4-.1-
3.1.6-.4.9-.8 1.7-.8 1.2 0 2.5 0 3.1.2h.1c-.1.6-.2 1.1-.4 1.7-.5-
.2-1-.4-1.5-.4-.3 0-.6 0-.9.1-.2 0-.3.1-.4.2-.2.2-.2.5 0
.7l.5.4c.4.2.8.4 1.1.6.5.3 1 .8 1.1 1.4.2.9-.1 1.7-.9 2.3-.5.4-
.7.6-1.4.6-1.4 0-2.5.1-3.4-.2-.1.2-.1.2-.2.1zm-3.5.3c.1-.7.1-.7.2-
1 .5-2.2 1-4.5 1.4-6.7.1-.2.1-.3.3-.3H18c-.2 1.2-.4 2.1-.7 3.2-.3
1.5-.6 3-1 4.5 0 .2-.1.2-.3.2M5 8.2c0-.1.2-.2.3-.2h3.4c.5 0 .9.3 1
.8l.9 4.4c0 .1 0 .1.1.2 0-.1.1-.1.1-.1l2.1-5.1c-.1-.1 0-.2.1-
.2h2.1c0 .1 0 .1-.1.2l-3.1 7.3c-.1.2-.1.3-.2.4-.1.1-.3 0-.5
0H9.7c-.1 0-.2 0-.2-.2L7.9 9.5c-.2-.2-.5-.5-.9-.6-.6-.3-1.7-.5-
1.9-.5L5 8.2z"
```

```
</>
```

```
</svg>
```

```
</div>
```

```
<div>
```

```
<svg
```

```
xmlns="http://www.w3.org/2000/svg"
```

```
width="38"
```

```
height="24"
```

```
aria-labelledby="pi-master"
```

```
viewBox="0 0 38 24">
```

```
<path
```

```
d="M35 0H3C1.3 0 0 1.3 0 3v18c0 1.7 1.4 3 3
3h32c1.7 0 3-1.3 3-3V3c0-1.7-1.4-3-3z"
```

```
opacity="0.07"
```

```
</>
```

```
<path
```

```
fill="#fff"
```

```
d="M35 1c1.1 0 2 .9 2 2v18c0 1.1-.9 2-2 2H3c-
1.1 0-2-.9-2-2V3c0-1.1.9-2 2-2h3z"
```

```
</>
```

```
<circle cx="15" cy="12" r="7" fill="#EB001B" />
```

```
<circle cx="23" cy="12" r="7" fill="#F79E1B" />
```

```
<path
```

```
fill="#FF5F00"
```

```
d="M22 12c0-2.4-1.2-4.5-3-5.7-1.8 1.3-3 3.4-3
5.7s1.2 4.5 3 5.7c1.8-1.2 3-3.3 3-5.7z"
```

```
</>
```

```
</svg>
```

```
</div>
```

```
<div>
```

```
<svg
```

```
viewBox="0 0 38 24"
```

```
xmlns="http://www.w3.org/2000/svg"
```

```
width="38"
```



```

    );
  }

export const layout = {
  areaId: "body",
  sortOrder: 1,
};

```

## Newsletter.jsx

```

import React, {useRef, useState} from "react";
import {Box, Modal} from "@mui/material";

const style = {
  position: 'absolute',
  top: '50%',
  left: '50%',
  transform: 'translate(-50%, -50%)',
  width: 400,
  bgcolor: 'background.paper',
  border: '2px solid #000',
  boxShadow: 24,
  pt: 2,
  px: 4,
  pb: 3,
};

function Newsletter() {
  const [isWrong, setIsWrong] = useState()
  const [open, setOpen] = useState(false)
  const inputRef = useRef(null)
  const [email, setEmail] = useState()
  const onClick = () => {
    if(inputRef.current.value.split('@').length>1 ) {
      inputRef.current.value = ''
      setIsWrong(false)
      setOpen(true)
    } else {
      setIsWrong(true)
    }
  }
  return (
    <div className="mt-[50px]">
      <div className="page-width newsletter">
        <div className="rounded p-[30px] bg-[#E8D5C4]">
          <div className="mb-3">

```

```

    <h2 className="text-center">Розсилка</h2>
    <p className="text-center">
      Підпишись на нашу розсилку і отримай 10% на перше
замовлення!
    </p>
  </div>
  <div className="flex justify-center items-center">
    <div className="flex gap-2 ">
      <input
        ref={inputRef}
        className="border border-divider rounded"
        type="text"
        placeholder="Емейл"
        onChange={(e) => setEmail(e.target.value) }
      />
      {isWrong && (
        <span className={'text-critical'}>Невірний
формат емейл</span>
      )}
      <button className="button"
onClick={onClick}>Підписатися</button>
    </div>
  </div>
</div>

<Modal
  open={open}
  onClose={() => setOpen(false)}
  aria-labelledby="parent-modal-title"
  aria-describedby="parent-modal-description"
>
  <Box sx={{ ...style, width: 450 }}>
    <h2 id="parent-modal-title">Ви успішно підписалися на
розсилку!</h2>
    <p id="parent-modal-description">
      Менеджер вишле вам промокод найблизчим часом на емейл
{email}!
    </p>
  </Box>
</Modal>
</div>
);
}

export default Newsletter;

```

```
export const layout = {
  areaId: "content",
  sortOrder: 50,
};
```

## FeaturedCategories.jsx

```
import React from "react";
import './FeatureCategories.scss'

function FeaturedCategories() {

  const categoryTurismUrl = '/tourism'
  const categoryStangaUrl = '/weight'
  const categoryClothesUrl = '/clothes'
  return (
    <div className="page-width feature-categories">
      <div className="mb-2 mt-3">
        <h2 className="text-center" id={'feature-
categories'}>Популярні Категорії</h2>
      </div>
      <div className="grid grid-cols-1 md:grid-cols-2 gap-2">
        <div className="relative col-span-1 row-span-2 men-cat">
          <a
            className="underline top-[20px] left-[20px] bg-
white px-2"
            href={categoryClothesUrl}>
            
            <span className={'absolute'}>Спортивний одяг</span>
          </a>
        </div>
        <div className="relative col-span-1 row-span-1 women-cat">
          <a
            className="underline top-[20px] left-[20px] bg-white
px-2"
            href={categoryStangaUrl}>
            
            <span className={'absolute'}>Штанги та гантелі</span>
          </a>
        </div>
        <div className="relative col-span-1 row-span-1 kid-cat">
          <a
            className="underline top-[20px] left-[20px] bg-white
px-2"
            href={categoryTurismUrl}>
            
            <span className={'absolute'}>Туризм</span>
          </a>
        </div>
      </div>
    </div>
  );
}
```

```

        </a>
      </div>
    </div>
  </div>
);
}

export default FeaturedCategories;

export const layout = {
  areaId: "content",
  sortOrder: 5,
};

```

### FeaturedProducts.jsx

```

import ProductList from
"@components/frontStore/catalog/product/list/List";
import { _ } from "@evershop/evershop/src/lib/locale/translate";
import PropTypes from "prop-types";
import React from "react";

export default function FeaturedProducts({ collection }) {
  if (!collection) {
    return null;
  }
  return (
    <div className="pt-3">
      <div className="page-width">
        <h2 className="mt-3 mb-3 text-center uppercase tracking-
widest">
          {collection.name}
        </h2>
        <ProductList products={collection.products.items}
countPerRow={3} />
      </div>
    </div>
  );
}

FeaturedProducts.propTypes = {
  collection: PropTypes.shape({
    collectionId: PropTypes.number.isRequired,
    name: PropTypes.string.isRequired,
    products: PropTypes.shape({
      items: PropTypes.arrayOf(

```

```

PropTypes.shape({
  productId: PropTypes.number.isRequired,
  name: PropTypes.string.isRequired,
  price: PropTypes.shape({
    regular: PropTypes.shape({
      value: PropTypes.number.isRequired,
      text: PropTypes.string.isRequired,
    }).isRequired,
    special: PropTypes.shape({
      value: PropTypes.number.isRequired,
      text: PropTypes.string.isRequired,
    }).isRequired,
  }).isRequired,
  image: PropTypes.shape({
    alt: PropTypes.string.isRequired,
    url: PropTypes.string.isRequired,
  }).isRequired,
  url: PropTypes.string.isRequired,
})
).isRequired,
}).isRequired,
}).isRequired,
};

FeaturedProducts.defaultProps = {
  featuredProducts: [],
};

export const layout = {
  areaId: "content",
  sortOrder: 15,
};

export const query = `
query query {
  collection (code: "homepage") {
    collectionId
    name
    products (filters: [{key: "limit", operation: eq, value:
"6"}]) {
      items {
        productId
        name
        price {
          regular {
            value
            text

```

```

        }
        special {
            value
            text
        }
    }
    image {
        alt
        url: listing
    }
    url
}
}
}
}
};

```

## FreeShippingBar.jsx

```

import React from "react";

function FreeShippingBar() {
    return (
        <div className="page-width">
            <div className="grid grid-cols-1 md:grid-cols-3 md:divide-x
border-divider border my-3">
                <div className="p-2 border-divider">
                    <h2>Безкоштовна доставка</h2>
                    <p>Отримай безкоштовну доставку при замовленні від 1200
Грн.</p>
                </div>
                <div className="p-2 border-divider">
                    <h2>Найкращі ціни</h2>
                    <p>Ми пропонуємо найкращі ціни на спорт товари на
ринку</p>
                </div>
                <div className="p-2 border-divider">
                    <h2>Чудовий сервіс</h2>
                    <p>Ми підтримуємо нашій клінтів 24/7.</p>
                </div>
            </div>
        </div>
    );
}

export default FreeShippingBar;

```



```
export const layout = {
  areaId: "content",
  sortOrder: 2,
};
```

## MainBanner.jsx

```
import React from "react";
import "./MainBanner.scss";
const { buildUrl } =
require('@evershop/evershop/src/lib/router/buildUrl');

function MainBanner() {
  return (
    <div className="main-banner-home flex items-center">
      <div className="page-width grid grid-cols-2 md:grid-cols-2
gap-2">
        <div className="text-center md:text-left px-2 ">
          <h2 className="h1 ">Відкрий Новий Сезон Спорту!

          </h2>
          <p>Зустрічай нову колекцію спортивного одягу - стиль,
комфорт та якість для кожного тренування. Поринь у світ спорту з
новими моделями, створеними для перемог.

          Почни свою історію успіху вже сьогодні! </p>
          <p></p>
          <a className="button button-primary" href="#feature-
categories">
            Дивитися товари
          </a>
        </div>
      <div />
    </div>
  </div>
);
}

export default MainBanner;

export const layout = {
  areaId: "content",
  sortOrder: 1,
};
```

## MainBanner.scss

```
.main-banner-home {
  background-image: url("/sport-banner1.png");
  background-size: cover;
  background-repeat: no-repeat;
  background-position: right center;
  height: 490px;
  position: relative;
  z-index: 1;
  &:after {
    position: absolute;
    left: 0;
    top: 0;
    background-color: rgba(0,0,0,0.3);
    width: 100%;
    z-index: -1;
    height: 100%;
    content: "";
  }

  .container {
    z-index: 10;
  }
  .button {
    margin-top: 27px;
  }
  p{

    color: #fff;
  }
  h2.h1 {
    color: #fff;
    letter-spacing: 0;
    font-size: 3.2rem;
    line-height: 4.2rem;
    @media screen and (min-width: 768px) {
      font-size: 5.5rem;
      line-height: 6.5rem;
    }
  }
}

#app {
  .header {
    margin-bottom: 0;
  }
  .button {
```

```
    text-transform: uppercase;
    font-size: 1.2rem;
    font-weight: bold;
    letter-spacing: 0.1rem;
  }
}
```

## FeatureCategories.scss

```
.feature-categories {

  a {
    position: relative;
    width: 100%;
    display: block;
    padding: 0;
    overflow: hidden;

    img {
      transition: all 300ms;
    }
    &:hover {

      img {
        transform: scale(1.1);
      }
    }

    span {
      bottom: 0;
      width: 100%;
      height: 100px;
      background-color: rgba(0,0,0,0.5);
      color: #FFF;
      font-size: 24px;
      max-width: 100%;
      text-align: center;
      padding: 25px 0;
      display: block;
    }
  }
}
```

## Menu.jsx

```
import PropTypes from "prop-types";
import React from "react";

export default function Menu({ menu: { items } }) {
  return (
    <div className="main-menu self-center hidden md:block">
      <ul className="nav flex space-x-275 justify-content-center">
        {items.map((i, index) => (
          // eslint-disable-next-line react/no-array-index-key
          <li className="nav-item" key={index}>
            <a className="nav-link hover:underline" href={i.url}>
              {i.name}
            </a>
          </li>
        ))}
        <li className="nav-item">
          <a className="nav-link hover:underline"
href={"/contact"}>
            Контакты
          </a>
        </li>
      </ul>
    </div>
  );
}

Menu.propTypes = {
  menu: PropTypes.shape({
    items: PropTypes.arrayOf(
      PropTypes.shape({
        name: PropTypes.string.isRequired,
        url: PropTypes.string.isRequired,
      })
    ).isRequired,
  }).isRequired,
};

export const layout = {
  areaId: "header",
  sortOrder: 1,
};

export const query = `
  query {
    menu {
```

```
        items {
            name
            url
        }
    }
} `;
```

## Layout.scss

```
:root {
  --primary: #3a3a3a;
  --secondary: #111213;
  --surface: #111213;
  --onSurface: #111213;
  --interactive: #2c6ecb;
  --critical: #d72c0d;
  --warning: #ffc453;
  --highlight: #5bcdda;
  --success: #008060;
  --decorative: #ffc96b;
  --border: #8c9196;
  --background: #f6f6f7fc;
  --icon: #5c5f62;
  --textSubdued: #6d7175;
  --divider: #e1e3e5;
  --text: #202223;
  --formFieldBorder: #c9cccf;
  --formFieldBorderShadow: #aeb4b9;
}

$font-size-sm: 1.4rem;
$font-size-md: 1.4rem;
$font-size-lg: 1.6rem;

$line-height-sm: 1.25;
$line-height-md: 1.5;
$line-height-lg: 2;

$font-size-base: $font-size-sm;
$h1-font-size: $font-size-base * 2.1;
$h2-font-size: $font-size-base * 1.9;
$h3-font-size: $font-size-base * 1.65;
$h4-font-size: $font-size-base * 1.45;
$h5-font-size: $font-size-base * 1.15;
$h6-font-size: $font-size-base;

html {
  font-size: 62.5%;
```

```
input[type='email'],
input[type='password'],
input[type='tel'],
input[type='text'],
select,
textarea {
  padding: 1rem 1.2rem;
}

.button {
  padding: 1rem 2rem;
  font-weight: 400;
  border-radius: 2px;
}
}
body {
  color: #3a3a3a;
}
p {
  margin-bottom: 1rem;
}
@media screen and (max-width: 767px) {
  $font-size-base: $font-size-sm;
  body {
    font-size: $font-size-base;
    line-height: $line-height-sm;
    .h1,
    h1 {
      font-size: $h1-font-size;
    }
    .h2,
    h2 {
      font-size: $h2-font-size;
    }
    .h3,
    h3 {
      font-size: $h3-font-size;
    }
    .h4,
    h4 {
      font-size: $h4-font-size;
    }
    .h5,
    h5 {
      font-size: $h5-font-size;
    }
  }
}
```

```

    .h6,
    h6 {
        font-size: $h6-font-size;
    }
}

@media (min-width: 768px) and (max-width: 1535px) {
    $font-size-base: $font-size-md;
    $h1-font-size: $font-size-base * 2.1;
    $h2-font-size: $font-size-base * 1.9;
    $h3-font-size: $font-size-base * 1.65;
    $h4-font-size: $font-size-base * 1.45;
    $h5-font-size: $font-size-base * 1.15;
    $h6-font-size: $font-size-base;

    body {
        font-size: $font-size-base;
        line-height: $line-height-md;
        .h1,
        h1 {
            font-size: $h1-font-size;
        }
        .h2,
        h2 {
            font-size: $h2-font-size;
        }
        .h3,
        h3 {
            font-size: $h3-font-size;
        }
        .h4,
        h4 {
            font-size: $h4-font-size;
        }
        .h5,
        h5 {
            font-size: $h5-font-size;
        }
        .h6,
        h6 {
            font-size: $h6-font-size;
        }
    }
}

@media (min-width: 1536px) {

```

```
$font-size-base: $font-size-lg;
$h1-font-size: $font-size-base * 2.1;
$h2-font-size: $font-size-base * 1.9;
$h3-font-size: $font-size-base * 1.65;
$h4-font-size: $font-size-base * 1.45;
$h5-font-size: $font-size-base * 1.15;
$h6-font-size: $font-size-base;

body {
  font-size: $font-size-base;
  line-height: $line-height-lg;
  .h1,
  h1 {
    font-size: $h1-font-size;
  }
  .h2,
  h2 {
    font-size: $h2-font-size;
  }
  .h3,
  h3 {
    font-size: $h3-font-size;
  }
  .h4,
  h4 {
    font-size: $h4-font-size;
  }
  .h5,
  h5 {
    font-size: $h5-font-size;
  }
  .h6,
  h6 {
    font-size: $h6-font-size;
  }
}
h1,
h2,
h3,
h4,
h5,
h6 {
  font-weight: 600;
}
.wrapper .content {
  min-height: calc(100vh - 50px);
}
```



```
}
.header {
  padding: 2.5rem !important;
  border-bottom: 1px solid #ebebeb;
  margin-bottom: 3rem;
  @media only screen and (min-width: 1200px) {
    padding-left: 50px;
    padding-right: 50px;
  }
}

.container {
  margin: 0 auto;
}

.page-width {
  max-width: 1200px;
  margin: 0 auto;
  padding: 0 22px;
  @media screen and (min-width: 1200px) {
    padding: 0 55px;
  }
}

.footer {
  background-color: #f6f6f6;
  margin-top: 9rem;
  padding-top: 1rem;
  padding-block-end: 1rem;
  border-top: 1px solid #ddd;
  .card-icons {
    svg {
      height: 4rem;
    }
  }
  .copyright {
    font-size: 13px;
  }
}

table {
  width: 100%;
  border-collapse: collapse;
  td,
  th {
    border: 1px solid var(--divider);
    padding: 0.5rem;
    text-align: left;
  }
}
```

```
th {
  vertical-align: top;
}
&.listing {
  td,
  th {
    border-left: 0;
    border-right: 0;
    padding: 1rem;
  }
  th {
    border: 0;
    padding-top: 2rem;
    padding-bottom: 2rem;
  }
}
}

button.outline {
  outline: none;
}

html .newsletter {
  input {
    border: #9ca3af 1px solid;
    border-radius: 5px;
  }
}
```