

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

**Автоматизації і інформаційних технологій**  
(факультет)

**Управління проєктами**  
(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Управління проєктом створення інтернет-магазину одягу та мобільного  
додатку до нього»

---

**ЛЕВИЩЕНКО ОЛЕКСАНДР ЯРОСЛАВОВИЧ**

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

**Автоматизації і інформаційних технологій**  
(факультет)  
**Управління проєктами**  
(кафедра)

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри УП  
Бушуєв С.Д.

\_\_\_\_\_

„\_\_\_” \_\_\_\_\_ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Управління проєктом створення інтернет-магазину одягу та мобільного  
додатку до нього»

**ЛЕВІЩЕНКО ОЛЕКСАНДР ЯРОСЛАВОВИЧ**  
(прізвище, ім'я та по батькові студента повністю)

Виконав: студент 4-го курсу, групи ІСТ-УП-20.

Спеціальності: 126 «Інформаційні системи та тех-  
нології».  
(шифр і назва напрямку підготовки, спеціальності)

Освітньо-професійна програма : «Управління проєк-  
тами»

Левіщенко О.Я.  
(прізвище та ініціали)

Керівник: к.т.н, доцент Бойко Є.Г.  
(прізвище та ініціали)

Рецензент \_\_\_\_\_:  
(прізвище та ініціали)

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: Автоматизації і інформаційних технологій .  
Кафедра: Управління проєктами .  
Освітній рівень: «бакалавр за ОП» .  
Спеціальність: 126 «Інформаційні системи та технології» .  
Освітньо-професійна програма: Управління проєктами .

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри УП  
Бушуєв С.Д.

„\_\_\_” \_\_\_\_\_ 2024 року

**З А В Д А Н Н Я  
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ  
НА ЗДОБУТТЯ ОСВІТЬОГО РІВНЯ «БАКАЛАВР»**

Левіщенко Олександр Ярославович

1. Тема роботи: Управління проєктом створення інтернет-магазину одягу та мобільного додатку до нього.

затверджена наказом ректора КНУБА № 357/2 від «15» 02 2024 р.

2. Керівник роботи: Бойко Євгенія Григоріївна, кандидат технічних наук, доцент кафедри управління проєктами.

3. Строк подання студентом роботи до захисту: червень 2024 року .

4. Зміст пояснювальної записки за розділами:

Р.1. Аналіз предметної області та постановка задачі.

Р.2. Розробка інформаційного забезпечення.

Р.3. Проєктні та технічні рішення.

Р.4. Управління проєктом.

5. Інформаційні слайди:

С.1. Титульна сторінка

- С.2. Актуальність теми дослідження
- С.3. Цілі та задачі
- С.4. Вибір методології
- С.5. Управління проєктом
- С.6. Етап розробки
- С.7. База даних
- С.8. Огляд фінального продукту
- С.9. Висновки

#### 6. Календарний план виконання АВР

Види робіт та їх зміст	Дата виконання
Р.1. Аналіз предметної області та постановка задачі	Травень 2024 р.
Р.2. Розробка інформаційного забезпечення	Травень 2024 р.
Р.3. Проєктні та технічні рішення	Червень 2024 р.
Р.4. Управління проєктом	Травень 2024 р.
Остаточне оформлення роботи	Червень 2024 р.
Попередній захист роботи на кафедрі	Червень 2024 р.

#### 7. Консультанти розділів АВР

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Ергономіка інформаційних технологій	д.т.н. проф. Терентьев О.О.		
Прийом програмного продукту	к.т.н., доц. Бойко Є.Г.		

8. Дата видачі завдання: 15 лютого 2024 року

Керівник

\_\_\_\_\_ (підпис)

Євгенія БОЙКО

\_\_\_\_\_ (ім'я та прізвище)

Бакалавр

\_\_\_\_\_ (підпис)

Олександр ЛЕВІЩЕНКО

\_\_\_\_\_ (ім'я та прізвище)

## АНОТАЦІЯ

Левіщенко О.Я. Управління проектом розробки Інтернет-магазину одягу та мобільного застосунку до нього.

Атестаційна випускна робота бакалавра за спеціальністю: 126 «Інформаційні системи і технології», освітня програма «Управління проектами». – Київський національний університет будівництва та архітектури. – Київ, 2024.

Робота присвячена створенню Інтернет-магазину одягу та взуття. Розроблена система дозволяє автоматизувати процеси продажу, обліку товарів, взаємодії з клієнтами та забезпечити безперебійний доступ до онлайн-магазину з мобільних пристроїв. Це забезпечує зручність для користувачів та оптимізує роботу адміністраторів магазину.

За результатами виконання атестаційної роботи було детально проаналізовано великий набір вхідних та вихідних даних, що дозволило отримати повну картину необхідних інформаційних потоків. Було детально описано організацію інформаційної бази. Розроблено логічну та фізичну моделі даних, які відображають структуру та взаємозв'язки між різними елементами системи, що дозволяє ефективно зберігати та обробляти інформацію.

Для проектування системи застосовано сучасні підходи з управління проектами, які включають планування ресурсів (фізичних та людських) та контроль за виконанням завдань. Ці підходи дозволили ретельно спланувати етапи розробки, визначити критичні точки та забезпечити своєчасне впровадження системи. Важливою складовою було також оптимізація процесів та ресурсів, що сприяло досягненню головної мети – створення ефективної та надійної системи.

Ключові слова: Інтернет-магазин, мобільний застосунок, управління проектами, інформаційні технології, інформаційні системи.

## SUMMARY

Levishchenko O.Y. Project management of the development of an online clothing store and a mobile application for it.

Bachelor's thesis for a bachelor's degree in specialty: 126 “Information Systems and Technologies”, educational program “Project Management.” - Kyiv National University of Construction and Architecture - Kyiv, 2024.

The work is dedicated to the creation of an online clothing and footwear store. The developed system allows automating sales processes, goods accounting, customer interaction, and providing uninterrupted access to the online store from mobile devices. This provides convenience for users and optimizes the work of store administrators.

Based on the results of the certification work, a large set of input and output data was analyzed in detail, which allowed us to get a complete picture of the necessary information flows. The organization of the information base was described in detail. Logical and physical data models were developed that represent the structure and relationships between different elements of the system, which allows for efficient storage and processing of information.

Modern project management approaches were used to design the system, including resource planning (physical and human) and task control. These approaches allowed us to carefully plan the development stages, identify critical points, and ensure timely implementation of the system. An important component was also the optimization of processes and resources, which contributed to the achievement of the main goal - the creation of an efficient and reliable system.

Keywords: online store, mobile application, project management, information technology, information systems.

## ЗМІСТ

<b>ВСТУП</b> .....	9
<b>Розділ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ</b> .....	11
<b>1.1 Характеристика діючої системи управління та проектування</b> .....	11
<b>1.2 Аналіз області застосування об'єкта проектування</b> .....	13
1.2.1 Досягнуті можливості на даний момент часу .....	13
1.2.2 Ідентифікація проблемних моментів .....	16
1.2.3 Визначення мети дипломного проектування .....	17
<b>1.3 Основні цілі автоматизованого проектування</b> .....	18
<b>1.4 Дерево цілей проектування</b> .....	20
<b>1.5 Аналіз поставленого завдання</b> .....	20
1.5.1 Огляд відомих досліджень з теми дипломної роботи .....	21
1.5.2 Перші згадки матеріалу з джерел .....	22
1.5.3 Висновки щодо можливості використання відомих рішень або необхідності проєктування оригінальних рішень .....	22
<b>1.6 Класифікація, основні характеристики об'єкта дослідження та інші відомості</b> .....	23
<b>1.7 Формулювання постановки задачі</b> .....	24
<b>1.8 Висновки до розділу</b> .....	25
<b>Розділ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ</b> .....	27
<b>2.1 Методологія розробки програмного рішення</b> .....	27
2.1.1 Загальні відомості .....	27
2.1.2 Існуючі рішення .....	27
2.1.3 Порівняння методологій .....	31
2.1.4 Вибір методології .....	36
<b>2.2 Метод розробки</b> .....	37
2.2.1 Загальні відомості .....	37
2.2.2 Існуючі рішення .....	37
2.2.3 Порівняння систем .....	38
<b>2.3 Інструменти та технології для проєктування архітектури</b> .....	40
<b>2.4 Висновки до розділу</b> .....	42
<b>Розділ 3 Проєктні та технічні рішення</b> .....	43
<b>3.1 Загальні положення</b> .....	43

<b>3.2 Реалізація програмного забезпечення</b> .....	43
<b>3.2.1 Розробка прототипу інтерфейсу</b> .....	43
<b>3.2.2 Фронтенд</b> .....	45
<b>3.2.3 Бекенд</b> .....	45
<b>3.3 База даних</b> .....	48
<b>3.4 Функціональні можливості</b> .....	50
<b>3.4.1 Сценарій користувача</b> .....	50
<b>3.4.2 Сторінка акаунту</b> .....	56
<b>3.4.3 Пошук товару</b> .....	58
<b>3.5 Висновки до розділу</b> .....	61
<b>Розділ 4 УПРАВЛІННЯ ПРОЄКТОМ</b> .....	62
<b>4.1 Загальний огляд проєкту</b> .....	62
<b>4.2 Огляд задач проєкту</b> .....	63
<b>4.3 Огляд ресурсів проєкту</b> .....	68
<b>4.4 Аналіз зацікавлених сторін</b> .....	72
<b>4.5 SWOT-аналіз</b> .....	73
<b>4.6 Цілі компанії</b> .....	74
<b>4.7 Висновки до розділу</b> .....	76
<b>ЗАГАЛЬНІ ВИСНОВКИ</b> .....	77
<b>ДЖЕРЕЛА</b> .....	80
<b>ДОДАТКИ</b> .....	81

## ВСТУП

В ХХІ столітті, де інформаційні технології стають невід'ємною частиною нашого повсякденного життя, електронна комерція набуває все більшого значення. Особливо швидкими темпами розвивається ринок інтернет-магазинів, що надає можливість купувати товари без виходу з дому, з будь-якої точки світу, у будь-який час доби. Однією з найпопулярніших категорій товарів в інтернет-магазинах є одяг, що пояснюється як різноманітністю асортименту, так і зручністю процесу покупки для споживачів.

Дана дипломна робота присвячена дослідженню теми “Управління проєктом розробки інтернет-магазину одягу”.

**Актуальність теми дослідження** обумовлена зростаючим попитом на електронні магазини, які надають можливість споживачам легко і швидко знаходити необхідні товари, порівнювати ціни та характеристики, а також здійснювати покупки з максимальною зручністю. З іншого боку, підприємцям це дає змогу знижувати витрати на утримання фізичних магазинів, розширювати ринки збуту та покращувати взаємодію з клієнтами.

**Метою роботи** є розробка інтернет-магазину одягу, що забезпечує зручний та інтуїтивно зрозумілий інтерфейс для користувачів, а також ефективне управління товарним асортиментом і замовленнями.

### **Задачі дослідження:**

1. Проаналізувати ринок інтернет-магазинів
2. Розробити проєкт
3. Розробити веб-сайт

У процесі дослідження будуть розглянуті основні етапи створення інтернет-магазину, починаючи від аналізу ринку та вимог користувачів, до технічної реалізації та впровадження функціональних можливостей. Також, в цю роботу входить управління проєктом.

При написанні роботи були використані наступні методи: огляд літератури

(в процесі дослідження було вивчено понад 16 джерел літератури, в тому числі навчальна література з означеної теми), використання різних методів для розробки проєкту (дерево цілей, SWOT-аналіз, календарне планування, організаційна структура проєкту, ресурсний план проєкту, розрахунку кошторису витрат за проєктом, розрахунок економічної ефективності проєкту та ін.), використання прикладних програм (Microsoft Word, Microsoft Access, Microsoft Project).

## РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Характеристика діючої системи управління та проектування

Для кращого розуміння області інтернет-магазинів одягу, важливо оцінити характеристики та функціонал існуючих систем. На сьогоднішній день на ринку існує CRM-система, яка необхідна при створенні інтернет магазину.

Спочатку ми розберемось, що таке CRM-система, а потім розберемо найпопулярніші в Україні системи.

CRM — це програма або онлайн-сервіс для управління бізнес-процесами в інтернет-магазині та їх автоматизації. CRM потрібні, щоб керувати всіма складовими взаємодії з клієнтом і роботи інтернет-магазину.

Основні функції CRM-систем для е-commerce та інших сфер бізнесу:

- Збір контактів і управління ними
- Моніторинг взаємодій
- Електронний документообіг
- Інтеграція з email
- Облік складських залишків
- Інтеграція з сервісами колл-трекінгу
- Аналітика та візуалізація продажів

Частина цих функцій є і у безкоштовних CRM. Але на певному етапі розвитку бізнесу базових можливостей уже буде недостатньо.

Тепер давайте розглянемо найпопулярніші в Україні CRM-системи, які можуть підійти до нашого інтернет-магазину.

1. Bitrix24 - це одна з найпопулярніших CRM-систем в Україні. Bitrix24 має широкий спектр функцій, включаючи управління клієнтами, комунікацію, завдання та проекти. Вона може бути легко інтегрована з інтернет-магазинами.
2. RetailCRM - ця CRM спеціалізується на роздрібній торгівлі та інтернет-магазинах. Вона надає інструменти для автоматизації продажів, управління клієнтами та аналітики.

3. AmoCRM - ця CRM також має функціонал, спрямований на розвиток інтернет-бізнесу. Вона дозволяє відстежувати контакти, створювати ліди та ефективно керувати продажами.
4. Salesforce - це одна з провідних CRM-платформ у світі, яка також користується популярністю в Україні. Salesforce пропонує рішення для автоматизації продажів, маркетингу та обслуговування клієнтів для різних типів бізнесу, включаючи інтернет-магазини.
5. HubSpot - ця CRM-платформа знаменита своїми інструментами інбаунд-маркетингу та автоматизації маркетингових кампаній. Вона також може бути корисною для інтернет-магазинів для управління клієнтськими відносинами.

Всі наведені вище системи можуть надавати широкий перелік можливостей, такі як зберігання інформації про клієнтів, сегментація клієнтів за різними критеріями для персоналізованого підходу, створення та відстеження угод і лідів, автоматизація процесів продажів, створення і відстеження маркетингових кампаній, автоматизовані повідомлення, збір та аналіз даних про продажі, створення аналітичних звітів для прийняття управлінських рішень, можливість інтеграції з іншими інструментами та засоби для внутрішньої комунікації в команді та співпраці з клієнтами.

На сьогоднішній день, з розвитком такої сфери як ШІ, стала можливість впроваджувати подібні системи до інтернет-магазинів. Дані системи використовуються для аналізу дій відвідувачів сайту. Це допомагає власникам інтернет-магазинів зрозуміти, як покращити свої продажі та привернути увагу нових клієнтів.

Подібні системи можуть бути використані в Інтернет-магазинах для поліпшення користувацького досвіду, оптимізації управління запасами, персоналізації рекомендацій і багато іншого. Нижче буде наведений перелік можливостей даних систем:

- Системи рекомендацій
- Чат-боти для обслуговування клієнтів
- Аналіз відгуків клієнтів

- Прогнозування попиту
- Персоналізоване маркетингове впливання
- Візуальне пошукове оброблення
- Автоматизовані системи оформлення замовлень

Враховуючи зазначені характеристики, можна зробити висновок про значну роль та потенціал Інтернет-магазинів в житті людей.

## **1.2 Аналіз області застосування об'єкта проектування**

Зважаючи на актуальність впровадження сучасних технологій у сфері Інтернет-магазинів в Україні, необхідно провести аналіз області застосування об'єкта проектування.

Цей аналіз допоможе розкрити контекст та потреби, які стоять перед реалізацією проекту "UStock" у сфері Інтернет-магазинів одягу та взуття. Детальне вивчення цих аспектів дозволить не лише визначити напрямки подальшого розвитку, а й забезпечить максимальну відповідність проекту реальним потребам і можливостям цільової аудиторії.

### **1.2.1 Досягнуті можливості на даний момент часу**

#### **Функціонал.**

Користувачі мають можливість зареєструватися в системі, створивши обліковий запис, або увійти в систему, якщо вони вже зареєстровані, для доступу до свого особистого кабінету. Вони можуть вільно переглядати широкий асортимент товарів, які представлені в інтернет-магазині, а також здійснювати пошук за різними параметрами, такими як розмір, колір, стиль та інше. Обрані товари можна додавати в кошик для подальшого оформлення замовлення. Процес оформлення замовлення включає вибір необхідних товарів, введення інформації для доставки та вибір методу оплати.

У особистому кабінеті користувачі можуть переглядати історію своїх замовлень, а також змінювати особисті дані, такі як адреса доставки, контактні дані тощо. Адміністратори мають доступ до панелі адміністрування, де вони можуть переглядати нові замовлення, оновлювати їх статус та здійснювати інші дії щодо управ-

ління замовленнями. Система також надає можливість надсилання повідомлень користувачам про статус їх замовлень та інші важливі події, наприклад, про спеціальні пропозиції або акції.

Крім того, адміністратори можуть керувати товарами, включаючи їх додавання, видалення та редагування інформації про товари, таку як опис, ціна, фотографії тощо. Вони також мають можливість збирати та аналізувати різноманітні дані, такі як дані про продажі, відвідуваність сайту, поведінку користувачів тощо, для здійснення аналізу та прийняття відповідних рішень щодо оптимізації роботи інтернет-магазину та поліпшення користувацького досвіду.

Система надає можливість адміністраторам виконувати рутинні задачі, такі як управління запасами, виявлення популярних товарів та здійснення акційних пропозицій для залучення клієнтів. Покупці отримують повну інформацію про товари, включаючи опис, характеристики, відгуки попередніх покупців та фотографії товарів. Система також може надавати рекомендації покупцям на основі їхньої попередньої активності та вподобань.

Щоб забезпечити безпеку, система також має можливості зберігання та обробки особистих даних користувачів відповідно до вимог законодавства про захист персональних даних. Це включає в себе шифрування даних, контроль доступу та заходи безпеки для запобігання несанкціонованому доступу до особистої інформації користувачів.

Загалом, інтернет-магазин забезпечує зручну та безпечну платформу для покупців, де вони можуть здійснювати покупки з будь-якого місця та в будь-який час, а адміністратори мають інструменти для ефективного управління бізнесом і забезпечення задоволення потреб клієнтів.

### **Інтеграція з технологіями.**

Розробка Інтернет-магазину одягу та мобільного застосунку може бути збагачена різноманітними інтеграціями з технологіями. Перш за все, важливо підключити платіжні системи які доступні в Україні, такі як Mastercard, PayPal або Visa, для зручної та безпечної оплати замовлень. Також можна додати спосіб оплати через електронні гаманці, найпопулярніші в Україні, це ApplePay та GooglePay. Далі,

інтеграція з системами управління складом, такими як ERP-системи або SAP, допоможе автоматизувати процеси управління запасами та замовленнями.

Також варто врахувати можливість підключення аналітичних інструментів, які допоможуть відстежувати та аналізувати поведінку користувачів, інтеграцію з соціальними медіа для розширення аудиторії та залучення нових клієнтів через платформи, такі як Facebook або Instagram.

Для ефективного управління взаємодією з клієнтами, важливо розглянути інтеграцію з системами CRM, такими як Salesforce або HubSpot.

Крім того, маркетингові інструменти, такі як Mailchimp або Klaviyo, можуть допомогти в проведенні рекламних кампаній та підвищенні маркетингового потенціалу.

Інтеграція з системами управління контентом, такими як WordPress або Drupal, дозволить ефективно керувати контентом вашого магазину. Також в планах даного проєкту є розробка мобільного застосунку, який буде мати такий самий функціонал, як сайт. Єдина відмінність від веб-сайту те, що користувачеві, регулярно, будуть надходити промокоди на знижку або додаткові бонуси при покупці. Важливо підключити мобільні платформи iOS та Android для оптимізації його під кожен з цих платформ.

### **Аналітика та штучний інтелект**

В рамках даного проєкту, може бути ряд можливостей у сфері аналітики та штучного інтелекту. Використання алгоритмів машинного навчання дозволяє проводити аналіз поведінки користувачів для персоналізації рекомендацій та пропозицій, а також прогнозування попиту на товари для оптимізації запасів та планування виробництва. Автоматизована класифікація товарів спрощує процес інвентаризації, а використання штучного інтелекту у процесі оброблення замовлень допомагає зменшити час обробки та уникнути помилок. Додатково, застосування чат-ботів із штучним інтелектом для надання підтримки користувачам у будь-який час доби, а також персоналізованої реклами на основі аналізу даних, забезпечує покращений користувацький досвід та ефективність маркетингових зусиль.

### **Електронна комунікація з клієнтами**

Управління проектом розробки Інтернет-магазину одягу та мобільного застосунку до нього передбачає ряд можливостей для електронної комунікації з клієнтами. Це включає в себе відправлення автоматизованих повідомлень клієнтам через електронну пошту про підтвердження замовлень, статус доставки тощо. Використання чат-ботів на сайті або в мобільному додатку для надання відповідей на запитання клієнтів у реальному часі. Взаємодію з клієнтами через соціальні медіа для відповіді на їх запитання та інформування про нові товари або акції. Розміщення форм зворотного зв'язку на сайті та в мобільному додатку для збору відгуків та пропозицій від клієнтів. Створення особистого кабінету для клієнтів, де вони можуть переглядати свої замовлення, редагувати особисту інформацію, зберігати список бажань тощо. Ці можливості сприяють поліпшенню спілкування з клієнтами та забезпечують їм зручний сервіс, що може позитивно вплинути на їх задоволеність і лояльність до бренду.

### **1.2.2 Ідентифікація проблемних моментів**

Розробка інтернет-магазину одягу та мобільного додатку може зустрічати різноманітні технічні та стратегічні виклики.

Ринок онлайн-продажу одягу насичений численними магазинами, тому вирізнитися та привернути увагу клієнтів - завдання не просте. Важливо провести дослідження ринку, щоб з'ясувати, які унікальні особливості чи послуги можуть зробити ваш інтернет-магазин привабливішим для клієнтів. Також, аналіз конкурентів допоможе визначити їхні переваги та недоліки, що стане основою для формування стратегії конкурентоспроможності проекту.

Розробка інтернет-магазину вимагає спеціалізованих технічних знань. Необхідно забезпечити належну інтеграцію з платіжними системами, системами управління запасами та іншими сервісами. Тестування також є критичним етапом, щоб переконатися у працездатності та безпеці програмного забезпечення перед випуском на ринок.

Важливо забезпечити, щоб мобільний додаток працював на різних версіях операційних систем (Android, iOS) та різних пристроях з різною роздільною здатністю екрану. Використання адаптивного дизайну допоможе забезпечити коректне відображення на різних пристроях та платформах.

Створення зручного та привабливого інтерфейсу користувача є ключовим завданням для успішного інтернет-магазину та мобільного додатку. Важливо врахувати простоту навігації, швидкість завантаження сторінок, доступність інформації та інші аспекти користувацького досвіду.

Забезпечення захисту особистої інформації користувачів та даних про платежі є критичним аспектом для будь-якого інтернет-магазину. Використання шифрування та надійних систем платежів є необхідним для забезпечення безпеки та довіри клієнтів.

Не менш важливим є належна організація логістики та доставки. Швидка та надійна доставка товарів до клієнтів - це великий плюс для вашого бізнесу. Необхідно буде розглянути можливості співпраці з доставчиками та службами доставки для оптимізації процесу.

Навіть, якщо були прийняті достатні міри та вірогідність виникнення проблем, що були наведенні вище зведені до мінімуму, без вдалої маркетингової стратегії Інтернет-магазин не знайде свою аудиторію. Це ключ до привертання уваги потенційних клієнтів та розбудови бази постійних покупців. Необхідно використовувати онлайн-маркетинг, соціальні медіа, рекламу та інші канали для просування інтернет-магазину.

Розробка інтернет-магазину одягу та мобільного додатку - це складний та багатогранний процес, що вимагає уваги до численних аспектів. Однак, ретельне розглядання та врахування проблемних моментів, може допомогти забезпечити успішне впровадження та подальший успіх проєкту. Ретельне планування, вивчення ринку та користувацьких потреб, а також вдосконалення процесів управління, дозволять максимізувати ефективність та досягти поставлених цілей.

### **1.2.3 Визначення мети дипломного проектування**

Мета данної дипломної роботи, це розробка Інтернет-магазину одягу «Ustock» та мобільного застосунку до нього.

Ключові завдання проекту включають:

1. Розробка веб-сайту, який включає в себе функціонал для пошуку та перегляду товару, замовлення товару, що продає магазин.
2. Забезпечення безпеки та конфіденційності даних користувачів, використовуючи сучасні технології шифрування та захисту даних.
3. Інтеграція з іншими системами, такими як мобільний додаток, для залучення нової аудиторії.
4. Забезпечення ефективної електронної комунікації з клієнтами, що включає сповіщення про замовлення, доставку та оплату.

Метою проекту є створення магазину «Ustock».

### **1.3 Основні цілі автоматизованого проектування**

Кожного дня інформаційно-комунікативні системи стають більш розвиненими. Це відкриває безліч можливостей для автоматизації, перше що спадає на думку – це е-commerce.

Основна мета автоматизованого проектування полягає в тому, щоб спростити і прискорити процес розробки продукту чи системи за допомогою використання комп'ютерних програм і інструментів. Це дозволяє зменшити витрати часу та людських зусиль, пов'язаних з проектуванням, а також підвищити точність і якість результатів.

Для автоматизованого проектування інтернет-магазину одягу можна використати різноманітні методи та інструменти. Ось деякі з них:

Використання баз даних: В даному проєкті плануються використовуватись дві бази даних – одна для клієнтів, інша для товарів які продає магазин. В першій базі даних будуть дані клієнтів, а саме ім'я (ПІБ), номер телефона, електронна пошта, пароль та домашня адреса. В іншій базі даних будуть дані про товар який продає магазин. Ця база даних буде включати в себе назву товару, код товару, назва

виробника та розміри. Завдяки цій базі даних користувач зможе отримати інформацію про товар та його наявність. Ці бази даних будуть сумісні не тільки з сайтом, а ще з мобільним додатком.

Використання технологій штучного інтелекту: В проєкті буде використовуватись система ШІ, яка зможе покращити досвід користувача, пропонуючи товар, який опирається на його інтереси, які беруться в свою чергу через діяльність клієнта на веб-сайті.

Методи аналізу та проектування систем: Методи аналізу та проектування допомагають зрозуміти потреби користувачів, визначити функціональні вимоги та ефективно спланувати та реалізувати інтернет-магазин.

Використання даних методів сильно допоможе створити інтернет-магазин, який не буде поступатися якістю та зручністю конкурентам.

## 1.4 Дерево цілей проектування

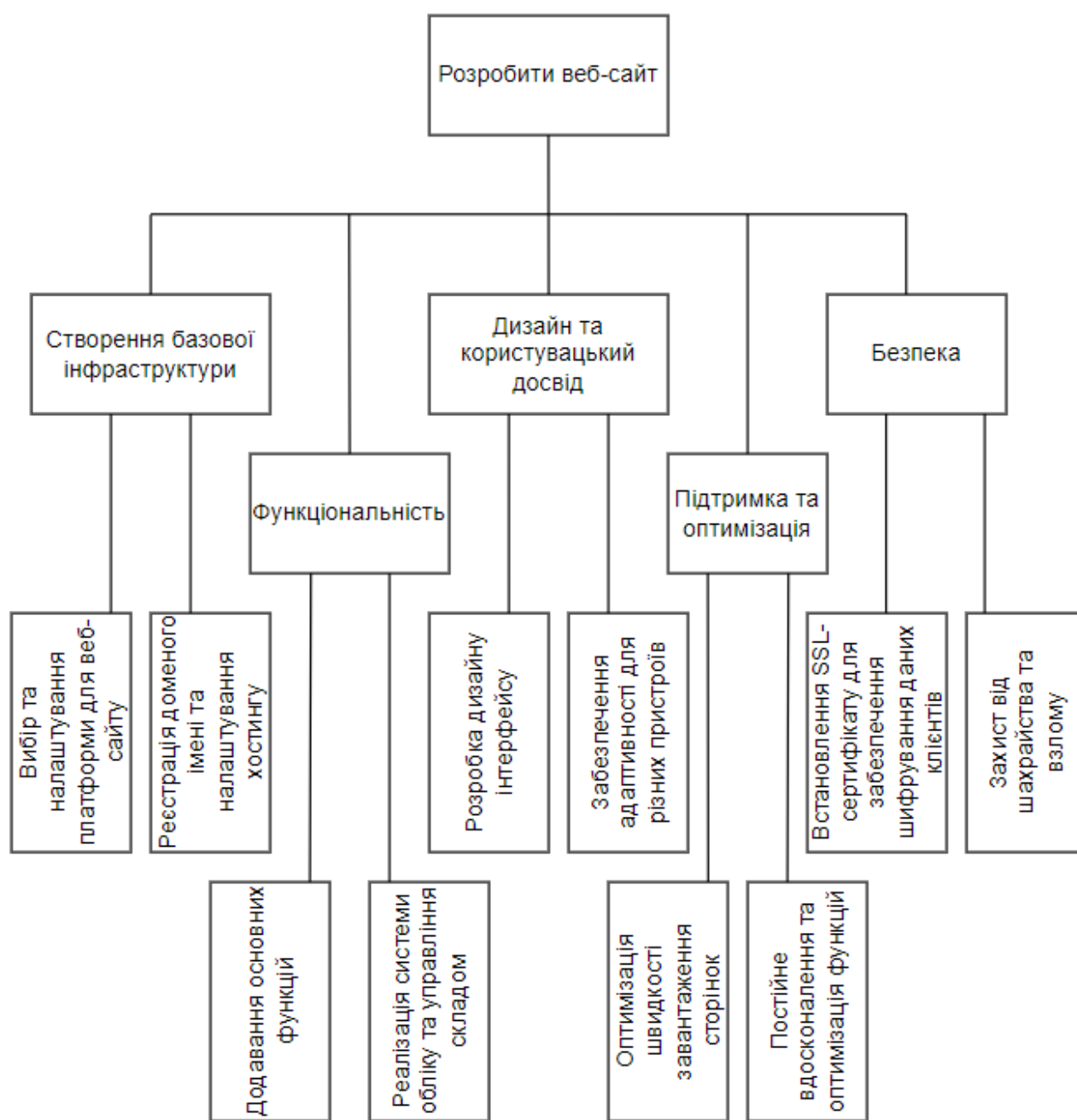


Рисунок 1.1 Дерево цілей

## 1.5 Аналіз поставленого завдання

Для цього розділу було ретельно проаналізовано всі аспекти завдання проекту та визначено його основні характеристики, складнощі та можливі ризики. Ось загальний опис аналізу:

1. Характеристика завдання: Завдання полягає в розробці Інтернет-магазину одягу та взуття «Ustock», а саме веб-сайт та мобільний додаток до нього.

2. Аналіз потреб користувачів: Завдяки проведеному аналізу потенційних клієнтів, можна визначити чого саме вони очікують від проєкту, а саме: інтуїтивно-зрозумілий інтерфейс, різноманітний товар, який може угодити найвимогливішому клієнту та зручна система оплати замовлення.

3. Оцінка технічних вимог: Проаналізувавши технічні вимоги до проєкту, можна зробити висновок, що перш за все треба забезпечити безпеку даних користувачів. Також необхідно оптимізувати всі процеси веб-сайту, щоб мінімізувати «Фрізи»

4. Виявлення складнощів та ризиків: В цій сфері головна потенційна проблема це конкуренція, також треба звернути увагу спроби взлому бази даних користувачів зі злим наміром.

5. Аналіз необхідності додаткових досліджень: Для успішної реалізації проєкту може знадобитися додаткове дослідження в галузі захисту даних, аналізу вимог користувачів та технічних можливостей системи.

### **1.5.1 Огляд відомих досліджень з теми дипломної роботи**

Основна мета цього аналізу - визначення стану галузі на даний момент часу та виявлення проблем, які можуть бути вирішені за допомогою дипломного проєкту.

#### **Огляд літератури та досліджень**

Проведений огляд наукових досліджень та літератури показав показав низький інтерес до розробки Інтернет-магазинів. Проте Інтернет наповнений безліччю курсами та відеороликами «Як створити Інтернет-магазин та запустити його». Зазвичай в цих джерелах інформації, Інтернет-магазин створюється за допомогою конструкторів, такі як: Wordpress, Shop-express і так далі.

#### **Аналіз аналогічних проєктів**

Додатково було проведено дослідження аналогічних проєктів, які вже існують на ринку та в майбутньому будуть створювати конкуренцію:

- PRM.com – Український Інтернет-магазин одягу та взуття, що належить Answer.com. Цей веб-сайт включає в себе інтуїтивно-зрозумілий та стильний

дизайн, функціонал, який не гірше аналогічних магазинів, регулярні розпродажі та широкий асортимент товару.

- [yesoriginal.com.ua](http://yesoriginal.com.ua) – Український Інтернет-магазин одягу та взуття. Його головна перевага це мінімалістичний дизайн, який не має в собі нічого зайвого. Хочеться, також звернути увагу на те, що на основній сторінці веб-сайту є посилання на Youtube-канал, де є багато відеороликів, що просвітляють глядача в культурі одягу.

### **Виявлення прогалин та перспектив**

Аналіз показав, що існуючі Інтернет-магазини мають встановлений стандарт, проте «скільки людей, стільки і думок», і неможливо зробити систему веб-сайт, який зможе задовольнити потреби всіх користувачів. Ми визначили головні «діри» проекту, вирішення яких зможе мінімізувати число незадоволених користувачів:

- Погана оптимізація
- Неясність інструкцій
- Недостатній рівень безпеки даних користувачів

### **1.5.2 Перші згадки матеріалу з джерел**

У цьому розділі вказані початкові посилання та згадки на матеріали із різних джерел, які були використані під час огляду літератури та проведення досліджень для цієї дипломної роботи. Це допомагає читачеві ознайомитися з ключовими джерелами інформації, що стали основою для цього розділу.

- "Effective Project Management: Traditional, Agile, Extreme" авторства Robert K. Wysocki.
- PMBOK (Project Management Body of Knowledge) від PMI (Project Management Institute).
- ISO 21500:2012 "Guidance on Project Management".
- Agile Manifesto та документація по методологіям Scrum, Kanban, та Lean.

### **1.5.3 Висновки щодо можливості використання відомих рішень або необхідності проектування оригінальних рішень**

**Використання відомих рішень:**

В процесі аналізу були замічені певні Інтернет-магазини, а саме PRM.com та yesoriginal.com.ua. Вони відповідають всім стандартам, які характерні для Інтернет-магазинів одягу. Дивлячись на них, можна взяти особливості які можна використати для розробки.

### **Необхідність проектування оригінальних рішень:**

Щоб відрізнятись від конкурентів та не бути копією інших Інтернет-магазинів, необхідно розробити ряд оригінальних рішень, що буде виділяти веб-сайт на фоні конкурентів.

## **1.6 Класифікація, основні характеристики об'єкта дослідження та інші відомості**

У цьому розділі буде проводитись аналіз об'єкта дослідження даного дипломного проекту, ми розглянемо класифікацію системи, її основні характеристики та інші відомості.

Ми розглянемо різні аспекти системи, включаючи її функціональні можливості, архітектуру, інтеграцію з іншими технологіями, аналітику та штучний інтелект, а також електронну комунікацію з клієнтами. Таким чином, ми зможемо сформулювати повний огляд системи та підготувати фундамент для подальшого розвитку та втілення нашого дипломного проекту.

Даний проєкт передбачає створення веб-сайту з функціональними можливостями, яка включає в себе систему реєстрації та аутентифікації користувачів, перегляд каталогу товарів з можливістю фільтрації та сортування, додавання товарів до кошика та оформлення замовлення, різноманітні способи оплати та доставки, особистий кабінет користувача з історією замовлень та можливістю зміни особистої інформації, а також адміністративний розділ для управління товарами, замовленнями та клієнтами. Цей проєкт буде базуватися на клієнт-серверній архітектурі з веб-сервером для інтернет-магазину та мобільним додатком для клієнтів, а також використовуватиме мікросервісну архітектуру для розділення функціональності на незалежні модулі та бази даних для збереження інформації про користувачів, товари, замовлення тощо. Інтеграція з іншими технологіями передбачатиме зв'язок з платіжними шлюзами для обробки платежів, поштовими службами для відправки

сповіщень про замовлення та доставку, а також з системами управління запасами для автоматичного оновлення наявності товарів. З метою аналізу та оптимізації діяльності інтернет-магазину передбачається використання аналітики та штучного інтелекту, зокрема збір та аналіз даних про покупки користувачів для рекомендаційних систем, аналіз покупкової поведінки для прогнозування популярності товарів та планування запасів, використання алгоритмів машинного навчання для виявлення шахраїв та зменшення кількості відмінених замовлень. Електронна комунікація з клієнтами буде забезпечуватися за допомогою відправлення сповіщень користувачам про стан їх замовлень, акції та нові надходження, системи тікетів для спілкування з покупцями та вирішення їх запитів, а також за допомогою відгуків та оцінок товарів від клієнтів для створення довіри та покращення продуктів.

### **1.7 Формулювання постановки задачі**

Основна мета:

Розробити і реалізувати Інтернет-магазин по продажу одягу та взуття для отримання подальшого прибутку. Також в планах проекту є розробка мобільного додатку, але його розробка буде залежати від успішності самого веб-сайту. Тобто, якщо Інтернет-магазин буде користуватись достатньою популярністю, тоді в розробці мобільного додатку, з ціллю залучення нової аудиторії, не буде необхідності.

Основні завдання:

1. Аналіз сучасного стану та проблем предметної області:
  - Вивчення і аналіз існуючих Інтернет-магазинів, які діють на території України.
  - Виявлення проблемних моментів та недоліків існуючих Інтернет-магазинів, які потребують вирішення.
2. Проектування та розробка веб-сайту:
  - Вивчення стандартів, які притамані подібним веб-сайтам.
  - Розробка фронтендної та бекендної частини.
  - Розробка баз даних
  - Реалізація функцій, що притамані для торгівельних веб-сайтів.
3. Тестування:

- Проведення тестування.
  - виправлення помилок, які були замічені під час тестування.
4. Реліз веб-сайту та підтримка:
- Запуск веб-сайту на ринок Інтернет-магазинів одягу та взуття
  - Надання технічної підтримки та навчання персоналу
5. Аналіз діяльності
- Проведення аналізу успішності Інтернет-магазину, дивлячись на кількість замовлень на веб-сайті.
  - Прийняття рішень про розробку мобільного додатку.
6. Розробка мобільного додатку
- Вивчення стандартів, які притамані подібним мобільним додаткам.
  - Розробка фронтендної та бекендної частини.
  - Розробка баз даних.
  - Реалізація функцій, що притамані для додатків для Інтернет-магазинів.
7. Тестування:
- Проведення тестування.
  - виправлення помилок, які були замічені під час тестування.
8. Реліз додатку та підтримка:
- Реліз мобільного додатку на ринку (Google Play та App Store)
  - Надання технічної підтримки

### **1.8 Висновки до розділу**

- Було розглянуто, що таке CRM-система та розглянули найпопулярніші системи, які можуть підійти до нашого сайту.
- Було розглянуто можливості, які доступні на даний момент часу, а саме: функціонал веб-сайту, інтеграції з технологіями, аналітика та штучний інтелект та електронна комунікація з клієнтами
- Визначено проблемні моменти, які можуть негативно вплинути на успішність веб-сайту.
- Визначено головне завдання проєкту та вказали ключові завдання, які необхідно виконати.

- Визначено основну мету автоматизованого проєктування та розглянули різні методи та інструменти, які можуть позитивно вплинути на кінцевий продукт.
- Побудовано дерево цілей проєктування
- Проведено ряд аналізів, а саме: характеристику завдання, аналіз потреб користувачів, оцінка технічних вимог, виявлення складнощів та ризиків, аналіз необхідності додаткових досліджень. Також ми проглянули дослідження, а саме: літературу, дослідження та аналогічні проєкти.

## **РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ**

### **2.1 Методологія розробки програмного рішення**

#### **2.1.1 Загальні відомості**

Методологія розробки програмного забезпечення – сукупність методів, які використовуються на різних стадіях життєвого циклу розробки програмного забезпечення, що мають спільний філософський підхід та дозволяють забезпечити найкращу ефективність процесів розробки. Кожна методологія характеризується: філософським підходом або основними принципами. Ці принципи, від яких залежить ефективність всієї методології, зазвичай можна коротко сформулювати і легко пояснити узгодженою множиною моделей та методів, які реалізують визначену методологію.

Можна простежити три шляхи виникнення методологій. По-перше, вони можуть бути результатом практичного досвіду. По-друге, методології можуть походити від однієї з чотирьох моделей алгоритму: абстрактна машина Тюрінга (імперативне програмування), рекурсивні функції Гільберта і Аккермана (структурне програмування), лямбда-числення Черча (функціональне програмування), нормальні алгоритми Маркова (логічне програмування). По-третє, методології можна пояснити через візуалізацію структури та моделей відповідно до мов моделювання та програмування. Складовими частинами можуть бути структура даних, структура управління і логіка. Кожна з моделей визначає частину методології. Вибір методології відповідає особливостям проекту, вимогам замовника, враховує досвід розробників та складність розробки та її запровадження.

#### **2.1.2 Існуючі рішення**

Кожна модель розробки програмного забезпечення має свої унікальні особливості, переваги та недоліки. Визначити, яка з них краща – не можна, оскільки під різні завдання, продукти та ідеї обирається свій принцип розробки. Давайте розберемо основні види, що використовуються у програмуванні.

**"Waterfall Model" (каскадна модель або "водоспад")**

Це класична модель, яка використовувалася на початку епохи розробки, але продовжує активно застосовуватись й сьогодні. Її принцип роботи досить простий: кожний наступний етап виконується лише тоді, коли повністю завершено попередній. Існує чітке розподілення етапів, й технологія реалізується ніби каскадом, поступово рухаючись від першого до останнього етапу. Ця модель розробки ПЗ досить жорстка, має суворі правила. Вона чітко визначає терміни проходження кожного етапу. Але є недолік: зробити крок назад дуже складно. Внесення правок до існуючого проекту коштує дуже дорого й проблематично. Такий спосіб підходить лише для чітко розписаних проектів, де є повне розуміння того, що створюється, для яких цілей та які вимоги виставляються. Можна використовувати такий підхід у тому випадку, якщо у проекті є докладний прототип або вже існуюча програма. Крім того, каскадну модель дуже люблять державні установи - у роботі з ними важливо суворо складати звітність, дотримуватися графіку та не відхилятися від заданого спочатку плану.

### **"Incremental Model" (інкрементна модель)**

Інкрементна модель розробки програмного забезпечення підходить у тому випадку, якщо є чіткий план дій, але продукт потрібно запустити досить швидко, а зміни можна буде вносити потім. Її суть полягає в тому, що спочатку розробляється план дій та сегментується на невеликі завдання. Далі кожен блок розробляється за традиційною каскадною моделлю. Спочатку робиться «базовий» продукт з мінімальними, але важливими функціями. Поступово він доповнюється завдяки розробці інших компонентів, які називаються інкрементами. Процес зациклюється, доки не буде повністю зібраної єдиної системи. Кожна частина роботи в такому проекті є готовим елементом. Іноді його можна використовувати окремо. Як правило, він розробляється таким чином, щоб потім його не переробляти. Саме тому й використовується каскадна модель всередині інкрементної моделі. Що пропонує така методологія розробки програмного забезпечення? Насамперед, мінімізуються ризики, забезпечується швидкий реліз та запуск продукту. Крім того, базовий функціонал вже працюватиме й приносить користь для бізнесу, а за необхідності завжди можна впровадити нові сформовані інструменти.

## **"Iterative Model" (ітеративна або ітераційна модель)**

Такий підхід за своєю конструкцією дещо нагадує попередній. Суть ітеративної методології розробки програмного забезпечення полягає в тому, щоб створити базовий функціонал та поступово його покращувати. Поки що звучить схоже з попереднім варіантом, але є різниця. Інкрементну модель можна порівняти з пазлом, де всі елементи викладаються один за одним й поступово збираються до єдиної картини. Ітеративна модель - це ескіз художника, який спочатку робить малюнок олівцем, далі бере в руки фарби, а потім вимальовує деталі й створює картину. Такий метод підходить для великих проєктів, у якому визначаються основні завдання та є загальне уявлення, що має вийти. Але при цьому деталі не зрозумілі, не до кінця зрозуміло, як працюватиме та чи інша функція. За таким методом розробляються, наприклад, соціальні мережі, великі корпоративні платформи.

Загальний функціонал зрозумілий, але кількість функцій та можливості не визначені до кінця. Такий підхід дозволяє поступово додавати нові компоненти та покращувати наявні.

## **"Agile Model" (гнучка методологія розробки)**

Гнучка методологія розробки програмного забезпечення – це чудове рішення для створення продукту, який не до кінця сформований у своїй ідеї. Особливість даного методу полягає в тому, що замовник може одразу спостерігати за змінами у розробці та коригувати дії. Це можливо завдяки визначенню спринтів – відрізків, в яких виконуються завдання. Під час обговорення ставляться цілі, визначається час відрізка, на якому розробники виконують завдання. Далі відбувається обговорення, вносяться зміни, призначається новий відрізок. Метод гнучкої розробки дуже ефективний, але має недоліки. Через те, що неможливо визначити точні результати та зрозуміти, скільки знадобиться часу на реалізацію ідеї, не можна наперед визначити точну вартість. Якщо проєкт налаштований на тривалий життєвий цикл, повинен мати адаптивність до змін на ринку, Agile метод відмінно підходить. Він дозволяє підлаштовуватися під нові вимоги та постійно розвивати продукт.

## **"Spiral Model" (спіральна модель)**

Спіральні моделі розробки ПЗ підходять для великих проектів, де припущення помилок призводить до великих втрат та негативних наслідків. Акцент робиться на оцінці ризиків та опрацюванні конкретних бізнес-завдань. Принцип роботи наступний: Робота йде по спіралі, на кожному витку якої здійснюється 4 основні дії: створення плану, аналіз ризиків, розробка та конструювання, оцінка результату та збирання відгуків. Якщо всі 4 етапи успішно пройдені, то технологія переходить на новий виток. А сам перехід на новий виток нагадує інкрементну модель, де кожен блок розробляється незалежно та приєднується до спочатку створеного базового функціоналу. Особливість підходу полягає в тому, що більше часу йде на аналіз, розрахунки та оцінку наслідків, ніж безпосередньо на саму розробку. Розробники іноді набагато швидше реалізують функцію, ніж вона вивчається аналітиками. Спіральна модель підходить великим компаніям, оскільки коштує недешево. Крім того, розробка забирає досить тривалий час. Цей метод для тих, хто готовий ризикнути й хоче забезпечити собі можливість зробити "крок назад", чи мінімізувати втрати у випадку можливого провалу ризикового проекту.

## **"V-Model" (V-подібна модель)**

Особливість підходу полягає в наголосі на тестування та перевірку працездатності систем під час розробки. Тести виконуються одночасно із самим процесом створення продукту. Сам принцип успадковує базовий підхід при каскадній розробці. Процес іде покроково, існує чіткий план дій, складається суворе технічне завдання. Але паралельно виконуються тести, у разі виявлення помилок вони одразу виправляються, незалежно від етапу розробки. Особливість цієї методології розробки ПЗ полягає у тому, що на ранніх стадіях створення проводяться тести, а до нової стадії можна перейти лише тоді, коли усуваються всі помилки. При цьому на новій стадії тести аналізують не лише новий етап, а й усі попередні. Це дозволяє контролювати взаємозв'язок компонентів та їхню працездатність.

## **"RAD Model" (rapid application development model або швидка розробка додатків)**

Методологія розробки програмного забезпечення RAD підходить тим компаніям, які хочуть максимально швидко запуснути продукт. Суть у тому, що це етапи створення додатка розподіляються на кілька окремих блоків, з кожним із яких працює окрема команда розробників. Після цього робочі невеликі модулі збираються до єдиної системи та утворюють робочий прототип, який можна показати клієнту. Далі збираються відгуки та вносяться зміни. Метод дозволяє швидко розробляти навіть складні продукти, але має низку особливостей:

- клієнт має брати активну участь в розробці та постійно спостерігати за результатами;
- вартість використання такого методу висока, оскільки доведеться наймати великий штат розробників;
- Необхідно чітко розуміння, що має вийти у результаті, щоб кожна команда знала, яку задачу вирішує її модуль.

Такий підхід дає можливість швидко протестувати ідею, вивести новий продукт на ринок протягом короткого часу, створювати потужні додатки з великим функціоналом.

### **2.1.3 Порівняння методологій**

#### **1. Гнучкість**

Agile:

- Висока гнучкість завдяки коротким ітераціям (спринтам).
- Можливість швидко реагувати на зміни вимог.
- Постійна взаємодія з клієнтом дозволяє регулярно коригувати напрямок розробки.

Spiral:

- Гнучка завдяки багаторазовим ітераціям та можливості оцінки ризиків на кожному етапі.
- Можливість внесення змін в кожній ітерації на основі аналізу ризиків і зворотного зв'язку.

RAD:

- Висока гнучкість завдяки швидкому створенню прототипів і інтерактивному процесу розробки.
- Активна участь користувачів у розробці дозволяє швидко враховувати їхні вимоги та побажання.

#### Incremental:

- Гнучка, оскільки функціональність додається поступово.
- Можливість змін вимог між інкрементами.

#### Iterative:

- Помірна гнучкість, оскільки кожна ітерація дозволяє переглянути та поліпшити попередню роботу.

#### Waterfall:

- Низька гнучкість, оскільки кожна фаза має бути завершена перед переходом до наступної.
- Важко врахувати зміни після завершення певної фази.

#### V-Model:

- Низька гнучкість, аналогічна Waterfall.
- Внесення змін на пізніх стадіях є складним і дорогим.

## **2. Чітка структура**

#### Waterfall:

- Чітко визначені етапи з послідовним виконанням кожної фази.
- Легко керувати процесом та документувати кожен етап.

#### V-Model:

- Чітка структура з визначеними етапами розробки та тестування.
- Кожна фаза розробки пов'язана з відповідною фазою тестування.

#### Incremental:

- Структурована, але з можливістю розділення проекту на окремі інкременти.
- Кожен інкремент є самостійною частиною, яка додається до попередніх.

#### Iterative:

- Чітка структура, яка включає багаторазові ітерації з циклом аналізу, дизайну, реалізації та тестування.

Agile:

- Менш структурована у порівнянні з Waterfall і V-Model.
- Фокус на адаптацію та швидке реагування на зміни.

Spiral:

- Комбінація чіткої структури (ітерації з визначеними фазами) та гнучкості (оцінка ризиків).

RAD:

- Чітка структура для створення швидких прототипів, але менш формальна у порівнянні з Waterfall.

### **3. Фокус на ризиках**

Spiral:

- Основний акцент на аналізі та управлінні ризиками.
- Кожна ітерація включає оцінку ризиків, що дозволяє зменшити ймовірність провалу проекту.

Agile:

- Відсутній формалізований процес оцінки ризиків, але регулярні зустрічі та зворотний зв'язок дозволяють швидко виявляти та вирішувати проблеми.

Iterative:

- Регулярний перегляд результатів кожної ітерації дозволяє виявляти та усувати ризики.

Waterfall, V-Model, Incremental, RAD:

- Не мають явного фокусу на ризиках. Ризики можуть бути оцінені на початку проекту, але не на кожній фазі.

### **4. Швидка доставка**

Agile:

- Короткі спринти (зазвичай 2-4 тижні) дозволяють швидко постачати робоче програмне забезпечення.

- Постійні релізи забезпечують швидке надання функціональності користувачам.

RAD:

- Фокус на швидкому створенні прототипів забезпечує швидку доставку початкових версій програмного забезпечення.

Incremental:

- Регулярні випуски інкрементів дозволяють швидко доставляти нову функціональність.

Iterative:

- Кожна ітерація включає створення та тестування робочого програмного забезпечення.

Spiral:

- Швидкість доставки залежить від тривалості ітерацій, але можливість швидкого внесення змін існує.

Waterfall, V-Model:

- Швидкість доставки нижча, оскільки готовий продукт доставляється лише після завершення всіх етапів.

## **5. Етапи тестування**

V-Model:

- Чітка структура тестування на кожному етапі розробки.
- Вимоги — План тестування вимог, Системний дизайн — План тестування системного дизайну і т.д.

Iterative:

- Тестування проводиться в кожній ітерації, що дозволяє виявляти дефекти на ранніх стадіях.

Agile:

- Тестування є інтегрованою частиною кожного спринту.
- Автоматизація тестування часто використовується для прискорення процесу.

Spiral:

- Тестування проводиться на кожному витку спіралі, що дозволяє постійно оцінювати якість продукту.

Waterfall:

- Тестування проводиться після завершення етапу реалізації.
- Помилки виявляються на пізніх стадіях, що може ускладнити їх виправлення.

Incremental:

- Тестування проводиться для кожного інкремента, що дозволяє поступово виявляти і виправляти помилки.

RAD:

- Активне залучення користувачів до тестування прототипів.
- Регулярне тестування прототипів дозволяє швидко виявляти та усувати проблеми.

## **6. Залучення користувачів**

Agile:

- Постійна взаємодія з клієнтами та користувачами.
- Клієнти беруть участь у плануванні спринтів і надають зворотний зв'язок.

RAD:

- Користувачі активно залучені до процесу розробки, особливо на етапах прототипування.

Incremental, Iterative, Spiral:

- Регулярне залучення користувачів для оцінки та надання зворотного зв'язку.

Waterfall, V-Model:

- Залучення користувачів на початкових етапах для визначення вимог.
- Обмежене залучення на інших етапах.

## **7. Оцінка часу та витрат**

Waterfall, V-Model:

- Легше прогнозувати час і витрати, оскільки вимоги і планування здійснюються на початку проекту.

Agile, RAD, Incremental, Iterative:

- Складніше оцінити час і витрати через постійні зміни та адаптації.
- Необхідність у регулярних переоцінках і коригуваннях бюджету та графіку.

Spiral:

- Врахування ризиків на кожному етапі допомагає більш точно оцінити витрати.
- Оцінка часу може змінюватися залежно від виявлених ризиків.

### 2.1.4 Вибір методології

Тепер коли ми розглянули існуючі методології та порівняли їх, ми можемо перейти до вибору методології, яку ми будемо використовувати в нашому проєкті. Допомогти нам в цьому може алгоритм вибору методології.

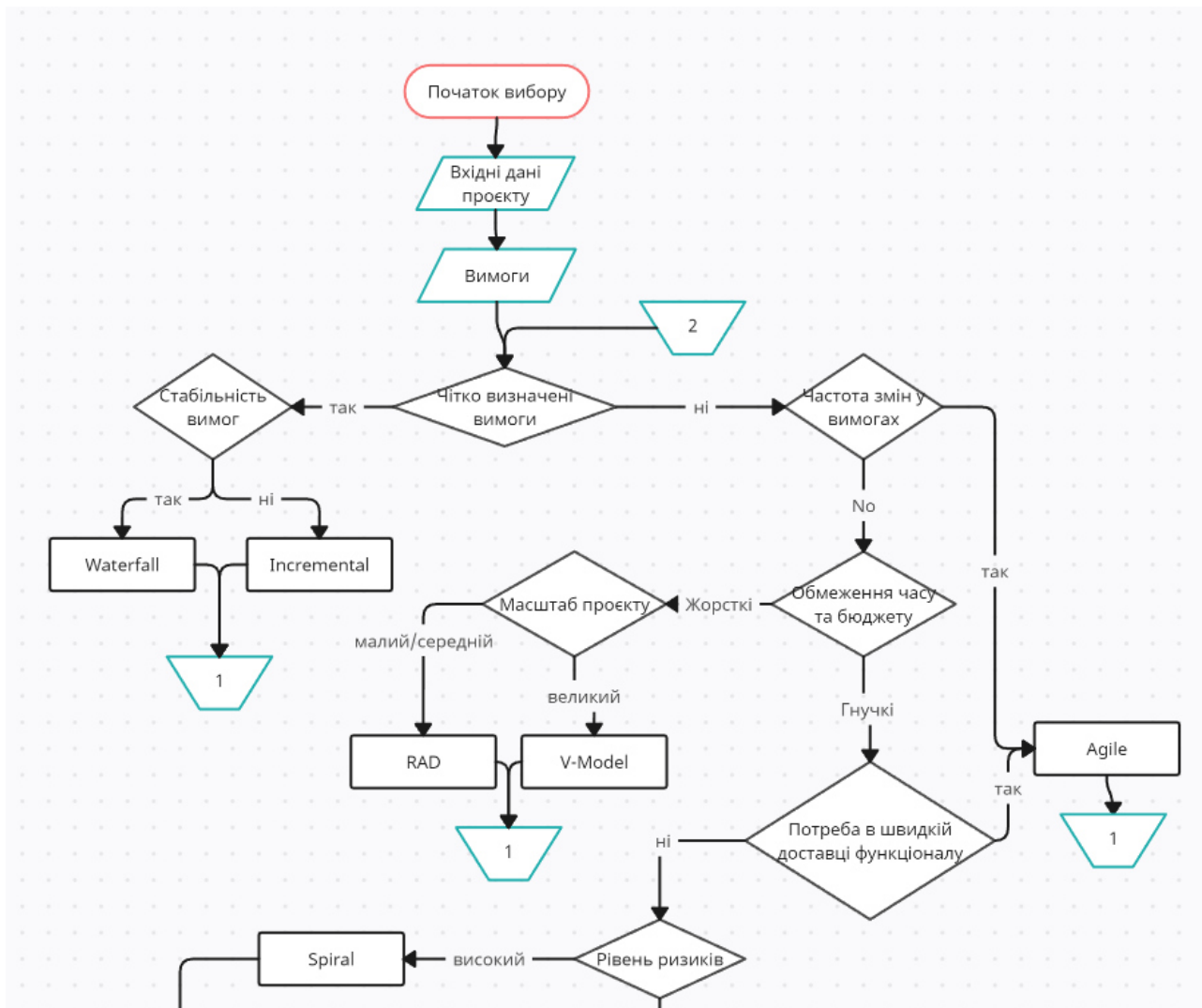


Рисунок 2.1 Алгоритм вибору методологій

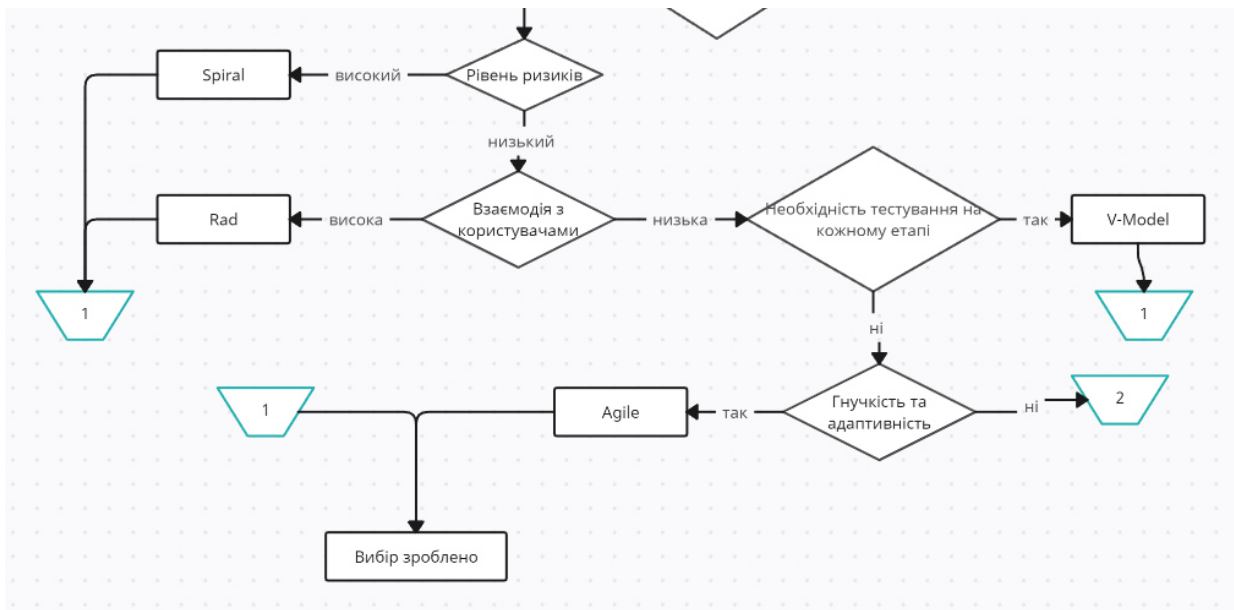


Рисунок 2.2 Алгоритм вибору методологій

Виконавши алгоритм ми прийшли до висновку, що найкращий варіант для нашого проекту – каскадна модель. Тому, що в нас є чітко визначені вимоги.

## 2.2 Метод розробки

### 2.2.1 Загальні відомості

В даному проекті ми будемо використовувати додаток для розробки інтерфейсу і прототипування.

### 2.2.2 Існуючі рішення

Виконавши аналіз додатків для розробки інтерфейсу та прототипування, в нас з'являються три варіанти: Figma, InVision та Adobe XD

Figma - це потужний інструмент для розробки інтерфейсів, що дозволяє працювати над проектами у реальному часі кільком користувачам одночасно. Він відомий своїм зручним для співпраці середовищем, де розробники і клієнти можуть працювати разом над проектами без потреби у встановленні додаткового ПЗ. Сервіс доступний при підписці, проте є безкоштовний тарифний план для одного користувача.

InVision - це інструмент для прототипування і спільної роботи, який дозволяє розробникам створювати інтерактивні макети, ділитися ними з командою та клієнтами, отримувати зворотний зв'язок і проводити користувацькі тестування.

InVision також пропонує додаткові інструменти, такі як InVision Studio, для створення більш складних анімацій та інтерфейсів.

Adobe XD - це універсальний інструмент для розробки та прототипування інтерфейсу від Adobe. Він надає широкі можливості для створення як статичних макетів, так і інтерактивних прототипів. XD інтегрується з іншими продуктами Adobe, що робить його особливо зручним для користувачів, які вже працюють з іншими продуктами Adobe Creative Cloud.

### **2.2.3 Порівняння систем**

Figma

Співпраця в реальному часі:

Перевага: Figma дозволяє кільком користувачам одночасно працювати над тим самим проектом у реальному часі, що робить його ідеальним для командної роботи.

Недолік: Потрібне постійне підключення до інтернету для повноцінної роботи.

Кросплатформеність:

Перевага: Figma працює прямо у браузері, що означає, що його можна використовувати на будь-якій операційній системі (Windows, macOS, Linux) без встановлення додаткового ПЗ.

Недолік: Веб-інтерфейс може бути менш продуктивним у порівнянні з настільними програмами при роботі з великими проектами.

Плагіни та інтеграції:

Перевага: Figma має багату екосистему плагінів, що дозволяє розширювати функціональність інструменту.

Недолік: Деякі специфічні плагіни можуть не бути доступні.

InVision

Прототипування та інтерактивність:

Перевага: InVision спеціалізується на створенні інтерактивних прототипів з анімаціями, переходами та інтерактивними елементами.

Недолік: Менше можливостей для детального дизайну інтерфейсів порівняно з Figma чи Adobe XD.

Співпраця та обговорення:

Перевага: InVision забезпечує потужні інструменти для коментарів та співпраці, що дозволяє командам обговорювати проекти безпосередньо на платформі.

Недолік: Обмеження на одночасну роботу над макетами порівняно з реальним часом у Figma.

Інтеграції та підтримка:

Перевага: Інтеграція з популярними інструментами, такими як Sketch, Photoshop і Slack.

Недолік: Інтеграції можуть бути складнішими в налаштуванні порівняно з більш інтуїтивними підходами в Figma чи Adobe XD.

Adobe XD

Інтерфейс та зручність використання:

Перевага: Adobe XD має інтуїтивний інтерфейс, що забезпечує легкість використання для новачків та ефективність для досвідчених користувачів.

Недолік: Деякі користувачі можуть відчувати обмеження у функціональності порівняно з Figma.

Прототипування та дизайн:

Перевага: Adobe XD поєднує в собі інструменти для дизайну та прототипування в одній програмі, що забезпечує цілісний підхід до роботи над проектом.

Недолік: Обмежені можливості для співпраці в реальному часі у порівнянні з Figma.

Інтеграція з Adobe Creative Cloud:

Перевага: Глибока інтеграція з іншими продуктами Adobe, що дозволяє безшовно використовувати ресурси з Photoshop, Illustrator та інших інструментів.

Недолік: Потребує підписки на Adobe Creative Cloud, що може бути дорогим варіантом для деяких користувачів.

Для розробки фронтенду ми будемо використовувати додаток Figma, так як він найкраще підійде для роботи в команді та велика кількість плагінів може «розв'язати руки» розробникам.

## 2.3 Інструменти та технології для проєктування архітектури

Для проєктування архітектури нашої системи ми будемо використовувати UML-діаграми, а саме діаграми класів, компонентів та послідовностей.

### А) Діаграма класів

Діаграма класів використовується для моделювання структури системи та візуалізації відносин між різними об'єктами. Відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів.

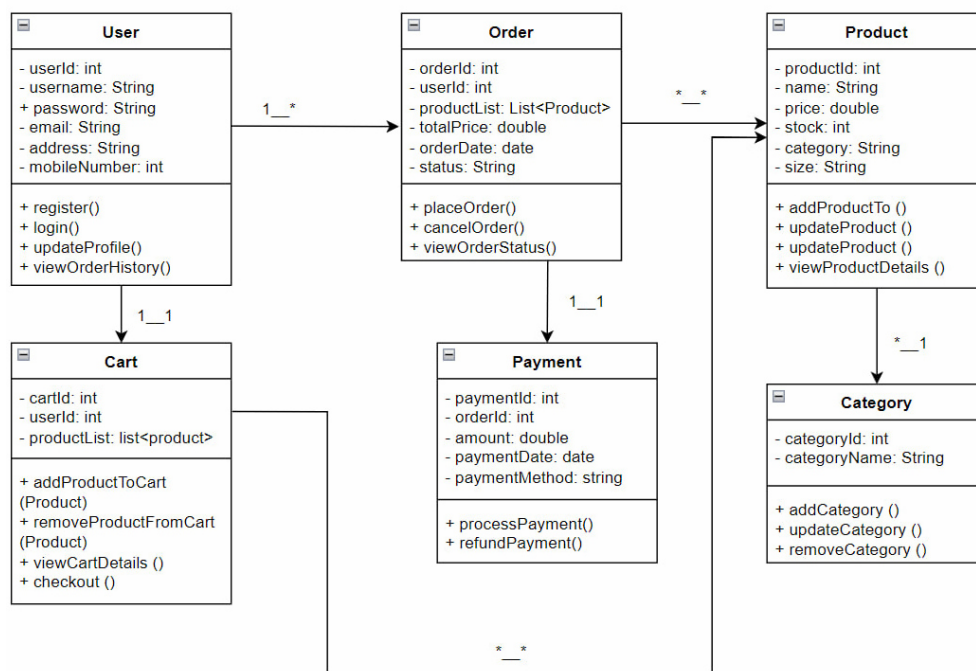


Рисунок 2.3 Діаграма класів

### Б) Діаграма компонентів

Діаграма компонентів використовується для моделювання фізичної архітектури системи. Вона допомагає відобразити, як різні частини системи (компоненти) взаємодіють і які залежності існують між ними.

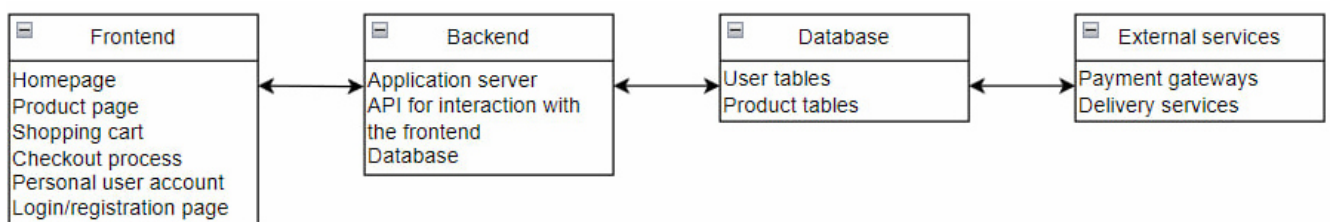


Рисунок 2.4 Діаграма компонентів

### В) Діаграми послідовностей

Діаграма послідовності - це важливий інструмент у розробці програмного забезпечення, який використовується для моделювання та візуалізації послідовності взаємодії між об'єктами в системі.

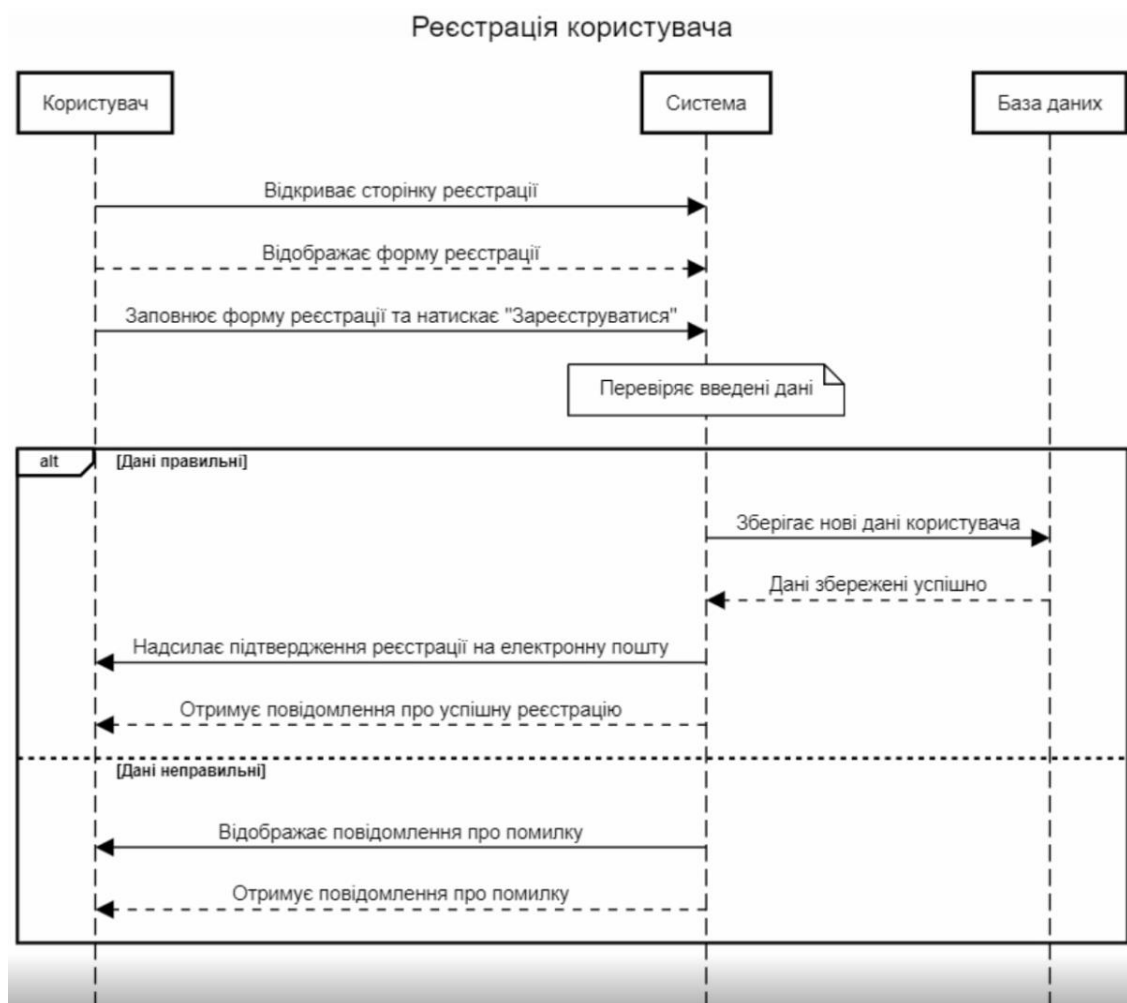


Рисунок 2.5 Діаграма послідовностей «реєстрація користувача»

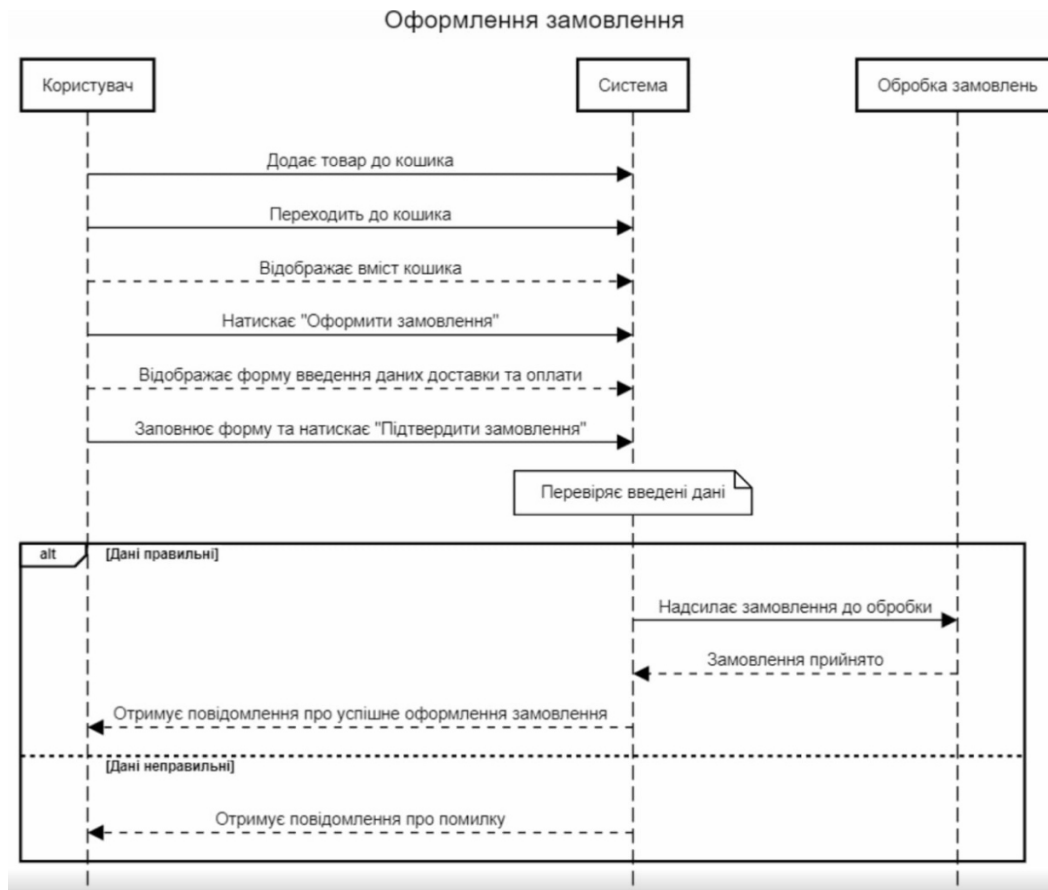


Рисунок 2.6 Діаграма послідовностей «оформлення замовлення»

## 2.4 Висновки до розділу

- Було розглянуто, що таке методологія програмного і яку роль вона відіграє в проєкті.
- Було розглянуто існуючі методології.
- Було порівняно існуючі методології та обрали модель, яка найкраще підходить до проєкту.
- Було розглянуто який тип програмного забезпечення буде використовуватись в розробці продукту.
- Було розглянуто додатки для розробки інтерфейсу і прототипування та порівняли їх.
- Було обрано додаток який буде використовуватись в розробці продукту.
- Було розглянуто інструменти та технології для проєктування архітектури які будуть використовуватись.
-

## РОЗДІЛ 3 ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ

### 3.1 Загальні положення

У цьому розділі розглядається процес програмної реалізації інтернет-магазину Ustock, що включає детальний опис реалізації основних компонентів, бази даних та функціонал веб-сайт з точки зору рядового користувача. Кожен з цих аспектів є важливим для забезпечення успішного впровадження та функціонування веб-сайту.

### 3.2 Реалізація програмного забезпечення

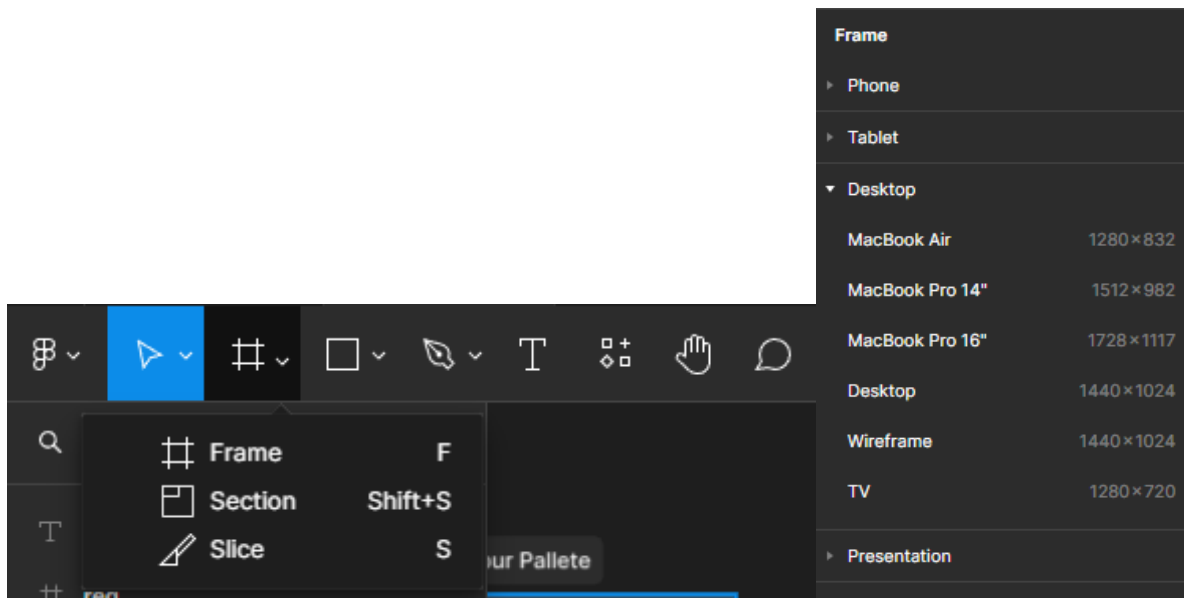
У цьому підрозділі буде розглянута реалізація основних компонентів інтернет-магазину Ustock, включаючи функціональні можливості, такі як реєстрація користувача, вхід в систему, пошук товару за назвою, сортування товарів та інші. Для початку ми створимо прототип інтерфейсу. Потім опираючись на інтерфейс ми зробимо фронтенд інтернет-магазину. Останій етап, буде заключатися в реалізації бекендної складової.

#### 3.2.1 Розробка прототипу інтерфейсу

Для розробки прототипу інтерфейсу буде використовуватись додаток Figma. В цій частині ми повині розробити дизайн веб-сайту, який буде прикладом для подальшої реалізації в наступному етапі.

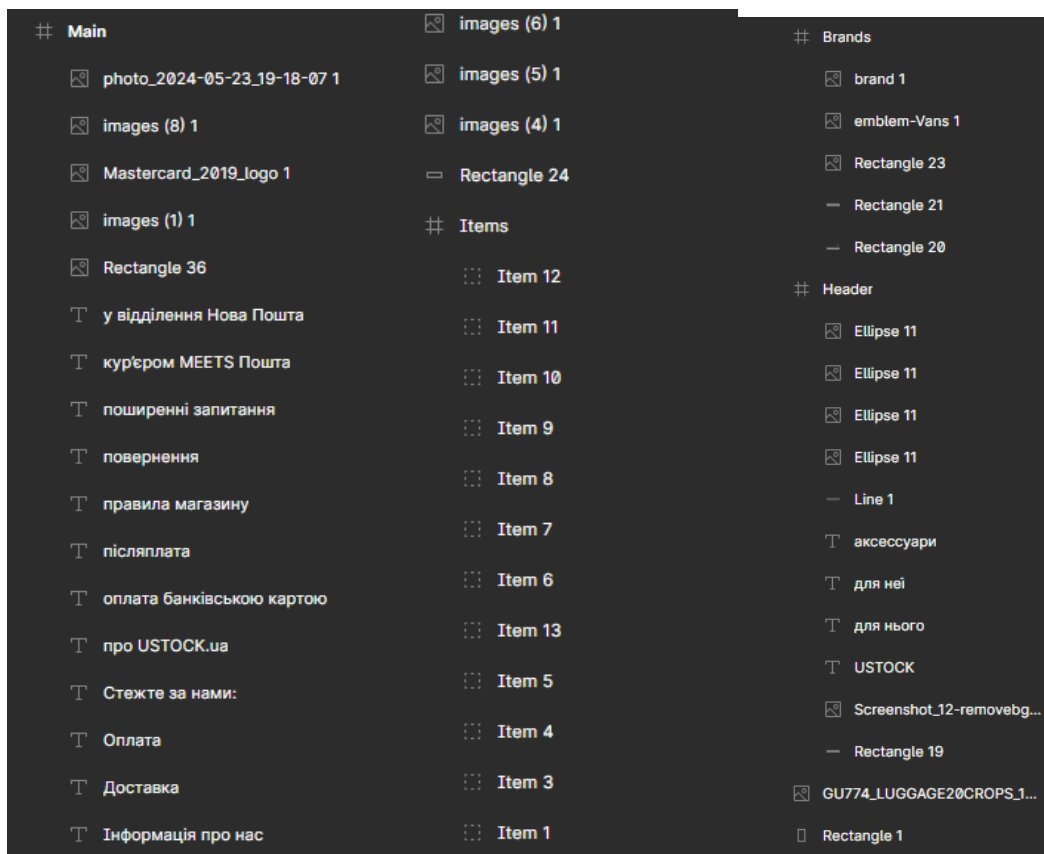
Ми оглянемо лише створення головної сторінки веб-сайту, так як інші сторінки створюються за таким же принципом.

Для створення сторінки, на панелі інструментів ми обираємо опцію «Frame». Потім, на панелі керування, ми обираємо розмір сторінки. Для початку, розмір сторінки буде складати 1718x1117 пікселів. Проте в будь-який момент можна змінити розмір сторінки, в нашому випадку, це буде 1728x 3206 пікселів, так ми зможемо вмістити всі необхідні елементи, які будуть на цій сторінці.



Рисунки 3.1 та 3.2 Створення фрейму

Коли ми створили рамку, ми починаємо заповнювати її елементами, які притаманні інтернет-магазинам – це header, список товарів та footer. Ми будемо додавати та редагувати елементи, які в результаті створюють загальний вигляд веб-сайту.



Рисунки 3.3, 3.4 та 3.5 Список шарів

### 3.2.2 Фронтенд

Далі, маючи прототип інтерфейсу, ми переходимо до його реалізації. Використовувати ми будемо HTML+CSS. Опираючись на прототип, ми будемо створювати повноцінний інтерфейс сайту. Фрагмент коду буде наведений в додатках.

### 3.2.3 Бекенд

Тут ми розглянемо реалізацію функціональних можливостей інтернет-магазину Ustock такі як реєстрація користувача, вхід в систему, пошук товару за назвою, сортування товарів та інші.

#### 3.2.3.1 Реєстрація та вхід користувача

Щоб замовити товар на сайті користувачеві спочатку потрібно зареєструватися. Він, звісно, може сортувати та переглядати товар але, щоб користуватися всім функціоналом сайту необхідний акаунт. Після того, як користувач створив акаунт при наступному заході на сайт, він зможе зайти в свій акаунт.

Налаштування середовища

#### Пакети

- `express`: Веб-фреймворк для Node.js, який використовується для створення веб-додатків та API.
- `mongoose`: ORM (Object-Relational Mapping) бібліотека для MongoDB та Node.js.
- `body-parser`: Middleware для парсингу тіла HTTP-запитів.
- `bcrypt`: Бібліотека для хешування паролів.
- `jsonwebtoken`: Бібліотека для створення та перевірки JSON Web Token (JWT).

Модель користувача (`models/User.js`)

#### Структура моделі

- Поля моделі: `firstName`, `lastName`, `mobileNumber`, `email`, `password`, `region`, `city`, `street`, `postalCode`, `houseNumber`.
- Перед збереженням користувача: Використовується middleware `pre('save')` для хешування пароля за допомогою `bcrypt`.

## Методи моделі

- `comparePassword`: Метод для порівняння введеного пароля з хешованим паролем в базі даних.

## Сервер і маршрути (app.js)

### Налаштування серверу

- Імпорт пакетів: Імпортуються необхідні пакети (`express`, `mongoose`, `body-parser`, `bcrypt`, `jsonwebtoken`, `User`).
- Підключення до MongoDB: Використовується `mongoose.connect` для підключення до локальної бази даних MongoDB.
- Middleware: `body-parser.json()` для парсингу JSON-запитів.

### Маршрут для реєстрації (/register)

- Отримання даних: Отримуються дані з тіла запиту (`req.body`).
- Створення нового користувача: Створюється новий екземпляр користувача `newUser`.
- Збереження користувача: Викликається метод `save()` для збереження користувача в базі даних.
- Обробка результату: Повертається відповідь з повідомленням про успішну реєстрацію або помилку.

### Маршрут для входу (/login)

- Отримання даних: Отримуються дані з тіла запиту (`req.body`).
- Пошук користувача: Використовується `findOne` для пошуку користувача за `email` або номером телефону.
- Перевірка пароля: Використовується метод `comparePassword` для порівняння введеного пароля з хешованим паролем в базі даних.
- Створення токена: Якщо пароль вірний, створюється JWT токен за допомогою `jsonwebtoken.sign`.
- Обробка результату: Повертається відповідь з повідомленням про успішний вхід або помилку.

### 3.2.3.2 Зміна даних користувача

Користувач може змінити свої дані, які він ввів при реєстрації, якщо вони змінились.

Маршрут для оновлення даних користувача (/users/:id)

- Отримання даних: Отримуються параметри маршруту (req.params) та дані для оновлення (req.body).
- Пошук та оновлення користувача: Використовується метод findByIdAndUpdate для пошуку користувача за ID та оновлення його даних. Параметри new: true повертають оновлений документ, а runValidators: true забезпечують валідацію даних перед оновленням.
- Обробка результату: Повертається відповідь з повідомленням про успішне оновлення користувача або помилку.

### 3.2.3.3 Пошук та сортування товару

Користувач зможе відсортувати товар за критеріями або знайти його за назвою. Це дозволить йому зекономити час на пошук бажаного товару.

**Модель товару (models/Product.js)**

- Поля моделі: name, description, price, category, stock.
- Схема Mongoose: Визначає структуру документа товару в базі даних.

**Сервер і маршрути (app.js)**

- Підключення пакетів: Імпортуються необхідні пакети (express, mongoose, body-parser, Product).
- Підключення до MongoDB: Використовується mongoose.connect для підключення до локальної бази даних MongoDB.
- Middleware: body-parser.json() для парсингу JSON-запитів.

**Маршрут для додавання товару (/products)**

- Отримання даних: Отримуються дані з тіла запиту (req.body).
- Створення нового товару: Створюється новий екземпляр товару newProduct.
- Збереження товару: Викликається метод save() для збереження товару в базі даних.

- Обробка результату: Повертається відповідь з повідомленням про успішне додавання або помилку.

### **Маршрут для пошуку товарів за назвою (/search)**

- Отримання даних: Отримуються параметри запиту (req.query).
- Перевірка параметра: Перевіряється, чи є параметр name.
- Пошук товарів: Використовується метод find для пошуку товарів з назвою, яка відповідає регулярному виразу (незалежно від регістру).
- Обробка результату: Повертається список знайдених товарів або помилка.

### **Маршрут для отримання товарів за категорією та підкатегорією (/products)**

- Отримання даних: Отримуються параметри запиту (req.query).
- Перевірка параметра: Перевіряється наявність параметра category.
- Формування запиту: Створюється об'єкт query, який включає категорію та, якщо вказана, підкатегорію.
- Пошук товарів: Використовується метод find для пошуку товарів за категорією та підкатегорією в базі даних.
- Обробка результату: Повертається список знайдених товарів або повідомлення про помилку.

## **3.3 База даних**

База даних є критично важливим компонентом в роботі інтернет-магазину. Вона включає в себе таблиці для зберігання даних користувачів, товару який доступний для придбання, замовлення та товару який буде включатись в замовлення. Всі таблиці, які будуть описані нижче продубльовані в Microsoft Access для кращого розуміння.

В першій таблиці будуть наведені дані, які користувач ввів при реєстрації. Вона в себе включає: id користувача, ім'я, прізвище, електрону пошту, мобільний телефон, область, назву населеного пункту, назва вулиці, номер будинку, поштавий індекс та пароль. При вході в акаунт будуть звірятись такі дані: електрона пошта або номер телефону та пароль.

Ідентифікація	user_id	first_name	last_name	email	mob_number	region	city/village	street	house_number	postal_code	password
1	00000001	Олександр	Левіщенко	Levishenkoy@	+380688389065	Київ	Васильків	Сагайдачного	38	08602	f36211966
2	00000002	Артем	Гузлік	Guzlika@gmail	+380984047685	Київ	Київ	просп. Академ	53	01001	s1234512
(Новий)											

Рисунок 3.1 Дані користувача

Друга таблиця створена для товарів які наявні або ненаявні на сайті. Вона включає в себе id товару, принадлежність товару (він/вона), тип товару (футболки, худі і т.д.), ім'я товару, бренд, розмір, ціна та кількість товару. Якщо кількість дорівнює нулю, то цей товар недоступний для покупки.

Ідентифікація	item_id	gender	item_type	item_name	Brand	size	price	quantity
1	000000001	he	Jackets	Carhartt WIP M	Carhartt	M	7099	10
2	000000002	he	Jackets	Carhartt WIP M	Carhartt	S	7099	10
3	000000003	he	Jackets	Carhartt WIP M	Carhartt	L	7099	10
4	000000004	he	Jackets	Carhartt WIP M	Carhartt	XL	7099	10
5	000000005	he	Jackets	Carhartt WIP M	Carhartt	XXL	7099	10
6	000000011	he/she	Sneakers	Nike Dunk Higl	Nike	40	8650	9
7	000000012	he/she	Sneakers	Nike Dunk Higl	Nike	41	8650	8
8	000000013	he/she	Sneakers	Nike Dunk Higl	Nike	42	8650	10
9	000000014	he/she	Sneakers	Nike Dunk Higl	Nike	43	8650	5
10	000000015	he/she	Sneakers	Nike Dunk Higl	Nike	44	8650	12
11	000000016	he/she	Sneakers	Nike Dunk Higl	Nike	45	8650	1
12	000000020	he/she	Accessories	Vans Belt	Vans	One size	699	20
13	000000030	he/she	Accessories	Carhartt WIP M	Carhartt	One size	1390	15
14	000000040	he/she	Sneakers	Vans SK8-Hi	Vans	40	3500	10
15	000000041	he/she	Sneakers	Vans SK8-Hi	Vans	41	3500	9
16	000000042	he/she	Sneakers	Vans SK8-Hi	Vans	42	3500	6
17	000000043	he/she	Sneakers	Vans SK8-Hi	Vans	43	3500	10
18	000000044	he/she	Sneakers	Vans SK8-Hi	Vans	44	3500	12
19	000000045	he/she	Sneakers	Vans SK8-Hi	Vans	45	3500	8
20	000000050	he	T-shirts	Carhartt WIP C	Carhartt	S	1399	10
21	000000051	he	T-shirts	Carhartt WIP C	Carhartt	M	1399	6
22	000000052	he	T-shirts	Carhartt WIP C	Carhartt	L	1399	15
23	000000053	he	T-shirts	Carhartt WIP C	Carhartt	XL	1399	14
24	000000054	he	T-shirts	Carhartt WIP C	Carhartt	XXL	1399	10
25	000000060	he	T-shirts	Nike Solo Swo	Nike	S	1750	10
26	000000061	he	T-shirts	Nike Solo Swo	Nike	M	1750	8
27	000000062	he	T-shirts	Nike Solo Swo	Nike	L	1750	5
28	000000063	he	T-shirts	Nike Solo Swo	Nike	XL	1750	6
29	000000064	he	T-shirts	Nike Solo Swo	Nike	XXL	1750	10
30	000000070	he	Hoodies and s	Nike Hoodie N	Nike	S	4015	8

Рисунок 3.2 Дані товару

Наступна таблиця присвячена для замовлень. Коли користувач оформляє замовлення в таблицю надходять такі дані: id замовлення, id користувача, дата замовлення, статус замовлення та ціна засовлення.

Ідентифікатор	order_id	user_id	order_date	status	order_price
1	0000001	00000002	31.05.2024	accepted	4 254,00 €
2	0000002	00000001	01.06.2024	considered	6 999,00 €
*	(Новий)				0,00 €

Рисунок 3.3 Дані замовлення

Остання таблиця створена для товарів, які входять для замовлення. Вона включає в себе id замовлення, id товару, розмір, кількість та ціна товару.

Ідентифікатор	order_id	item_id	size	quantity_of	price
1	0000001	000000003	<u>l</u>	1	4 299,00 €
*	(Новий)			0	0,00 €

Рисунок 3.4 Дані товарів до замовлення

### 3.4 Функціональні можливості

В даному підрозділі ми розглянемо функціонал який буде доступний рядовому користувачеві.

#### 3.4.1 Сценарій користувача

Для початку ми візьмемо сценарій, який покаже більшу частину функціоналу веб-сайту. А інші функціональні можливості ми розглянемо окремо. Отже, сценарій такий:

1. Реєстрація користувача на веб-сайті
2. Вибір товару
3. Додавання в кошик
4. Оформлення замовлення

При заході на веб-сайт, користувач потрапляє на головну сторінку де він може обрати товар. До функціоналу цієї сторінки входить: вибір товару, пошук товару, перехід на сторінку акаунту, перехід на сторінку вподобань, перехід на сторінку кошику та можливість сортування товару (для нього, для неї та аксесуари).

Проте перш за все, по сценарію, користувач повинен зареєструватися на веб-сайті або мобільному додатку.

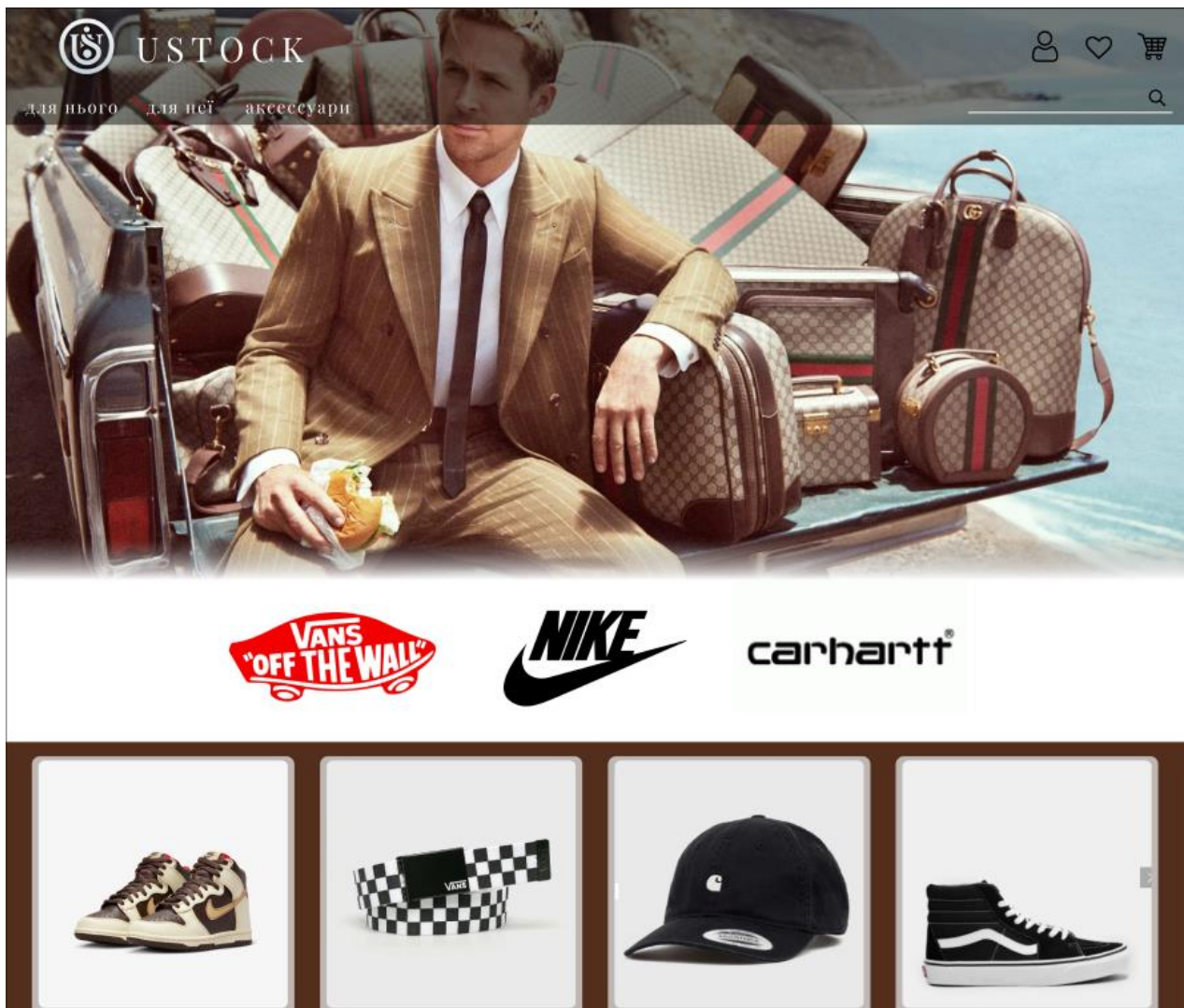


Рисунок 3.5 Головна сторінка веб-сайту

Далі користувач потрапляє на сторінку входу в акаунт на веб-сайт. Він повинен ввести свою адресу електронної пошти або номер телефону. Проте, в користувача немає акаунту, тож він натискає на посилання «Зареєструватись». Відкривається вкладка реєстрація, де користувач повинен ввести своє ім'я, прізвище, адресу електронної пошти, номер телефону та пароль. Також користувач повинен ввести свої дані для доставки, а саме: область, населений пункт, поштовий індекс, вулицю та номер будинку. Обов'язково треба погодитись з правилами магазину. Після заповнення всіх даних, користувач повинен натиснути на кнопку «Створити акаунт».

Після цього, акаунт створено і користувач може перейти до покупок. Його повертає на головну сторінку.

USTOCK

Електронна пошта або номер телефону

Пароль

[Завми пароль?](#)

У вас немає акаунту? [Зареєструйтесь](#)

**Ввійти**

Інформація про нас  
про USTOCK,на  
правила магазину  
повернення  
поширенні запитання

Доставка  
курьером MEETS Пошта  
у відділення Nova Пошта

Оплата  
оплата банківською картою  
післяплата

Стежте за нами:

Рисунок 3.6 Сторінка входу в акаунт

Рисунок 3.7 Сторінка реєстрації акаунту

Повернувшись на головну сторінку, користувач, по сценарію, повинен обрати товар. В нього є кілька варіантів, як це зробити:

- Відсортувати товари та знайти бажаний товар
- Ввести в рядку пошуку назву товару
- Через головну сторінку

Перший варіан найкраще підходить у випадку, якщо користувач знає який саме товар він хоче придбати. Другий варіант підходить у випадку, якщо користувач знає який тип товару він хоче купити (наприклад кросівки або футболка). Останій варіант показує всі товари, які доступні на веб-сайті. Користувач буде використовувати, третій варіант.

Знайшовши бажаний товар, потенційний клієнт потрапляє на сторінку товару, де він зможе подивитись інформацію про товар, наявні розміри та фото товару. Обравши бажаний розмір, користувач натискає на кнопку «Додати в кошик» і товар додається у кошик. Користувач, може додати незлічену кількість товару до свого кошику та придбати їх, як одне замовлення.

USTOCK

Введіть промокод SummerVibes до 01.06.24, та отримаєте знижку 15%

Carhartt WIP Manu Jacket

S M **L** XL XXL

остання розмір!

[Таблиця розмірів](#)

4299 грн. 7099 грн.

**Додати в кошик**

Куртка з колекції Carhartt WIP. Неутеплена модель виконана з гладкого матеріалу. Для створення моделі були використані високоякісні матеріали.

- Модель без утеплення.
- Прямий фасон.
- Прорізані кишені.
- Модель з коміром.
- Застібється на блискавку.
- Модель на підкладці.

Склад  
100% Неолон

ID Товару  
99KK-KUMOPK\_99X

Код виробника  
1031480

Рисунок 3.8 Сторінка товару

Натиснувши на інтуїтивно зрозумілу кнопку «кошик», що знаходиться в верхній-правій частині екрану. Користувач потрапляє на сторінку кошику, де він може побачити товари, які були додані в кошик, ціну товару або суму товарів без знижки, знижку та ціну товару або товарів, вже зі знижкою. Користувач може видалити товар з кошика, натиснувши на інтуїтивно зрозумілу кнопку видалити з кошика. Якщо користувача все влаштовує, він може натиснути на кнопку «Оформити замовлення» та перейти на сторінку де він буде оформлювати замовлення.

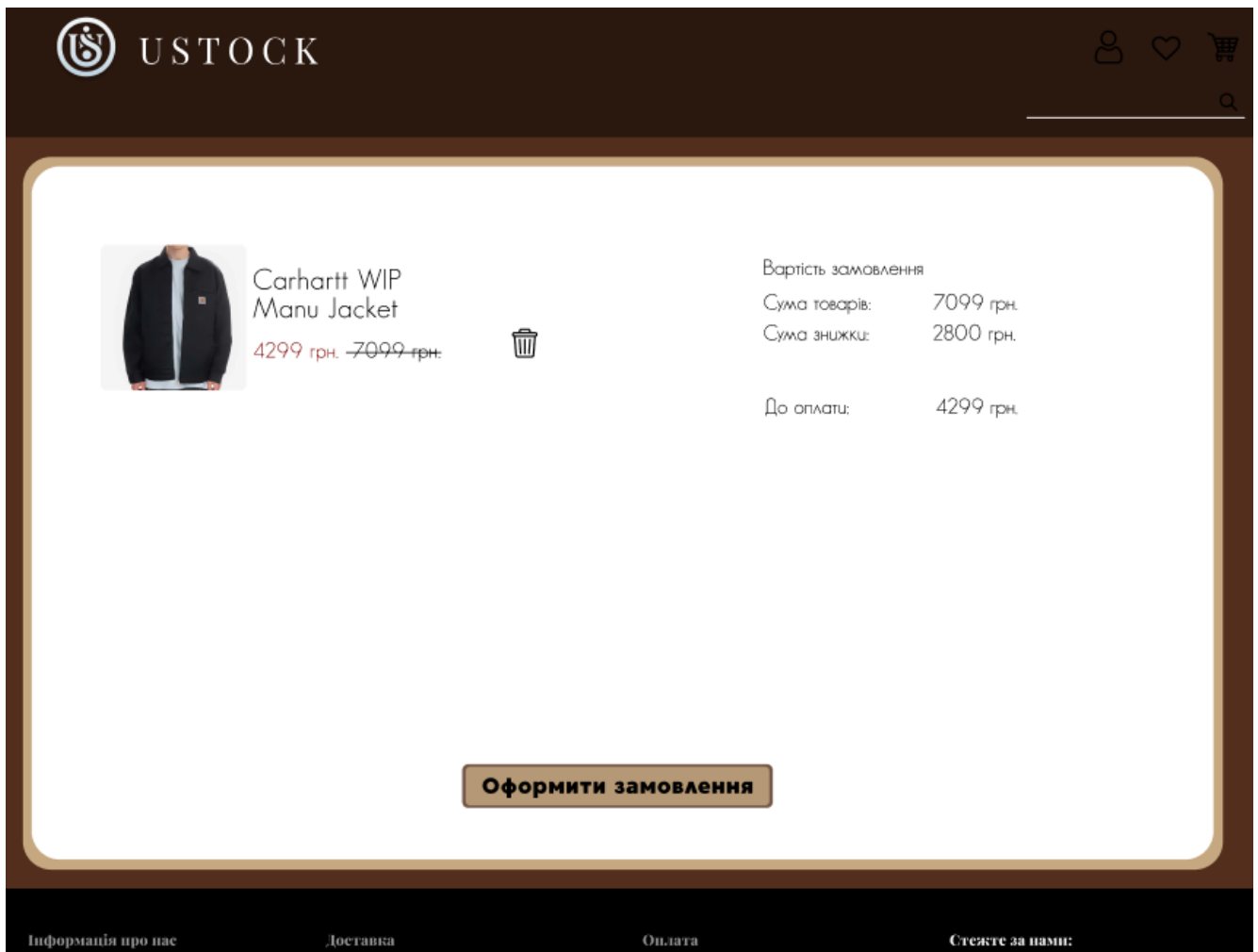


Рисунок 3.9 Сторінка кошику

На сторінці «Оформлення замовлення». Користувач може побачити свої дані, які він вводив при реєстрації та вартість замовлення. Користувач може обрати варіант доставки та оплати. Також користувач може ввести промокод на знижку. Обравши варіант доставки та оплати, користувач повинен натиснути на кнопку купити, після чого на електронну пошту (яку користувач ввів при реєстрації) приходить повідомлення, що його замовлення відправлене в обробку. Якщо, замовлення прийнято або відхилено, користувачеві на електронну пошту приходить повідомлення, що його товар укомплектовують (у випадку, якщо замовлення прийнято). Згодом, користувачеві приходить повідомлення, що його товар був переданий в службу доставки.

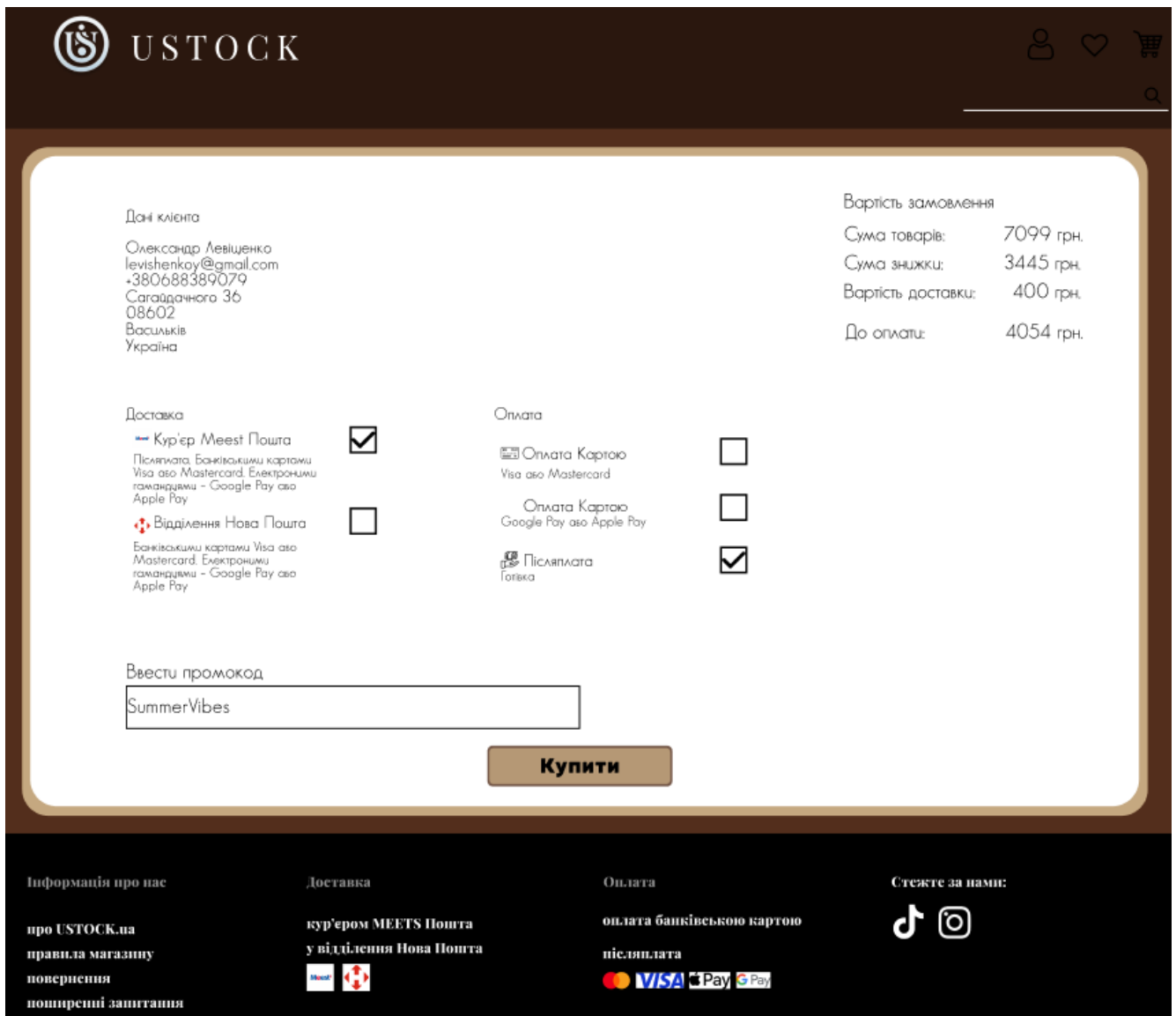


Рисунок 3.10 Сторінка «Оформлення замовлення»

### 3.4.2 Сторінка акаунту

Ми розглянули типовий сценарій рядового користувача. Тепер перейдемо до функціоналу, що не описувався в рамках сценарію.

Якщо користувач зайшов в свій акаунт і натиснув на кнопку «акаунт», його посилас на сторінку акаунту. На ній клієнт зможе:

- переглянути та змінити свої дані
- переглянути замовлення, які вже були виконані або виконуються
- зв'язатися зі службою підтримки

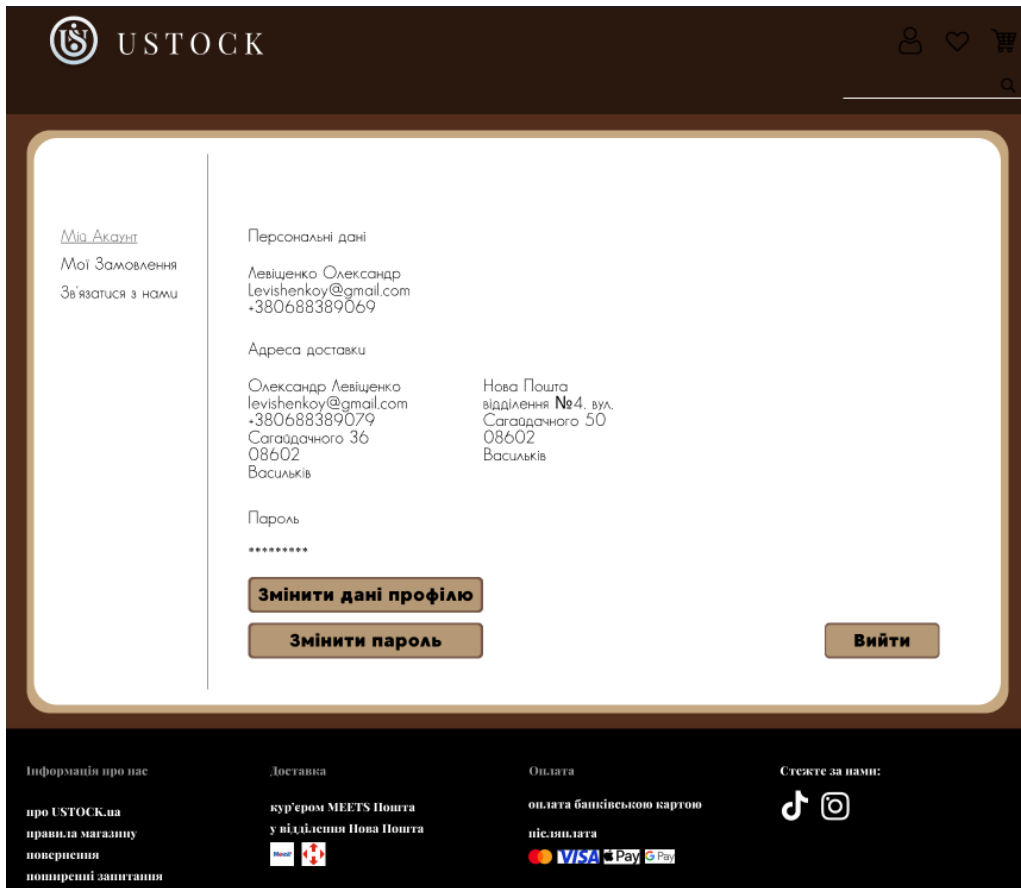


Рисунок 3.11 Сторінка «Акаунт» відділ «Мій акаунт»

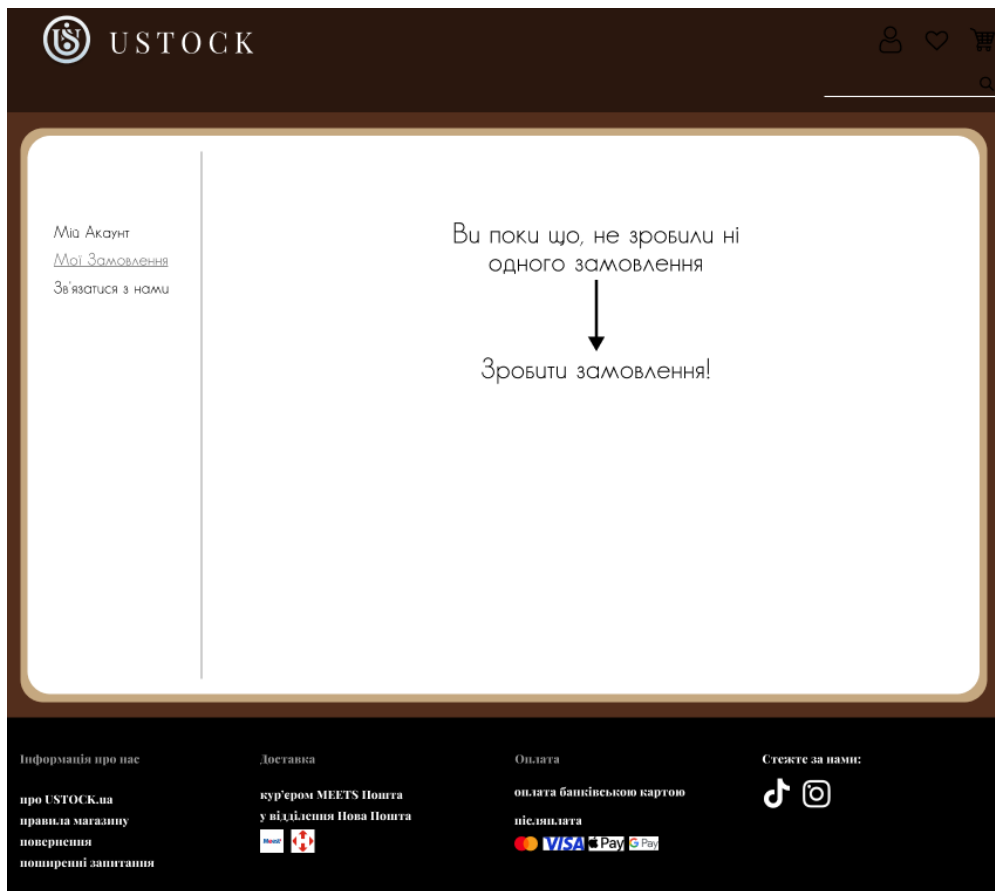


Рисунок 3.12 Сторінка «Акаунт» відділ «Мої замовлення»

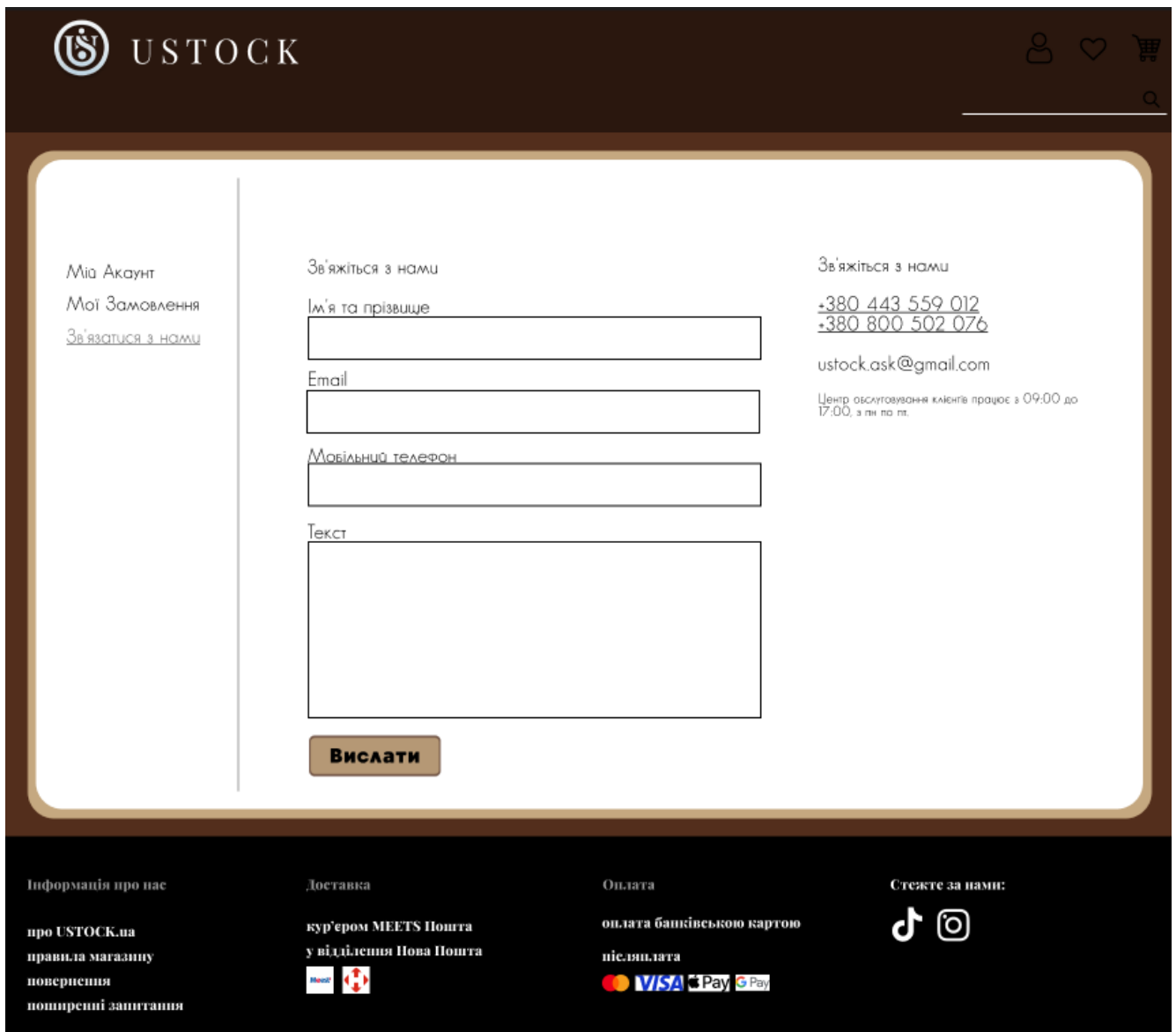


Рисунок 3.13 Сторінка «Акаунт» відділ «Зв'язок зі службою підтримки»

### 3.4.3 Пошук товару

В сценарії у користувача було 3 варіанти пошуку товару. Тепер ми розглянемо варіанти, які не були розглянуті в сценарії.

Користувач може відсортувати товар. Для цього користувач повинен навести курсор миші на розділ «для нього», «для неї» або «аксесуари». Як тільки юзер наведе курсор миші на один з цих розділів, йому відкриється вікно, в якому він зможе обрати тип товару. Натиснувши на тип товару, користувача перекине на сторінку з відсортованими товарами.



Рисунок 3.14 Сортування товарів

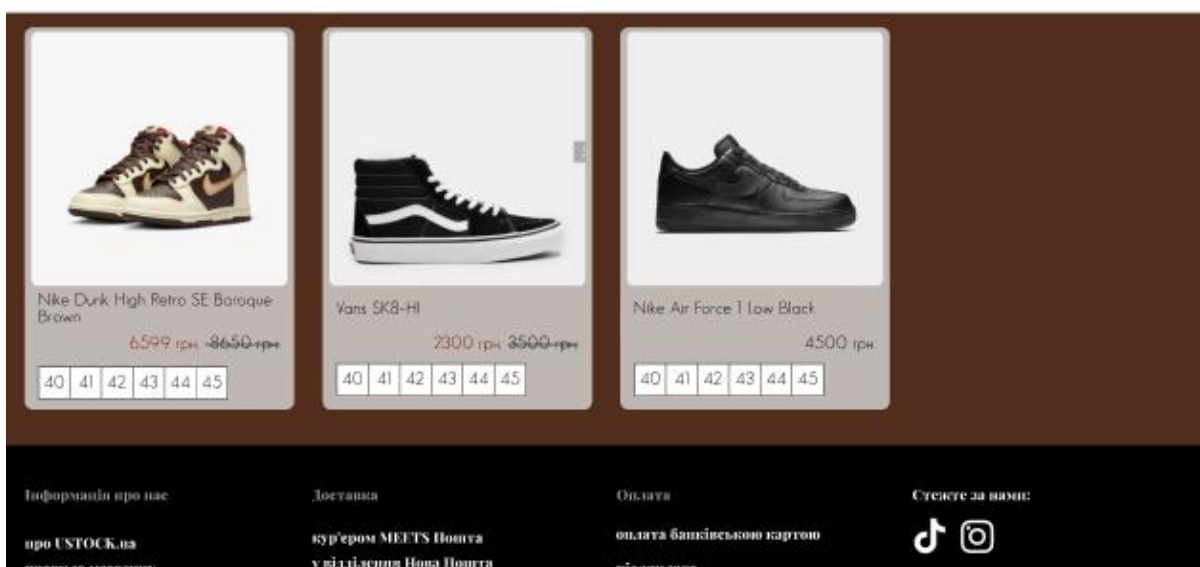


Рисунок 3.15 Головна сторінка з відсортованими товарами

Також користувач може ввести в рядку пошуку назву товару, якщо є співпадіння по назві відкриється вікно в якому будуть відображенні товари, що співпадають по назві.



Рисунок 3.16 Пошук товару через рядок пошуку

#### 3.4.4 Вподобання

Тепер ми розглянемо останню функцію яка буде доступна користувачеві, а саме список вподобань. Користувач зможе додати товар до вподобань, щоб йому не доводилося витратити зайвий час на пошук бажаного товару.

Опинившись на сторінці вподобань, користувач баче товар який був добавлений до списку, він може перейти на сторінку з товаром або видалити його зі списку вподобань.

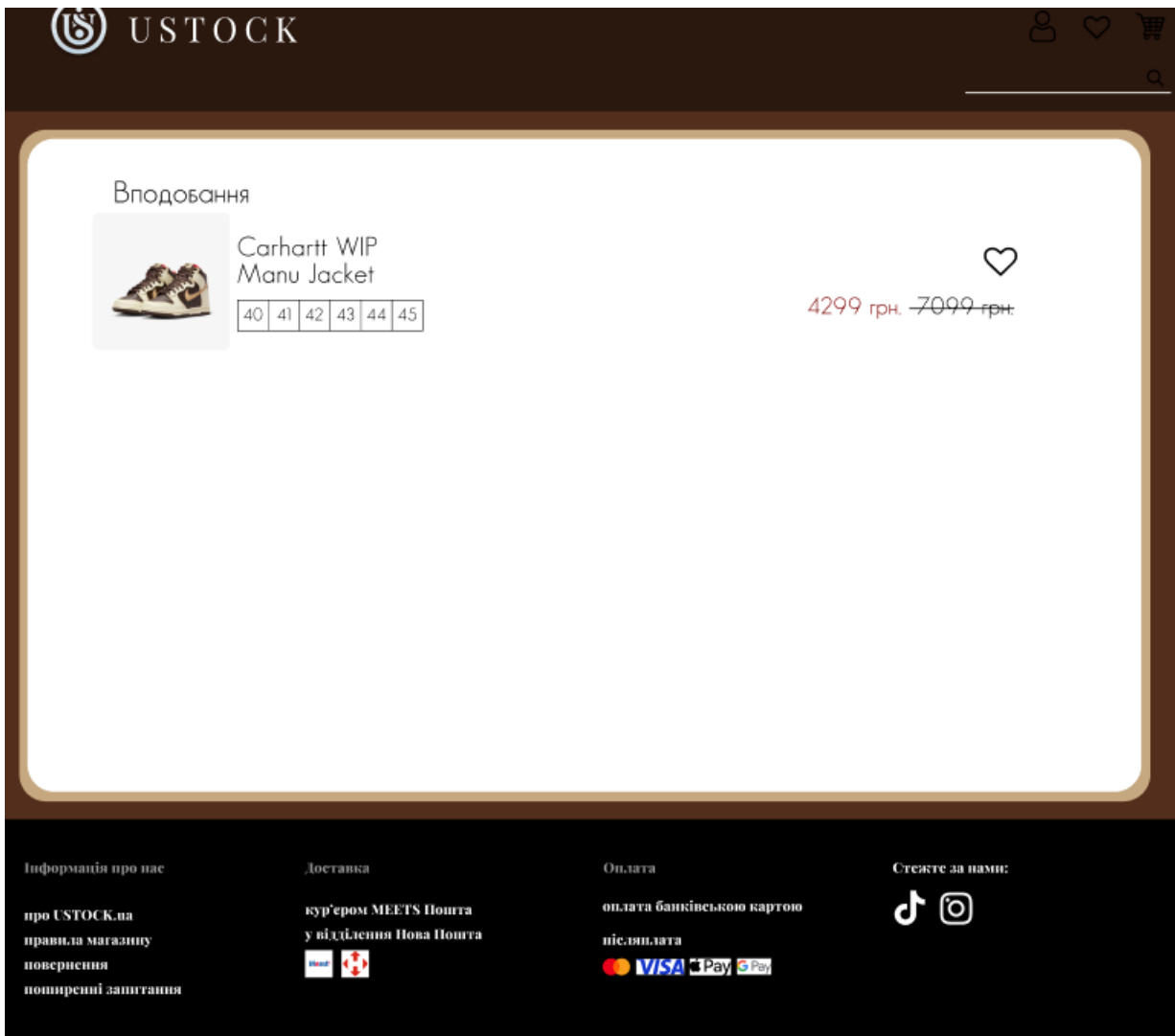


Рисунок 3.17 Сторінка вподобань

### 3.5 Висновки до розділу

- Був розглянутий процес розробки прототипу інтерфейсу
- Був розглянутий процес релізації фронтендної частини
- Було розглянуто реалізацію основних компонентів інтернет-магазину

#### Ustock

- Була розглянута процедура реєстрації та входу користувача на сайт
- Була розглянута процедура реалізації зміни даних користувача
- Була розглянута процедура реалізації пошуку та сортування товару на сайті
- Було розглянуто базу даних та таблиць які використовуються в ній
- Були розглянуті функціональні можливості, які доступні рядовому користувачеві

## РОЗДІЛ 4 УПРАВЛІННЯ ПРОЄКТОМ

### 4.1 Загальний огляд проєкту

Перш ніж почати цей підрозділ, нам необхідно визначити що з себе представляє управління проєктом та сам проєкт.

Управління проєктами - область знань з планування, організації та управління ресурсами з метою успішного досягнення цілей та завершення завдань проєкту.

Проєкт — це обмежений часовими рамками процес, що має визначений початок та кінець, зазвичай обмежений датою, але також може обмежуватися фінансуванням або досягненням результатів, який здійснюється для реалізації унікальних цілей та завдань, зазвичай, щоб призвести до вигідних змін або створення доданої вартості. Тимчасова природа проєктів контрастує з бізнесом (процесами), які є повторюваною, постійною або частково постійною діяльністю з виробництва продуктів або послуг. На практиці, управління вищезазначеними двома системами часто різняться і таким чином вимагає розвитку окремих технічних навичок та використання розподіленого управління ними.

Головним завданням проєктного управління є досягнення всіх цілей та виконання завдань проєкту, одночасно виконуючи зобов'язання щодо наперед визначених обмежень проєкту. Типовими обмеженнями є межі та зміст проєкту, час, бюджет. Другорядним завданням, але більш амбітним, є оптимізація, розподілення та інтеграція завдань, необхідних для досягнення наперед визначених цілей.

Тепер ми можемо перейти до проєкту який виконувався в рамках даної дипломної роботи. Наш проєкт це управління розробкою Інтернет-магазину одягу та мобільного застосунку до нього, даний проєкт був описаний в Microsoft Project Pro. Нам необхідно визначити бюджет проєкту, час необхідний для розробки та ресурси, які включають в себе як людські так і матеріальні ресурси. В наступних підрозділах ми розглянемо ці аспекти більш детально.

Варто зауважити, що розробка мобільного додатку це опціональне рішення, що спрямоване на підвищення популярності Інтернет-магазину. Через 2 місяці пі-

сля запуску веб-сайту буде проведений аналіз активності користувачів. Якщо кількість користувачів або замовлень на веб-сайті не буде відповідати цілям, то буде прийняте рішення про створення мобільного додатку, що може залучити нову аудиторію та зробити Інтернет-магазин більш конкурентоспроможним.

## 4.2 Огляд задач проєкту

В даному підрозділі ми більш детально розглянемо які задачі нам необхідно виконати для створення веб-сайту та мобільного додатку. Всі завдання були поділені на групи, так як в даному форматі легше орієнтуватись.

На першому рисунку (4.1) ми бачимо дві групи завдань, а саме: ініціація та планування. Ці завдання необхідні для планування часу, бюджету, ресурсів, ризиків, якості та комунікації співробітників.

Спочатку розглянемо групу «Ініціація», в даній групі є два завдання – це визначення зацікавлених сторін або стейкхолдерів та розробка статуту проєкту. Для виконання першого завдання нам треба 8 днів, а на друге завдання – 26. Тобто, в сумі на виконання першої групи завдань нам треба виділити 34 дні.

Далі розглянемо групу «Планування», в дана група включає в себе 8 завдань. Це документи вимог, обсяг проєкту, бюджет проєкту, план якості, план людських ресурсів, план комунікацій, план управління ризиками та план управління проєктом. Виконання першого завдання потребує 44 дні. Обсяг проєкту – 12. Наступні завдання, а саме бюджет проєкту, план якості, план людських ресурсів, план комунікацій та план управління ризиками, можуть виконуватись паралельно так як кожним завданням займається свій спеціаліст та вони незалежать один від одного. На виконання цих завдань потребується виділити 20 днів. Останнє завдання в цій групі – план управління проєктом, потребує виконання всіх вище зазначених завдань та вимагає 12 днів на виконання. Виконання цієї групи завдань потребує 88 днів.

2	→	▸ Ініціювання	34 днів	01.04.24	16.05.24	
3	→	Визначення зацікавлених	8 днів	01.04.24	10.04.24	
4	→	Статут проєкту	26 днів	11.04.24	16.05.24	3
5	→	▸ Планування	88 днів	17.05.24	17.09.24	
6	→	Документи вимог	44 днів	17.05.24	17.07.24	4
7	→	Обсяг проєкту	12 днів	18.07.24	02.08.24	6
8	→	Бюджет проєкту	13 днів	05.08.24	21.08.24	7
9	→	План якості	20 днів	05.08.24	30.08.24	7
10	→	План людських ресурсів	14 днів	05.08.24	22.08.24	7
11	→	План комунікацій	8 днів	05.08.24	14.08.24	7
12	→	План управління ризиками	18 днів	05.08.24	28.08.24	7
13	→	План управління проектом	12 днів	02.09.24	17.09.24	7;8;9;10;11;12

Рисунок 4.1 Ініціювання та планування

На рисунку 4.2 ми бачимо дві групи завдань – це розробка веб-сайту та його запуск. На виконання першої групи нам треба 195 днів, а для другої 2 дні.

Перша група завдань складається з двох підгруп, а саме розробка дизайну та розробка веб-сайту. Ці дві підгрупи будуть виконуватись паралельно, так як над завданнями будуть працювати команди різних спеціальностей.

Почнемо з першої підгрупи. Вона в себе включає два завдання які виконуються послідовно – це розробка концепту та розробка дизайну. Виконання першого завдання вимагає 10 днів. Виконання другого завдання потребує 30 днів. Тобто виконання першої підгрупи вимагає 40днів.

Перейдемо до другої підгрупи, вона містить в собі два завдання які будуть виконуватись паралельно, а саме: розробка фронтендної частини та розробка бекендної частини. Розробка фронтенду вимагає 150 днів, розробка бекенду – 190 днів. Виконання цієї підгрупи займає 190 днів. Через те, що підгрупи виконуються паралельно, час виконання цієї групи завдань співпадає з часом виконання другої підгрупи, тобто 190 днів.

Наступна група завдань це тестування. Вона включає в себе всього одне завдання тестування та виправлення помилок. Виконання цієї підгрупи вимагає 60 днів.

Наступна група завдань це запуск веб-сайту, вона включає в себе 3 завдання, які виконуються послідовно – це вибір та реєстрація доменного імені (3 дні), вибір та реєстрація хостинг-провайдера (3 дні) та реєстрація веб-сайту в пошукових системах (3 дні). На виконання цих трьох завдань необхідно виділити 9 днів.

Після виконання цієї групи завдань, наш Інтернет-магазин готовий до роботи.

14	→	▲ Розробка веб-сайту	643 днів	18.09.24	05.03.27	
15	→	▲ Дизайн веб-сайту	40 днів	18.09.24	12.11.24	
16	→	Концепт	10 днів	18.09.24	01.10.24	13
17	→	Розробка	30 днів	02.10.24	12.11.24	16
18	→	▲ Розробка веб-сайту	190 днів	18.09.24	10.06.25	
19	→	Розробка фронтенду	150 днів	18.09.24	15.04.25	13
20	→	Розробка бекенду	190 днів	18.09.24	10.06.25	13
21	→	▲ Тестування	453 днів	11.06.25	05.03.27	
22	→	Тестування та виправлення помилок	60 днів	11.06.25	02.09.25	20
23	→	▲ Запуск веб-сайту	9 днів	03.09.25	15.09.25	
24	→	Вибір та реєстрація доменного	3 днів	03.09.25	05.09.25	22
25	→	Вибір та реєстрація	3 днів	08.09.25	10.09.25	24
26	→	Реєстрація веб-сайту в пошукових системах	3 днів	11.09.25	15.09.25	25

Рисунок 4.2 Розробка веб-сайту та запуск веб-сайту

Через 2 місяці нам треба зробити аналіз успішності нашого веб-сайту, щоб зрозуміти, чи треба робити мобільний додаток, чи ні. Дані аналізи ми будемо порівнювати з цілями, які ми розглянемо в підрозділі 4.6. Якщо, цілі будуть виконані та веб-сайт буде достатньо популярним в розробці мобільного додатку, не буде сенсу і можна буде переходити до закриття проєкту.

На третьому рисунку (4.3) в нас буде одна група – аналіз успішності. Завдання в цій групі будуть виконуватись послідовно, окрім першого та другого. На

виконання першого та другого завдання – це веб-аналітика та SEO-аналіз, нам необхідно виділити 10 днів. Наступні завдання це аналіз відгуків користувачів (4 дні), аналіз конверсії (3 дні), аналіз продуктивності (6 днів) та конкурентний аналіз (6 днів). На виконання цих завдань нам треба виділити 16 днів. Тобто, виконання цієї групи завдань вимагає 26 днів.

26	→	▲ Аналіз успішності	26 днів	12.09.25	17.10.25	
27	→	Веб-аналітика	10 днів	12.09.25	25.09.25	25FS+60 днів
28	→	SEO-аналіз	10 днів	12.09.25	25.09.25	25FS+60 днів
29	→	Аналіз відгуків користувачів	4 днів	26.09.25	01.10.25	28
30	→	Аналіз конверсії	3 днів	02.10.25	06.10.25	29
31	→	Аналіз продуктивності	6 днів	07.10.25	14.10.25	30
32	→	Конкурентний аналіз	3 днів	15.10.25	17.10.25	31

Рисунок 4.3 Аналіз успішності

Далі, в залежності від виконаних цілей компанії, буде прийняте рішення розробляти мобільний застосунок чи ні. Ми візьмемо найгірший сценарій – Інтернет-магазин не зміг привабити необхідну аудиторію та виконати цілі. В результаті було прийняте рішення про створення мобільного додатку, що може покращити становище.

На рисунку 4.4 ми бачимо дві групи завдань – це розробка мобільного додатку та його реліз. В свою чергу, розробка мобільного додатку буде складатися з двох підгруп, а саме: розробка дизайну та розробка додатку.

Почнемо з розробки мобільного додатку, а конкретніше з першої підгрупи. Вона в себе включає два завдання які виконуються послідовно – це розробка концепту та розробка дизайну. Виконання першого завдання вимагає 10 днів. Виконання другого завдання потребує 30 днів. Тобто виконання першої підгрупи вимагає 40днів.

Перейдемо до другої підгрупи, вона містить в собі два завдання які будуть виконуватись паралельно, а саме: розробка фронтендної частини та розробка бекендної частини та тестування. Розробка фронтенду вимагає 180 днів, розробка бекенду – 210 днів.

Наступна група завдань це тестування. Вона включає в себе всього одне завдання тестування та виправлення помилок. Виконання цієї підгрупи вимагає 75 днів.

Виконання цієї підгрупи займає 210 днів. Через те, що підгрупи виконується паралельно, час виконання цієї групи завдань співпадає з часом виконання другої підгрупи, тобто 210 днів.

Наступна група завдань це запуск проєкту, вона включає в себе лише одне завдання – це реліз мобільного додатку на торгові майданчики, а саме App Store та Google Play. Це завдання вимагає 2 дні на своє виконання.


34		▲ Розробка мобільного додатку	298 днів	14.01.26	05.03.27	
35		▲ Розробка дизайну	298 днів	14.01.26	05.03.27	
36		Концепт	10 днів	14.01.26	27.01.26	33
37		Розробка	30 днів	28.01.26	10.03.26	36
38		▲ Розробка додатку	210 днів	14.01.26	03.11.26	
39		Розробка фронтенду	180 днів	14.01.26	22.09.26	33
40		Розробка бекенду	210 днів	14.01.26	03.11.26	33
41		▲ Тестування	75 днів	04.11.26	16.02.27	
42		Тестування та виправлення помилок	75 днів	04.11.26	16.02.27	40
43		▲ Реліз мобільного додатку	2 днів	17.02.27	18.02.27	
44		Реліз мобільного додатку	2 днів	17.02.27	18.02.27	42

Рисунок 4.4 розробка мобільного додатку та його реліз

Після релізу мобільного додатку ми можемо переходити до останньої групи завдань, а саме закриття проєкту, який ми бачимо на рисунок 4.5. Ця група завдань буде в себе включати 5 завдань, які будуть виконуватись послідовно. Перше завдання це остаточне прийняття проєкту, на його виконання необхідно виділити 3

дні. Друге завдання це оцінка задоволеності зацікавлених сторін на його виконання необхідно виділити 1 день. Наступне завдання – це закриття запису проєкту, на це завдання можна виділити 1 день. Наступне завдання – це перегляд і узгодження фінансових показників, це завдання вимагає 3 дні. І, нарешті, останнє завдання – це оформлення договору субпідрядника, на це завдання треба виділити 3 дні. Після цього завдання наш проєкт завершений.

43		▲ Закриття	11 днів	04.09.26	18.09.26	
44		Остаточне прийняття проєкту	3 днів	04.09.26	08.09.26	42
45		Оцінка задоволеності зацікавлених сторін	1 день	09.09.26	09.09.26	44
46		Закрити записи проєкту	1 день	10.09.26	10.09.26	45
47		Перегляд і узгодження фінансових показників	3 днів	11.09.26	15.09.26	46
48		Договір субпідрядника	3 днів	16.09.26	18.09.26	47
49			Завершення проєкту	0 днів	16.09.26	16.09.26

Рисунок 4.5 закриття проєкту

Для виконання проєкту, якщо ми не будемо розробляти мобільний додаток, нам необхідно виділити 435 днів. Проте, якщо веб-сайт не виконає цілі та нам необхідно буде розробляти мобільний додаток, кількість днів буде збільшено на 210. Тобто на проєкт необхідно буде виділити 645 днів.

### 4.3 Огляд ресурсів проєкту

В цьому підрозділі ми будемо розглядати ресурси та бюджет нашого проєкту. Ми будемо розглядати кожен ресурс (незалежно робочий чи матеріальний ресурс), його ціну або ставку та яку роль він відіграє в нашому проєкті. Також ми розглянемо витрати, які будуть на кожному завданні.

Почнемо з людських ресурсів. Необхідно звернути увагу на те, що в нас 8-годинний робочий день і в робочого ресурса в нас плата погодинна. На рисунках 4.6 та 4.7 в нас будуть наведені людські ресурси, їхня ставка та понаднормова ставка.

	Ім'я ресурсу	Тип	Одиниця вимірювання матеріалів	Ініціали	Група	Макс. одиниць	Звич. ставка	Понад. ставка	Витрати	Нараху	Основний календар
1	Керівник проекту	Робота		К		100%	227,00 €/год	244,00 €/год	0,00 €	Пропорц	Standard
2	Проектний менеджер	Робота		П		100%	198,00 €/год	215,00 €/год	0,00 €	Пропорц	Standard
3	Менежер з управління ризиками	Робота		М		100%	187,00 €/год	204,00 €/год	0,00 €	Пропорц	Standard
4	Фінансовий менежер	Робота		Ф		100%	204,00 €/год	221,00 €/год	0,00 €	Пропорц	Standard
5	Менежер з управління ресурсів	Робота		М		100%	193,00 €/год	210,00 €/год	0,00 €	Пропорц	Standard
6	Менежер з управління якості	Робота		М		100%	181,00 €/год	198,00 €/год	0,00 €	Пропорц	Standard
7	Аналітик	Робота		А		100%	230,00 €/год	260,00 €/год	0,00 €	Пропорц	Standard

Рисунок 4.6 аркуш ресурсів

7	Аналітик	Робота		А		100%	230,00 €/год	260,00 €/год	0,00 €	Пропорц	Standard
8	Команда дизайнерів 1	Робота		К		100%	511,00 €/год	596,00 €/год	0,00 €	Пропорц	Standard
9	Команда фронтенд розробників 1	Робота		К		100%	1 135,00 €/год	1 250,00 €/год	0,00 €	Пропорц	Standard
10	Команда бекенд розробників 1	Робота		К		100%	1 278,00 €/год	1 360,00 €/год	0,00 €	Пропорц	Standard
11	Команда тестувальників 1	Робота		К		100%	339,00 €/год	408,00 €/год	0,00 €	Пропорц	Standard
12	Команда дизайнерів 2	Робота		К		100%	562,00 €/год	630,00 €/год	0,00 €	Пропорц	Standard
13	Команда фронтенд розробників 2	Робота		К		100%	1 250,00 €/год	1 335,00 €/год	0,00 €	Пропорц	Standard
14	Команда бекенд розробників 2	Робота		К		100%	1 306,00 €/год	1 392,00 €/год	0,00 €	Пропорц	Standard
15	Команда тестувальників 2	Робота		К		100%	392,00 €/год	426,00 €/год	0,00 €	Пропорц	Standard

Рисунок 4.7 аркуш ресурсів

1. Керівник проекту: його звичайна ставка 227 грн/год, а понаднормова ставка – 244 грн/год. Він приймає участь в таких завданнях: визначення зацікавлених сторін, документи вимог, вибір та реєстрація доменого імені, вибір та реєстрація хостинг-провайдера, реєстрація веб-сайту в пошукових системах,

- реліз мобільного додатку, остаточне прийняття проєкту, оцінка задоволеності зацікавлених сторін, закриття записів проєкту, перегляд і узгодження фінансових показників та оформлення договору субпідрядника.
2. Проєктний менежер: його звичайна ставка 198 грн/год, а понаднормова ставка – 215 грн/год. Він приймає участь в таких завданнях: визначення зацікавлених сторін, статут проєкту, документи вимог, обсяг проєкту, план комунікацій, план управління проєктом, остаточне прийняття проєкту і оцінка задоволеності зацікавлених сторін.
  3. Менежер з управління ризиками: його звичайна ставка 187 грн/год, а понаднормова ставка – 204 грн/год. Він приймає участь лише в одному завданні – план управління ризиками
  4. Фінансовий менежер: його звичайна ставка 204 грн/год, а понаднормова ставка – 221 грн/год. Він приймає участь в таких завданнях: бюджет проєкту та перегляд і узгодження фінансових показників.
  5. Менежер з управління ресурсами: його звичайна ставка 193 грн/год, а понаднормова ставка – 210 грн/год. Він приймає участь в таких завданнях: план людських ресурсів та план комунікацій.
  6. Менежер з якості: його звичайна ставка 181 грн/год, а понаднормова ставка – 198 грн/год. Він приймає участь лише в одному завданні – план якості.
  7. Аналітик: його звичайна ставка 230 грн/год, а понаднормова ставка – 260 грн/год. Він приймає участь в таких завданнях: веб-аналітика, SEO-аналіз, аналіз відгуків користувачів, аналіз продуктивності, конкурентний аналіз та оцінка задоволеності зацікавлених сторін
  8. Команда дизайнерів 1: її звичайна ставка 511 грн/год, а понаднормова ставка – 596 грн/год. Варто зауважити, що ця команда складає з себе 3-ьох дизайнерів. Вони приймають участь в таких завданнях: розробка концепту веб-сайту та розробка дизайну веб-сайту.

9. Команда дизайнерів 2: її звичайна ставка 562 грн/год, а понаднормова ставка – 630 грн/год. Варто зауважити, що ця команда складає з себе 3-ьох дизайнерів. Вони приймають участь в таких завданнях: розробка концепту мобільного застосунку та розробка дизайну мобільного застосунку.
10. Команда фронтенд розробників 1: її звичайна ставка 1135 грн/год, а понаднормова ставка – 1250 грн/год. Варто зауважити, що ця команда складає з себе 5-ьох розробників. Вони приймають участь в одній задачі – це розробка фронтендної частини веб-сайту.
11. Команда фронтенд розробників 2: її звичайна ставка 1250 грн/год, а понаднормова ставка – 1335 грн/год. Варто зауважити, що ця команда складає з себе 5-ьох розробників. Вони приймають участь в одній задачі – це розробка фронтендної частини мобільного застосунку.
12. Команда бекенд розробників 1: її звичайна ставка 1278 грн/год, а понаднормова ставка – 1360 грн/год. Варто зауважити, що ця команда складає з себе 5-ьох розробників. Вони приймають участь в одній задачі – це розробка бекендної частини веб-сайту.
13. Команда бекенд розробників 2: її звичайна ставка 1306 грн/год, а понаднормова ставка – 1392 грн/год. Варто зауважити, що ця команда складає з себе 5-ьох розробників. Вони приймають участь в одній задачі – це розробка бекендної частини мобільного застосунку.
14. Команда тестувальників 1: її звичайна ставка 339 грн/год, а понаднормова ставка – 408 грн/год. Варто зауважити, що ця команда складає з себе 2-ьох тестувальників. Вони приймають участь в одній задачі – це тестування веб-сайту.
15. Команда тестувальників 1: її звичайна ставка 392 грн/год, а понаднормова ставка – 426 грн/год. Варто зауважити, що ця команда складає з себе 2-ьох тестувальників. Вони приймають участь в одній задачі – це тестування мобільного додатку.

Тепер ми можемо перейти до огляду матеріальних ресурсів. Варто зауважити, що ми не будемо орендувати офіс та купувати необхідну техніку, кожен співробітник буде працювати зі свого пристрою віддалено. Тобто єдині витрати це програмне забезпечення для розробки веб-сайту та мобільного додатку.

1. Програмне забезпечення яке ми будемо використовувати в розробці веб-сайту буде коштувати 130000 грн. (ця сума розрахована на 5 розробників, тобто, одна цифрова копія ПЗ коштує 26000 грн)
2. Програмне забезпечення яке ми будемо використовувати в розробці мобільного додатку є безкоштовним продуктом.
3. Підписка на програмне забезпечення яке ми будемо використовувати для розробки дизайну буде коштувати 24000 грн на рік. (ця сума розрахована на 3 дизайнерів, тобто, одна цифрова копія ПЗ коштує 8000 грн)

Витрати на даний проєкт складають 10048248 грн, це при тих умовах, якщо веб-сайт не виконає цілі які будуть наведені в підрозділі 4.6, але, якщо веб-сайт буде популярним і цілі виконані, то в розробці мобільного застосунку не буде сенсу і він не буде розроблюватись. Ціна виходить 4818848 грн. Проте нам варто врахувати обставини, які можуть збільшити час розробки веб-сайту або мобільного додатку. Тобто витрати також збільшаться. Зазвичай, на обставини, які збільшать вартість розробки додають 30% від витрат, тож наш бюджет – 13019760 грн.

#### **4.4 Аналіз зацікавлених сторін**

Аналіз зацікавлених сторін є ключовим елементом управління проєктами, оскільки він допомагає визначити, хто впливає на проєкт і на кого впливає проєкт. Є два види зацікавлених сторін: внутрішні та зовнішні зацікавлені сторони. Різниця між внутрішніми та зовнішніми зацікавленими сторонами полягає в їхньому положенні та ролі щодо організації та проєкту. Далі ми розглянемо зацікавлені сторони та їхні цілі в проєкті.

Внутрішні зацікавлені сторони:

**Керівник проєкту (власник):** Фінансовий успіх проєкту, розвиток та позиціонування на ринку.

**Команда менеджерів:** Вчасне та успішне виконання проєкту в межах бюджету та графіку, ефективна комунікація з командою та стейкхолдерами.

**Команда проєкту:** можливість для творчого вираження та розвитку професійних навичок, успішна реалізація ідеї та внесок у створення конкурентоспроможного продукту.

Зовнішнє оточення проєкту:

**Конкуренти:** збільшення своєї конкурентної переваги, моніторинг стратегій та нововведень конкурентів, реакція на зміни в ринковому середовищі.

**Держава:** розвиток малого та середнього бізнесу, підтримка електронної комерції та інноваційних проєктів, забезпечення прав споживачів та дотримання законодавства.

**Постачальник одягу:** збільшення обсягів продажу через новий канал збуту, партнерство з успішним інтернет-магазином, можливість просування власного бренду через інтернет магазин.

**Соціальні мережі:** збільшення аудиторії та залучення нових клієнтів через рекламні кампанії.

**Клієнти:** закупівля якісного одягу за демократичним цінами.

#### 4.5 SWOT-аналіз

SWOT-аналіз (Strengths, Weaknesses, Opportunities, Threats) є важливим інструментом стратегічного планування і може бути використаний у різних контекстах, включаючи управління проєктами, розробку бізнес-стратегії, маркетинг, та інші сфери. В рамках даної дипломної роботи ми дізнаємось про сильні та слабкі сторони нашого Інтернет-магазину, а також можливості та загрози, які можуть виникнути.

**SWOT-аналіз проєкту**

Сильні сторони	Слабкі сторони
<ol style="list-style-type: none"> <li>1. Великий потенціал клієнтської бази: Зацікавленість багатьох клієнтів у покупці одягу через інтернет.</li> <li>2. Доступ до якісного одягу: Здатність співпрацювати з надійними постачальниками одягу забезпечить високу якість продукції для клієнтів.</li> <li>3. Маркетинговий потенціал соціальних мереж: Здатність використовувати соціальні мережі для просування інтернет магазину та привертання клієнтів.</li> <li>4. Досвідна команда проєкту: Наявність кваліфікованої команди з досвідом у розробці та управлінні інтернет-проєктами.</li> </ol>	<ol style="list-style-type: none"> <li>1. Високий рівень конкуренції: Інтенсивна конкуренція в галузі одягу та електронної комерції може складати труднощі в просуванні.</li> <li>2. Технічні обмеження: Можливі труднощі з технічною реалізацією інтернет-магазину та додатку.</li> <li>3. Залежність від постачальників: Залежність від постачальників одягу може вплинути на наявність товарів та цінову політику.</li> <li>4. Недостатня усвідомленість: Інтернет магазин може потребувати значних зусиль для побудови своєї репутації та визнання серед клієнтів.</li> </ol>
Можливості:	Загрози:
<ol style="list-style-type: none"> <li>1. Розширення асортименту: Можливість розширення асортименту товарів та послуг для задоволення потреб різних клієнтів.</li> <li>2. Розвиток мобільного додатку: Розробка та впровадження мобільного додатку може підвищити зручність для клієнтів та збільшити їхню лояльність.</li> <li>3. Використання аналітики даних: Використання даних про клієнтів для вдосконалення маркетингових стратегій та асортименту товарів.</li> <li>4. Експансія на нові ринки: Можливість розширення бізнесу на нові географічні ринки або використання нових каналів збуту.</li> </ol>	<ol style="list-style-type: none"> <li>1. Конкуренція з боку великих гравців: Змагання з великими онлайн-ритейлерами може ускладнити просування бренду та здобуття частки ринку.</li> <li>2. Технологічні загрози: Зміни у технологіях або нові технологічні тенденції можуть потребувати постійного оновлення інтернет-магазину та додатку.</li> <li>3. Зміни в економіці: Економічні коливання, зростання вартості ресурсів або зміни курсів валют можуть впливати на витрати та прибуток проєкту.</li> <li>4. Проблеми з безпекою даних: Загрози стосовно кібербезпеки та захисту персональних даних клієнтів можуть вплинути на довіру споживачів до інтернет-магазину.</li> </ol>

Таблиця 4.1 SWOT-аналіз

**4.6 Цілі компанії**

В даному підрозділі ми розглянемо цілі, які повинен виконати веб-сайт впродовж 2 місяців після запуску. Якщо цілі будуть виконанні, то ми не будемо розробляти мобільний додаток. Проте, якщо веб-сайт не буде популярним та цілі, які

будуть наведені нижче не будуть виконані, то буде прийняте рішення про створення мобільного додатку.

1. Забезпечення задоволення клієнтів:

- Мінімізувати кількість незадоволених клієнтів.
- Відстежувати негативні відгуки через коментарі.
- Відповідати на коментарі (як на позитивні, так і на негативні).
- Подальші цілі будуть впливати на дану ціль.
- Забезпечити співвідношення позитивних до негативних коментарів як 3/1, і краще

2. Постійне вдосконалення:

- Проводити аналіз відгуків клієнтів раз в 2 неділі та впроваджувати необхідні зміни для поліпшення якості.
- Раз в місяць оновлювати мобільний додаток для забезпечення сучасних функцій та високої продуктивності.
- QA-тестувальники під час розроблення будуть тестувати нововведення та «фікси»
- На закінченні проекту реалізувати технічний потенціал продукту

3. Збільшення кількості користувачів:

- Забезпечити аудиторію сайту до 2000 зареєстрованих користувачів
- Дана ціль забезпечується достатньо ефективною рекламною кампанією

4. Збільшення замовлень на сайті:

- Продати товар на суму 300000 грн.
- Дана ціль забезпечується достатньо ефективною рекламною кампанією, широким асортиментом товарів та демократичними цінами.

5. Дотримання вимог та стандартів:

- Забезпечувати відповідність усіх процесів і виробів вимогам та стандартам якості.
- Проводити планові систематичні перевірки для переконання у дотриманні встановлених стандартів.

#### **4.7 Висновки до розділу**

- Створено проєкт в MS Project Pro
- Визначено час, який необхідно на виконання проєкту та розглянули його задачі
- Розглянуто ресурси, які необхідні для виконання проєкту. Обчислено витрати та встановили бюджет проєкту
- Виконано аналіз зацікавлених сторін та стало відомо хто зможе вплинути на проєкт.
- Зроблено SWOT-аналіз, щоб дізнатись про сильні та слабкі сторони проєкту
- Встановлено цілі, які повинен виконати веб-сайт після релізу

## ЗАГАЛЬНІ ВИСНОВКИ

Завданням даної дипломної роботи було управління проектом розробки Інтернет-магазину одягу Ustock. Розробка була поділена на кілька етапів. Для створення прототипу веб-сайту ми використовували додаток Figma. Для реалізації фронтендної частини використовувалась формальна мова декодування й опису зовнішнього вигляду CSS та стандартизована мова гіпертекстової розмітки документів для перегляду веб-сторінок у браузері HTML. Для розробки бекенду середовище розробки Microsoft Visual Studio 2022 та мову програмування javascript. Це дало змогу забезпечити високу швидкодію та надійність системи. Створення та розробка програмного забезпечення проводилась із застосуванням підходів та методів управління проектами.

В даній роботі представлено аналіз поставленої задачі, вивчено специфіку предметної області, опрацьовано теоретичний матеріал, викладено суть та теоретичні основи досліджуваної проблеми, обрано інструментальні засоби, визначенні основні завдання проекту та проведено огляд сучасної літератури, присвяченої питанням, що розглядаються.

В результаті роботи було описано процес проектування і розробки веб-сайту, опис структури даних, інтерфейсу користувача, а також основних алгоритмів.

Створений програмний продукт, що відповідає визначеним вимогам в плані швидкодії, надійності та дизайну і реалізовує поставлені цілі в повній мірі.

В рамках даної роботи було розроблено Інтернет-магазин одягу Ustock, який можна використовувати в побудові бізнесу, з ціллю отримання подальшого продукту.

В цій дипломній роботі були виконані дані дії:

- Було розглянуто, що таке CRM-система та розглянули найпопулярніші системи, які можуть підійти до нашого сайту.
- Було розглянуто можливості, які доступні на даний момент часу, а саме: функціонал веб-сайту, інтеграції з технологіями, аналітика та штучний інтелект та електронна комунікація з клієнтами

- Визначено проблемні моменти, які можуть негативно вплинути на успішність веб-сайту.
- Визначено головне завдання проєкту та вказали ключові завдання, які необхідно виконати.
- Визначено основну мету автоматизованого проєктування та розглянули різні методи та інструменти, які можуть позитивно вплинути на кінцевий продукт.
- Побудовано дерево цілей проєктування
- Проведено ряд аналізів, а саме: характеристику завдання, аналіз потреб користувачів, оцінка технічних вимог, виявлення складнощів та ризиків, аналіз необхідності додаткових досліджень. Також ми проглянули дослідження, а саме: літературу, дослідження та аналогічні проєкти.
- Було розглянуто, що таке методологія програмного і яку роль вона відіграє в проєкті.
- Було розглянуто існуючі методології.
- Було порівняно існуючі методології та обрали модель, яка найкраще підходить до проєкту.
- Було розглянуто який тип програмного забезпечення буде використовуватись в розробці продукту.
- Було розглянуто додатки для розробки інтерфейсу і прототипування та порівняли їх.
- Було обрано додаток який буде використовуватись в розробці продукту.
- Було розглянуто інструменти та технології для проєктування архітектури які будуть використовуватись.
- Був розглянутий процес розробки прототипу інтерфейсу
- Був розглянутий процес реалізації фронтендної частини
- Було розглянуто реалізацію основних компонентів інтернет-магазину Ustock
- Була розглянута процедура реєстрації та входу користувача на сайт
- Була розглянута процедура реалізації зміни даних користувача
- Була розглянута процедура реалізації пошуку та сортування товару на сайті
- Було розглянуто базу даних та таблиць які використовуються в ній

- Були розглянуті функціональні можливості, які доступні рядовому користувачеві
- Створено проєкт в MS Project Pro
- Визначено час, який необхідно на виконання проєкту та розглянули його задачі
- Розглянуто ресурси, які необхідні для виконання проєкту. Обчислено витрати та встановили бюджет проєкту
- Виконано аналіз зацікавлених сторін та стало відомо хто зможе вплинути на проєкт.
- Зроблено SWOT-аналіз, щоб дізнатись про сильні та слабкі сторони проєкту
- Встановлено цілі, які повинен виконати веб-сайт після релізу

## **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ:**

1. PMBOK (Project Management Body of Knowledge) від PMI (Project Management Institute).
2. ISO 21500:2012 «Guidance on Project Management».
3. «Effective Project Management: Traditional, Agile, Extreme» автор Robert K. Wysocki.
4. «Ecommerce: Business, Technology, Society» автори Kenneth C. Laudon та Carol Guercio Traver.
5. «JavaScript: The Good Parts» автор Douglas Crockford.
6. «CSS: The Definitive Guide» автор Eric A. Meyer.
7. «Node.js Design Patterns» автори Mario Casciaro та Luciano Mammino.
8. «Microsoft Project 2019 Step by Step» автор Timothy Johnson.
9. «ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДОЛОГІЙ РОЗРОБКИ ІТ ПРОДУКТУ» автори О.О. Коваленко, А.В. Денисюк та В.М. Бажан.
10. "Digital Marketing Excellence: Planning, Optimizing and Integrating Online Marketing" авторів Dave Chaffey та PR Smith.
11. "Designing Data-Intensive Applications" автор Martin Kleppmann.
12. "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses" автор Eric Ries.
13. "Designing Data-Intensive Applications" автор Martin Kleppmann.
14. "Microservices: Flexible Software Architecture" автор Eberhard Wolff.
15. "Learning React: Functional Web Development with React and Redux" автори Alex Banks та Eve Porcello.
16. "E-commerce and Information Technology in Hospitality and Tourism" авторів Zongqing Zhou.

# ДОДАТКИ

## Додаток 1

### Фрагмент коду головної сторінки

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      rel="stylesheet"
      href="https://fonts.googleapis.com/css2?family=Inter:wght@500"
    />
    <title>HTML Preview</title>
    <link rel="stylesheet" type="text/css" href="styles.css" />
    <link rel="stylesheet" type="text/css" href="compiled-styles.css" />
  </head>
  <body style="margin: 0px">
    <!-- This only removes default margins in our renderer iframe, you may remove -->
    <div id="firejet-html-app">
      <div class="component-main-wrapper">
        <main class="component-main-main">
          <section class="component-main-rectangle-1">
            <header
              class="component-main-t-gu-774_luggage-20crops_150dpi_52000x-
11251tbrands" >
              <nav class="component-main-g-u774_luggage20crops_150dpi_52000x-11251" >
                <footer
                  class="component-main-ttt-screenshot_12removebg-preview-
11ttustock-ttellipse-11ellipse-11tline-1tellipse-11bg-rectangle-19" >
                  <aside class="component-main-tt-screenshot_12removebg-preview-1" >
                    <article class="component-main-t-screenshot_12removebg-preview-1" >
                      <address
                        class="component-main-screenshot_12removebg-preview-1"
                      >></address>
                    </article>
                    <div class="component-main-component"><p>для нього</p></div>
                  </aside>
                  <div
                    class="component-main-ttt-ustock-ttellipse-11ellipse-11tline-1tellipse-11" >
                    <div class="component-main-tt-ustock-t">
                      <p class="component-main-para-0">USTOCK</p>
                    </div>
                  </div>
                </nav>
              </header>
            </section>
          </main>
        </div>
      </div>
    </body>
  </html>
```

```
<div class="component-main-t">
  <div class="component-main-component-0">
    <p>для неї</p>
  </div>
  <div class="component-main-component-1">
    <p>аксесуари</p>
  </div>
</div>
</div>
<div class="component-main-t-line-1tellipse-11">
  <object
    data="/assets/Line1.svg"
    class="component-main-line-1"
  >>/object>
  <div class="component-main-t-ellipse-11">
    <object
      data="/assets/Ellipse11.svg"
      class="component-main-ellipse-11"
    >>/object>
  </div>
</div>
<div class="component-main-t-ellipse-11ellipse-11">
  <object
    data="/assets/Ellipse12.svg"
    class="component-main-ellipse"
  >>/object>
  <object
    data="/assets/Ellipse13.svg"
    class="component-main-ellipse-0"
  >>/object>
</div>
<div class="component-main-t-ellipse">
  <object
    data="/assets/Ellipse14.svg"
    class="component-main-ellipse-1"
  >>/object>
</div>
</div>
</footer>
```

```
</nav>
<div class="component-main-t-brands">
  <object
    data="/assets/Brands.svg"
```

## Додаток 2

### Модель користувача:

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;
const bcrypt = require('bcrypt');

const userSchema = new Schema({
  firstName: { type: String, required: true },
  lastName: { type: String, required: true },
  mobileNumber: { type: String, required: true },
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  region: { type: String, required: true },
  city: { type: String, required: true },
  street: { type: String, required: true },
  postalCode: { type: String, required: true },
  houseNumber: { type: String, required: true }
});

// Hash password before saving
userSchema.pre('save', async function (next) {
  if (this.isModified('password') || this.isNew) {
    this.password = await bcrypt.hash(this.password, 10);
  }
  next();
});

userSchema.methods.comparePassword = function (password) {
  return bcrypt.compare(password, this.password);
};

const User = mongoose.model('User', userSchema);
module.exports = User;
```

### Сервер і маршрути для реєстрації та входу:

```

const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const bcrypt = require('bcrypt');
const jwt = require('jsonwebtoken');
const User = require('./models/User');

const app = express();
const PORT = process.env.PORT || 3000;
const SECRET_KEY = 'your_secret_key';

// Middleware
app.use(bodyParser.json());

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/registration', {
  useNewUrlParser: true,
  useUnifiedTopology: true });

mongoose.connection.once('open', () => {
  console.log('Connected to MongoDB');
}).on('error', (error) => {
  console.log('Connection error:', error);
});

// Registration route
app.post('/register', async (req, res) => {
  const { firstName, lastName, mobileNumber, email, password, region, city, street, postalCode,
houseNumber } = req.body;

  // Create a new user
  const newUser = new User({
    firstName,
    lastName,
    mobileNumber,
    email,
    password,
    region,
    city,
    street,
    postalCode,
    houseNumber
  });

```

```

    try {
      const savedUser = await newUser.save();
      res.status(201).json({ message: 'User registered successfully', user: savedUser });
    } catch (error) {
      res.status(400).json({ message: 'Error registering user', error });
    }
  });

// Login route
app.post('/login', async (req, res) => {
  const { emailOrPhone, password } = req.body;

  try {
    // Find user by email or phone number
    const user = await User.findOne({ $or: [{ email: emailOrPhone }, { mobileNumber: emailOrPhone
}] });
    if (!user) {
      return res.status(401).json({ message: 'Authentication failed. User not found.' });
    }

    // Check password
    const isMatch = await user.comparePassword(password);
    if (!isMatch) {
      return res.status(401).json({ message: 'Authentication failed. Wrong password.' });
    }

    // Create token
    const token = jwt.sign({ id: user._id, email: user.email }, SECRET_KEY, { expiresIn: '1h' });

    res.json({ message: 'Login successful', token });
  } catch (error) {
    res.status(500).json({ message: 'Error logging in', error });
  }
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});

```

## Оновлення серверу і маршрути (app.js):

```

const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const User = require('./models/User');
const Product = require('./models/Product');

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(bodyParser.json());

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/shop', { useNewUrlParser: true, useUnifiedTopology: true
});

mongoose.connection.once('open', () => {
  console.log('Connected to MongoDB');
}).on('error', (error) => {
  console.log('Connection error:', error);
});

// Route to add a product (for testing purposes)
app.post('/products', async (req, res) => {
  const { name, description, price, category, subcategory, stock } = req.body;

  const newProduct = new Product({
    name,
    description,
    price,
    category,
    subcategory,
    stock
  });

  try {
    const savedProduct = await newProduct.save();
    res.status(201).json({ message: 'Product added successfully', product: savedProduct });
  } catch (error) {
    res.status(400).json({ message: 'Error adding product', error });
  }
});

```

```

// Route to get products by category and subcategory
app.get('/products', async (req, res) => {
  const { category, subcategory } = req.query;

  if (!category) {
    return res.status(400).json({ message: 'Category query parameter is required' });
  }

  try {
    let query = { category: category };

    if (subcategory) {
      query.subcategory = subcategory;
    }

    const products = await Product.find(query);
    res.json(products);
  } catch (error) {
    res.status(500).json({ message: 'Error fetching products', error });
  }
});

// Route to update user data
app.put('/users/:id', async (req, res) => {
  const { id } = req.params;
  const updateData = req.body;

  try {
    const updatedUser = await User.findByIdAndUpdate(id, updateData, { new: true, runValidators:
true });
    if (!updatedUser) {
      return res.status(404).json({ message: 'User not found' });
    }
    res.json({ message: 'User updated successfully', user: updatedUser });
  } catch (error) {
    res.status(400).json({ message: 'Error updating user', error });
  }
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});

```

## Модель товару

```

const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const productSchema = new Schema({
  name: { type: String, required: true },
  description: { type: String },
  price: { type: Number, required: true },
  category: { type: String, required: true },
  subcategory: { type: String, required: true },
  stock: { type: Number, default: 0 }
});

const Product = mongoose.model('Product', productSchema);
module.exports = Product;

```

## Сервер і маршрут для пошуку товару:

```

const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const Product = require('./models/Product');

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(bodyParser.json());

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/shop', { useNewUrlParser: true, useUnifiedTopology: true
});

mongoose.connection.once('open', () => {
  console.log('Connected to MongoDB');
}).on('error', (error) => {
  console.log('Connection error:', error);
});

// Route to add a product (for testing purposes)
app.post('/products', async (req, res) => {

```

```

const { name, description, price, category, stock } = req.body;

const newProduct = new Product({
  name,
  description,
  price,
  category,
  stock
});

try {
  const savedProduct = await newProduct.save();
  res.status(201).json({ message: 'Product added successfully', product: savedProduct });
} catch (error) {
  res.status(400).json({ message: 'Error adding product', error });
}
});

// Route to search products by name
app.get('/search', async (req, res) => {
  const { name } = req.query;

  if (!name) {
    return res.status(400).json({ message: 'Name query parameter is required' });
  }

  try {
    const products = await Product.find({ name: new RegExp(name, 'i') });
    res.json(products);
  } catch (error) {
    res.status(500).json({ message: 'Error searching for products', error });
  }
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});

```

## Створення маршруту для сортування товарів

```

const express = require('express');
const mongoose = require('mongoose');

```

```

const bodyParser = require('body-parser');
const Product = require('./models/Product');

const app = express();
const PORT = process.env.PORT || 3000;

// Middleware
app.use(bodyParser.json());

// MongoDB connection
mongoose.connect('mongodb://localhost:27017/shop', { useNewUrlParser: true, useUnifiedTopology: true
});

mongoose.connection.once('open', () => {
  console.log('Connected to MongoDB');
}).on('error', (error) => {
  console.log('Connection error:', error);
});

// Route to add a product (for testing purposes)
app.post('/products', async (req, res) => {
  const { name, description, price, category, subcategory, stock } = req.body;

  const newProduct = new Product({
    name,
    description,
    price,
    category,
    subcategory,
    stock
  });

  try {
    const savedProduct = await newProduct.save();
    res.status(201).json({ message: 'Product added successfully', product: savedProduct });
  } catch (error) {
    res.status(400).json({ message: 'Error adding product', error });
  }
});

// Route to get products by category and subcategory
app.get('/products', async (req, res) => {
  const { category, subcategory } = req.query;

```

```
if (!category) {
  return res.status(400).json({ message: 'Category query parameter is required' });
}

try {
  let query = { category: category };

  if (subcategory) {
    query.subcategory = subcategory;
  }

  const products = await Product.find(query);
  res.json(products);
} catch (error) {
  res.status(500).json({ message: 'Error fetching products', error });
}
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```