

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету )  
Кафедра комп'ютерних наук  
(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка інтернет-магазину “Hoodie customize” за допомогою технологій Python

Виконав: студент IV курсу, групи СНС-41  
спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Богуцький М. І.  
(підпис) (прізвище та ініціали)

Керівник Приймак М. В.  
(підпис) (прізвище та ініціали)

Нормоконтроль \_\_\_\_\_  
(підпис) (прізвище та ініціали)

Завідувач кафедри \_\_\_\_\_  
(підпис) (прізвище та ініціали)

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

Тернопіль  
2023

Міністерство освіти і науки України  
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
Завідувач кафедри

\_\_\_\_\_  
(підпис) (прізвище та ініціали)  
«\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Богущькому Михайлові Івановичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-сайту “Hoodie Customize” за допомогою технологій Python

Керівник роботи \_\_\_\_\_  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 7 » лютого 2023 року № 4/7-13X

2. Термін подання студентом завершеної роботи \_\_\_\_\_

3. Вихідні дані до роботи \_\_\_\_\_

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. РОЗДІЛ 1. Постановка задачі розробки веб-сайту “Hoodie customize”, 1.1 Аналіз предметної області, 1.2 Формування вимог розробки веб-сайту “Hoodie customize”, 1.3 Пошук актантів та варіантів використання веб-сайту “Hoodie customize”, 1.3.1 Пошук актантів, 1.3.2 Варіанти використання веб-сайту “Hoodie customize”, 1.4 Опис ключових варіантів використання веб-сайту “Hoodie customize”, 1.5 Вибір середовища розробки веб-сайту “Hoodie customize”, 1.5.1 Мова програмування, 1.5.2 База даних, 1.6 Обґрунтування використовуваних технологій розробки веб-сайту “Hoodie customize”, 1.7 Висновки до першого розділу, 2. Проектування та реалізація веб-сайту “Hoodie customize”, 2.1 Архітектура веб-сайту “Hoodie customize”, 2.2 Вибір основних залежностей для розробки веб-сайту “Hoodie customize”, 2.3 Файлова структура веб-сайту “Hoodie customize”, 2.4 Структура та особливості використаної БД для розробки веб-сайту “Hoodie customize”, 2.5 Адміністрування веб-сайту “Hoodie customize” 2.6 Розгортання веб-сайту “Hoodie customize” на хостингових платформах, 2.7 Тестування та використання веб-сайту “Hoodie customize”, 2.7.1 Тестування веб-сайту “Hodie Customize”, 2.7.2 Використання веб-сайту “Hoodie customize”, 2.8 Висновки до другого розділу, 3. Безпека життєдіяльності, основи хорони праці, 3.1 Методи боротьби з монотонністю праці на виробництві, 3.2 Загальні вимоги безпеки з охорони праці для користувачів ПК, 3.3 Висновки до третього розділу, Висновки, Перелік використаних джерел, Додатки.



## 5. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи хорони праці			

6. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		<i>Виконано</i>
2.	Пошук джерел щодо розробки та створення веб-сайтів		<i>Виконано</i>
3.	Опрацювання обраних джерел		<i>Виконано</i>
4.	Аналіз області розробки веб-сайту, підбір технологій для розробки		<i>Виконано</i>
5.	Розробка веб-сайту “Hoodie customize”		<i>Виконано</i>
6.	Оформлення розділу «Постановка задачі розробки веб-сайту “Hoodie customize”»		<i>Виконано</i>
7.	Оформлення розділу «Проектування та реалізація веб-сайту “Hoodie customize”»		<i>Виконано</i>
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»		<i>Виконано</i>
9.	Виконання завдання до підрозділу «Основи хорони праці»		<i>Виконано</i>
10.	Оформлення кваліфікаційної роботи		<i>Виконано</i>
11.	Нормоконтроль		<i>Виконано</i>
12.	Перевірка на плагіат		<i>Виконано</i>
13.	Попередній захист кваліфікаційної роботи		<i>Виконано</i>
14.	Захист кваліфікаційної роботи		<i>Виконано</i>

Студент

\_\_\_\_\_

(підпис)

Богуцький М. І.

\_\_\_\_\_

(прізвище та ініціали)

Керівник роботи

\_\_\_\_\_

(підпис)

\_\_\_\_\_

(прізвище та ініціали)

## АНОТАЦІЯ

Розробка веб-сайту “Hoodie Customize” засобами Django, Python, HTML та CSS // Кваліфікаційна робота освітнього рівня «Бакалавр» // Богуцький Михайло Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СНс-41 // Тернопіль, 2023 // С. 75, рис. – 15, табл. – 2, кресл. – 0 , додат. – 4, бібліогр. – 15.

**Ключові слова:** django, python, sql, веб-сайт, merch, web, hoodie shop, customize.

Кваліфікаційна робота присвячена розробці веб-сайту “Hoodie customize” засобами Django, Python, HTML та CSS.

Метою даної роботи є розробка веб-сайту інтернет-магазину продажу та їх купівлі кастомізованого худі.

В першому розділі кваліфікаційної роботи проведено аналіз предметної області, сформовано основні вимоги для розробки, вибравши при цьому необхідні інструменти, описано взаємодію акторів з системою.

В другому розділі кваліфікаційної роботи розглянуто архітектуру розроблюваного веб-сайту, описано основні залежності та структуру, розглянуто розгортання на хостингових платформах, проведено тестування.

В третьому розділі, було розглянуто пожежну профілактику та пожежну безпеку на виробництві та профілактику нещасних випадків, професійних захворювань і отруєнь на виробництві.

## ANNOTATION

Development of the “Hoodie Customize” website using Django, Python, HTML and CSS // Qualification work of Bachelor educational degree // Boguzkiy Mikhailo Ivanovich // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Computer Science Department, SNs-41 group // Ternopil, 2023 // Pages – 75, figures – 15 , tables – 2 , sketches – 0, addendums – 4, references – 15.

**Keywords:** django, python, sql, website, merch, web, hoodie shop, customize.

The qualification work is dedicated to the development of the "Hoodie customize" website using Django, Python, HTML and CSS.

The purpose of this work is to develop a website for an online store selling and buying customized hoodies.

In the first section of the qualification work, an analysis of the subject area was carried out, the main requirements for the development were formed, while choosing the necessary tools, the interaction of actors with the system was described.

In the second section of the qualification work, the architecture of the developed website was considered, the main dependencies and structure were described, the deployment on hosting platforms was considered, and testing was carried out.

In the third section, fire prevention and fire safety at work and prevention of accidents, occupational diseases and poisoning at work were considered.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CSS – Cascading Style Sheets – це мова стилів, яка використовується для опису вигляду і форматування документів HTML.

HTML – Hypertext Markup Language – це мова розмітки гіпертексту, яка використовується для створення структури і вмісту веб-сторінок.

PY – Код Python може включати інструкції, що взаємодіють з операційною системою, зчитують і записують дані, виконують обчислення, створюють функції та багато іншого.

DB – Database – це система для зберігання і організації великих обсягів даних, яка дозволяє ефективно зберігати, керувати та отримувати доступ до інформації.

Стек (Stack) – це набір технологій, який використовується для розробки веб-сайтів або додатків. Він може включати фреймворки, бібліотеки, мови програмування та інші компоненти.

Баг (Bug) – це помилка або несправність, яка може бути виявлена під час розробки або використання програмного забезпечення.

Трек (Track) – це запис музичного треку або аудіофайлу.

Шарінг (Share) – це процес поширення або обміну інформацією, файлами, ресурсами тощо між користувачами або системами.

Use case – це діаграма або опис сценаріїв використання, яка показує взаємодію між користувачами та системою і описує функціональність продукту.

Крос-платформеність – це властивість програмного забезпечення працювати на різних апаратних платформах або операційних системах без необхідності значних модифікацій.

DOM – Document Object Model – це програмний інтерфейс, який представляє структуру і зміст веб-документа і дозволяє програмам змінювати його вміст, стиль і структуру.

SPA – Single Page Application – це тип веб-додатка, який завантажує весь вміст на одну HTML сторінку і оновлює його динамічно за допомогою JavaScript, не перезавантажуючи сторінку під час використання.

UI – User Interface – це набір екранів, сторінок і візуальних елементів, таких як кнопки та значки, які дозволяють користувачу взаємодіяти з продуктом або сервісом.



## ЗМІСТ

ВСТУП .....		9
<b>РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ВЕБ-САЙТУ “HOODIE CUSTOMIZE”</b> .....		
1.1	Аналіз предметної області .....	10
1.2	Формування вимог розробки веб-сайту “Hoodie customize” .....	11
1.3	Пошук актантів та варіантів використання веб-сайту “Hoodie customize” .....	13
1.3.1	Варіанти використання веб-сайту “Hoodie customize” .....	15
1.4	Опис ключових варіантів використання веб-сайту “ Hoodie customize ” .....	17
1.5	Вибір середовища розробки веб-сайту “ Hoodie customize ” .....	18
1.5.1	Мова програмування .....	19
1.5.2	База даних.....	20
1.6	Обґрунтування використовуваних технологій розробки веб-сайту “Hoodie customize” .....	22
1.7	Висновок до першого розділу .....	25
2.	<b>ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ “HOODIE CUSTOMIZE”</b> ..	27
2.1	Клієнт-серверна архітектура веб-сайту “Hoodie customize” .....	27
2.2	Вибір основних залежностей для розробки веб-сайту “Hoodie customize” .. <b>Error!</b> <b>Bookmark not defined.</b>	
2.3	Файлова структура веб-сайту “Hoodie customize” .....	31
2.4	Структура та особливості використаної БД для розробки веб-сайту“ Hoodie Customize ” .....	33
2.5	Адміністрування веб-сайту “Hoodie customize” .....	35
2.6	Розгортання веб-сайту “Hoodie customize” на хостингових платформах.....	38
2.7	Тестування та використання веб-сайту “Hoodie customize” .....	43
2.7.1	Тестування веб-сайту “Hoodie customize” .....	43
2.7.2	Використання веб-сайту “Hoodie Customize” .....	46
2.8	Висновки до другого розділу .....	50
3.	<b>БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ</b> .....	52
3.1	Пожежна профілактика та пожежна безпека на виробництві .....	52

3.2	Профілактика нещасних випадків, професійних захворювань і отруєнь на виробництві.....	54
3.3	Висновок до третього розділу .....	57
	ВИСНОВКИ.....	59
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
	Додаток А.....	62
	Лістинг 1 – Код файлу «manage.py».....	62
	Додаток Б .....	63
	Лістинг 2 – Код файлу «models.py» .....	63
	Додаток В.....	64
	Лістинг 3 – Код файлу «admin.py» .....	64
	Додаток Г .....	65
	Лістинг 4 – Код файлу «views.py».....	65

## ВСТУП

**Актуальність теми.** Розробка для магазину худі є дуже актуальною, особливо з огляду на зростаючу популярність електронної комерції та онлайн-шопінгу. Споживачі все більше звертаються до покупок в Інтернеті, включаючи покупку одягу та аксесуарів.

Розробка магазину худі дозволяє підприємствам, брендам або підприємцям представити свою продукцію широкій аудиторії через Інтернет. Вона надає можливість зручного перегляду, вибору та замовлення худі, а також забезпечує зручну систему оплати та доставки.

Популярність магазинів худі також пояснюється тим, що худі є популярним елементом гардеробу, який використовується як у повсякденному, так і у спортивному стилі одягу. Це дозволяє привернути широкий спектр клієнтів, що впливає на популярність та прибутковість таких магазинів.

Крім того, розробка магазину худі надає можливості для персоналізації та налаштування продуктів, що дозволяє клієнтам створювати унікальні дизайни своїх худі, додавати елементи, зображення або текст за власними бажаннями. Це створює індивідуальний досвід покупця та сприяє збільшенню залучення та задоволення клієнтів.

**Мета і задачі дослідження.** Метою цієї кваліфікаційної роботи освітнього рівня “Бакалавр”, є реалізація веб-сайту інтернет-магазину для продажу кастомізованих худі та їх купівлі, використовуючи при цьому сучасний інструментарій розробки програмного забезпечення.

Для успішної реалізації задачі, було сформовано ряд наступних завдань:

- вибрати сучасний інструментарій для розробки (стек);
- провести розробку як серверної частини, так і клієнтської;
- перевірити змістовність та сумісність вибраного інструментарію;
- провести тестування для виявлення багів та їх усунення.

# РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ВЕБ-САЙТУ “HOODIE CUSTOMIZE”

## 1.1 Аналіз предметної області

У сучасному світі електронна комерція стає все більш популярною і доступною для широкого кола споживачів. Розроблення інтернет-магазинів стає однією з найважливіших галузей веб-розробки. У цьому розясненні буде проведено аналіз предметної області "Розроблення інтернет-магазину з кастомізованими худі" для розуміння важливості, особливостей та переваг таких проектів.

### 1. Визначення інтернет-магазину з кастомізованими худі.

Інтернет-магазин з кастомізованими худі - це електронний ресурс, де споживачі можуть замовляти худі з можливістю індивідуалізації, тобто налаштувати дизайн, кольори, розміри тощо, щоб створити унікальний продукт, відповідний їхнім потребам та вподобанням.

### 2. Особливості інтернет-магазинів з кастомізованими худі.

Однією з основних особливостей інтернет-магазинів з кастомізованими худі є можливість персоналізації продукту. Споживачі можуть вибрати кольори, дизайн, матеріал, розміри тощо, що дає їм можливість створити худі, яке відповідає їхньому унікальному стилю та виразності.

Крім того, інтернет-магазини з кастомізованими худі зазвичай мають інтуїтивно зрозумілий інтерфейс, що дозволяє легко налаштувати худі за допомогою інтерактивних інструментів. Такі магазини також забезпечують візуалізацію кастомізації, що дозволяє споживачам побачити, як буде виглядати їхній кінцевий продукт перед замовленням.

### 3. Переваги інтернет-магазину з кастомізованими худі:

- Індивідуальний підхід: Створення кастомізованих худі дає споживачам можливість виразити свою особистість та стиль через одяг. Клієнти можуть створити унікальний продукт, який підходить саме їхнім потребам, та відчути особисту зв'язок з брендом.

- Розширені можливості продажу: Для бізнесу розробка інтернет-магазину з кастомізованими худі відкриває нові можливості для збільшення обсягів продажу. Споживачі більш схильні придбати продукт, який вони можуть персоналізувати, що стимулює продажі і збільшує задоволеність клієнтів

- Мінімізація виробничих втрат: Для виробника кастомізованого одягу

виробництво відбувається лише після отримання замовлення, що дозволяє уникнути непроданих запасів і зменшити витрати на зберігання. Це може забезпечити оптимізацію виробничих процесів та зниження витрат на складське утримання.

#### 4. Технічні аспекти розроблення інтернет-магазину з кастомізованими худі.

Для розробки інтернет-магазину з кастомізованими худі необхідно врахувати ряд технічних аспектів. Основними елементами є:

- Веб-дизайн і UX/UI: Забезпечення зручного та привабливого інтерфейсу для користувачів, що дає можливість легко налаштувати та візуалізувати худі перед замовленням.
- Конфігуратор продукту: Розробка інтерактивного інструменту, який дозволяє споживачам вибрати параметри кастомізації, такі як колір, дизайн, розмір, вишивка тощо, і отримати візуальне представлення кінцевого продукту.
- Інтеграція платіжних систем: Забезпечення безпечних та зручних способів оплати для клієнтів, що включають різноманітні платіжні шлюзи та онлайн-платформи.
- Оптимізація продуктивності та безпеки: Врахування вимог до швидкості завантаження, безпеки даних, масштабованості та інших технічних аспектів, щоб забезпечити високу якість роботи інтернет-магазину.

Розробка інтернет-магазину з кастомізованими худі - це цікаве та перспективне напрямком в електронній комерції. Враховуючи індивідуальні потреби та вподобання клієнтів, такі магазини можуть стати успішними бізнес-проектами, забезпечуючи унікальний продукт і високу задоволеність споживачів. Правильна реалізація технічних аспектів і створення зручного інтерфейсу - ключові фактори успіху такого проекту.

## 1.2 Формування вимог розробки веб-сайту “Hoodie customize”

Веб-сайт "Hoodie Customize" спеціалізується на наданні можливості кастомізації худі для своїх клієнтів. Процес формування вимог для розробки цього веб-сайту є критично важливим для досягнення успіху проекту та задоволення потреб споживачів. У цьому підрозділі будуть розглянуті ключові аспекти формування вимог розробки веб-сайту "Hoodie Customize" для створення зручного та персоналізованого досвіду для користувачів.

1. Вивчення цільової аудиторії. Першим кроком у формуванні вимог є ретельне дослідження та аналіз цільової аудиторії "Hoodie Customize". Це включає вивчення їхніх вподобань, потреб і очікувань щодо кастомізації худі. Наприклад, можливо, клієнти бажають мати можливість вибирати кольори, дизайн, матеріал, розмір і додаткові деталі.

2. Визначення функціональності та особливостей. На основі дослідження аудиторії визначаються основні функціональні можливості та особливості веб-сайту. До них можуть входити:

- Конфігуратор продукту: Розробка інтерактивного інструменту, що дозволяє клієнтам вибирати різні параметри кастомізації, такі як кольори, дизайн, матеріали та інші деталі, і отримувати візуальне відображення результату.

- Кошик та оформлення замовлення: Реалізація системи кошика, де клієнти можуть переглядати, зберігати та оформляти замовлення своїх кастомізованих худі, включаючи вибір розміру, кількості та інших опцій.

- Інтеграція платіжних систем: Забезпечення безпечних та зручних способів оплати, які включають різні платіжні шлюзи та онлайн-платформи.

- Збереження та керування замовленнями: Розробка системи, яка дозволяє зберігати та керувати замовленнями клієнтів, включаючи статус замовлення, доставку та сповіщення.

3. Дизайн та користувацький інтерфейс. Дизайн та користувацький інтерфейс є важливими аспектами веб-сайту "Hoodie Customize". Необхідно визначити естетику, шрифти, колірну палітру та взаємодію елементів веб-сторінок. Дизайн має бути привабливим, сучасним та легким у сприйнятті. Користувацький інтерфейс повинен бути інтуїтивно зрозумілим, легким у використанні та надавати зручну навігацію.

4. Вимоги до технічного розробки. Для успішної розробки веб-сайту "Hoodie Customize" необхідно встановити вимоги до технічного розробки. Це включає вибір платформи розробки, мов програмування, бази даних, хостингу та інших технічних деталей. Додатково, необхідно забезпечити високу продуктивність, безпеку та масштабованість веб-сайту.

Формування вимог розробки веб-сайту "Hoodie Customize" є ключовим етапом для створення успішного та зручного сервісу для клієнтів. Це вимагає ретельного вивчення аудиторії, визначення функціональних можливостей, розробки дизайну та інтерфейсу, а також встановлення технічних вимог. Правильна формулювання вимог дозволить

створити унікальний та персоналізований досвід для клієнтів "Hoodie Customize" і забезпечить успіх проекту.

### **1.3 Пошук актантів та варіантів використання веб-сайту "Hoodie customize"**

Веб-сайт є потужним інструментом для комунікації та взаємодії з актантами - учасниками системи, які впливають на функціонування та успіх проекту. Пошук актантів та вивчення їхніх потреб та вимог є важливим етапом у розробці веб-сайту. У даному підрозділі будуть розглянуті процес пошуку актантів та розглянуті різні варіанти використання веб-сайту.

1. Пошук актантів. Пошук актантів включає виявлення та ідентифікацію основних груп користувачів та інших учасників, які мають взаємодіяти з веб-сайтом. Для ефективного пошуку актантів можна використовувати наступні методи:

- Аналіз цільової аудиторії: Вивчення потенційних користувачів, їхніх характеристик, потреб та поведінки. Це може включати дослідження ринку, проведення опитувань або спостереження.
- Консультації зі зацікавленими сторонами: Спілкування з експертами, потенційними клієнтами, партнерами та іншими особами, які можуть мати інформацію про актантів та їхні потреби.
- Використання аналітичних інструментів: Використання даних та аналітики для вивчення поведінки користувачів, їхніх взаємодій з веб-сайтом та отримання інсайтів щодо їхніх потреб.

2. Варіанти використання веб-сайту. Після виявлення актантів та розуміння їхніх потреб можна розглянути різні варіанти використання веб-сайту "Hoodie Customize". Основні варіанти включають:

- Онлайн-продажі: Веб-сайт може використовуватись як платформа для продажу кастомізованих худі. Клієнти можуть переглядати різні моделі та опції кастомізації, робити замовлення та здійснювати оплату онлайн.
- Креативний конструктор: Веб-сайт може надати клієнтам можливість використовувати інтерактивний конструктор для створення своїх унікальних дизайнів

худі. Це може включати вибір кольорів, графічних елементів, тексту та інших деталей.

- Візуалізація результату: Веб-сайт може надати візуальне відображення результату кастомізації, дозволяючи клієнтам побачити, як буде виглядати їхнє худі з обраними параметрами.

- Спільнота та обмін: Веб-сайт може створити спільноту користувачів, де вони можуть ділитися своїми дизайнами, досвідом та враженнями від кастомізації худі.

Пошук актантів та вивчення їхніх потреб є важливим етапом у розробці веб-сайту. Розуміння актантів дозволяє створити веб-сайт, який задовольняє їхні потреби та надає цінність. Різні варіанти використання веб-сайту "Hoodie Customize", такі як онлайн-продажі, креативний конструктор, візуалізація результату та спільнота, дозволяють створити унікальний та персоналізований досвід для клієнтів та підвищити конкурентоспроможність бізнесу.

Модель використання веб-сайту - це концептуальна структура, яка описує, які основні функції та можливості надаються користувачам на веб-сайті та як вони можуть взаємодіяти з ним.

Модель використання веб-сайту включає такі елементи:

- Навігація: Користувачі можуть використовувати різні засоби навігації, такі як меню, пошукові поля, посилання та інші елементи, щоб переходити між сторінками та розділами веб-сайту.

- Функціональність: Веб-сайт може надавати різні функції, такі як можливість реєстрації користувачів, створення профілю, здійснення покупок, заповнення форм, коментування, завантаження файлів тощо.

- Аутентифікація та авторизація: Веб-сайт може вимагати від користувачів виконання процесу аутентифікації, щоб перевірити їхню ідентичність та надати доступ до певних ресурсів або функцій.

- Взаємодія з інтерфейсом користувача: Користувачі можуть взаємодіяти з різними елементами інтерфейсу, такими як кнопки, поля введення, списки, перемикачі та інші, для виконання певних дій або введення інформації.

- Модель використання веб-сайту допомагає розуміти, як користувачі будуть сприймати та використовувати функціонал веб-сайту. Вона визначає, які можливості та обмеження будуть доступні для користувачів та як вони будуть отримувати очікувані результати від використання веб-сайту.



Зазвичай, випадки використання пов'язані з "акторами", які можуть бути людьми або машинами, які взаємодіють з системою. Роль акторів у системі визначається набором доступних їм варіантів використання. [5]

Виходячи з цього діаграма варіантів використання веб-сайту “Hoodie customize”, матиме наступний вигляд (див. рис. 1.1):

Рисунок 1.1 – діаграма варіантів використання (Use case) веб-сайту “Hoodie customize”

### 1.3.1 Варіанти використання веб-сайту “Hoodie customize”

Веб-сайт "Hoodie Customize" є платформою, спеціалізованою на кастомізації худі. Він надає користувачам можливість створити унікальний дизайн своїх худі з різними параметрами та елементами. Ось декілька варіантів використання цього веб-сайту:

**Перегляд та вибір варіантів:** Користувачі можуть переглядати доступні моделі худі та варіації кастомізації. Вони можуть ознайомитись з різними стилями, кольорами, розмірами та іншими параметрами, щоб обрати той, який найкраще відповідає їхнім потребам.

**Замовлення та покупка:** Після завершення кастомізації, користувачі можуть додати обране худі до кошика та здійснити покупку. Вони можуть обрати необхідну кількість, вказати доставку та здійснити оплату онлайн.

**Візуалізація результату:** Веб-сайт може надати візуальне відображення того, як буде виглядати кастомізоване худі з обраними параметрами. Це допомагає користувачам побачити перед покупкою, як виглядатиме їхнє худі з обраними змінами.

**Спільнота та обмін:** Веб-сайт може створити спільноту користувачів, де вони можуть ділитися своїми створеними дизайнами худі, залишати відгуки, обмінюватися думками та враженнями від кастомізації. Це створює сприятливу атмосферу для взаємодії та інспірації серед користувачів.

Загалом, веб-сайт "Hoodie Customize" надає можливість користувачам створити свої унікальні худі та насолодитись персоналізованим досвідом покупки.

Виходячи з діаграми варіантів використання розроблювана музична платформа повинна у повній мірі забезпечувати вище зазначений функціонал, Тому з опису основних обов'язків актора можна сформулювати ключові варіанти використання системи.

Короткий опис варіантів використання наведено в таблиці 1.1.

Таблиця 1.1 – Варіантів використання веб-сайту “Hoodie customize”

Актор	Найменування	Формулювання
Unauthorized user	Sign in / sign up	Для можливість доступу використання функціоналу веб-сайту, автентифікації
Authorized user	Перегляд, додавання, редагування, видалення, прослуховування, вподобання треків	Перегляд, купівля, користування кошиком, порівняння
Admin	Перегляд, додавання, редагування, видалення	Перегляд, додавання, редагування, видалення, в тому числі й інших користувачів

Таким чином, виходячи з табличних даних варіантів використання веб-сайту, можна зрозуміти як кожен з акторів повинен взаємодіяти з системою.

#### **1.4           Опис ключових варіантів використання веб-сайту “ Hoodie customize ”**

Взаємодія кожного з акторів на веб-сайті відбувається по-різному, кожен актор має свою власну схему взаємодії та очікує від системи конкретної поведінки та реакції. Цю взаємодію акторів можна проілюструвати за допомогою UML діаграми.

Таким чином, для неавторизованого користувача UML діаграма виглядатиме наступним чином:

Рисунок 1.2 – діаграма варіантів використання не авторизованого користувача  
Авторизувавшись, користувач має значно більше можливостей:

Рисунок 1.3 – діаграма варіантів використання авторизованого користувача

Окрім можливостей, доступних користувачам, адміністратор має додаткові повноваження щодо керування користувачами.

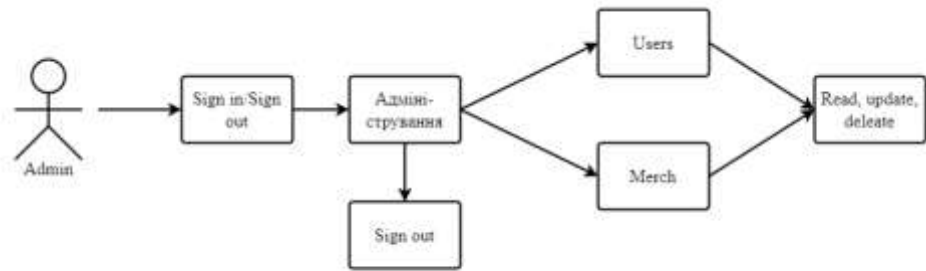


Рисунок 1.4 – діаграма варіантів використання адміністратора

Порівнявши діаграми, можна зробити висновок, що неавторизований користувач обмежений у доступі до функціоналу веб-сайту, тоді як адміністратор має більш широкі повноваження порівняно з користувачем. Адміністратор має можливість керувати записами користувачів, зокрема редагувати та видаляти їх облікові записи.

## 1.5 Вибір середовища розробки веб-сайту “Hoodie customize”

Вибір середовища та технологій розробки веб-сайту "Hoodie Customize".

У сучасному світі веб-сайти є важливою складовою успішного бізнесу. Один з ключових етапів розробки веб-сайту - вибір середовища та технологій, які будуть використовуватися для створення функціонального, зручного та ефективного веб-сайту. У цьому підрозділі буде розглянуто вибір середовища розробки та технологій, які були використані при створенні веб-сайту "Hoodie Customize".

Веб-сайт "Hoodie Customize" спеціалізується на кастомізації худі, де користувачі можуть створити свій унікальний дизайн та замовити персоналізовані худі. Для розробки такого веб-сайту було зроблено розумний вибір середовища розробки та технологій, щоб забезпечити швидку розробку, гнучкість, безпеку та ефективність сайту.

Середовище розробки, в якому був створений веб-сайт "Hoodie Customize", - Visual Studio Code. Це популярний та потужний текстовий редактор, який забезпечує зручне та продуктивне робоче середовище для розробки веб-додатків. Visual Studio Code має широкі можливості для розробників, такі як підсвічування синтаксису, автодоповнення, вбудована система керування версіями, розширення та інструменти для розробки на різних мовах програмування, включаючи Python. Visual Studio Code забезпечує зручний

інтерфейс та навігацію, що сприяє швидкому та продуктивному процесу розробки. [3]

Технології, використані у розробці веб-сайту "Hoodie Customize", включають Python та Django. Python є потужною та популярною мовою програмування, відомою своєю простотою та читабельністю коду. Django, у свою чергу, є потужним фреймворком для розробки веб-додатків на Python. Він надає готові компоненти та інструменти, що спрощують розробку, такі як система маршрутизації, управління базами даних та адміністративний інтерфейс.

Об'єднання Python та Django дозволило розробникам ефективно створити веб-сайт "Hoodie Customize" з використанням кастомізованої функціональності та забезпеченням безпеки даних. Python забезпечує гнучкість та можливість розширення, а Django допомагає управляти різними аспектами веб-сайту, включаючи обробку запитів, маршрутизацію та взаємодію з базою даних. [1]

Вибір середовища розробки Visual Studio Code та використання технологій Python та Django дозволили розробникам створити функціональний та ефективний веб-сайт "Hoodie Customize" з високим рівнем користувацького досвіду та можливістю кастомізації худі для клієнтів.

### **1.5.1 Мова програмування**

Мова програмування для розробки веб-сайту "Hoodie Customize". Вибір правильної мови може вплинути на продуктивність, швидкодію, безпеку та розширюваність веб-сайту. У цьому підрозділі буде розглянуто мову програмування, яка була використана для розробки веб-сайту "Hoodie Customize".

Мовою програмування, обраною для розробки веб-сайту "Hoodie Customize", є Python. Python - це високорівнева, інтерпретована мова програмування, відома своєю простотою, елегантністю та читабельністю коду. Вона має широкий спектр застосувань, включаючи веб-розробку, наукові обчислення, штучний інтелект та багато іншого.

Python має численні переваги, які зробили його привабливим вибором для розробки веб-сайту "Hoodie Customize". По-перше, Python має велику активну спільноту розробників, яка надає підтримку, документацію та розширення, що полегшують розробку. Крім того, Python має багато сторонніх бібліотек і фреймворків, що спрощують роботу з веб-розробкою, таких як Django, Flask, Pyramid та багато інших.

Одним з ключових фреймворків Python для веб-розробки є Django, який був використаний у розробці веб-сайту "Hoodie Customize". Django забезпечує широкий функціонал для розробки веб-додатків, включаючи обробку запитів, маршрутизацію, роботу з базами даних та адміністративний інтерфейс. Використання Django спрощує розробку, полегшує підтримку та забезпечує безпеку даних.

Крім того, Python має чудову інтеграцію з іншими технологіями та мовами програмування, що робить його гнучким вибором для веб-розробки. Наприклад, можна використовувати JavaScript для взаємодії з користувачем на стороні клієнта, а Python використовувати на серверній стороні для обробки запитів та взаємодії з базою даних.

## 1.5.2 База даних

Веб-сайт "Hoodie Customize" потребує потужної та ефективної бази даних для зберігання та управління інформацією про користувачів, замовлення, налаштування худі та інші важливі дані. У цьому підрозділі розглянемо роль та вибір бази даних для цього веб-сайту.

Одним з популярних виборів для бази даних у розробці веб-сайтів є реляційні бази даних (RDBMS). Такі бази даних використовують структуровану схему, яка складається з таблиць, колонок та відношень між ними. Один з відомих RDBMS - PostgreSQL, може бути відмінним вибором для "Hoodie Customize".

PostgreSQL - це потужна та надійна відкрита реляційна база даних, яка надає широкий функціонал та розширення. Вона підтримує мову запитів SQL, що дозволяє ефективно працювати з даними та виконувати складні операції. PostgreSQL також має розширення, які дозволяють працювати з географічними даними, повнотекстовим пошуком та багатьма іншими функціями, що можуть бути корисними для "Hoodie Customize".

Однак, розробка веб-сайту може також вимагати розгляду інших типів баз даних. Наприклад, NoSQL бази даних можуть бути корисними для зберігання неструктурованих даних, таких як зображення, відео чи аудіофайли. MongoDB є одним з відомих NoSQL рішень, яке може бути використано для таких потреб.

Вибір бази даних для "Hoodie Customize" також залежатиме від потреб та масштабу проекту. Якщо веб-сайт очікується мати велику кількість користувачів, то можливо,

варто розглянути розподілені бази даних, які забезпечують горизонтальне масштабування та високу доступність.

Крім вибору бази даних, важливо також звернути увагу на оптимізацію запитів, індексування та забезпечення безпеки даних. Налаштування бази даних, вибір відповідних індексів та використання кешування можуть значно покращити продуктивність веб-сайту та відповідати потребам користувачів.

У підсумку, вибір бази даних для веб-сайту "Hoodie Customize" є важливим кроком у розробці. Реляційні бази даних, такі як PostgreSQL, або NoSQL бази даних, такі як MongoDB, можуть бути розглянуті залежно від потреб проекту. Незалежно від вибору, оптимізація та забезпечення безпеки даних є ключовими аспектами для успішного функціонування веб-сайту "Hoodie Customize".

База даних для веб-сайту "Hoodie Customize" була обрана на основі вбудованої підтримки SQL бази даних в фреймворку Django. Django пропонує використання реляційної бази даних для зберігання та управління даними. Вбудований SQL двигун Django дозволяє розробникам працювати з реляційними базами даних, такими як SQLite, PostgreSQL, MySQL та іншими, зручним та стандартизованим способом.

Один із варіантів бази даних, який може бути використаний з Django, це SQLite. Він є простим та легким використовувати, ідеальним для розробки та тестування локально на початкових етапах проекту. SQLite зберігає всю базу даних в одному файлі, що спрощує розгортання та управління.

Однак, для більш масштабованих проектів, де передбачається велика кількість користувачів та операцій з базою даних, можна розглянути використання інших SQL баз даних, таких як PostgreSQL або MySQL. Ці бази даних забезпечують більшу масштабованість та продуктивність, а також можуть бути легко налаштовані для резервного копіювання та відновлення даних. [2]

Джерело даних в Django представляє собою моделі, які визначають структуру та взаємозв'язки між таблицями бази даних. Вони використовуються для створення, зчитування, оновлення та видалення даних в базі даних. Django надає ORM (Object-Relational Mapping), яке дозволяє працювати з базою даних за допомогою об'єктів та методів, замість написання прямих SQL запитів.

Особливістю використання SQL бази даних в Django є автоматичне створення та оновлення схеми бази даних на основі визначених моделей. Django забезпечує міграції

даних, які автоматично створюють та оновлюють таблиці бази даних з урахуванням змін у моделях. Це дозволяє зручно розгортати та зберігати консистентну структуру бази даних протягом розвитку проекту.

Крім вбудованої підтримки SQL бази даних, Django також надає можливості роботи з іншими базами даних та додатковими розширеннями. Він може інтегруватись з NoSQL базами даних, такими як MongoDB або Redis, для вирішення певних завдань, які вимагають нереляційного підходу до зберігання даних.

У підсумку, вибір SQL бази даних в Django для веб-сайту "Hoodie Customize" є хорошим рішенням, оскільки Django надає зручний та потужний спосіб роботи з реляційними базами даних. Використання вбудованої підтримки SQL дозволяє зосередитись на розробці функціональності веб-сайту, забезпечуючи одночасно надійне та ефективно зберігання та управління даними.

## **1.6 Обґрунтування використаних технологій розробки веб-сайту "Hoodie customize"**

Вибір середовища розробки є важливим етапом при створенні веб-сайту "Hoodie Customize". У даному випадку, Visual Studio Code було обрано як основне середовище розробки з кількох причин.

1. Мультиплатформенність: Visual Studio Code підтримує роботу на різних операційних системах, таких як Windows, macOS та Linux. Це дозволяє розробникам працювати у зручному середовищі на платформі свого вибору.

2. Зручний інтерфейс: Visual Studio Code має інтуїтивно зрозумілий інтерфейс з багатьма корисними функціями, такими як автодоповнення, відладка коду, керування версіями та інше. Це дозволяє розробникам покращити продуктивність та ефективність роботи.

3. Розширюваність: Visual Studio Code підтримує велику кількість розширень, які дозволяють налаштувати середовище розробки під свої потреби. Це дозволяє використовувати різноманітні інструменти та розширення, що полегшують розробку веб-сайту.

4. Підтримка Python та Django: Visual Studio Code надає потужну підтримку для розробки на мові програмування Python та фреймворку Django. Вона включає функції,



такі як підсвічування синтаксису, автодоповнення коду, відлагодження, підтримку віртуальних середовищ та багато іншого. Це дозволяє розробникам зручно працювати з Python та Django, використовуючи Visual Studio Code як основне середовище.

5. Активна спільнота та підтримка: Visual Studio Code має велику та активну спільноту розробників, яка забезпечує підтримку, навчання та обмін досвідом. Це дає змогу швидко вирішувати проблеми та отримувати актуальну інформацію про розробку веб-сайту.

Враховуючи всі ці переваги, вибір Visual Studio Code для розробки веб-сайту "Hoodie Customize" є обґрунтованим. Він надає зручне та потужне середовище розробки, підтримку Python та Django, а також багато інших інструментів та розширень, що допомагають зробити процес розробки більш ефективним та продуктивним.[3]

Обґрунтування використання мови програмування Python для розробки веб-сайту "Hoodie Customize" базується на кількох ключових факторах:

1. Простота та зрозумілість: Python є однією з найбільш зрозумілих та легких у вивченні мов програмування. Його синтаксис наближений до англійської мови, що робить код читабельним і зрозумілим. Це дозволяє розробникам швидко оволодіти мовою та приступити до створення функціоналу веб-сайту без зайвих ускладнень.

2. Велика кількість готових бібліотек та фреймворків: Python має широкий вибір бібліотек та фреймворків, які значно спрощують розробку веб-сайтів. Django, який був використаний для розробки "Hoodie Customize", є одним з найпопулярніших фреймворків Python для веб-розробки. Він надає потужні інструменти для швидкого створення веб-додатків, забезпечує безпеку, масштабованість та інші корисні функції.

3. Велика активна спільнота: Python має велику та активну спільноту розробників, яка постійно вносить внески у вдосконалення мови та розробку додаткових інструментів. Це означає, що розробники отримують широку підтримку, актуальну документацію, відповіді на питання та рішення проблем.

4. Масштабованість та розширюваність: Python є мовою програмування, яка добре підходить для розробки масштабованих веб-сайтів. Вона підтримує архітектурні принципи, такі як об'єктно-орієнтоване програмування (ООП), що дозволяє легко розбити функціонал на модулі та компоненти. Крім того, Python дозволяє легко розширювати функціонал веб-сайту шляхом підключення сторонніх бібліотек та модулів.

5. Широке застосування: Python є універсальною мовою програмування, яка знайшла застосування в багатьох галузях, включаючи веб-розробку. Це означає, що наявність досвідчених розробників Python більш ймовірна, а також є багато матеріалів та ресурсів для навчання.

Загалом, використання мови програмування Python для розробки веб-сайту "Hoodie Customize" було обґрунтованим через її простоту, широкі можливості, наявність популярних фреймворків та активну спільноту розробників. Вона дозволила ефективно створити функціональний та масштабований веб-сайт для задоволення потреб користувачів. [4]

Обґрунтування використання фреймворку Django для розробки веб-сайту "Hoodie Customize" базується на декількох важливих факторах:

1. Прискорення розробки: Django пропонує широкий набір готових компонентів і інструментів, що дозволяє розробникам ефективно створювати функціональні веб-сайти швидше. Готові модулі, такі як аутентифікація, управління формами, маршрутизація та адміністративний інтерфейс, допомагають прискорити процес розробки і знизити кількість написаного коду.

2. Масштабованість: Django розроблений з урахуванням масштабованості веб-додатків. Він пропонує вбудовану підтримку кешування, реплікації бази даних, масштабування горизонтально та вертикально, що дозволяє збільшити продуктивність та обробляти великий потік запитів.

3. Безпека: Django має вбудовані механізми безпеки, які допомагають уникнути багатьох типових вразливостей, таких як SQL-ін'єкція, CSRF-атаки, XSS-атаки тощо. Вбудований механізм аутентифікації та авторизації дозволяє контролювати доступ користувачів до функціоналу сайту.

4. Спільнота та документація: Django має широку та активну спільноту розробників, що забезпечує підтримку, допомогу та постійний розвиток фреймворку. Крім того, Django має добре організовану та актуальну документацію, що спрощує вивчення та використання фреймворку.

5. Підтримка бази даних: Django надає розширену підтримку для різних типів баз даних, включаючи SQLite, PostgreSQL, MySQL та інші. Це дозволяє використовувати потужні та надійні бази даних залежно від потреб проекту.

Усі ці фактори роблять Django відмінним вибором для розробки веб-сайту "Hoodie

Customize". Фреймворк надає зручність, продуктивність, безпеку та масштабованість, що допомагають створити потужний та функціональний веб-сайт з високоякісними функціями та гарним дизайном.

## **1.7 Висновок до першого розділу**

У цьому чаті ми розглянули різні аспекти розробки веб-сайту "Hoodie Customize". Розпочавши з обговорення предметної області і формування вимог до сайту, ми перейшли до аналізу актантів та варіантів використання, а також до опису вибраних технологій та середовищ розробки.

Під час аналізу предметної області "Розроблення інтернет-магазину з кастомізованими худі" було розглянуто ключові варіанти використання веб-сайту, що дозволяють користувачам налаштовувати та замовляти персоналізовані худі. Окрему увагу було приділено акторам, які взаємодіють з системою, та їх ролям у функціонуванні веб-сайту.

Далі ми розглянули вибір технологій для розробки веб-сайту "Hoodie Customize". Було обґрунтовано використання Visual Studio Code як середовища розробки, яке надає зручні інструменти для написання та налагодження коду. Також було обґрунтовано використання мови програмування Python, зокрема фреймворку Django, як потужного і простого у використанні засобу для розробки веб-додатків. Python дозволяє швидко розробляти функціонал сайту і має велику спільноту розробників, а Django надає готові рішення для широкого спектру завдань, включаючи управління користувачами, взаємодію з базою даних та безпеку.

Окремо було розглянуто використання бази даних у веб-сайті "Hoodie Customize". Було вибрано базу даних, яка є вбудованою в Django, а саме SQL базу даних. Це дозволяє зручно працювати з даними сайту, використовувати міграції для збереження структури бази даних та забезпечувати надійність і швидкість доступу до інформації.

Крім того, було обґрунтовано вибір фреймворку Django для розробки веб-сайту "Hoodie Customize". Django надає широкий функціонал, гнучкість у розробці, безпеку та масштабованість. Він допомагає створити потужний веб-сайт з ефективною адміністрацією, зручною навігацією та безпечною обробкою даних.

У висновку, вибір Visual Studio Code, мови програмування Python та фреймворку

Django для розробки веб-сайту "Hoodie Customize" обґрунтований їхніми перевагами, які допомагають прискорити процес розробки, забезпечити безпеку та масштабованість, а також надати зручний та функціональний фронтенд та бекенд для користувачів. Використання цих технологій допоможе створити веб-сайт, який задовольнить потреби користувачів і забезпечить їм зручний та персоналізований досвід.

## **2. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-САЙТУ “HOODIE CUSTOMIZE”**

### **2.1 Клієнт-серверна архітектура веб-сайту “Hoodie customize”**

Клієнт-серверна архітектура є ключовим компонентом багатьох сучасних веб-сайтів, в тому числі і веб-сайту "Hoodie Customize". Ця архітектура дозволяє ефективно розподілити функціональність та взаємодію між клієнтською та серверною сторонами системи, що забезпечує гнучкість, масштабованість та безпеку.

На початку взаємодії клієнтська сторона (браузер користувача) ініціює запит до серверної сторони веб-сайту "Hoodie Customize". Цей запит може бути пов'язаний з отриманням статичного контенту, такого як HTML, CSS та JavaScript-файли, або з виконанням динамічних операцій, таких як створення замовлення або оновлення користувацької інформації.

Серверна сторона, побудована з використанням фреймворку Django, отримує цей запит і обробляє його. За допомогою Python-скриптів та Django-функцій, сервер генерує відповідь, яка може містити додатковий HTML-код, дані з бази даних або іншу інформацію, яка потрібна для клієнта.

Після отримання відповіді сервер передає її клієнту, який відображає її на стороні користувача. Браузер інтерпретує HTML, виконує JavaScript і застосовує CSS для відображення сторінки. Таким чином, користувач отримує актуальну інформацію та може взаємодіяти з веб-сайтом, роблячи замовлення, змінюючи налаштування чи переглядаючи товари.

Після взаємодії клієнта з веб-сайтом "Hoodie Customize" шляхом створення замовлення, зміни користувацької інформації або перегляду товарів, дані відправляються з браузера користувача на серверну сторону для подальшої обробки і збереження.

Коли клієнт розміщує замовлення, дані про вибраний товар, кількість, розмір та інші параметри передаються через мережу на сервер. Серверна сторона веб-сайту отримує ці дані і виконує необхідні операції для обробки замовлення.

У випадку зміни користувацької інформації, такої як адреса доставки або контактні дані, браузер користувача надсилає введені дані на сервер. Серверна сторона обробляє ці

дані і оновлює відповідні записи користувача в базі даних. Це дозволяє зберігати актуальну інформацію про користувача і його налаштування.

Отримавши відповідь від сервера, браузер користувача інтерпретує HTML-код і відображає сторінку, використовуючи зазначені CSS-стилі для візуального оформлення. JavaScript-скрипти виконуються, надаючи додаткову динаміку сторінці та забезпечуючи можливість взаємодії з елементами веб-сайту. Наприклад, вони можуть забезпечувати можливість додавання товарів у кошик, переключення між сторінками, асинхронне завантаження даних та інші функції. Архітектура веб-сайту "Hoodie customize" показана на рисунку 2.1 нижче.

Одним з основних переваг використання клієнт-серверної архітектури є розділення відповідальності між клієнтською та серверною сторонами. Це дозволяє забезпечити гнучкість у розробці, так як розробники можуть працювати над фронтендом і бекендом незалежно один від одного. Крім того, така архітектура сприяє масштабованості, оскільки можна додавати додаткові сервери для обробки більшої кількості запитів. Також, це забезпечує безпеку, оскільки важлива логіка та обробка даних відбуваються на серверній стороні, що дозволяє зменшити ризик витоку інформації або зловживання доступом.

Таким чином, клієнт-серверна архітектура, використана у веб-сайті "Hoodie Customize", є ефективним та надійним рішенням. Вона дозволяє забезпечити ефективну взаємодію між користувачем і системою, забезпечує масштабованість, безпеку та гнучкість у розробці. Ця архітектура допомагає створити веб-сайт, який задовольняє потреби користувачів і забезпечує їм зручний та персоналізований досвід. [6]

Рисунок 2.1 – Архітектура веб-сайту на django "Hoodie customize"

## 2.2 Вибір основних залежностей для розробки веб-сайту “Hoodie customize”

В процесі розробки веб-сайту "Hoodie Customize" були використані різні залежності, що допомогли забезпечити необхідну функціональність та ефективність сайту. Основні залежності, які були використані, включають:

1. Django (версія 3.2.19): Django є основним фреймворком, на якому ґрунтується веб-сайт. Ця залежність дозволяє швидко розробляти потужні веб-додатки, надає готові компоненти для роботи з базою даних, URL-рутингу, автентифікації та багато іншого.

2. Pillow (версія 9.5.0): Pillow є бібліотекою для обробки зображень у Python. Вона дозволяє працювати зі зображеннями, змінювати їх розмір, обрізати, робити зміни в їх форматі тощо. У веб-сайті "Hoodie Customize" використовується Pillow для роботи з зображеннями худі.

3. requests (версія 2.25.1): Бібліотека requests дозволяє здійснювати HTTP-запити з Python. Вона використовується для взаємодії з зовнішніми сервісами або API, що можуть бути необхідними для функціонування веб-сайту "Hoodie Customize".

4. python-decouple (версія 3.4): Ця бібліотека дозволяє зберігати конфігураційні параметри в окремому файлі. Вона допомагає зберігати конфіденційну інформацію, таку як ключі API або параметри підключення до бази даних, окремо від вихідного коду.

5. Залежність `sqlparse` є однією з ключових компонентів, використовуваних для розробки веб-сайту "Hoodie Customize". Вона має на меті аналізувати та розбирати SQL-запити, що використовуються в системі.

Основна функціональність `sqlparse` полягає в розборі SQL-запитів на компоненти, такі як ключові слова, оператори, ідентифікатори, значення та інші елементи, які складають запит. Це дозволяє розробникам легко аналізувати та обробляти SQL-запити, забезпечуючи більшу гнучкість та контроль над взаємодією з базою даних.

Деякі основні можливості `sqlparse` включають:

– Розпізнавання структури запиту: `sqlparse` дозволяє розпізнавати структуру SQL-запиту та виділяти його основні компоненти, такі як SELECT, FROM, WHERE, GROUP BY, ORDER BY тощо.

- Форматування SQL-коду: `sqlparse` забезпечує можливість форматування SQL-коду за заданими стандартами або правилами форматування.
  - Валідація та перевірка синтаксису: `sqlparse` може перевіряти правильність синтаксису SQL-запитів та виявляти можливі помилки.
  - Розбір вкладених запитів: `sqlparse` може розбирати вкладені SQL-запити та виділяти їх окремо.
  - Залежність `sqlparse` інтегрується з іншими компонентами веб-сайту, особливо з фреймворком Django та його ORM (Object-Relational Mapping).
6. інші залежності: `asgiref`, `certifi`, `chardet`, `django-admin-honeypot`, `django-admin-thumbnails`, `django-session-timeout`, `idna`, `pytz`, `six`, `urllib3`. Ці залежності включають різні компоненти, які підтримують роботу фреймворку Django та забезпечують додаткові можливості та функціональність.

Використання цих залежностей допомогло спростити процес розробки веб-сайту "Hoodie Customize" та забезпечити необхідні функціональність та ефективність. Вони дозволили швидко створювати веб-додатки, працювати з базою даних, зображеннями та взаємодіяти з іншими сервісами. Залежності забезпечили гнучкість, розширюваність та безпеку веб-сайту "Hoodie Customize". [7]

Крім основних залежностей, зазначених вище, також були використані інші технології та компоненти для розробки веб-сайту "Hoodie Customize":

- HTML/CSS: HTML є основною мовою розмітки для створення структури веб-сторінок, а CSS використовується для стилізації та візуального оформлення. Вони необхідні для побудови користувацького інтерфейсу та забезпечення зручного та привабливого вигляду веб-сайту.
- JavaScript: JavaScript використовується для реалізації динамічного та інтерактивного поведінки на веб-сторінках. Він дозволяє додавати веб-сайту функції, включаючи анімацію, перехоплення подій, валідацію форм та взаємодію з користувачем.
- Bootstrap: Bootstrap є популярним фреймворком CSS, який надає готові компоненти та шаблони для швидкої розробки веб-інтерфейсів. Bootstrap дозволяє створювати привабливий дизайн, забезпечуючи однорідний вигляд на різних пристроях.
- Git: Git є системою керування версіями, що дозволяє відстежувати зміни в коді та співпрацювати з іншими розробниками. Використання Git допомагає управляти розробкою, вносити зміни та відновлювати попередні версії коду.



Використання цих технологій та компонентів дозволило забезпечити надійність, ефективність та зручність розробки веб-сайту "Hoodie Customize". Кожна з них виконує свою специфічну роль у створенні інтерактивного та привабливого веб-додатку.

Список залежностей показаний на рисунку 2.2 нижче:

Рисунок 2.2 – Список залежностей веб-сайту “Hoodie customize”

### 2.3 Файлова структура веб-сайту “Hoodie customize”

Файлова структура веб-сайту "Hoodie Customize" грає важливу роль у організації та управлінні всіма необхідними файлами, директоріями та налаштуваннями, що використовуються під час розробки та функціонування веб-сайту.

Рисунок 2.3 – Файлова структура веб-сайту “Hoodie Customize”

Якщо розглядати більше детально то ця структура буде виглядати наступним чином:

Папка "accounts": Ця папка містить файли, пов'язані з аутентифікацією, реєстрацією та управлінням користувачами на веб-сайті. Вона може містити файли для обробки реєстрації, входу, виходу, профілю користувача та інших відповідних функцій.

Папка "carts": У цій папці зберігаються файли, що стосуються кошиків покупок на веб-сайті. Вона може включати файли для додавання товарів до кошика, оновлення кількості товарів, видалення товарів та інших операцій, пов'язаних з кошиком.

Папка "category": Ця папка містить файли, пов'язані з категоріями товарів на веб-сайті. Вона може містити файли для відображення категорій, додавання нових категорій, редагування та видалення існуючих категорій.

Папка "env": У цій папці зберігаються файли, що містять конфігураційні параметри та змінні середовища для веб-сайту. Наприклад, файл `.env` може містити конфігураційні параметри, які потрібні для підключення до бази даних, налаштування розсилки електронних листів тощо.

Папка "greatkart": Ця папка є основною папкою проекту та містить налаштування, маршрутизацію, моделі, представлення та інші складові, пов'язані з основним функціоналом веб-сайту.

Папка "media": У цій папці зберігаються медіа-файли, такі як зображення, відео чи аудіо, які використовуються на веб-сайті. Вона може містити файли товарів, зображень

користувачів та інших відповідних медіа-елементів.

Папка "orders": Ця папка містить файли, що стосуються замовлень на веб-сайті. Вона може включати файли для оформлення замовлень, відстеження статусу замовлень, підтвердження платежів та інших функцій, пов'язаних з замовленнями.

Папка "static": У цій папці зберігаються статичні файли, такі як CSS-стилі, JavaScript-скрипти та інші ресурси, які використовуються для візуального оформлення та функціональності веб-сайту.

Папка "store": Ця папка містить файли, пов'язані з відображенням та управлінням продуктами на веб-сайті. Вона може містити файли для перегляду деталей продукту, додавання продукту до кошика, пошуку та фільтрації продуктів та інших відповідних функцій.

Папка "templates": У цій папці знаходяться шаблони HTML-сторінок, які використовуються для відображення контенту на веб-сайті. Кожен шаблон може мати свій власний файл HTML, що відповідає конкретній сторінці або компоненту веб-сайту.

Окрім папок, у файловій структурі також присутні деякі файлові елементи, які варто відзначити:

`.env` та `.env_sample`: Це файли конфігурації, які містять конфіденційні дані та змінні середовища, які використовуються в проєкті. Файл `.env` містить реальні значення, тоді як `.env_sample` містить приклади та шаблони для заповнення.

`.gitignore`: Цей файл використовується для вказівки системі контролю версій Git про файли та папки, які необхідно ігнорувати під час збереження змін.

`db.sqlite3`: Це файл бази даних SQLite, який використовується для збереження даних веб-сайту.

`manage.py`: Цей файл є виконуваним скриптом Django і використовується для взаємодії з різними аспектами веб-сайту, такими як міграції бази даних, запуск сервера розробки тощо.

`README`: Це файл, який містить коротку документацію або інструкції щодо веб-сайту.

`requirements.txt`: Цей файл містить список залежностей та пакетів Python, які необхідні для встановлення та запуску веб-сайту.

Файлова структура "Hoodie Customize" добре організована та структурована, що дозволяє розробникам легко зорієнтуватись у проєкті та здійснювати розширення та

підтримку веб-сайту. Кожна папка та файл має своє визначення та призначення, що сприяє зручності та ефективності розробки.

Структура веб-сайту "Hoodie Customize" можна уявити як дерево, де початкові вузли залишаються незмінними. Ці початкові вузли представлені макетами програми, які є основою сайту. Далі, у кожній папці відбувається поділ на різні гілки, які можуть містити додаткові папки або файли, які виконують роль додаткових гілок структури. Цей поділ утворює ієрархічну структуру, де кожен рівень відображає розгалуження та розширення можливостей сайту. Кожна гілка може містити свої власні підгілки та додаткові файли, створюючи багаторівневу структуру, яка відображає взаємозв'язки та організацію різних компонентів сайту.

## **2.4 Структура та особливості використаної БД для розробки веб-сайту “Hoodie Customize”**

Веб-сайт "Hoodie Customize" використовує базу даних (БД) для збереження та управління інформацією, пов'язаною з продуктами, користувачами, замовленнями та іншими аспектами функціональності сайту. У розробці цього веб-сайту була використана вбудована БД, яка пропонується самим фреймворком Django - SQLite.

SQLite - це легка, вбудована, реляційна система керування базами даних, яка зберігає дані у локальних файлових сховищах. Одна з особливостей SQLite полягає в тому, що вона не вимагає окремого сервера баз даних для роботи, а замість цього працює безпосередньо з файловою системою. Це робить її легко використовувати та налаштовувати, зокрема для проектів невеликого розміру або випадків, коли не потрібна розподілена архітектура бази даних.

У структурі проекту "Hoodie Customize" можна знайти файл з розширенням `.sqlite3`, який представляє собою сам файловий контейнер бази даних SQLite. Цей файл містить таблиці, які відповідають сутностям (таким як продукти, користувачі, замовлення тощо), а також колонки, які описують поля цих сутностей.

З використанням мови програмування Python та фреймворку Django, розробники здійснюють маніпуляції з БД, такі як створення таблиць, зміна схеми, додавання,

видалення або оновлення записів. Django забезпечує ORM (об'єктно-реляційне відображення), яке дозволяє виконувати операції з базою даних за допомогою об'єктно-орієнтованого підходу, що полегшує роботу з БД та спрощує розробку.

Така структура БД на основі SQLite і фреймворку Django забезпечує надійне та ефективне зберігання даних, а також простоту в розробці та підтримці. Вона дозволяє зберігати всю необхідну інформацію, що стосується худі, користувачів, замовлень та інших аспектів функціональності "Hoodie Customize", і забезпечує швидкий доступ до цих даних для оптимального функціонування веб-сайту.

Концептуальну схему бази даних веб-сайту "Hoodie Customize" можна відобразити наступним чином (див рис. 2.4):

Рисунок 2.4 – Концептуальна схема бази даних веб-сайту на Django

Файловий контейнер бази даних цього веб-сайту, який використовує базу даних SQLite, містить таблиці, які представляють різні сутності та їх взаємозв'язки. Основні таблиці, присутні у цьому контейнері, включають:

- Таблиця "Користувачі" (Users): Ця таблиця зберігає інформацію про зареєстрованих користувачів веб-сайту, такі як їх ідентифікатор, ім'я, електронну пошту, хешований пароль тощо. Вона дозволяє здійснювати аутентифікацію та авторизацію користувачів.

- Таблиця "Профілі користувачів" (User Profiles): Ця таблиця містить додаткові дані про користувачів, такі як їхнє місце проживання, контактну інформацію, дату народження тощо. Вона дозволяє розширити профіль користувача та зберігати додаткові деталі.

- Таблиця "Варіанти" (Options): Ця таблиця містить інформацію про різні варіанти та налаштування, які доступні користувачам при кастомізації худі. Вона включає такі поля, як назва варіанту, опис, зображення тощо.

- Таблиця "Відгуки" (Reviews): Ця таблиця дозволяє користувачам залишати відгуки та оцінки про куплені худі. Вона містить інформацію про користувача, текст відгуку, оцінку і дату публікації.

- Таблиця "Галереї" (Galleries): Ця таблиця зберігає дані про різні галереї

зображень, пов'язаних з худі. Вона містить ідентифікатор галереї, назву, опис та шлях до зображень.

– Таблиця "Товари" (Products): Ця таблиця містить інформацію про доступні товари (худі) на веб-сайті. Вона включає поля, такі як назва товару, опис, ціна, зображення тощо.

Файловий контейнер бази даних використовує структуру таблиць та їх взаємозв'язки для зберігання та організації даних веб-сайту "Hoodie Customize". Ця структура дозволяє ефективно управляти даними користувачів, варіантів, відгуків, галерей та товарів, що підтримує функціональність веб-сайту та забезпечує задоволення потреб користувачів. [8]

## **2.5 Адміністрування веб-сайту "Hoodie customize"**

Адміністрування веб-сайту "Hoodie Customize" включає в себе керування та підтримку його роботи, забезпечення безпеки та виконання різних адміністративних завдань.

Адміністративна панель або адмін-панель - це набір функцій, доступних через спеціальний інтерфейс для адміністраторів. Ці функції сприяють виконанню різних завдань, таких як створення та редагування контенту, додавання настроюваних полів до контенту, управління користувачами та багато іншого.

Адміністративна панель дозволяє адміністраторам зручно керувати різними аспектами веб-сайту. Вона забезпечує доступ до різноманітних інструментів, які спрощують процес управління сайтом, забезпечують швидкий доступ до важливих функцій і забезпечують зручний інтерфейс для виконання різноманітних завдань.

Адміністративна панель дозволяє адміністраторам ефективно працювати з контентом, налаштовувати сайт під свої потреби, встановлювати права доступу для користувачів та забезпечувати потрібні функціональні можливості для оптимального управління веб-сайтом.

Основні обов'язки адміністратора веб-сайту включають:

– Керування користувачами: Адміністратор веб-сайту має можливість керувати користувачами, включаючи створення нових облікових записів, редагування інформації про користувачів, скидання паролів та видалення облікових записів.

– Керування змістом: Адміністратор веб-сайту відповідає за додавання, редагування та видалення змісту, такого як товари, категорії, відгуки, зображення тощо. Він також забезпечує належне форматування та відображення змісту на сайті.

– Безпека: Адміністратор веб-сайту відповідає за забезпечення безпеки сайту. Це включає установку оновлень, захист від зломів та зловмисного програмного забезпечення, резервне копіювання даних та моніторинг системи на наявність потенційних загроз.

– Налаштування сервера: Адміністратор веб-сайту виконує налаштування та підтримку сервера, на якому працює сайт. Це може включати конфігурування веб-сервера, бази даних, налаштування електронної пошти, моніторинг ресурсів сервера тощо.

– Відстеження проблем: Адміністратор веб-сайту відповідає за виявлення та вирішення проблем, які можуть виникнути на сайті. Він має знання про технічні аспекти сайту і здатність вирішувати проблеми швидко та ефективно.

– Моніторинг та аналітика: Адміністратор веб-сайту відстежує продуктивність сайту, моніторить трафік, швидкість завантаження сторінок та інші метрики. Він також здійснює аналіз даних для виявлення тенденцій, удосконалення функціональності та покращення користувацького досвіду.

Узагальнюючи, адміністрування веб-сайту "Hoodie Customize" передбачає керування користувачами, змістом, безпекою, налаштування сервера, відстеження проблем та моніторинг продуктивності. Відповідальність адміністратора полягає в забезпеченні безперебійної та ефективної роботи сайту, щоб задовольнити потреби користувачів та досягти поставлених цілей.

На цьому веб-сайті адміністрування відбувається за допомогою вбудованої адміністративної панелі Django. Django забезпечує готовий інструментарій для швидкого створення адміністративних інтерфейсів, що дозволяє зручно управляти різними аспектами веб-сайту. [9]

Адміністрування в панелі Django надає широкий набір функціональних можливостей для ефективного управління веб-сайтом. Вона забезпечує зручне та інтуїтивно зрозуміле інтерфейс користувача для виконання адміністративних завдань.

Адмін панель зображено на рисунку 2.5.

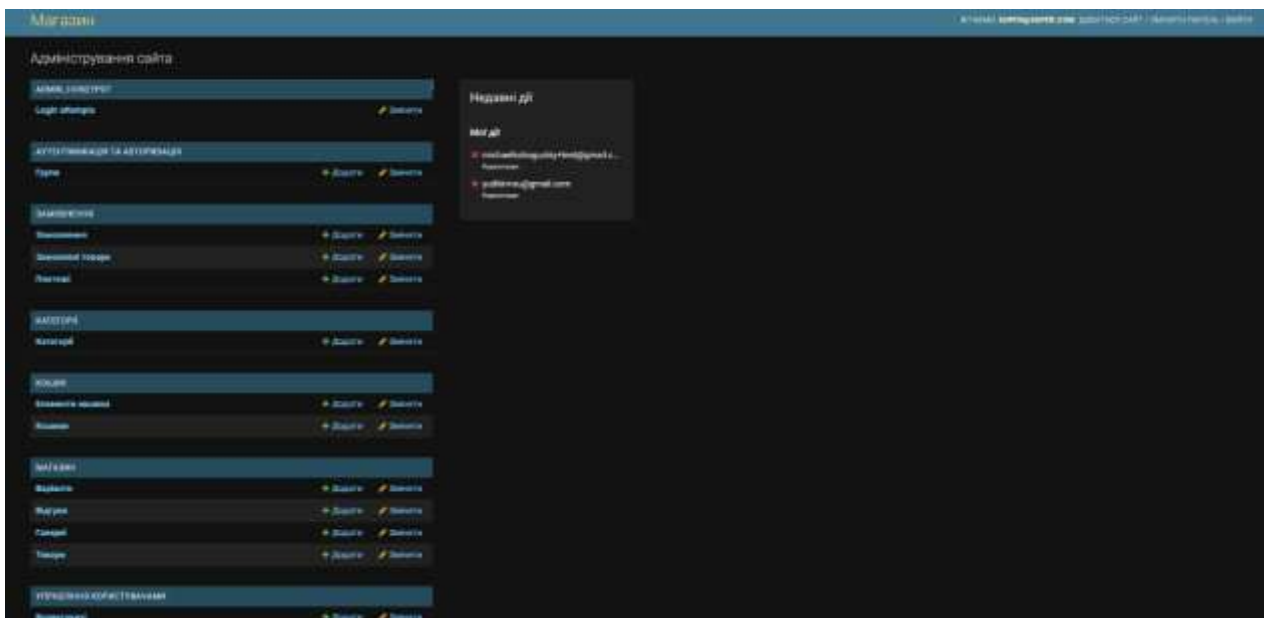


Рисунок 2.5 – Адмін панель Django для сайту “Hoodie Customize”

Ключові функції, які присутні в панелі адміністрування Django:

- **ADMIN\_HONEYPOT:** Це додатковий захисний шар, який може бути використаний для виявлення спроб несанкціонованого доступу до панелі адміністрування. Цей інструмент дозволяє виявити та заблокувати спроби вторгнення або неавторизованого доступу.
- **Login attempts:** Ця функція відстежує спроби входу до панелі адміністрування. Вона показує кількість спроб, час і місце з яких вони були здійснені. Це дозволяє адміністратору виявляти підозрілу активність та приймати заходи щодо забезпечення безпеки.
- **Аутентифікація та авторизація:** Панель адміністрування Django надає потужні засоби для управління аутентифікацією та авторизацією користувачів. Вона включає можливість створення користувачів, надання їм ролей та дозволів, а також контроль доступу до різних частин сайту.
- **Групи:** Ця функція дозволяє групувати користувачів зі схожими правами доступу до панелі адміністрування. Адміністратор може створювати групи, призначати їм ролі та дозволи, що спрощує управління доступом та забезпечує більш гнучку систему авторизації.
- **Замовлення:** Панель адміністрування Django надає можливість керувати замовленнями, що здійснюються на веб-сайті. Адміністратор може переглядати, редагувати, видаляти замовлення та керувати станом замовлення.

– Категорії: Ця функція дозволяє адміністратору керувати категоріями товарів на веб-сайті. Він може створювати нові категорії, редагувати існуючі, встановлювати їх порядок та здійснювати інші операції з категоріями.

– Кошик: Панель адміністрування дозволяє адміністратору переглядати та керувати елементами кошика користувачів. Він може переглядати замовлені товари, видаляти елементи кошика, керувати станом та вартістю кошика.

– Магазин: В панелі адміністрування є розділ, присвячений керуванню товарами в магазині. Адміністратор може додавати нові товари, редагувати їх властивості, встановлювати ціни, змінювати наявність та проводити інші операції, пов'язані з управлінням товарів.

– Управління користувачами: Панель адміністрування надає зручні інструменти для управління користувачами сайту. Адміністратор може переглядати, редагувати, блокувати або видаляти користувачів. Він також може керувати профілями користувачів та змінювати їх дані.

Панель адміністрування Django забезпечує зручний та потужний інтерфейс для виконання адміністративних завдань. Вона дозволяє адміністраторам ефективно керувати різними аспектами веб-сайту, такими як аутентифікація, авторизація, замовлення, категорії, кошик, магазин та користувачі. [9]

## **2.6 Розгортання веб-сайту “Hoodie customize” на хостингових платформах**

Розгортання веб-сайту "Hoodie Customize" на хостинг є важливим етапом в процесі його публікації та доступу для відвідувачів. Для розуміння розгортання варто ознайомитися з поняттями хостингу, домену та їх значенням.

Хостинг - це сервіс, який надає комп'ютерні ресурси та послуги для зберігання веб-сайту та його доступності через Інтернет. Провайдер хостингу забезпечує необхідне обладнання (сервери), програмне забезпечення та мережеві з'єднання для зберігання файлів веб-сайту та обробки запитів користувачів. [10]

Домен - це унікальна адреса, яка ідентифікує веб-сайт у мережі Інтернет. Він служить для легкого запам'ятовування і доступу до веб-сайту користувачами. Наприклад, домен "hoodiecustomize.com" може бути використаний для доступу до веб-



сайту "Hoodie Customize". [10]

Розгортання веб-сайту на хостинг передбачає кілька можливих варіантів:

**Спільний хостинг (Shared Hosting):** Веб-сайт розміщується на сервері, який використовується також іншими клієнтами хостинг-провайдера. Це економічний варіант, але обмежена кількість ресурсів.

**Віртуальний приватний сервер (Virtual Private Server - VPS):** Сервер фізично розділяється на декілька віртуальних серверів, кожен з яких має власні ресурси. Цей варіант надає більшу гнучкість та контроль над сервером.

**Власний фізичний сервер:** Клієнт купує або орендує фізичний сервер, на якому розміщує веб-сайт. Це найбільш гнучкий варіант, але вимагає більших витрат на обладнання та обслуговування.

Розгортання веб-сайту на хостингу дозволяє зробити його доступним для користувачів з усього світу. Це дає можливість просувати бренд, представляти продукти або послуги, спілкуватися з клієнтами та забезпечувати їм доступ до необхідного функціоналу. Крім того, хостинг забезпечує надійне зберігання даних, резервне копіювання та захист від втрати інформації.

Отже, розгортання веб-сайту на хостингу є необхідним етапом для публікації та доступу до нього. Вибір відповідного хостинг-провайдера та планування ресурсів залежить від потреб проекту, бюджету та масштабів діяльності. Це дозволяє забезпечити стабільну роботу веб-сайту і задоволення користувачів.

Вибір хостингу для веб-сайту "Hoodie Customize" залежить від декількох факторів, таких як розмір проекту, очікувана кількість відвідувачів, потреби в ресурсах та бюджет. Ось кілька популярних хостинг-провайдерів, які варто розглянути:

- **SiteGround:** Це ще один популярний вибір для хостингу. SiteGround пропонує швидкий та надійний хостинг з хорошою підтримкою. Вони також спеціалізуються на хостингу WordPress, що може бути корисним, якщо веб-сайт побудований на цій платформі.

- **DigitalOcean:** Якщо ви шукаєте більш гнучкий підхід до хостингу, DigitalOcean може бути хорошим варіантом. Вони пропонують хмарні сервери, де ви можете налаштовувати ресурси відповідно до потреб. Це підходить для розширення веб-сайту з часом.

Перед вибором хостингу рекомендується вивчити їх плани, ціни, підтримку,

надійність та можливості масштабування. Також корисно прочитати відгуки користувачів та звернутися до фахівців для отримання конкретних рекомендацій, що відповідають потребам та бюджету.

SiteGround і DigitalOcean є двома популярними хостинг-провайдерами, які мають свої особливості і підходять для різних типів проектів. Ось порівняння цих двох хостинг-провайдерів:

#### SiteGround:

- Орієнтований на веб-сайти на основі платформи WordPress і має оптимізовані плани хостингу для WordPress.
- Надає швидкі та надійні сервери з високою продуктивністю.
- Має дружній та легкий у використанні інтерфейс, що підходить для новачків.
- Забезпечує надійну підтримку клієнтів з швидкими відповідями на запити.

#### DigitalOcean:

- Зосереджений на хмарному хостингу і надає вам більшу гнучкість та контроль над серверами.
- Має можливість масштабування, дозволяючи вам збільшувати або зменшувати ресурси відповідно до потреб проекту.
- Надає вам доступ до кореневого рівня сервера, що дозволяє вам налаштовувати його згідно з потребами.
- Має розширений набір інструментів та можливостей для розробників.

Остаточний вибір між SiteGround і DigitalOcean залежить від конкретного проекту та потреб. Якщо ви шукаєте спрощений і оптимізований хостинг для веб-сайтів на WordPress, SiteGround може бути кращим варіантом. У разі, якщо ви бажаєте більшу гнучкість, контроль та можливість масштабування, DigitalOcean може відповідати потребам краще. [11]

Основні відмінності між платформами наведені в таблиці 2.1.

Таблиця 2.1 – Основні відмінності між DigitalOcean та SiteGround

	<b>SiteGround</b>	<b>DigitalOcean</b>
Опис	Орієнтований на веб-сайти на платформи	Зосереджений на хмарному хостингу
Дата заснування	2004 рік	2011 рік
Материнська компанія	Відсутня	Limited Liability Company
Кількість працівників	189	260
Гнучкість	Середня	Висока
Ціна	Підписка	Підписка
Масштабованість	Присутня	Присутня
Ідеально підходить для	Сучасної автоматизації веб-сайтів	Створення хмарних рішень, в тому числі БД
Постійна інтеграція	+	+
Фокус	Front-end	Back-end + Front-end
Тип проекту	Web	Web
Зауваження	-	-

Інструкція по встановленню сайту на хостинг DigitalOcean може виглядати так:

[12]

Крок 1: Реєстрація та налаштування облікового запису DigitalOcean.

- Відкрити веб-сайт DigitalOcean та створити обліковий запис.
- Зареєструватися та підтвердити електронну пошту.
- Створити SSH-ключі для безпечного з'єднання з сервером.

Крок 2: Створення та налаштування віртуального сервера (Droplet).

- У панелі керування DigitalOcean вибрати "Create" (Створити) і "Droplets" (Віртуальні сервери).
- Вибрати дистрибутив операційної системи, який має бути використовуваним (наприклад, Ubuntu).
- Вибрати розмір Droplet, та вказавши необхідні ресурси (пам'ять, процесор і т.д.).

- Вибрати регіон для розташування сервера.
- Додати SSH-ключ, щоб появилася можливість підключатися до сервера через SSH.

- Натиснути "Create" (Створити) для створення Droplet.

Крок 3: Підключення до сервера через SSH.

- Запустити термінал на комп'ютері.
- Ввести наступну команду, щоб підключитися до сервера:

```
ssh root@<IP-адреса-сервера>
```

Обов'язково замінити <IP-адреса-сервера> на фактичну IP-адресу сервера DigitalOcean.

- Ввести пароль, який було отримано від DigitalOcean.
- Успішне підключення до сервера через SSH.

Крок 4: Налаштування сервера та встановлення залежностей.

Оновити пакети операційної системи за допомогою наступної команди:

```
sudo apt install python3 python3-pip python3-venv  
sudo apt install nginx
```

- Створити віртуальне середовище Python для проекту та активувати його.
- Встановити необхідні пакети Python для проекту за допомогою pip.

Крок 5: Налаштування веб-сервера і розгортання сайту.

- Створити конфігураційний файл Nginx для сайту.
- Налаштувати Nginx, щоб проксіювати запити до веб-сервера.
- Скопіювати файли веб-сайту на сервер, включаючи файли Django-проекту та статичні файли.
- Запустити веб-сервер та перевірити, чи працює сайт.

## **2.7 Тестування та використання веб-сайту “Hoodie customize”**

### **2.7.1 Тестування веб-сайту “Hoodie customize”**

Тестування веб-сайту "Hoodie Customize" є важливою складовою частиною процесу розробки та забезпечення якості. Воно дозволяє виявляти помилки, недоліки та потенційні проблеми, а також переконується, що сайт працює належним чином та задовольняє потреби користувачів. Для тестування "Hoodie Customize" можуть бути використані наступні підходи та методи.

1. Функціональне тестування: Цей тип тестування перевіряє, чи працюють основні функції сайту "Hoodie Customize" належним чином. Тестувачі перевіряють, чи можна здійснити покупку товарів, налаштувати параметри худі, додати відгук або здійснити інші дії, які передбачені функціоналом сайту. Це включає перевірку роботи реєстрації користувачів, авторизації, додавання товарів у кошик, розміщення замовлення та інші функції, що є ключовими для сайту.

2. Тестування користувацького інтерфейсу: Цей тип тестування перевіряє зручність, естетику та доступність користувацького інтерфейсу сайту. Тестувачі аналізують, як легко користувачі можуть навігувати по сайту, знайти потрібну інформацію, взаємодіяти з різними елементами і виконувати дії. Також перевіряється, чи сайт працює на різних пристроях та браузерах, адаптується до різних розмірів екранів та забезпечує коректне відображення контенту.

3. Тестування продуктивності: Цей тип тестування оцінює швидкість та продуктивність сайту "Hoodie Customize". Виконуються навантажувальні тести, які демонструють, як швидко сайт відповідає на запити, як добре він витримує велику кількість одночасних користувачів та які обсяги даних він може обробляти. Це включає оцінку часу завантаження сторінок, швидкості обробки запитів сервером та оптимізацію продуктивності для забезпечення як найкращої користувацького досвіду.

4. Тестування безпеки: Важливо проводити тестування безпеки для переконання, що сайт "Hoodie Customize" має високий рівень захисту від потенційних загроз. Це включає перевірку наявності вразливостей, перехоплення даних, можливості несанкціонованого доступу та інших потенційних проблем безпеки. Тестувачі використовують різні методи, такі як сканування вразливостей, тестування на

проникнення та перевірку безпеки даних.

5. Тестування сумісності: Виконується тестування сумісності, щоб переконатися, що сайт "Hoodie Customize" працює на різних операційних системах, браузерах та пристроях. Це включає перевірку наявності проблем у відображенні, функціональності та інших аспектах взаємодії залежно від використаного середовища.

6. Тестування бази даних: Виконується тестування бази даних, щоб переконатися, що дані на сайті "Hoodie Customize" зберігаються, оновлюються та використовуються належним чином. Перевіряється коректність запитів до бази даних, відповідність структури бази даних вимогам та інші аспекти збереження даних.

Усі ці типи тестування допомагають забезпечити якість та надійність сайту "Hoodie Customize". Вони дозволяють виявляти помилки, оптимізувати продуктивність, забезпечувати безпеку та гарантувати задоволення користувачів. Результати тестування допомагають вдосконалити сайт та забезпечити його успішну експлуатацію у реальних умовах.

Для тестування веб-сайту на різноманітні аспекти використання та якості можна використовувати наступні сервіси:

- Google PageSpeed Insights: Цей інструмент від Google допомагає оцінити швидкість та продуктивність сайту на мобільних та настільних пристроях. Він надає рекомендації щодо оптимізації швидкості завантаження сторінки, кешування ресурсів, оптимізації зображень та іншого.

- GTmetrix: Цей сервіс надає докладні звіти щодо продуктивності сайту, зокрема швидкості завантаження, оптимізації зображень, кешування, стиснення ресурсів і багато іншого. Він також надає рекомендації для покращення продуктивності. [13]

- WebPageTest: Цей інструмент дозволяє тестувати швидкість завантаження сторінки з різних місць світу та на різних пристроях. Він надає докладну інформацію про час завантаження, розмір сторінки, зображень, запитів до сервера та інших метрик продуктивності.

- Pingdom Tools: Цей сервіс дозволяє перевіряти швидкість завантаження сторінки з різних серверів у різних місцях світу. Він надає детальні звіти про різні етапи завантаження сторінки та рекомендації щодо оптимізації.

- Sitechecker: Цей інструмент перевіряє сайт на різні аспекти оптимізації, включаючи SEO, швидкість завантаження, безпеку та інші технічні показники. Він надає

звіти з рекомендаціями щодо вдосконалення веб-сайту.

Ці інструменти можуть допомогти виявити проблеми та забезпечити рекомендації щодо покращення продуктивності, швидкості та якості веб-сайту.

Веб-сайт "Hoodie Customize" був підданий ретельному тестуванню з метою забезпечення його високої якості, продуктивності та безпеки. Для досягнення цих цілей були використані різні методи тестування та інструменти, включаючи GTmetrix, результат зображено на рисунку 2.6.

Рисунок 2.6 – Результати тестування GTmetrix

Перш за все, було проведено функціональне тестування, щоб переконатися, що всі функції сайту працюють належним чином. Команда тестувальників перевіряла різні сценарії взаємодії з сайтом, такі як додавання товарів до кошика, оформлення замовлень, відгуки користувачів та інші функціональні можливості. Помилки та некоректність були виявлені і виправлені, забезпечуючи належну роботу сайту.

Також було проведено тестування продуктивності, щоб переконатися, що сайт "Hoodie Customize" працює швидко та ефективно навіть при великому навантаженні. За допомогою спеціальних інструментів для тестування продуктивності, було виміряно час завантаження сторінок, швидкість обробки запитів та інші метрики продуктивності. Були внесені оптимізації та вдосконалення для покращення продуктивності сайту.

Аспекти безпеки також були важливим об'єктом тестування. Проводилося тестування на наявність вразливостей, перехоплення даних та несанкціонований доступ. Тестувачі використовували методи сканування вразливостей та тестування на проникнення, щоб забезпечити високий рівень безпеки сайту.

Для оцінки загальної продуктивності та визначення областей для подальшого вдосконалення був використаний інструмент GTmetrix. Цей інструмент дозволяє аналізувати швидкість завантаження сторінок, розмір файлів, оптимізацію зображень та інші параметри. Результати аналізу допомогли виявити можливі проблеми та недоліки, які були виправлені з метою поліпшення продуктивності сайту.

В цілому, завдяки ретельному тестуванню та використанню різних методів та інструментів, веб-сайт "Hoodie Customize" був оптимізований, забезпечуючи високу якість, продуктивність та безпеку. Тестування дозволило виявити й виправити

потенційні проблеми, що забезпечило задоволення користувачів та успішну експлуатацію сайту у реальних умовах.

### **2.7.2 Використання веб-сайту “Hoodie Customize”**

Розділи та підрозділи веб-сайту використовуються для групування відповідного контенту. Наприклад, якщо веб-сайт належить до магазину з одягом, можуть бути розділи, що присвячені різним категоріям одягу. Кожен розділ може містити власні підрозділи, де представлені конкретні товари або моделі.

Для зручності користувачів на кожній сторінці веб-сайту має бути наявна навігація, яка дозволяє їм швидко переміщатися між розділами та сторінками. Це може бути реалізовано за допомогою меню, пошукової панелі, хлібних крихток, які вказують на шлях користувача, а також внутрішніх посилань між пов'язаними сторінками. Ця навігація сприяє зручному взаємодії користувачів з веб-сайтом та допомагає їм легко знаходити потрібну інформацію та переходити до бажаних розділів та сторінок.

Веб-сайт "Hoodie Customize" є платформою, яка надає користувачам можливість створювати та налаштовувати свої власні худі з використанням різноманітних дизайнів, кольорів та елементів. Даний розділ пропонує детальний опис кожної зі сторінок цього веб-сайту, розкриваючи їх функціональність та значення для користувачів.

Сторінка "Головна": Головна сторінка є центральною точкою сайту "Hoodie Customize". Вона приваблює увагу користувачів за допомогою привітного дизайну та акційних пропозицій. На цій сторінці можуть бути представлені новини, актуальні колекції худі, найпопулярніші дизайни тощо. Вона також містить посилання на інші розділи сайту, щоб легко переходити до бажаної інформації. Головна сторінка зображена на рисунку 2.7.

Рисунок 2.7 – Головна сторінка.

Наступна сторінка "Магазин" на веб-сайті "Hoodie Customize" є однією з головних сторінок, де користувачі можуть переглядати доступні продукти та здійснювати покупки. Ця сторінка має на меті надати користувачам зручний спосіб ознайомитися з асортиментом товарів, знайти потрібні їм худі та зробити замовлення.



На сторінці "Магазин" можуть бути різні елементи та функції, які допомагають користувачам здійснювати покупки. Основні елементи цієї сторінки включають:

- Категорії: На сторінці можуть бути відображені різні категорії продуктів, наприклад, худі різних стилів, дизайнів або колекцій. Категорії допомагають організувати товари і дозволяють користувачам швидко знайти те, що їх цікавить.

- Картки товарів: Кожен продукт може мати свою картку, на якій відображаються його зображення, назва, ціна та короткий опис. Ці картки допомагають користувачам отримати швидкий огляд товару та прийняти рішення про покупку.

- Детальна інформація: При кліці на картку товару користувач може перейти на окрему сторінку з докладною інформацією про вибраний продукт. Тут можуть бути надані більш детальні характеристики, опис, варіанти розмірів та кольорів, а також можливість додати товар до кошика.

- Кнопка "Додати до кошика": На сторінці може бути присутня кнопка, яка дозволяє користувачам додавати товари до свого кошика. Після натискання на цю кнопку вибраний товар зберігається в кошику, і користувач може продовжити переглядати і додавати інші товари до кошика.

Сторінка "Магазин" на веб-сайті "Hoodie Customize" є центральною точкою для покупок худі та взаємодії користувачів з продуктами. Вона надає зручний і організований спосіб презентації товарів, дозволяє здійснювати пошук, фільтрацію та детальне ознайомлення з худі перед придбанням.

Сторінка "Магазин": Сторінка "Магазин" відображає широкий асортимент доступних худі для користувачів. Вона дозволяє користувачам переглядати різні варіанти худі, фільтрувати їх за розміром, кольором або стилем. Користувачі можуть обирати певний дизайн, додавати елементи та налаштовувати кольори за своїм смаком. Ця сторінка надає можливість докладного ознайомлення з продуктами та їх параметрами.

Дана сторінка зображена на рисунку 2.8.

Рисунок 2.8 – Сторінка «Магазин».

Для категорій було виділено сайд-бар. Категорії: В сайдбарі можуть бути відображені список доступних категорій товарів. Це дозволяє користувачам швидко

обрати певну категорію товарів і переглянути відповідні варіанти худі.

Сторінка "Кошик" на веб-сайті "Hoodie Customize" є важливою частиною функціональності сайту, де користувачі можуть переглядати, керувати та оформляти свої покупки перед остаточним замовленням.

Основні компоненти та функції, які можна знайти на сторінці "Кошик", включають:

- Список товарів: На сторінці "Кошик" відображається список всіх товарів, які були додані користувачем до кошика. Кожен товар може бути представлений з його зображенням, назвою, ціною та іншою інформацією, яка допомагає користувачу впізнати його.

- Кількість товарів: Для кожного товару в кошику користувач може вказати бажану кількість. Це дозволяє змінювати кількість одиниць товару перед оформленням замовлення.

- Видалення товарів: Користувач має можливість видалити товари зі свого кошика, якщо вони більше не потрібні або були помилково додані.

- Підсумкова вартість: На сторінці "Кошик" також відображається підсумкова вартість всіх товарів у кошику. Це допомагає користувачеві відстежувати загальну вартість своїх покупок та зробити відповідні рішення.

- Продовження оформлення замовлення: Коли користувач готовий оформити своє замовлення, сторінка "Кошик" надає можливість перейти до процесу оформлення, де користувач заповнює необхідні дані для доставки, обирає спосіб оплати та підтверджує своє замовлення.

Сторінка "Кошик" є важливою складовою електронної комерції, яка надає користувачам зручність та контроль над їхніми покупками. Сторінка "Кошик": Сторінка "Кошик" відображає вміст кошика покупок користувача. Тут користувачі можуть переглядати всі додані товари, змінювати їх кількість або видаляти з кошика. Крім того, на цій сторінці можуть бути відображені загальна сума замовлення, можливість використовувати промо-коди або купони на знижку. Кошик зображено на рисунку 2.9

Рисунок 2.9 – Кошик

Сторінка "Вхід" є частиною функціоналу веб-сайту "Hoodie Customize" і надає користувачам можливість увійти до свого облікового запису. Описано нижче деталі

сторінки "Вхід":

- Форма входу: На сторінці "Вхід" користувачі можуть бачити форму входу, де їм необхідно ввести свої облікові дані, такі як ім'я користувача або електронна пошта та пароль. Ця форма дозволяє забезпечити аутентифікацію користувачів та перевірити їхню ідентичність.

- Запам'ятовування облікових даних: На сторінці "Вхід" може бути присутня опція "Запам'ятати мене", яка дозволяє користувачам зберігати свої облікові дані на пристрої, щоб не потрібно було вводити їх кожного разу при наступному вході.

- Забули пароль?: Якщо користувач забув свій пароль, сторінка "Вхід" може містити посилання або форму для скидання пароля. Це дозволяє користувачам відновити доступ до свого облікового запису шляхом встановлення нового пароля.

- Реєстрація: На сторінці "Вхід" може бути посилання або кнопка, що перенаправляє користувачів на сторінку реєстрації, якщо вони ще не мають облікового запису. Це дозволяє новим користувачам створити обліковий запис на веб-сайті.

- Повідомлення про помилки: Якщо користувач ввів неправильні облікові дані або сталася інша помилка, сторінка "Вхід" може відображати відповідні повідомлення про помилку для користувача.

Сторінка "Реєстрація" також є важливою частиною функціоналу веб-сайту "Hoodie Customize" і дозволяє новим користувачам створювати облікові записи. Описано нижче деталі сторінки "Реєстрація":

- Форма реєстрації: На сторінці "Реєстрація" користувачі повинні заповнити форму з необхідними обліковими даними, такими як ім'я, електронна пошта, пароль тощо. Ця форма дозволяє новим користувачам створити свій особистий обліковий запис на веб-сайті.

- Правила реєстрації: Сторінка "Реєстрація" може містити посилання на правила та умови використання веб-сайту або політику конфіденційності. Це дозволяє користувачам ознайомитися з правилами, перед тим як створювати обліковий запис.

- Підтвердження облікового запису: В деяких випадках сторінка "Реєстрація" може включати механізм підтвердження облікового запису, де новим користувачам надсилається електронний лист з посиланням для активації облікового запису.

- Повідомлення про успішну реєстрацію: Після успішної реєстрації сторінка "Реєстрація" може відображати повідомлення про успішне створення облікового запису

та надати користувачу інструкції для продовження використання веб-сайту.

– Повідомлення про помилки: Якщо під час реєстрації сталася помилка, сторінка "Реєстрація" може відображати відповідні повідомлення про помилку для користувача, наприклад, якщо введені дані недійсні або вже використовуються.

Сторінка "Вхід": Сторінка "Вхід" надає користувачам можливість авторизуватися на сайті за допомогою свого облікового запису. Користувачі можуть ввести свої дані, такі як електронна пошта і пароль, для входу в особистий кабінет. Сторінку зображено на рисунку 2.10.

Рисунок 2.10 – Сторінка входу в особистий кабінет

Сторінка "Реєстрація": Сторінка "Реєстрація" надає новим користувачам можливість створити обліковий запис на сайті. Користувачі можуть ввести свої особисті дані та створити пароль для майбутнього використання. Це дає можливість отримати доступ до всіх функціональних можливостей сайту, таких як збереження налаштувань, замовлення та знижки. Сторінка реєстрації зображена на рисунку 2.11.

Рисунок 2.11 – Сторінка реєстрації

Ці дві сторінки, "Вхід" і "Реєстрація", є важливими для користувачів веб-сайту "Hoodie Customize", оскільки вони надають зручний спосіб увійти в обліковий запис або створити новий обліковий запис для доступу до всіх функцій та можливостей сайту.

Кожна з цих сторінок веб-сайту "Hoodie Customize" відіграє свою роль у створенні зручного та функціонального середовища для користувачів. Вони сприяють легкому пошуку, налаштуванню та замовленню худі, надають можливість входу та реєстрації, а також забезпечують інформаційну базу для користувачів. Завдяки цим сторінкам веб-сайт "Hoodie Customize" стає привабливим та зручним інструментом для створення власного унікального худі.

## **2.8 Висновки до другого розділу**

У цьому розділі було розглянуто різні аспекти веб-сайту "Hoodie Customize". Розпочавши з опису самого сайту та його основної концепції, було розглянуто різноманітні сторінки, які входять в його структуру. Були описані особливості і

функціонал кожної сторінки, такі як "Магазин", "Категорії", "Кошик", "Вхід" і "Реєстрація".

Далі було розглянуто про використання веб-сайту "Hoodie Customize" з точки зору адміністрування, тестування та розгортання на хостингу. Було розглянуто процес адміністрування сайту з використанням панелі адміністратора Django, а також інструменти тестування, такі як Lighthouse і GTmetrix.

Загалом, веб-сайт "Hoodie Customize" є комплексним проектом, який включає різноманітні сторінки і функціонал, які допомагають користувачам замовляти та налаштовувати свої унікальні худі. Розгортання і адміністрування сайту відбуваються з використанням відповідних інструментів і технологій, що дозволяють забезпечити безперебійну роботу сайту і зручну взаємодію з користувачами.

Загалом, розглянуті аспекти демонструють комплексний підхід до розробки, адміністрування та тестування веб-сайту "Hoodie Customize", що допомагає створити зручне та функціональне середовище для користувачів, які цінують індивідуальний підхід до свого одягу.

### **3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ХОРОНИ ПРАЦІ**

#### **3.1 Пожежна профілактика та пожежна безпека на підприємстві**

Забезпечення пожежної безпеки - невід'ємна частина державної діяльності з охорони життя й здоров'я людей, матеріальних цінностей та навколишнього середовища. Згідно зі статтею 4 Закону України "Про пожежну безпеку" (далі - Закон) державні органи виконавчої влади й органи самоврядування всіх рівнів у межах своєї компетенції організовують розробку й упровадження у відповідних галузях 1 регіонах організаційних й науково-технічних заходів щодо запобігання пожеж, їх гасіння, а також забезпечення пожежної безпеки населених пунктів та об'єктів.

Згідно з чинним законодавством відповідальність за утримання промислового підприємства у належному протипожежному стані лягає безпосередньо на керівника (власника).

Власники підприємств, закладів та організацій, а також орендарі зобов'язані:

- розробити комплексні заходи щодо забезпечення пожежної безпеки;
- згідно з нормативними актами пожежної безпеки розробляти й затверджувати положення, інструкції, інші нормативні акти, які діють у межах підприємств, здійснювати постійний контроль за їх дотриманням;
- забезпечувати дотримання протипожежних вимог стандартів, норм, правил, а також виконання вимог постанов органів державного пожежного нагляду;
- організувати навчання працівників правилам протипожежного захисту й зв'язку та пропаганду заходів з їх забезпечення;
- створювати у разі необхідності підрозділи пожежної охорони та необхідну для їх функціонування матеріально-технічну базу;
- надавати на вимогу державної пожежної охорони відомості та документи про стан пожежної безпеки об'єктів та вироблюваної ними продукції;
- здійснювати заходи щодо впровадження автоматичних засобів виявлення та гасіння пожеж;
- своєчасно інформувати пожежну охорону про несправності пожежної техніки, систем протипожежного захисту, водопостачання тощо;
- проводити службові розслідування випадків пожеж.

Згідно зі статтею 6 Закону громадяни України, іноземні громадяни та особи, які не мають громадянства, що перебувають на території України, зобов'язані:

- виконувати правила пожежної безпеки, забезпечувати будівлі, які належать їм на правах особистої власності, первинними засобами гасіння пожеж і протипожежним інвентарем;
- сповіщати пожежній охороні про виникнення пожежі і вживати заходи з її ліквідації, спасіння людей та майна.

Інструкції мають розроблятися на основі діючих правил та інших нормативних актів з пожежної безпеки будівель, споруд, технологічних процесів, технологічного та виробничого обладнання.

Вони мають встановлювати порядок та спосіб забезпечення пожежної безпеки, обов'язки і дії працівників у разі виникнення пожежі, включаючи порядок оповіщення людей та повідомлення пожежної охорони, порядок евакуації тварин і матеріальних цінностей, застосування засобів пожежогасіння та взаємодії з підрозділами пожежної охорони.

Комплекс протипожежних заходів для приміщення обладнаного персональними комп'ютерами з ВДТ розроблений згідно вимог «Правила пожежної безпеки в Україні».

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні, відповідно до вимог «Норми визначення категорій приміщень, будинків і зовнішніх установок з взривопожежної і пожежної безпеки», приміщення обладнане ВДТ належить до виробництв категорії «В» з пожежної небезпеки - простір у приміщенні, в якому знаходяться тверді горючі речовини та матеріали. Оскільки приміщення (офіс) обладнане ВДТ належить до виробництв категорії «В» з пожежної небезпеки, тому згідно вимог «Пожежна безпека об'єктів будівництва» вона має II ступінь вогнестійкості.

З технічних та організаційних заходів запобігання пожеж в приміщенні обладнаному персональними комп'ютерами з ВДТ передбачені наступні протипожежні заходи. На силовому обладнанні, силових та освітлювальних колах, згідно вимог пункту 3.1 «ПУЕ», встановлені захисні пристрої, що вимикають джерело живлення від ділянки електричного кола, у якій виникло коротке замикання. Згідно вимог «Правила улаштування та експлуатації систем оповіщення про пожежу та управління евакуацією людей в будинках та спорудах» і «Системи протипожежного захисту», в приміщенні

обладнаному персональними комп'ютерами з ВДТ встановлена система пожежної й охоронної сигналізації «Сигнал-ВК6». Яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика. Відповідно до вимог «Типові норми належності вогнегасників» для гасіння електрообладнання у приміщенні обладнаному персональними комп'ютерами з ВДТ, що знаходиться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-5 в кількості 2 штук. Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10м.

Передбачені для приміщення обладнаного персональними комп'ютерами з візуальними дисплейними терміналами заходи по забезпеченню безпеки, виробничої санітарії, гігієни праці і пожежної безпеки забезпечують безпечні та комфортні умови праці персоналу

### **3.2 Профілактика нещасних випадків, професійних захворювань і отруєнь на підприємстві**

Профілактика нещасних випадків, професійних захворювань і отруєнь на виробництві є важливою складовою безпеки та здоров'я працівників. Організації повинні приділяти належну увагу цій проблемі, розробляючи та впроваджуючи ефективні стратегії та заходи, щоб запобігти негативним наслідкам для працівників та ділової діяльності загалом.

Одним із ключових аспектів профілактики є розуміння потенційних ризиків та небезпек, пов'язаних з конкретним видом виробництва або робочим середовищем. Це може включати аналіз робочих процесів, ідентифікацію можливих джерел небезпеки, оцінку ризику та впровадження заходів з мінімізації цих ризиків.

Профілактика також передбачає належне навчання та підготовку працівників. Це охоплює надання інформації про безпеку та здоров'я, навичок безпечної роботи, використання необхідного захисного спорядження, а також навчання реагуванню на надзвичайні ситуації та першу допомогу. Крім того, необхідно забезпечити постійне навчання та оновлення знань з питань безпеки та здоров'я.

Важливим аспектом профілактики є належне обладнання та інфраструктура. Це охоплює правильне функціонування та обслуговування обладнання, проведення



періодичних перевірок безпеки, встановлення захисних пристроїв та систем, а також забезпечення належних умов праці, які включають правильну вентиляцію, освітлення та розміщення робочих місць.

Ефективна профілактика також передбачає впровадження системи моніторингу та контролю. Це може включати регулярний аудит безпеки, аналіз нещасних випадків і інцидентів, ведення статистики та звітності, а також виявлення і вирішення проблем, які можуть створювати ризики для працівників.

Загальний підхід до профілактики передбачає системний підхід до управління безпекою та здоров'ям на робочому місці. Це означає, що керівництво організації повинне виявити відповідальність за створення культури безпеки, розробку політики безпеки та створення відповідних процедур, а також залучення всіх працівників до процесу профілактики.

Актуальність профілактики нещасних випадків, професійних захворювань і отруєнь на виробництві є надзвичайно важливою для забезпечення безпеки та здоров'я працівників. Це допомагає попередити травми, захворювання та негативні наслідки, що можуть виникнути в результаті небезпечних робочих умов. Крім того, це сприяє поліпшенню репутації організації, збереженню робочої сили та покращенню продуктивності. Профілактика повинна бути постійним процесом, що охоплює всі аспекти роботи і залучає всіх учасників виробництва.

Аналіз причин виробничого травматизму свідчить, що від 33 до 47% нещасних випадків щорічно відбуваються внаслідок наявності небезпечних та шкідливих виробничих факторів.

Небезпечний виробничий фактор - це виробничий фактор, дія якого на працюючого в певних умовах приводить до травми або іншого погіршення здоров'я.

Шкідливий виробничий фактор - це виробничий фактор, дія якого на працюючого приводить до захворювання або зниження працездатності.

Небезпечні та шкідливі виробничі фактори поділяються на 4 класи:

- фізичні;
- хімічні;
- біологічні;
- психофізіологічні.

В результаті дії небезпечних та шкідливих виробничих факторів, шкідливих речовин можуть мати місце нещасні випадки і професійні захворювання.

Нещасні випадки на виробництві - це випадки з робітником, пов'язані з дією на нього шкідливого виробничого фактора при виконанні ним трудових обов'язків або завдання керівника робіт, які приводять до травми.

До травм відносяться: переломи, опіки, поранення, ураження електричним струмом, отруєння тощо.

Виробнича травма - це травма, отримана робітником на виробництві і пов'язана з недотриманням вимог безпеки праці.

Однак, якщо встановлено, що травма пов'язана з виробництвом, але головною технічною або організаційною причиною її стало порушення правил і норм ОП (незадовільний стан обладнання, проходів, освітлення, неправильна організація або відсутність контролю за виконанням робіт), то нещасний випадок повинен бути признаним пов'язаним з виробництвом.

Адміністрація несе повну відповідальність тільки за нещасні випадки, пов'язані з виробництвом. При порушенні норм виробничої санітарії працюючий може отримати професійне захворювання. Професійним називається захворювання, спричинене впливом на працівників шкідливих умов праці.

Міністерством охорони здоров'я затверджений перелік професійних захворювань та Інструкція по його застосуванню. Цей перелік містить специфічні професійні захворювання, які виникають під впливом шкідливих факторів, властивих даній професії, умовам праці, а також захворювання, які зустрічаються в даних умовах частіше за інші.

До професійних відносять всі хвороби, які виникли внаслідок впливу на працюючого шкідливих виробничих факторів. До них належать хвороби, які виникли внаслідок зміни атмосферного тиску (кесонна хвороба, гірна хвороба та ін.), хвороби внаслідок впливу професійного пилу (пневмоконіози), хвороби шкіри від дії подразнюючих і токсичних речовин (дерматити, язви) та інші хвороби.

Професійне захворювання, яке виникає на протязі малого проміжку часу (однієї зміни або доби) називається гострим, а яке виникає на протязі більш тривалого часу - хронічним. Професійне отруєння - це окремий випадок професійного захворювання. Отруєння може бути гострим, тобто таким, що сталося в результаті одноразового впливу

на людину великої дози отруйної речовини, і хронічним, тобто таким, що розвивалося протягом тривалого часу, наприклад кількох років, внаслідок систематичного впливу порівняно малих доз шкідливих речовин. Наявність професійного захворювання встановлюється компетентними медичними органами. Для розробки заходів попередження нещасних випадків необхідно виявити причини, що їх викликають.

### **3.3 Висновок до третього розділу**

Охорона праці є невід'ємною складовою будь-якої організації, включаючи веб-сайти, які надають послуги, наприклад, Hoodie Customize. Теми, які були розглянуті в розділі з охорони праці, включали профілактику нещасних випадків, професійних захворювань і отруєнь на виробництві, а також пожежну профілактику та пожежну безпеку на виробництві.

Застосування принципів охорони праці в контексті Hoodie Customize виконує важливу роль у забезпеченні безпеки та здоров'я працівників, а також захисті виробничих активів. Розробники, адміністратори та користувачі сайту повинні дотримуватися відповідних норм, стандартів і процедур, щоб уникнути негативних наслідків.

Профілактика нещасних випадків та професійних захворювань на виробництві повинна включати аналіз ризиків, навчання працівників правилам безпеки, встановлення захисного спорядження та систем моніторингу та контролю. В контексті Hoodie Customize це може означати надання інструкцій користувачам щодо безпечного користування сайтом, використання захисних програмних засобів та забезпечення конфіденційності даних.

Пожежна профілактика та пожежна безпека на виробництві важливі для забезпечення безпеки працівників і запобігання матеріальним збиткам. У випадку Hoodie Customize це може включати встановлення протипожежних систем на серверах, забезпечення резервного копіювання даних та захист від вторгнень.

Реалізація цих принципів охорони праці сприяє створенню безпечного та надійного середовища для працівників і користувачів Hoodie Customize. Забезпечення безпеки є однією з ключових вимог до функціональності та якості сайту, що позитивно впливає на репутацію організації та задоволення клієнтів.

Отже, охорона праці та пожежна безпека на виробництві є необхідними складовими для успішної роботи Hoodie Customize та забезпечення безпеки працівників і користувачів. Розуміння та дотримання відповідних норм і процедур сприяє створенню надійного та ефективного веб-сайту, який забезпечує задоволення потреб клієнтів та впевненість у безпеці.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи освітнього рівня “бакалавр”, розробки веб-сайту “Hoodie customize” засобами Django, Python, HTML та CSS, було виконано всі раніше поставлені завдання при дослідженні задачі.

Результати дослідження теми було інтерпретовано у вигляді написання трьох розділів кваліфікаційної роботи, два з яких технічні що мають на меті реалізацію веб-сайту.

В першому розділі було проаналізовано предметну область, сформовано вимоги до розробки веб-сайту, проведено пошук актантів та можливі варіанти використання веб-сайту, детальніше описано ключові. Вибрано стек для реалізації.

В другому розділі описано архітектуру веб-сайту “Hoodie customize”, детальніше про вибір технологій відповідно до стеку, файловою структурою веб-сайту, структурні особливості вибраної бази даних, розгортання на хостигових платформах, про тестування та використання розробленої системи.

В третьому розділі, було розглянуто пожежну профілактику та пожежну безпеку на виробництві та профілактику нещасних випадків, професійних захворювань і отруєнь на виробництві.

В процесі роботи над кваліфікаційною роботою:

- освоєно новий теоретичний матеріал досліджуваної теми, заповнено пробіли у наявних знаннях;
- розглянуто сучасний інструментарій розробки веб-сайтів та вибрано підходящий;
- проведено порівняння між функціональними можливостями веб-сайту “Hoodie customize” та функціональними можливостями конкурентів;
- на основі отриманої інформації веб-сайт було: розроблено та протестован;

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Стаття про Django [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/razrabotka-sajtov-na-python-django>
2. Робота з моделями на Django [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.2/topics/db/models/>
3. Visual Studio Code docs [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://code.visualstudio.com/docs>
4. Python docs [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://docs.python.org/3/>
5. The Use Case Model [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://sparxsystems.com/resources/tutorials/uml/use-case-model.html>.
6. Наукова робота по клієн серверній архітектурі на Django [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://ekmair.ukma.edu.ua/items/4a2ea6f2-6e3c-40e2-88f9-1752da8c2673>
7. Вибір залежностей [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.2/ref/contrib/>
8. Структура моделей Django [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://metanit.com/python/django/5.1.php>
9. Адміністрування Django [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://docs.djangoproject.com/en/4.2/ref/contrib/admin/>
10. Хостинг і домен [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://globalnet.com.ua/ua/blog/useful/chto-takoe-hosting-i-dome/>
11. Вибір хостингу [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://www.digitalocean.com/>
12. Інструкція по встановленню [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://www.digitalocean.com/community/tutorials/how-to-set-up-django-with-postgres-nginx-and-gunicorn-on-ubuntu-18-04>
13. Тестування сайту [Електронний ресурс] – 15.06. 2023– Режим доступу до ресурсу: <https://gtmetrix.com/>
14. Вимоги безпеки під час роботи на ПК [Електронний ресурс]. – Режим

доступу до ресурсу: <https://sites.google.com/site/ohoronapraci44/33-vimogi-bezpeki-pid-cas-roboti-na-pk>.

15. Вимоги охорони праці перед початком роботи на комп'ютері [Електронний ресурс]. – Режим доступу до ресурсу: <https://sites.google.com/site/ohoronapraci44/01-vimogi-ohoroni-praci-pered-pocatkom-roboti-na-komp-uteri>.

## ДОДАТКИ

### Додаток А

#### Лістинг 1 – Код файлу «manage.py»

```
import React from 'react';
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys
def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'greatkart.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```



**Лістинг 2 – Код файлу «models.py»**

```
from django.db import models
from store.models import Product, Variation
from accounts.models import Account
# Create your models here.
class Cart(models.Model):
    cart_id = models.CharField(max_length=250, blank=True,
verbose_name='ID')
    date_added = models.DateField(auto_now_add=True,
verbose_name='Дата додавання')
    def __str__(self):
        return self.cart_id
    class Meta:
        verbose_name = 'Кошик'
        verbose_name_plural = 'Кошкики'
        ordering = ['-date_added']
class CartItem(models.Model):
    user = models.ForeignKey(Account,
on_delete=models.CASCADE, null=True, verbose_name='Користувач')
    product = models.ForeignKey(Product,
on_delete=models.CASCADE, verbose_name='Товар')
    variations = models.ManyToManyField(Variation,
blank=True, verbose_name='Варіанти')
    cart = models.ForeignKey(Cart, on_delete=models.CASCADE,
null=True, verbose_name='Кошик')
    quantity = models.IntegerField(verbose_name='Кількість')
    is_active = models.BooleanField(default=True,
verbose_name='Активний')
    class Meta:
        verbose_name = 'Елемент кошика'
        verbose_name_plural = 'Елементи кошика'
        ordering = ['cart']
    def sub_total(self):
        return self.product.price * self.quantity
    def __unicode__(self):
        return self.product
```

**Лістинг 3 – Код файлу «admin.py»**

```
from django.contrib import admin
from .models import Cart, CartItem
# Register your models here.

class CartAdmin(admin.ModelAdmin):
    list_display = ('cart_id', 'date_added')

class CartItemAdmin(admin.ModelAdmin):
    list_display = ('product', 'cart', 'quantity', 'is_active')

admin.site.register(Cart, CartAdmin)
admin.site.register(CartItem, CartItemAdmin)
```

**Лістинг 4 – Код файлу «views.py»**

```

from django.shortcuts import render, redirect, get_object_or_404
from store.models import Product, Variation
from .models import Cart, CartItem
from django.core.exceptions import ObjectDoesNotExist
from django.contrib.auth.decorators import login_required

# Create your views here.
def _cart_id(request):
    cart = request.session.session_key
    if not cart:
        cart = request.session.create()
    return cart

def add_cart(request, product_id):
    current_user = request.user
    product = Product.objects.get(id=product_id) # get the product
    # If the user is authenticated
    if current_user.is_authenticated:
        product_variation = []
        if request.method == 'POST':
            for item in request.POST:
                key = item
                value = request.POST[key]

                try:
                    variation =
                    Variation.objects.get(product=product,
                    variation_category__iexact=key, variation_value__iexact=value)
                    product_variation.append(variation)
                except:
                    pass

            is_cart_item_exists =
            CartItem.objects.filter(product=product,
            user=current_user).exists()
            if is_cart_item_exists:
                cart_item = CartItem.objects.filter(product=product,
            user=current_user)
                ex_var_list = []
                id = []
                for item in cart_item:
                    existing_variation = item.variations.all()
                    ex_var_list.append(list(existing_variation))
                    id.append(item.id)

                if product_variation in ex_var_list:
                    # increase the cart item quantity
                    index = ex_var_list.index(product_variation)
                    item_id = id[index]
                    item = CartItem.objects.get(product=product,
            id=item_id)

```

```

        item.quantity += 1
        item.save()

    else:
        item = CartItem.objects.create(product=product,
quantity=1, user=current_user)
        if len(product_variation) > 0:
            item.variations.clear()
            item.variations.add(*product_variation)
        item.save()
    else:
        cart_item = CartItem.objects.create(
            product = product,
            quantity = 1,
            user = current_user,
        )
        if len(product_variation) > 0:
            cart_item.variations.clear()
            cart_item.variations.add(*product_variation)
        cart_item.save()
        return redirect('cart')
# If the user is not authenticated
else:
    product_variation = []
    if request.method == 'POST':
        for item in request.POST:
            key = item
            value = request.POST[key]

            try:
                variation =
Variation.objects.get(product=product,
variation_category__iexact=key, variation_value__iexact=value)
                product_variation.append(variation)
            except:
                pass

        try:
            cart = Cart.objects.get(cart_id=_cart_id(request)) #
get the cart using the cart_id present in the session
        except Cart.DoesNotExist:
            cart = Cart.objects.create(
                cart_id = _cart_id(request)
            )
            cart.save()

            is_cart_item_exists =
CartItem.objects.filter(product=product, cart=cart).exists()
            if is_cart_item_exists:
                cart_item = CartItem.objects.filter(product=product,
cart=cart)

                # existing_variations -> database
                # current variation -> product_variation
                # item_id -> database
                ex_var_list = []

```

```

id = []
for item in cart_item:
    existing_variation = item.variations.all()
    ex_var_list.append(list(existing_variation))
    id.append(item.id)

print(ex_var_list)

if product_variation in ex_var_list:
    # increase the cart item quantity
    index = ex_var_list.index(product_variation)
    item_id = id[index]
    item = CartItem.objects.get(product=product,
id=item_id)
    item.quantity += 1
    item.save()

else:
    item = CartItem.objects.create(product=product,
quantity=1, cart=cart)
    if len(product_variation) > 0:
        item.variations.clear()
        item.variations.add(*product_variation)
    item.save()
else:
    cart_item = CartItem.objects.create(
        product = product,
        quantity = 1,
        cart = cart,
    )
    if len(product_variation) > 0:
        cart_item.variations.clear()
        cart_item.variations.add(*product_variation)
    cart_item.save()
return redirect('cart')

def remove_cart(request, product_id, cart_item_id):

    product = get_object_or_404(Product, id=product_id)
    try:
        if request.user.is_authenticated:
            cart_item = CartItem.objects.get(product=product,
user=request.user, id=cart_item_id)
        else:
            cart = Cart.objects.get(cart_id=_cart_id(request))
            cart_item = CartItem.objects.get(product=product,
cart=cart, id=cart_item_id)
        if cart_item.quantity > 1:
            cart_item.quantity -= 1
            cart_item.save()
        else:
            cart_item.delete()
    except:
        pass
    return redirect('cart')

```

```

def remove_cart_item(request, product_id, cart_item_id):
    product = get_object_or_404(Product, id=product_id)
    if request.user.is_authenticated:
        cart_item = CartItem.objects.get(product=product,
user=request.user, id=cart_item_id)
    else:
        cart = Cart.objects.get(cart_id=_cart_id(request))
        cart_item = CartItem.objects.get(product=product,
cart=cart, id=cart_item_id)
    cart_item.delete()
    return redirect('cart')

def cart(request, tax=99, grand_total=0, total=0, quantity=0,
cart_items=None):
    try:
        if request.user.is_authenticated:
            cart_items = CartItem.objects.filter(user=request.user,
is_active=True)
        else:
            cart = Cart.objects.get(cart_id=_cart_id(request))
            cart_items = CartItem.objects.filter(cart=cart,
is_active=True)
        for cart_item in cart_items:
            total += (cart_item.product.price * cart_item.quantity)
            quantity += cart_item.quantity
        grand_total = total + tax
    except ObjectDoesNotExist:
        pass #just ignore

    context = {
        'total': total,
        'quantity': quantity,
        'cart_items': cart_items,
        'tax': tax,
        'grand_total': grand_total,
    }
    return render(request, 'store/cart.html', context)

@login_required(login_url='login')
def checkout(request, tax=99, grand_total=0, total=0, quantity=0,
cart_items=None):
    try:
        if request.user.is_authenticated:
            cart_items = CartItem.objects.filter(user=request.user,
is_active=True)
        else:
            cart = Cart.objects.get(cart_id=_cart_id(request))
            cart_items = CartItem.objects.filter(cart=cart,
is_active=True)
        for cart_item in cart_items:
            total += (cart_item.product.price * cart_item.quantity)
            quantity += cart_item.quantity
        grand_total = total + tax
    except ObjectDoesNotExist:

```

```
    pass #just ignore

context = {
    'total': total,
    'quantity': quantity,
    'cart_items': cart_items,
    'tax': tax,
    'grand_total': grand_total,
}
return render(request, 'store/checkout.html', context)
```