

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА

перший (бакалаврський) рівень вищої освіти

на тему:

«Розробка інтернет-магазину для продажу органічної продукції з використанням фреймворка Django»

Виконав: студент групи ІТ-41
спеціальності 126 «Інформаційні системи та
технології»

Гойсак А.В.

(прізвище та ініціали)

Керівник:

Желєзняк А.М.

(прізвище та ініціали)

ДУБЛЯНИ 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Рівень вищої освіти перший (бакалаврський)
Спеціальність 126 «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ (підпис)
д.т.н., професор, Тригуба А. М.
(вч. звання, прізвище, ініціали)
“ _____ ” _____ 202__ року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Гойсака Артура Володимировича
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка інтернет-магазину для продажу органічної продукції з використанням фреймворка Django»

керівник роботи к. е н., доцент., Желєзняк А.М.
(наук. ступінь, вч. звання, прізвище, ініціали)

затвержені наказом Львівського НУП № 641/к-с з 27.11.23р.

2. Строк подання студентом роботи 09.06.2024 р.

3. Вихідні дані: характеристика веб-сайту; вихідні вимоги та дані проекту, проектування баз даних веб-додатку, технічне завдання на створення інтернет-магазину органічних продуктів.

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

1. Теоретичні аспекти розробки інтернет-магазину органічної продукції

2. Основи розробки інтернет-магазину

3. Розробка інтернет-магазину з використанням фреймворку Django

4. Охорона праці та безпека в надзвичайних ситуаціях

Висновки

5. Перелік графічного матеріалу

Графічний матеріал подається у вигляді презентації

6. Консультанти розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата		Відмітка про виконання
		завдання видав	завдання прийняв	
1, 2, 3	<i>к. е н., доцент., Желєзняк А.М.</i>			
4	<i>к.т.н., доцент Городецький І.М.</i>			

7. Дата видачі завдання 30.10.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Терміни виконання етапів роботи	Примітка
1.	<i>Написання першого розділу.</i>	<i>30.10.2023 – 15.12.2023</i>	
2.	<i>Опис вимог, моделювання інтернет-магазину, розробка структури бази даних веб-сайту.</i>	<i>16.12.2023 – 05.02.2024</i>	
3.	<i>Обґрунтування вибору мови програмування.</i>	<i>06.02.2024 – 31.03.2024</i>	
4.	<i>Програмна реалізація проекту(розробка сайту, оцінка ефективності проекту)</i>	<i>01.04.2024 – 30.04.2024</i>	
5.	<i>Завершення оформлення розрахуноко-пояснювальної записки та підготовка презентаційного матеріалу.</i>	<i>01.05.2024 – 26.05.2024</i>	
6.	<i>Завершення роботи в цілому та підготовка до захисту дипломної роботи.</i>	<i>27.05.2024 – 10.06.2024</i>	

Студент

_____ (підпис)

Гойсак А. В.

_____ (прізвище та ініціали)

Керівник роботи

_____ (підпис)

Желєзняк А. М.

_____ (прізвище та ініціали)

УДК 004.738.5

Кваліфікаційна робота: 51 сторінок текстової частини, 3 таблиці, 16 рисунків, 20 джерел літератури, 3 додатки.

«Розробка Інтернет-магазину для продажу органічної продукції з використанням фреймворка Django» Гойсак А. В. – Кваліфікаційна робота. Кафедра інформаційних технологій. Дубляни, Львівський національний університет природокористування.

Показано веб-сайт органічної продукції, а саме магазину OrganicFood. Було проведено аналіз принцип розробки, визначено характеристики аналогів (конкурентів) описано детальні функціональні вимоги.

Реалізовано Бекенд частину сайту з урахуванням новітніх технологій, додавання дизайну сторінок проекту та підтримка адміністративної сторінки. Детальне пояснення товару на сайті, додавання продукції в кошик та зворотній зв'язок для допомоги користувачу.

Ключові слова: інтернет-магазин, бази даних, органічна продукція, дизайн, інформаційні технології

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ ОГРАНІЧНОЇ ПРОДУКЦІЇ	10
1.1 Основні поняття і підходи у веб-розробці	10
1.2 Особливості функціонування інтернет-магазинів.....	14
1.3 Аналіз прикладної сфери та існуючих практик.....	16
РОЗДІЛ 2 ОСНОВИ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ.....	20
2.1 Опис функціональних вимог та постановка завдання	20
2.2 Моделювання системи інтернет-магазину.....	22
2.3 Розробка архітектури інтернет-магазину	24
2.4 Проектування структури бази даних інтернет магазину	27
РОЗДІЛ 3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ВИКОРИСТАННЯМ ФРАЙМВОРКУ DJANGO.....	33
3.1 Розробка основних функціональних компонентів інтернет-магазину.....	33
3.2 Розробка back-end частини інтернет-магазину з використанням фреймворку Django	36
3.3 Налаштування та впровадження бази даних інтернет-магазину	40
3.4 Оцінка ефективності проекту	44
РОЗДІЛ 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	46
4.1 Аналіз небезпеки під час роботи за комп'ютером.....	46
4.2 Безпека в надзвичайних ситуаціях.....	47

4.3 Аналіз травмонебезпечних ситуацій під час виконання робіт	48
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	52
ДОДАТКИ	544

ВСТУП

У нинішньому світі, коли у людей майже завжди не вистачає часу, а іноді фізичних сил на відвідування магазинів і на довгі та нудні пересування ними. Найбільш простим варіантом є відвідування інтернет-магазину та вибору потрібної речі. Кожного дня ми користуємось Інтернетом і при потребі ми відвідуємо сайти. Сучасна цифрова епоха та стрімкий розвиток електронної комерції роблять важливим викликом створення ефективних та зручних інструментів для продажу якісних, екологічно чистих продуктів. Одним з ключових напрямків є розробка інтернет-магазинів, які не тільки гарантують ефективні та безпечні транзакції, але й підтримують принципи сталого розвитку. Інтернет-магазини стали важливим інструментом просування та продажу цих продуктів, особливо в умовах пандемії COVID-19, і багато людей шукають можливості купувати продукцію онлайн.

У цій роботі проаналізовано процес розробки магазину з продажу органічних продуктів, а саме використання фреймворку Django. Цей вибір був зроблений не тільки через його популярність та широкий спектр можливостей, але й через його ефективність у розробці веб-додатків, зокрема платформ електронної комерції. Органічні продукти користуються великою популярністю серед споживачів через їхній внесок у здоров'я та благополуччя. Дослідження спрямоване на вивчення інноваційних рішень у сфері електронної комерції, які зроблять конкретний внесок у розвиток технологій розробки веб-додатків для підтримки сталого споживання та екологічних ініціатив у сучасній електронній комерції. Актуальність даного дослідження полягає в необхідності розробки та реалізації інтернет-магазину по продажу органічних товарів. Збільшити потік клієнтів та надати покупцям повну інформацію про ціни та послуги.

У роботі описано основні етапи розробки, архітектурні особливості та можливості оптимізації для забезпечення високої продуктивності та задоволення потреб сучасного споживача у якісних та екологічно чистих продуктах.

Мета створення такого магазину є розробка повнофункціонального інтернет-магазину, який забезпечує комфортний процес покупок для користувачів та ефективно управління продуктами для адміністраторів, забезпечення високого рівня безпеки для користувачів та їхніх персональних даних, використовуючи засоби шифрування та заходи безпеки та зменшення витрат компанії. В міру простий інтерфейс сайту "OrganicFood" дає можливість користувачам швидко та легко робити покупки з електронною оплатою на сайті. Це питання продовжує залишатися актуальним у сучасному суспільстві.

Мета даної випускної кваліфікаційної роботи - розробка інтернет-магазину на основі фреймворку Django для торгівлі у мережі Інтернет. В кваліфікаційній роботі було детально показано зв'язок Django з базою даних, де буде помітно, що інструмент допомагає в роботі адміністраторам такого сайту.

Важливим аспектом цього дослідження є висвітлення того, як Django полегшує роботу адміністраторів веб-сайтів, надаючи їм інтуїтивно зрозумілий і потужний інтерфейс для управління контентом і структурування веб-сайтів.

Завдання дослідження:

1. Дослідити сучасний стан ринку органічних продуктів, його тенденції розвитку, конкурентну ситуацію, а також перспективи його зростання.
2. Встановити потреби цільової аудиторії у покупці органічної продукції через Інтернет, їхні вподобання та очікування від інтернет-магазину.
3. Визначити основні функції та можливості, які має мати інтернет-магазин для забезпечення зручного та ефективного процесу покупки органічних продуктів.

4. Обґрунтувати вибір фреймворка Django для розробки інтернет-магазину, розглянути інші технологічні компоненти, такі як бази даних, серверні та клієнтські технології.

5. Розробити архітектуру інтернет-магазину, включаючи моделі даних, структуру бази даних, взаємодію між компонентами та інші аспекти системи.

6. Створити користувацький інтерфейс, який буде зручним та привабливим для користувачів, з урахуванням їхніх потреб та вподобань.

Кваліфікаційна робота складається із вступу, чотирьох розділів, загальних висновків, списку використаних джерел та додатків.

Результати кваліфікаційної роботи були апробовані на Звітній студентській науковій конференції кафедри інформаційних технологій (Дубляни, 5 березня 2024 р).

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ ОГРАНІЧНОЇ ПРОДУКЦІЇ

1.1 Основні поняття і підходи у веб-розробці

На початку існування Інтернету веб-сайти були дуже простими і переважно текстовими. Розробники використовували таблиці для структурування контенту. Цей метод створення сайтів залишався популярним до 2000-х років, але з настанням ери мобільного інтернету застарів. З появою стандарту CSS робота веб-розробників значно спростилася. Тепер професіонали мають єдину стандартну систему правил для опису елементів HTML і XML. У створенні сайту є багато етапів. Кожен з них вимагає досвіду та спеціаліста певного профілю.

Якщо розглянути основні етапи створення сайтів, то всі вони працюють за однаковими принципами. Якщо хоча б один з етапів створення сайту виконати недбало або зовсім пропустити, то вся ідея втрачає сенс, оскільки ресурс не працює належним чином і не може виконувати поставлені перед ним завдання. Тож можна з упевненістю сказати, що створення яскравих веб-сайтів і веб-додатків сьогодні в моді. Чим більш барвистими та інтерактивними є результати розробки, тим привабливішим стає продукт. Кожен веб-сайт має свою власну структуру та індивідуальну архітектуру. Розробники сайтів прийшли до однієї з найкращих структур написання проектів, в якій є все, що потрібно для користувача. Все відштовхується від головної сторінки та знаходиться у «двох кроках» від неї. Схематично структуру веб-сайту можна відобразити наступним чином.

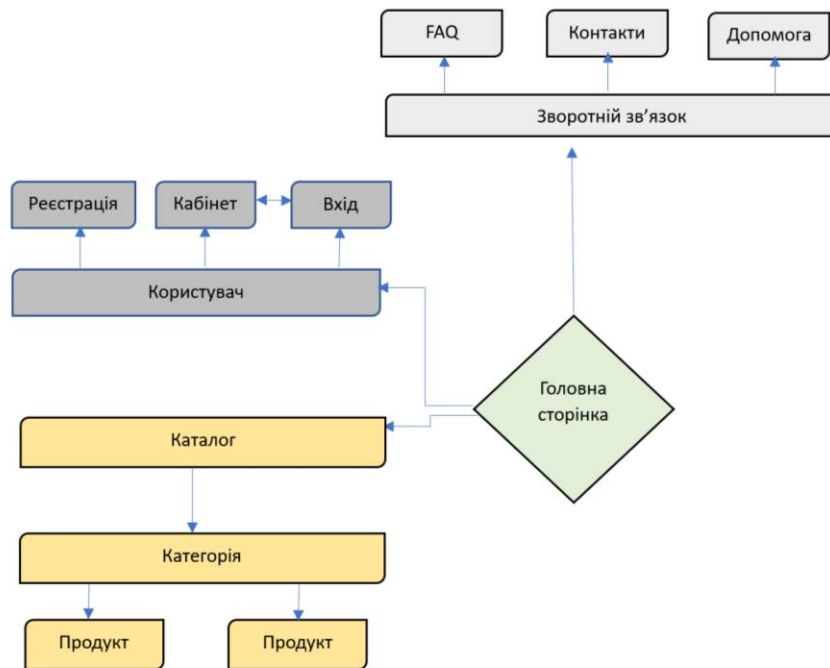


Рисунок 1.1. - Відображення структури сайту

Але не кожен сайт може користуватись такою архітектурою, адже сайти можуть бути різних типів, до основних можна віднести:

- Інформаційний сайт.
- Блог.
- Інтернет-магазин.
- Соціальна мережа.

Інформаційний сайт - це тип веб-сайтів, призначений для надання користувачам інформації на різноманітні теми. Такі веб-сайти часто використовуються компаніями, установами, організаціями або приватними особами, щоб надати більше інформації про свої послуги, продукти, проекти, новини чи події іншу цікаву інформацію.

Блог – це тип сайтів, в яких мета надавати свіжу та цікаву інформацію про події у світі, новинах, автомобільні новинки, новітні смартфони тощо. Для утримання аудиторії співпрацюють з відомими людьми та брендами.

Інтернет-магазин - це веб-сайт, де клієнти можуть відвідувати та купувати товари та послуги, які пропонуються на цьому веб-сайті, відповідно до їхніх уподобань. Клієнти можуть шукати потрібні товари та розміщувати замовлення, здійснюючи за них оплату.

Соціальна мережа - це онлайн-платформи, які дозволяють користувачам спілкуватися, обмінюватися інформацією, створювати профілі додаючи власні фото чи різні інші файли, які буде підтримувати платформа і взаємодіяти з іншими користувачами в інтернеті.

Розбираючи питання поняття веб-сайтів слід звертати увагу на підходи у розробці, для того щоб забезпечити ефективну та результативну роботу проекту.

Підходів у веб-розробці є безліч, можна винести 3 ключових підходи, а саме застосування методології DevOps, мікросервісів та безсерверної архітектури Serverless.

Методологія DevOps - це практика, яка поєднує розробку та експлуатацію для підвищення ефективності та автоматизації процесів розробки, тестування та розгортання програмного забезпечення. DevOps включає застосування автоматизації, контролю версій, тестування, моніторингу та інших інструментів для забезпечення швидкого та надійного продукту.

Можна виділити такі основні риси та елементи DevOps:

- 1) Continuous Integration (Continuous Integration, CI) - автоматичне об'єднання коду команди розробників, виявлення та виправлення конфліктів і помилок під час розробки.

2) Continuous Deployment (CD) - автоматизація розгортання та випуску нових версій програмного забезпечення протягом усього життєвого циклу проекту.

3) Infrastructure as Code (IaC)) - використання коду для автоматизації створення та управління інфраструктурою.

4) Collaboration and Communication - співпраця та комунікація між розробниками, тестувальниками та менеджерами для покращення розуміння та вирішення проблем.

Microservices (Мікросервіси) - це архітектурний підхід до розробки програмного забезпечення, який декомпозує програмні продукти на невеликі, автономні компоненти, відомі як мікросервіси. Кожен мікросервіс виконує певну функцію і може бути розгорнутий і розширений незалежно від інших сервісів.

До основних рис Microservices на нашу думку можна віднести:

1) Незалежність, оскільки кожен мікросервіс можна розгортати та розширювати незалежно від інших сервісів, що полегшує розробку та управління системою.

2) Автономність, адже мікросервіси можуть працювати автономно і взаємодіяти один з одним через API.

3) Простоту розробки та обслуговування. Зміни та оновлення завдяки їх застосуванню можна впроваджувати швидше та безпечніше, оскільки вони не впливають на інші частини системи.

4) Масштабованість - окремі мікросервіси можуть масштабуватися відповідно до навантаження.

Serverless (безсерверна архітектура) - це методологія розробки, в якій розробники відокремлені від управління серверами та інфраструктурою. У безсерверній архітектурі код виконується у відповідь на події, а вартість обчислень базується на фактичному споживанні ресурсів.

Розглянемо основні риси Serverless: безперервні функції (FaaS: Functions as a Service), автоматичне масштабування, відсутність управління інфраструктурою та зниження витрат.

У відповідності до цих характеристик під час застосування Serverless розробка та розгортання окремих функцій, що запускаються подіями, такими як HTTP-запити або завдання в черзі відбувається на основі безперервних функцій. Окрім того ресурси веб-сайту автоматично масштабуються відповідно до навантаження і при цьому розробники не несуть відповідальності за управління серверами або вирішення проблем масштабування. Внаслідок застосування даного підходу можна очікувати зниження витрат для проектів з низьким навантаженням, оскільки оплата здійснюється лише за ті ресурси, які фактично використовуються.

1.2 Особливості функціонування інтернет-магазинів

На сьогоднішній день створення інтернет магазину є хорошою практикою для власників товарів чи продуктів. Вони стають одними з найнеобхідніших інструментів у наш час, завдяки швидкості та легкості покупки для користувачам.

Клієнти можуть робити покупки онлайн з будь-якого пристрою з доступом до Інтернету, включаючи смартфони, планшети та ноутбуки. Інтернет-магазини можуть надавати онлайн-підтримку клієнтів через чат, електронну пошту або телефон для вирішення питань і проблем. Сайти можуть пропонувати широкий спектр товарів і послуг, від одягу та електроніки до продуктів харчування і послуг доставки додому.

Позитивною стороною інтернет-торгівлі є те, що витрати на утримання та організацію є нижчими, оскільки матеріально-технічна інфраструктура (будівлі, споруди та обладнання) та кількість персоналу є значно обмеженішими. Інтернет-магазини вигідні для комерційних компаній, які потребують повного контролю та

управління всіма процесами електронної комерції та різноманітними маркетинговими кампаніями. Незважаючи на значні початкові витрати, пов'язані зі створенням інтернет-магазину порівняно з веб-магазином, інтернет-магазин може бути більш рентабельним у міру зростання обсягів продажів. Варто не забувати про аналітику таких магазинів, придумуючи нові стратегії при падінні чи рості результатів. Ключовим чинником для функціонування магазинів є контекстна реклама, яка завдяки грамотному підходу та маркетингу йде на збільшення конверсії для магазину. Загалом виділяють чинники, які впливають на функціонування інтернет-магазину (табл.1.1.).

Таблиця 1.1. – Опис вимог до інтернет-магазину

Назва вимоги	Опис вимог
Дизайн	відповідний дизайн впливатиме на користувача з перших секунд відвідування сайту
Широкий асортимент товарів	чим більше товарів, тим більший % збільшення прибутку та рентабельності інтернет-магазину
Безпека	важливий елемент функціонування інтернет-магазину, захисту персональних даних та фінансових платежів
Комунікація	магазину слід підтримувати зв'язок з користувачами для надання оперативної допомоги, включаючи сервісне обслуговування, надання відгуків тощо
Користувацький досвід	інтернет-магазини можуть створити більш зручне і привабливе середовище для покупців, що призводить до підвищення конверсії, задоволеності клієнтів і ймовірності повторних покупок.

Визначення вимог, які впливають на роботу інтернет-магазину, важливі для того, щоб зрозуміти, як оптимізувати його роботу і досягти успіху. Розуміння факторів може допомогти визначити, на яких аспектах слід зосередитися в першу чергу, щоб оптимізувати ресурси та підвищити ефективність, слід постійно відстежувати та вдосконалювати роботу магазину, адаптовувати її до мінливих вимог та ринкових умов.

1.3 Аналіз прикладної сфери та існуючих практик

Аналіз прикладної сфери для розробки інтернет магазинів є важливим етапом при створенні такого проекту. Він дозволяє дізнаватись інформацію про вже розроблені рішення та робочий функціонал, які відмінно працюють. На даний момент існує безліч сайтів для продажу органічних продуктів. Велика частина з них це всесвітні сайти, які працюють для будь-якої точки світу.

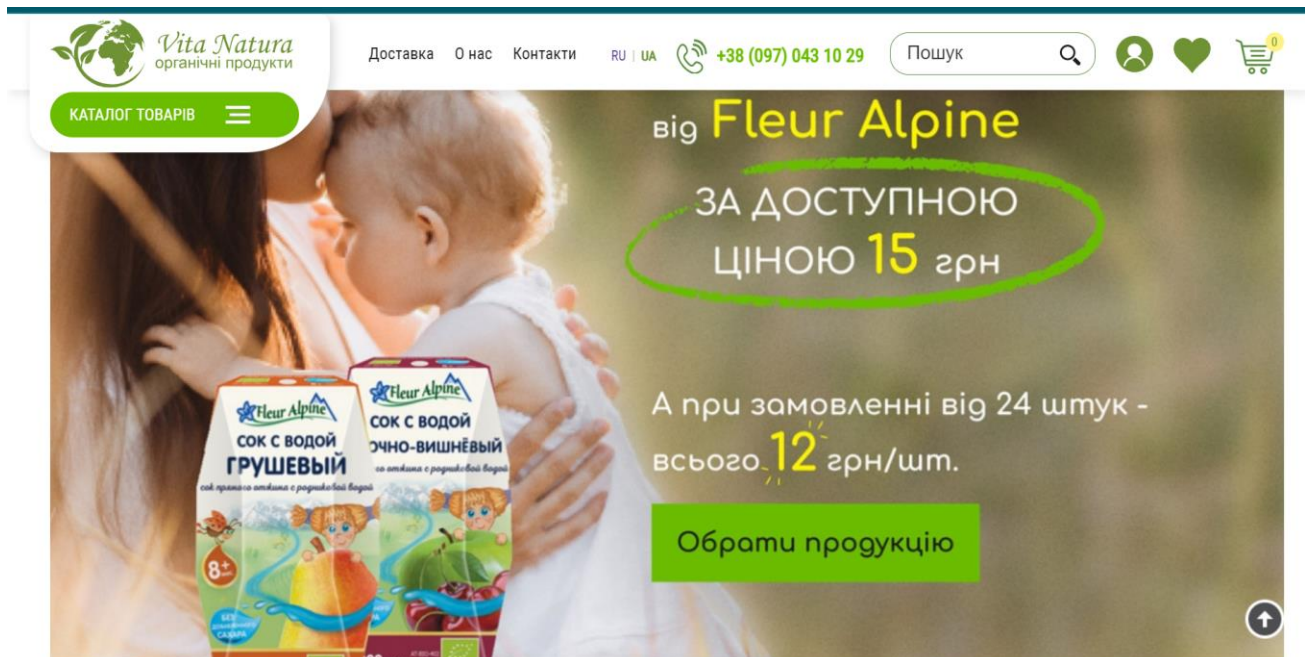


Рисунок 1.2 – Головна сторінка сайту VitaNatura

Для розробки ефективного інтернет-магазину доцільно проаналізувати існуючі аналоги на ринку, для виконання даної роботи були обрані наступні веб-сайти. Одним з лідерів продажу органічних продуктів є сайт [VitaNatura](#), вони пропонують невеликий асортимент товарів, а саме, дитяче харчування, продукти для здоров'я, натуральну косметику чи натуральні напої.

Також на сайті та простий дизайн, з додавання товарів у кошик, проста реєстрація та інформативна сторінка "Про нас". Для порівняння можна взяти сайт [FineOrganic](#), який також спеціалізується на продажі органічних товарів, які вирощуються та виробляються з дотриманням високих стандартів, вирощені без додавання хімікатів чи штучних добрив.

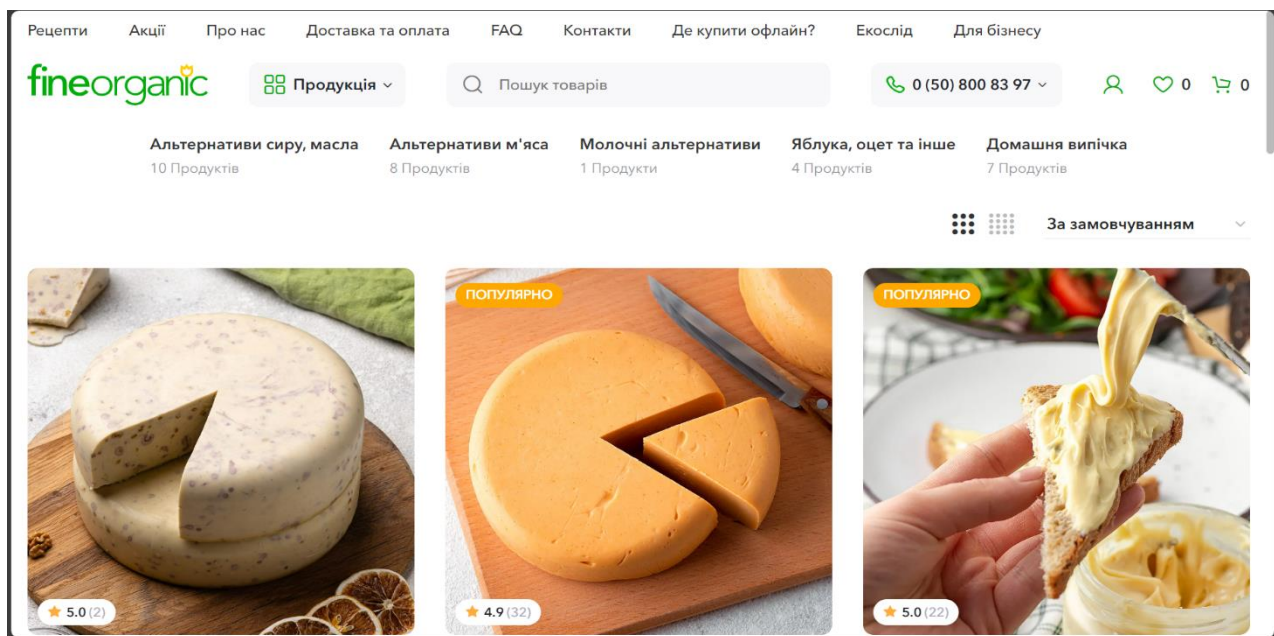


Рисунок 1.3 – Головна сторінка сайту FineOrganic

На сайті велика сітка товарів, яка заохочує користувача "подорожувати" по сторінках сайту все далі і далі. Магазин підтримує «прозорість» у своїх продуктах, що додає впевненості для покупки товарів з цього магазину.

Аналізуючи існуючі подібні продукти, важливо враховувати їхні сильні та слабкі сторони. А саме функціональність, доступність, користувацький інтерфейс, відгуки користувачів, ретельно вивчивши та проаналізувавши кожен наявний подібний продукт. Визначити його сильні та слабкі сторони і знайти розуміння, які функції є найбільш важливими для цього типу розробки. Загалом сайти, що продають органічні продукти, можна порівнювати за різними критеріями, які відображають якість, зручність та ефективність сайту, відобразивши її у вигляді схеми.

Таблиця 1.2- Порівняльна характеристика інтернет-магазинів органічних продуктів

Критерії	Інтернет-магазини	
	VitaNatura	FineOrganic
Ціни	Середні	Середні
Асортимент	Середній	Великий
Якість продуктів	Так	Так
Сервіс та підтримка клієнтів	Так	Так
Доставка	Так	Так
Новітні технології	Ні	Так
Акції	Ні	Так

Провівши аналіз обох магазинів, була визначена оцінка їх продуктивності, визначення слабких місць, перевірка швидкості підключень та завантажень даних веб-сайтів. Завдяки ресурсу [WebPageTest](#) який спеціалізується на швидкості завантаження будь-яких даних на сайті, було проведено швидкість завантаження обох сайтів, в результаті тестів виходить наступна інформація, сайт VitaNatura виявився

повільнішим в декілька разів, для прикладу завантаження першої сторінки зайняло ресурсу 13.9 секунд, що на 5 секунд довше ніж в FineOrganic. В цілому з обох магазинів можна зробити наступні висновки.

1) Vita Natura - це простий магазин органічних продуктів, з середнім візуальним виглядом, невеликим асортиментом товарів, призначений для простого продажу товарів, який

2) FineOrganic - це майбутній гігант з продажу продукції на ринку України, одним лише дизайном сайт заохочує переглядати сайт та перебування на ньому є приємним. Зручна навігація та велика кількість продуктів, які будуть збільшувати % прибутку власникам такого проекту, адже чим більше покупець на сайті, тим більший шанс, що він зробить більші покупки.

Отже, при розробці нашого проекту, слід врахувати сильні сторони сайтів конкурентів, а саме: кольорову гаму, можливість фільтрації продукції по категоріям, зручну реєстрацію, додавання елементів в корзину, та зручний опис продуктів для користувачів.

РОЗДІЛ 2 ОСНОВИ РОЗРОБКИ ІНТЕРНЕТ-МАГАЗИНУ

2.1 Опис функціональних вимог та постановка завдання

Розробка інтернет магазину з продажу органічної продукції має на собі за мету виконати практичне завдання із створення сайту з продажу якісних продуктів для людей, де застосовані найновітніші технології .Ці вимоги визначають основні функції та характеристики системи інтернет-магазину з використанням Django, які можуть бути налаштовані та розширені відповідно до конкретних потреб бізнесу. Визначення вимог до системи для інтернет-магазину з продажу органічної продукції через Django може бути розділене на кілька ключових аспектів.

Таблиця 2.1 – Визначення вимог інтернет-магазинів

Функціональні вимоги	Нефункціональні вимоги
<i>Реєстрація користувачів</i> - можливість легкої та зрозумілої реєстрації нових користувачів на веб-сайті.	<i>Безпека</i> - забезпечення захисту від несанкціонованого доступу, шифрування даних користувачів та платежів
<i>Авторизація та аутентифікація</i> - система авторизації користувачів для забезпечення доступу до особистого кабінету та замовлення товарів.	<i>Продуктивність</i> - забезпечення швидкої та ефективної роботи системи навіть при великому обсязі даних та трафіку.
<i>Пошук і фільтрація продуктів</i> - можливість знаходити продукти за ключовими словами, категоріями, ціною тощо.	<i>Масштабованість</i> - можливість легко розширювати функціональність та обсяги системи з ростом бізнесу.

Функціональні вимоги до органічних веб-сайтів включають широкий спектр можливостей, які дозволяють користувачам зручно шукати, переглядати та

купувати товари. Це лише деякі з основних функціональних вимог для успішної роботи інтернет-магазину. При розробці такого проекту, потрібно правильно розставити пріоритети та зробити все для зручного та подальшого підтримання такого сайту.

Почати слід з каталогу товарів, адже це основне за чим приходять користувач, каталог повинен бути розділений за категоріями (овочі, фрукти, соки, сушені продукти і т.д.) також повинна бути характеристика товарів, де користувач зможе легко прочитати вміст товару, такими характеристиками може бути: фото, вага, ціна, країна виробник, компанія виробника, опис товару тощо.

Наступним кроком йде створення кошика користувача та подальше оформлення замовлення, слід зробити зручне додавання товарів в кошик та легкий перегляд товарів в ньому, після чого користувач зможе оформити замовлення. Користувачам також потрібно мати можливість змінювати кількість товарів у кошику або видаляти їх.

Не мало важливим пунктом є реєстрація та вхід користувачів на такий сайт, користувачі повинні мати можливість створити власний обліковий запис та без проблем входити в систему де його буде очікувати його Власний Кабінет. Власний кабінет потрібен для перегляду товарів, які користувач взяв у магазині, редагування власного профілю та оформлення замовлень.

Правильним рішенням буде налаштувати адміністративну панель, для адміністраторів чи менеджерів сайту в якій можна швидко додати редагувати чи видалити інформацію про товар, слідкувати за кількістю товару чи оформлення замовлень.

Одним з найважливіших пунктів в описі постановки завдання є захист та безпека користувачів чи самого сайту від хакерських атак. Слід зробити акцент на шифруванні даних, використовуючи сторонні засоби безпеки або засоби які

вбудовано в технологію розробки сайту, заходи проти запобігання витоку даних про користувачів(паролі, телефони, банківські рахунки).

2.2 Моделювання системи інтернет-магазину

Моделювання системи інтернет-магазину важливий етап розробки проекту. Моделювання систем інтернет-магазинів сприяє більш ефективному та успішному впровадженню та функціонуванню бізнесу в онлайн середовищі. Широке використання інтернет-технологій при здійсненні покупок збільшило кількість відвідувачів інтернет-магазинів. Кількість відвідувачів інтернет-магазинів помітно йде на підвищення

Моделювання системи допомагає визначити функціональність, необхідну для інтернет-магазину. Сюди входить каталог товарів, система замовлень, система оплати та доставки, адміністративна панель тощо.

Архітектурне планування передбачає, що моделювання допоможе визначити архітектурну структуру системи, вибір технології для написання інтернет-магазину, з'єднання компонентів, бази даних, веб-сервери та технології безпеки.

Удосконалення процесів може базуватися на тому, що моделювання дозволить виявити можливості для оптимізації робочого процесу інтернет-магазину, додавання додаткових компонентів, підвищення якості обслуговування та скорочення часу обробки замовлень, додавання нових функцій для інтернет-магазину, по типу, оформлення знижок чи додавання промокодів, залишення відгуків тощо.

На основі моделі можна розробити та впровадити такі заходи безпеки як шифрування даних, двохфакторна автентифікація та моніторинг подій безпеки додавання токенів безпеки тощо.

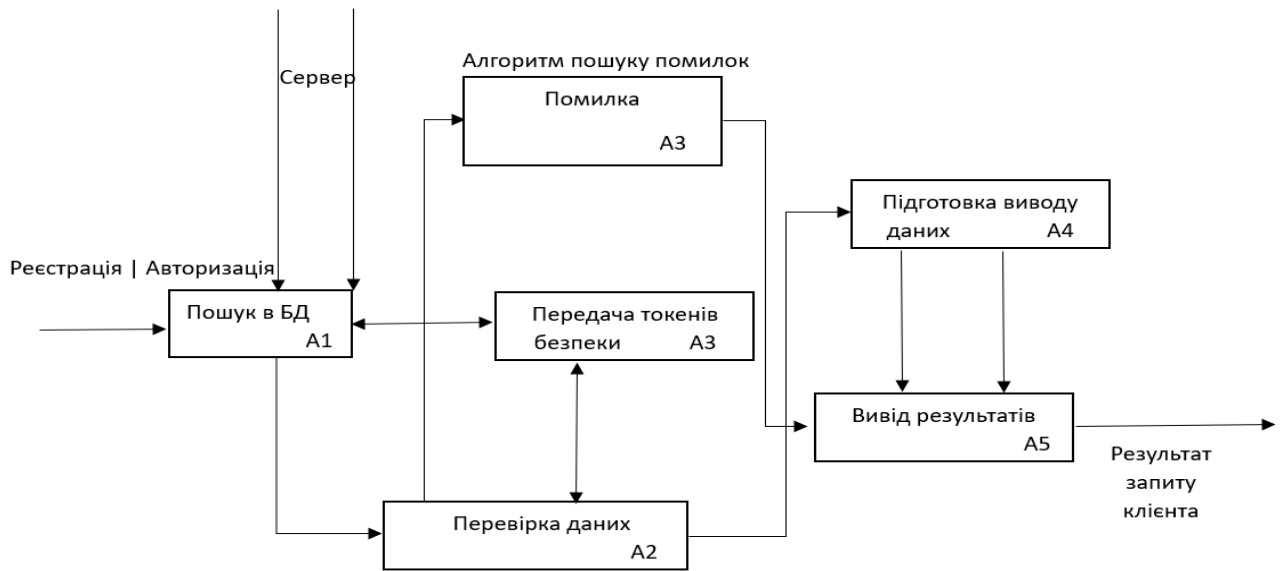


Рисунок 2.1 – Діаграма нульового рівня

Діаграма нульового рівню описує набір функцій взаємопов'язаних між собою для коректної роботи самого веб-сайту.

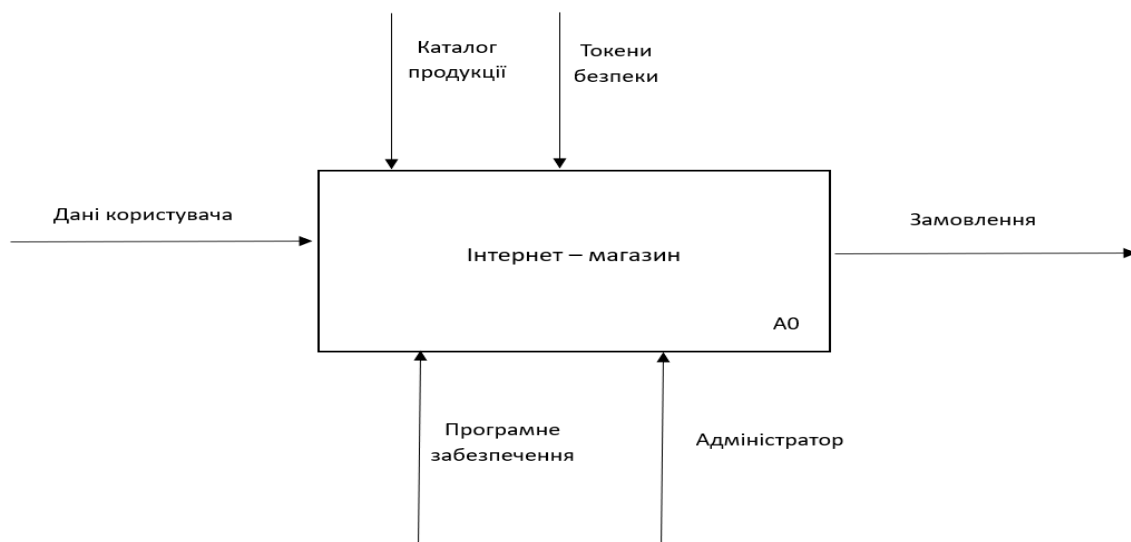


Рисунок 2.2 – Діаграма першого рівня

Моделювання системи інтернет - магазину є важливим кроком у створенні ефективної та надійної платформи електронної комерції. Він може оптимізувати бізнес-процеси, забезпечити зручність для користувачів, забезпечити масштабованість та адаптивність системи, інтегруватися з іншими критичними системами та покращити управління даними. За допомогою моделювання можна зменшити витрати на розробку та впровадження, забезпечити високий рівень безпеки даних та створити платформу, яка сприяє зростанню бізнесу та задоволеності клієнтів. Зрештою, це допоможе забезпечити успіх інтернет-магазину в конкурентному середовищі електронної комерції.

2.3 Розробка архітектури інтернет-магазину

Перед початком розробки магазину на тему "Органічні Продукти" слід сформулювати архітектуру такого сайту.

Цільова аудиторія - це група людей або організацій, які є потенційними споживачами або користувачами інтернет-магазину. Визначення цільової аудиторії є важливим кроком на шляху до успіху в бізнесі, оскільки дозволяє точно визначити на кого спрямовані ваші товари та послуги.

Потрібно визначити цільову аудиторію, розподіл цільової аудиторії за такими критеріями, як вік, рід занять, регіон тощо також важливим є розуміння основних потреб та очікувань користувачів від інтернет-магазину. Визначити, як повинен бути структурований каталог товарів, щоб користувачі могли легко знаходити потрібні їм продукти. Розробити зручний та інтуїтивно зрозумілий процес оформлення замовлення.

Вибір технологій важливий крок під час розробки проекту, також важливим кроком є правильно розроблений дизайн. Структура Back-End частини на фреймворку Django, розділенням самого проекту на міні додатки, реєстрація, оплата,

користувачі тощо. Структура FrontEnd-у використання принципів розділення FrontEnd-у на компоненти для підтримки легкої модифікації та розширення. Забезпечення реактивного дизайну для оптимізації взаємодії з користувачем. Забезпечення адаптивного дизайну для оптимального відображення на різних пристроях, розробка інтуїтивно зрозумілої системи навігації для зручного користування сайтом.

Швидкодія інтернет-магазину є важливою характеристикою, яка безпосередньо впливає на користувацький досвід та конверсію. При розробці архітектури інтернет-магазину важливо враховувати різні аспекти, які впливають на швидкодію.

Встановлення вимоги до швидкості завантаження сторінок і швидкості реакції сервера, щоб забезпечити хороший користувацький досвід. Врахувати потреби при можливості збільшення магазину, трафіку, кількості товару чи додавання іншого функціоналу сайту та визначити необхідні програмні та апаратні засоби для роботи інтернет-магазину.

Розробка архітектури бази даних інтернет-магазину передбачає детальне проектування структур даних, які забезпечують ефективне і безперебійне зберігання, оновлення та пошук інформації. Розроблення бази з уточненням всіх можливих зв'язків в ній. Слід сформулювати таблиці і зв'язати їх в правильний та зрозумілий код, та не забувати про масштабування проекту. Проектування слід робити таким чином:

1. Користувачі
2. Товари
3. Категорії
4. Корзина користувача

Безпека та захист є ключовими елементами розробки інтернет-магазину, оскільки важливо забезпечити конфіденційність, цілісність та доступність даних. Шифрування трафіку між клієнтом і сервером за допомогою HTTPS та CSRF-T.

Регулярне оновлення всіх компонентів, включаючи серверне програмне забезпечення, фреймворки та бібліотеки, щоб закрити потенційні вразливості. Використання системи безпеки, такі як брандмауери та аналізатори трафіку, для виявлення та запобігання DoS- і DDoS-атакам. Використовування мережі доставки контенту (CDN) для розподілу трафіку та зменшення навантаження на сервер. Використовування надійних методів автентифікації, такі як хешування паролів та двоетапна перевірка. Визначити права доступу користувачів і ролі для обмеження доступу до функцій сховища. Схему архітектури веб-сайту можна відобразити у такому вигляді.

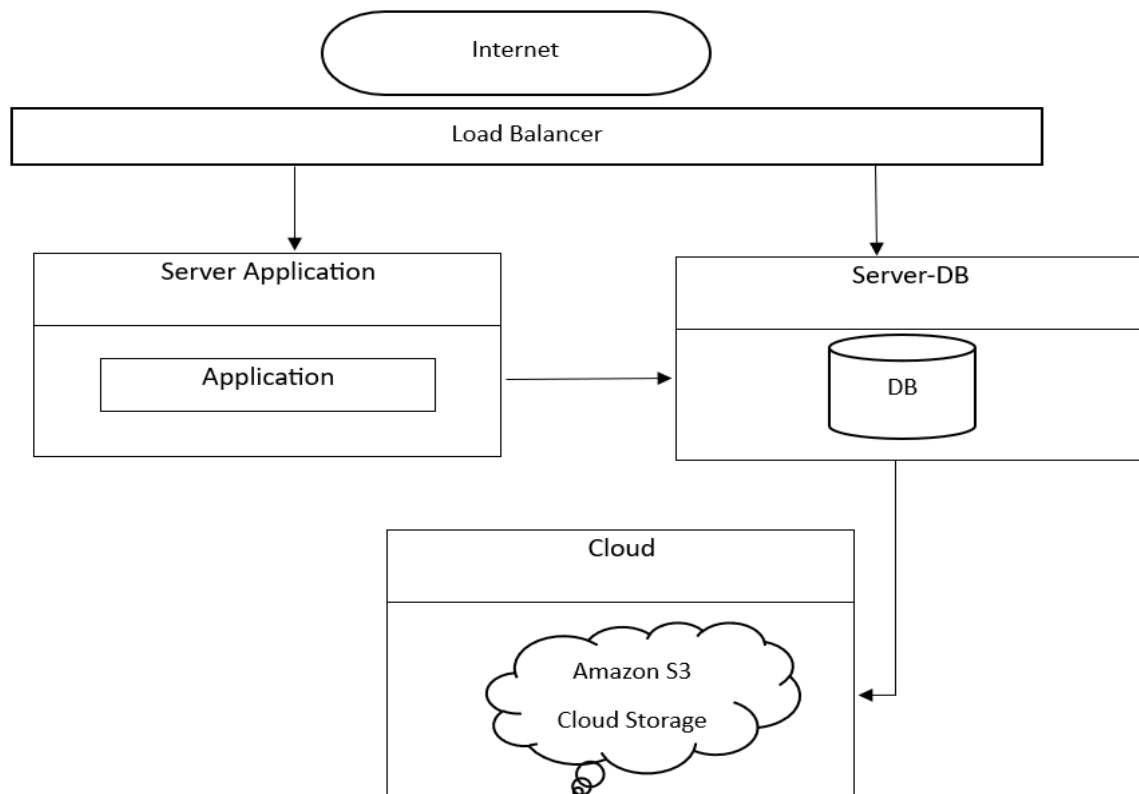


Рисунок 2.3 – Зображення архітектури сайту.

При з'єднанні до такого веб-сайту, йде завантаження даних сторінки, це може бути зроблено за допомогою бази даних (DB), або з хмарного сховища, яке представлено у вигляді Amazon S3 чи Cloud Storage. Як правило, дисковий простір досить дорогий, і цілком можливо, що ємність обраних дисків з часом вичерпається, і систему доведеться перенести на новий сервер. Це також потребуватиме додаткових робіт. Тому, якщо планується зберігати велику кількість медіаматеріалів або надавати своїм клієнтам можливість завантажувати матеріали, можна розглянути можливість зберігання такого контенту на зовнішньому сервісі. Більшість хмарних провайдерів пропонують послуги об'єктного зберігання.

2.4 Проектування структури бази даних інтернет магазину

Проектування структури бази даних для інтернет магазину OrganicFood з продажу органічних продуктів є надважливим етапом у розробці такого веб-сайту. Метою створення бази даних є зберігання даних, які будуть відображатись на сторінках сайту.

База даних складається з восьми основних таблиць:

1. Користувачі (Users)
2. Продукт (Product)
3. Готовий Набір (CompleteCart)
4. Країна Виробник (CountryProducingCategory)
5. Компанія Виробник (CompanyProducingCategory)
6. Категорія (Category)
7. Кошик (Cart)
8. Замовлення(Order)

Зображено блок схему бази для магазину. Зупинимось на ній більш детально щоб було зрозуміло. Тут зображено всі зв'язки в базі, усі моделі та їх спільну

взаємодію. Спершу слід визначити дані які будуть зберігатись. Це дає нам зрозуміти, що кожна таблиця має свої поля (такі таблиці як користувачі, продукти, категорії, та зв'язки з іншими таблицями).

Після цього визначається модель моєї бази даних, яка визначає структуру таблиць між собою. Кожна таблиця має різні поля, але вони можуть бути схожі, для таблиці "Продукт" є поля "вага", "ціна, "назва" та інші, для таблиці "Категорія" поля "назва" та "посилання".

В цілому під час проектування структури бази даних веб-сторінки часто необхідно ретельно проаналізувати вимоги до даних і встановити логічні зв'язки між даними, щоб забезпечити ефективну і надійну роботу веб-сторінки, для того, щоб забезпечити ефективну та безперебійну роботу веб-сторінок, необхідно встановити логічні зв'язки між даними.

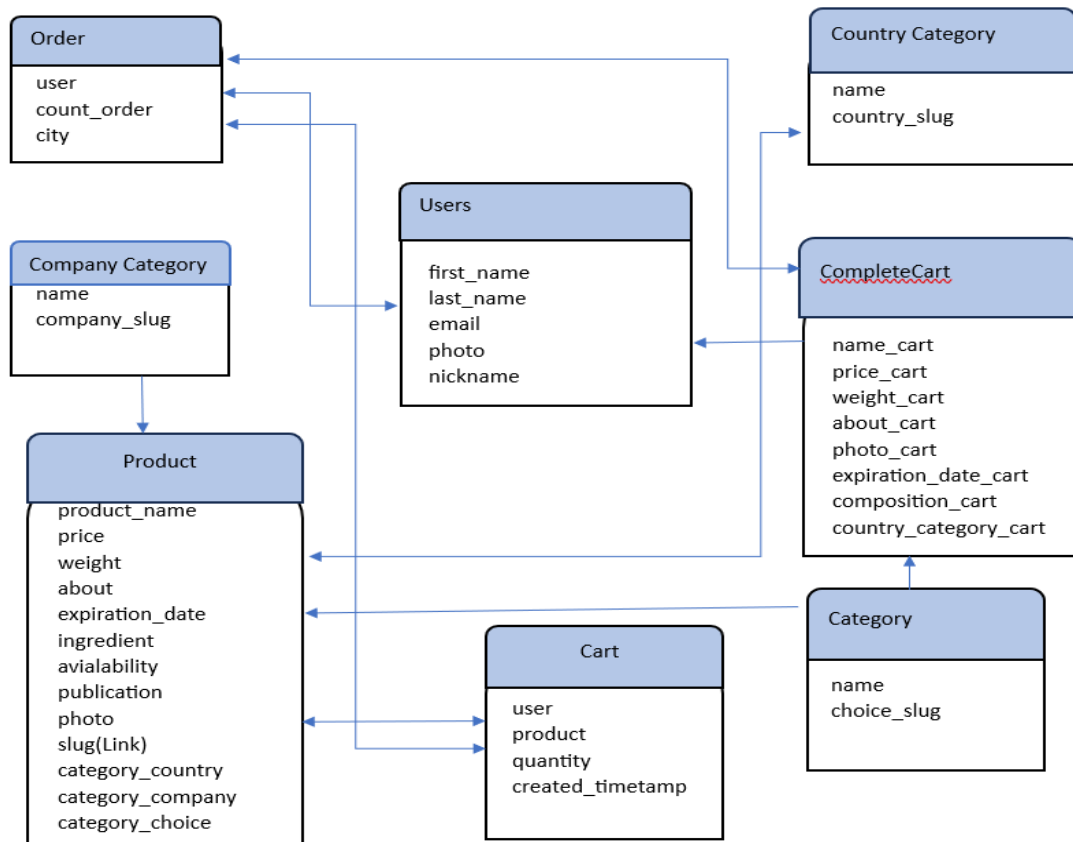


Рисунок 2.4 – Схема таблиць в базі даних

Базою для даного сайту було обрано SQLite, яку пропонує Django по замовчанню. Вона проста, швидка, надійна, зрозуміла та має хорошу швидкодію. SQLite - це система управління базами даних (СКБД), яка зазвичай використовується для розробки веб-сайтів. Зазвичай для веб-сайтів SQLite можна використовувати для управління самою базою чи онлайн замовлень або обробки замовлень.

На даний момент, в світі програмування є безліч інструментів для розробки сайтів. Одними з найпопулярніших фреймворків на мові програмування Python є Django, Flask, Tornado, FastAPI, на JavaScript Express.js, Laravel - на PHP. Мовою програмування було вибрано Python, тому слід розібратись, що вибрати серед двох надпотужних фреймворків для реалізації проекту.

Перед мною було поставлене завдання обрати фреймворк, виділивши плюси і мінуси та обрати кращий з них.



Рисунок 2.5 - Порівняння Django та Flask

Flask - це мікрофреймворк для створення веб-додатків на Python. Він поставляється з мінімальним набором інструментів, але підтримує розширення так, ніби вони реалізовані у Flask. Розширення цього мікрофреймворку дозволяють

взаємодіяти з базами даних, перевіряти форми, контролювати навантаження на сервер, змінювати автентифікацію та багато іншого.

Найкращі особливості фреймворку – це легка вага, відкритий вихідний код та мінімальна кількість коду для розробки програми. Більше підходить для менших та середніх проектів, де розробники хочуть більше свободи в обраній структурі та компонентах.

Простота використання фреймворку Flask пояснюється тим, що немає багатофункціональних компонентів, розробники можуть вибирати і додавати бібліотеки, які їм потрібні, відповідно до вимог свого проекту.

Завдяки своїй гнучкості Flask пропонує велику свободу вибору бібліотек, баз даних та інших інструментів. Розробники можуть використовувати лише ті компоненти, які їм потрібні для конкретного проекту. Інструмент дозволяє розширювати її функціональність за допомогою плагінів і модулів, які можна легко інтегрувати у проект.

В кожному інструменті є зворотня сторона, в фреймворку Flask на мові програмування Python можна виділити наступні недоліки.

Через відсутність готових рішень розробники можуть спочатку використовувати лише мінімальний функціонал Flask. Якщо їм потрібні розширені можливості, такі як автентифікація користувачів, їм потрібно реалізувати їх самостійно.

Немає вбудованої багатопотоковості - Flask розроблявся як однопотоковий фреймворк. Для керування багатопотоковими веб-додатками потрібно встановити додаткові бібліотеки.

Коли проекти починають рости і ускладнюються, збільшувати, тобто масштабувати такий проект може стати важко підтримувати необхідну функціональність. Для цього слід писати нові функції та думати різні вирішення проблем, а для цього потрібно потратити більше часу для написання.

Flask не має вбудованої адміністративної панелі, як його однокласник - Django. Для управління адміністративними даними потрібно створити власне рішення.

Django - це високорівневий веб-фреймворк, написаний мовою програмування Python, який прискорює розробку веб-додатків, надаючи готові до використання інструменти та структуру проекту. Django запроваджує концепцію "в коробці" і включає в себе набір компонентів і бібліотек, які забезпечують широкий спектр функціональних можливостей.

До плюсів Django можна віднести наступний факт, що автоматично генерується адміністративна панель під час створення програми. Це позбавляє розробника необхідності створювати адмінку вручну. За допомогою сторонніх програм дефолтну консоль керування можна вдосконалити та адаптувати під потреби свого проекту. Крім того, фреймворк дозволяє налаштовувати інтерфейс по замовчуванню для адміністративної панелі.

Популярні мови програмування мають бібліотеки, які допомагають вирішувати спеціальні завдання Django в цьому плані надає готові до використання рішення, такі як функції, класи та конфігурації під час використання бібліотек.. Ці рішення розширюють функціональність мови і спрощують розробку додатків.

Django підтримує використання бібліотек при розробці веб-додатків. Найчастіше використовуються такі бібліотеки:

- Django REST Framework - для простого використання API.
- Django CMS - корисний інструмент для управління контентом.
- Django-allauth - використовується для реалізації функцій реєстрації, аутентифікації та управління обліковими записами.

Django реалізує Object Relational Mapping (ORM), який дозволяє додаткам взаємодіяти з базами даних. ORM автоматично передає дані з баз даних, таких як

PostgreSQL або MySQL чи SQLite3, в об'єкти, що використовуються в коді програми.

Проміжне програмне забезпечення(Middleware) в Django може реалізовувати додаткову функціональність для обробки HTTP-запитів та відповідей. Його можна використовувати для виконання різних завдань, таких як автентифікація, ведення журналів та обробка помилок.

Та багато інших привілежій, які присутні в Django. В кожному інструменті є зворотня сторона, в фреймворку Django на мові програмування Python можна виділити наступні недоліки:

Можливості Django можуть бути занадто великими для невеликих проектів. Важливо підкреслити що в екосистемі Python є інші фреймворки, які можна використовувати в таких ситуаціях. Фреймворк має вбудовані компоненти і правила, які є потужними і ефективними для великих і складних проектів. Однак, для менших проектів, які не потребують всієї функціональності та структури Django, використання цього фреймворку може здатися невиправдано складним. В інших випадках, якщо легше розбити проект на меншу кількість компонентів, менший і легший фреймворк може бути більш доречним.

Деякі модулі Django, такі як ORM та форми, важко модифікувати. Розробникам доводиться витратити багато часу на зміну внутрішньої структури фреймворку та використання альтернативних інструментів.

РОЗДІЛ 3 РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ З ВИКОРИСТАННЯМ ФРАЙМВО- РКУ DJANGO

3.1 Розробка основних функціональних компонентів інтернет-магазину

Для ефективної роботи інтернет-магазину важливо не тільки мати чудовий UI/UX, але й надійний код, здатний витримувати високі навантаження. Під час створення такого проекту, його можна розділити на дві важливі категорії це Бекенд та Фронтенд.

Back-End - це серверна частина інтернет-магазину. Вона містить функціональний код і відповідає за правильну реалізацію бізнес-логіки. Коли мова заходить про розробку інтернет-магазину, мова Python стає однією з пріоритетних технологій, і ось чому.

Висока швидкість розробки у мові програмування Python надає розробникам доступ до багатьох стандартних бібліотек, включаючи готові функції, класи та модулі, що позитивно впливає на швидкість розробки. Крім того, Python - дуже компактна мова, і робота на ній вимагає написання меншої кількості рядків коду, ніж, наприклад, на Java.

Універсальність вибраної мови після отримання результатів дозволяє, отримати відчутні результати від нашого веб-сайту, ми можемо захотіти розширити свій бізнес і запустити мобільний додаток, наприклад. У цьому випадку ми можемо скористатися готовим Back-End-ом на Python і заощадити значну частину свого бюджету.

Навіть після релізу ми можемо вносити необмежену кількість змін до Python коду, як для впровадження нових модулів, так і для оптимізації існуючого функціоналу. Таким чином, з часом такий інтернет-магазин ніколи не втратить своєї актуальності і завжди буде відповідати очікуванням наших користувачів.

Оскільки діяльність інтернет-магазину пов'язана з фінансовими операціями, надійність коду є дуже важливим параметром, і Python-фреймворки, такі як Django і Flask, за замовчуванням використовують найсучасніші функції безпеки, щоб забезпечити захист ресурсів від SQL-ін'єкцій, XSS, клікджекінгу та інших видів атак.

Front-End - це клієнтська частина, де користувач взаємодіє з браузером. Для цього розробники виконують верстку, створюють анімацію, таблиці, графіку та інші компоненти інтерфейсу. Дизайн робить інтернет-магазини зручними, комфортними та такими, що запам'ятовуються. Але для цього потрібен грамотний підхід і дотримання сучасних тенденцій. Представлення FrontEnd частини зроблено за допомогою HTML CSS та JavaScript.

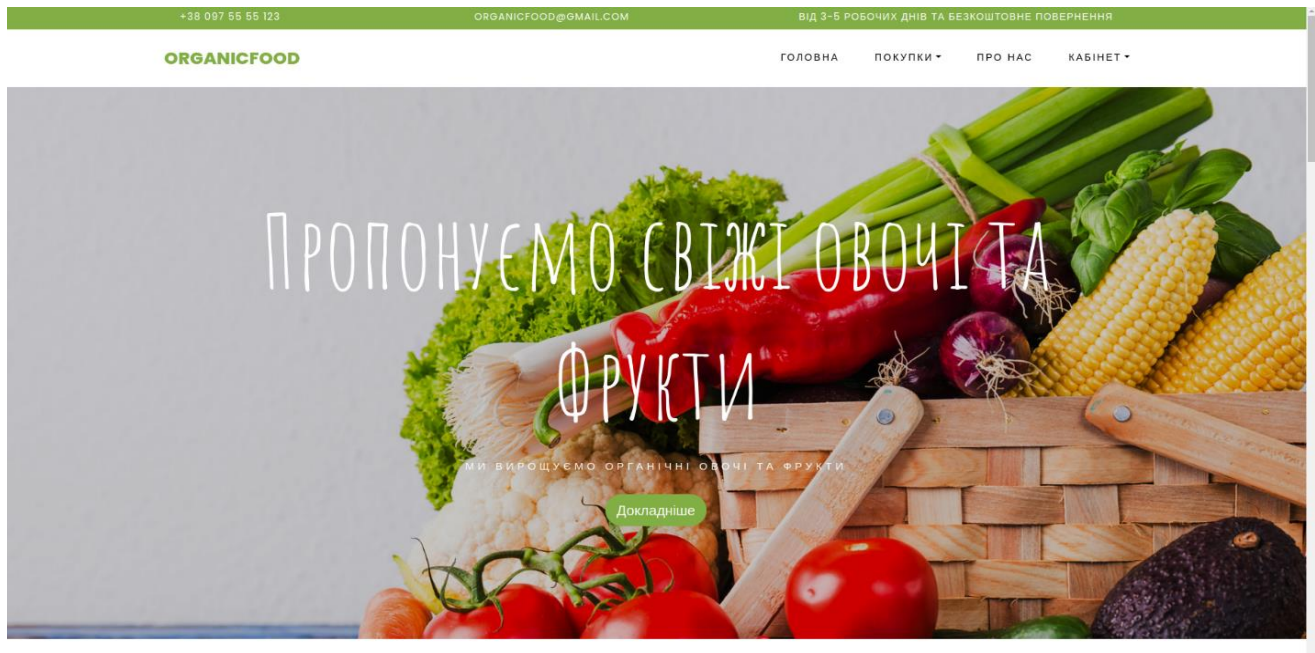


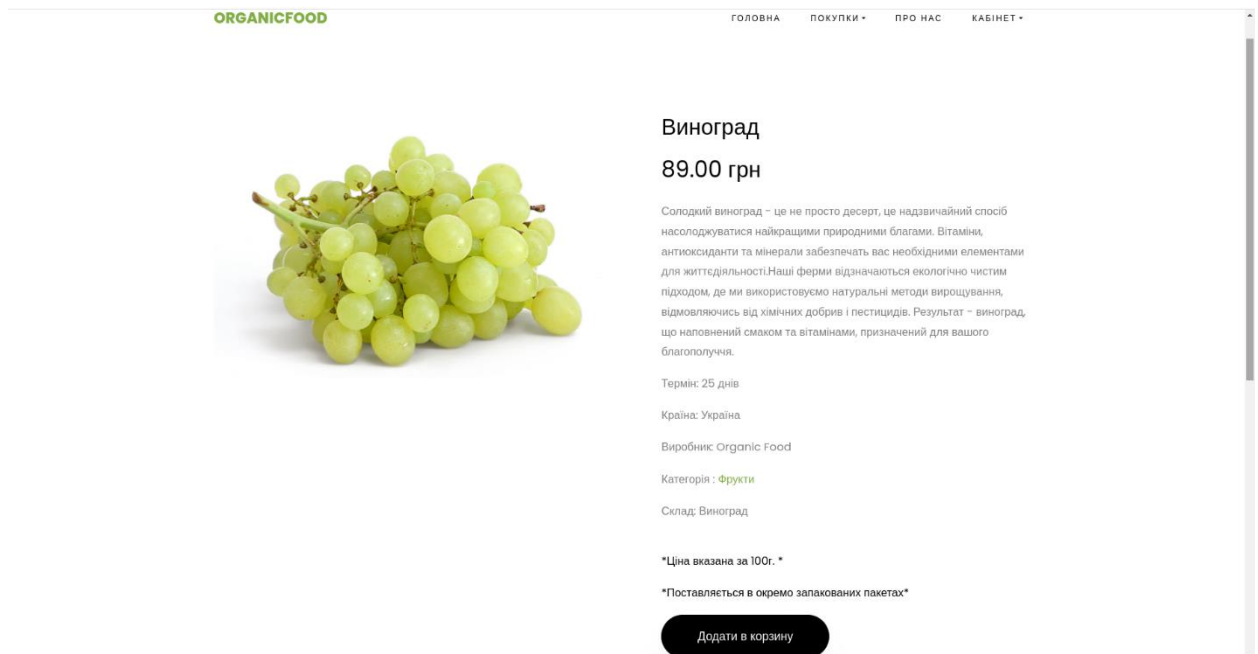
Рисунок 3.1 - Зовнішній вигляд сайту OrganicFood

Коли ми розібрались з технологіями слід приступити до компонентів онлайн магазину для найкращої роботи такого сайту.

Каталог товарів, те за чим приходять покупець, можна уявити себе покупцем, та зрозуміти чи подобається сайт особисто нам, тому потрібно забезпечити

зручний та доступний перегляд товарів і продуктів магазину, з можливістю фільтрації за категоріями. Сюди входять елементи навігації по сайту і меню пошуку товарів - вони завжди повинні бути перед очима клієнта.

Окремі сторінки для кожного товару - це частина веб-сайту, яку повинен мати інтернет-магазин. Покупець повинен знати, що він купує та бачити детальну інформацію продукту, тому слід реалізувати сторінки з деталями продукту, вага ціна, країна виробник та інші дані товару.



ORGANICFOOD ГОЛОВНА ПОКУПКИ • ПРО НАС КАБІNET •

Виноград

89.00 грн

Солодкий виноград – це не просто десерт, це надзвичайний спосіб насолоджуватися найкращими природними благами. Вітаміни, антиоксиданти та мінерали забезпечать вас необхідними елементами для життєдіяльності. Наші ферми відзначаються екологічно чистим підходом, де ми використовуємо натуральні методи вирощування, відмовляючись від хімічних добрив і пестицидів. Результат – виноград, що наповнений смаком та вітамінами, призначений для вашого благополуччя.

Термін: 25 днів
 Країна: Україна
 Виробник: Organic Food
 Категорія: Фрукти
 Склад: Виноград

Ціна вказана за 100г.
 Поставляється в окремо запакованих пакетах

[Додати в кошику](#)

Рисунок 3.2 - Вигляд опису продукту.

Весь процес з кошиком та продуктами повинен відповідати правилу: "чим простіше і швидше, тим краще". Можливість користувачам додавати продукти до кошика, який зберігатиметься в базі для зручного замовлення потім. Надання можливості користувачам змінювати кількість продуктів та видаляти їх з кошика перед оформленням замовлення.

Слід надати можливість користувачам створювати облікові записи та авторизуватися для оформлення покупок та налаштувань.

Розділ власного кабінету та особистих даних дає озвіл користувачам та оновлювати особисті дані в обліковому записі, перегляд продукції.

Інформація про доставку, способи оплати, повернення, гарантії та контактні дані повинна бути надана клієнтам у зрозумілій формі. Чим більше контактних даних, тим краще. При потребі користувач повинен мати змогу залишити питання побажання чи скарги для веб-сайту.

3.2 Розробка back-end частини інтернет-магазину з використанням фреймворку Django

Ця робота присвячена детальному огляду та аналізу процесу розробки бекенд-частини інтернет-магазину з використанням фреймворку Django. Метою є не тільки створення функціонально ефективного бекенду, а й визначення ключових міркувань, які можуть допомогти створити back-end, що відповідає найвищим стандартам безпеки, масштабованості та надійності, які є вимогами до інтернет-магазину.

Фреймворк Django, базований на мові програмування Python, забезпечує розробникам потужний інструментарій для швидкої та ефективної реалізації цих завдань.

Першим кроком у розробці є визначення моделі даних - Django використовує класи Python для створення моделі, яка представляє собою таблицю бази даних. Атрибути кожного класу визначають поля таблиці, а тип даних визначає тип цього поля. Фреймворк використовує ORM для спрощення взаємодії з базами даних, використання ORM дозволяє розробникам працювати з даними в об'єктно-орієнтованому підході, а не за допомогою прямих SQL запитів. Він надає потужну систему управління запитамі для складного відбору та фільтрації даних. Розробникам не потрібно писати складні SQL запити і вони можуть взаємодіяти з даними

за допомогою об'єктно-орієнтованого підходу. Для прикладу простий запит в БД буде виглядати так: *(Product.objects.filter(publication=True))*, яка означає, витягнути дані з таблиці Product за фільтром "публікація".

Django автоматично створює інтерфейс адміністратора для роботи з даними в базі даних. Адміністратори можуть легко додавати, редагувати та видаляти записи без прямого втручання в код. Також в адміністративній панелі можна редагувати статус користувача, назначивши його менеджером чи навпаки заблокувати обліковий запис, ще дає можливість переглядати останній вхід на сайт.

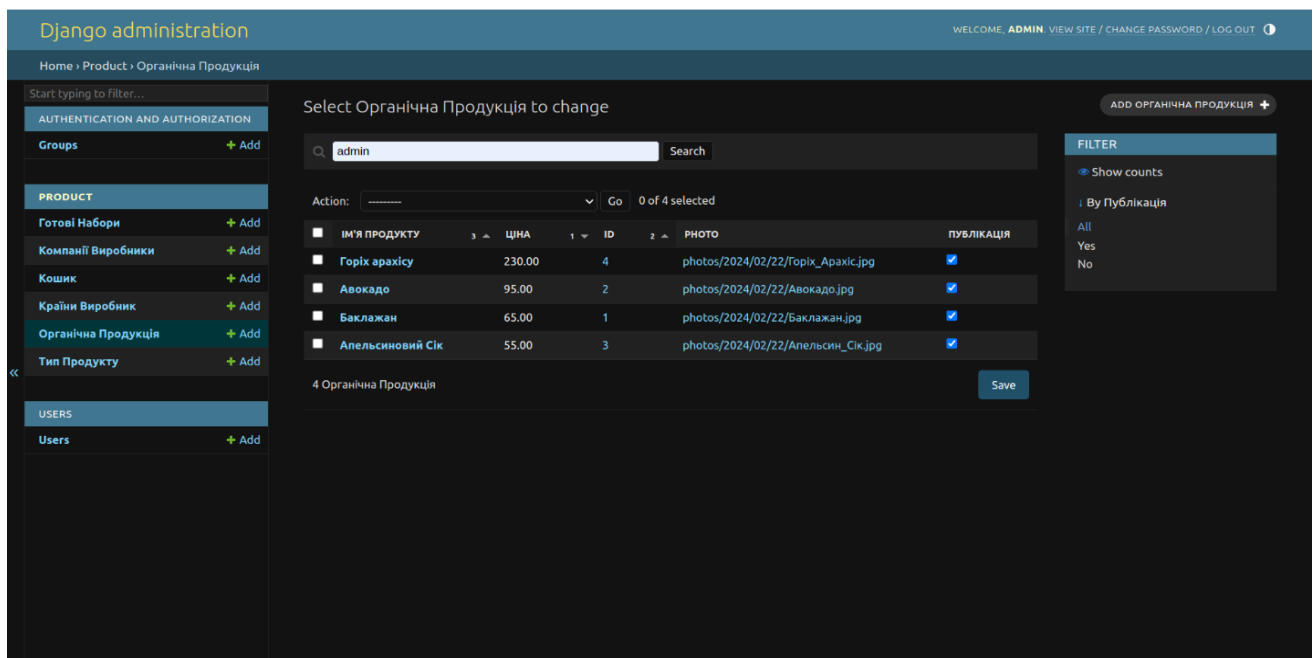


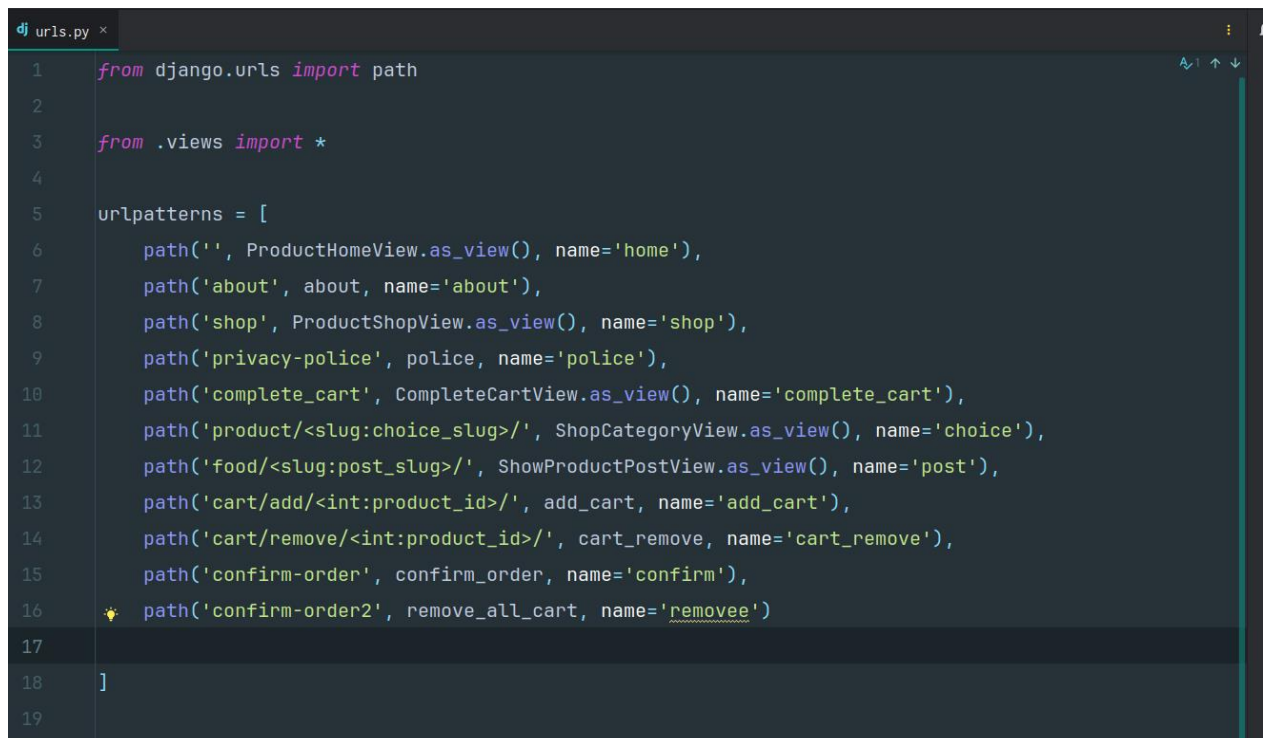
Рисунок. 3.3 - Фільтрування даних по спаданню цін

Система маршрутизації визначає, які частини коду викликаються для обробки певної URL-адреси. Маршрутизація визначається у файлі `urls.py`, в якому можуть бути вказані шляхи до різних частин BackEnd-у програми.

Реєстрація та авторизація є важливими аспектами розробки веб-додатків, особливо в контексті інтернет-магазинів. Django надає вбудовані інструменти для реалізації цих функцій, роблячи процес захисту користувачів і контролю доступу до ресурсів дуже простим і ефективним.

Система автентифікації забезпечує перевірку облікових даних користувачів та визначення дій, які може виконувати кожен користувач. Структура включає користувачів і групи, саму систему дозволів і вбудовану модель для визначення того, чи можуть користувачі виконувати завдання, які форми і екрани доступні авторизованим користувачам, а також доступ до контенту з обмеженим доступом. Адміністратор легко може поміняти права для користувачів, та назначити їм додаткові доступи на сайті.

Авторизація - це процес надання користувачеві певних прав доступу після успішної аутентифікації. Django використовує декоратори для встановлення прав доступу до певних подань. Декоратор `@login_required` може бути використаний, щоб гарантувати, що лише авторизовані користувачі можуть отримати доступ до певних ресурсів, наприклад перегляд власного кабінету.

A screenshot of a code editor window titled 'dj urls.py'. The code defines a list of URL patterns for a Django application. The patterns are: a root path for 'home', paths for 'about', 'shop', 'privacy-police', 'complete_cart', 'product/<slug:choice_slug>', 'food/<slug:post_slug>', 'cart/add/<int:product_id>', 'cart/remove/<int:product_id>', 'confirm-order', and 'confirm-order2'. Each path is associated with a specific view function and a name. The code is as follows:

```
1 from django.urls import path
2
3 from .views import *
4
5 urlpatterns = [
6     path('', ProductHomeView.as_view(), name='home'),
7     path('about', about, name='about'),
8     path('shop', ProductShopView.as_view(), name='shop'),
9     path('privacy-police', police, name='police'),
10    path('complete_cart', CompleteCartView.as_view(), name='complete_cart'),
11    path('product/<slug:choice_slug>', ShopCategoryView.as_view(), name='choice'),
12    path('food/<slug:post_slug>', ShowProductPostView.as_view(), name='post'),
13    path('cart/add/<int:product_id>', add_cart, name='add_cart'),
14    path('cart/remove/<int:product_id>', cart_remove, name='cart_remove'),
15    path('confirm-order', confirm_order, name='confirm'),
16    path('confirm-order2', remove_all_cart, name='removee')
17
18 ]
19
```

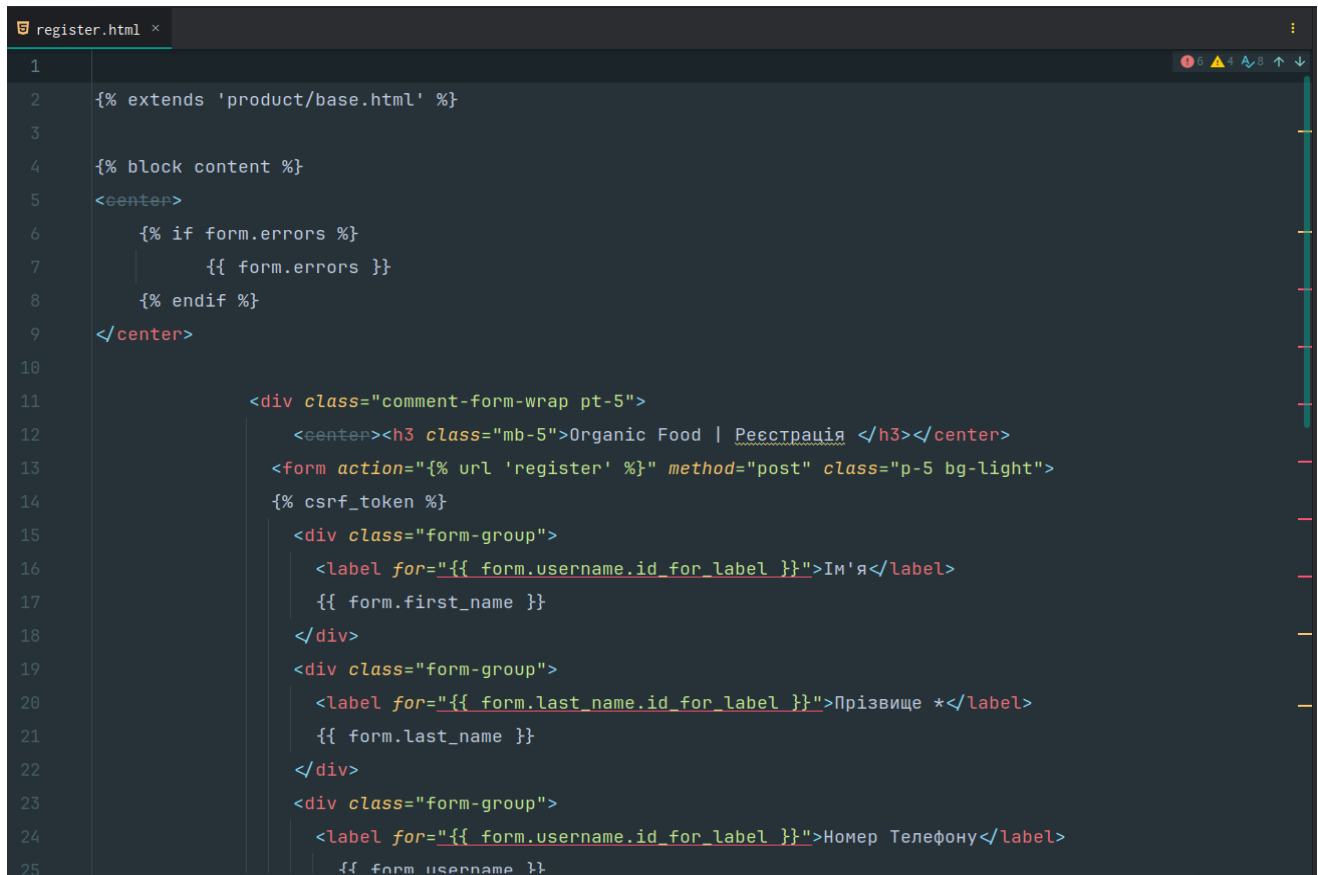
Рисунок. 3.4 - Відображення URL-адрес

На даному зображенні показані маршрути для користувача, при переході на якийсь з них буде відображена конкретна сторінка HTML. Для прикладу у стрічці

12 `food/<slug:post_slug>/` користувач перейде на вибраний товар, де буде відображено товар по "слагу", це допоміжне слово для зручного та читабельного показу нашого URL на сайті. За допомогою маршрутизації можна розділити логіку додатку і ефективно визначати, які дії виконувати для кожного запиту користувача.

Django забезпечує вбудовані механізми захисту від різноманітних атак, таких як SQL-ін'єкції, атаки на злам та інші загрози безпеки. Використання вбудованих засобів безпеки робить розробку Backend-у надійною. Дана робота захищається засобами які пропонує сам фреймворк, а саме CSRF-Token.

CSRF Token - це унікальний, рандомізований токен, який генерується сервером і додається до форми на сторінці, що показується користувачеві. Цей токен надсилається разом із запитом, і сервер перевіряє дійсність токена, перевіряючи, чи збігається токен у запиті з раніше виданим токеном.



```

1
2 {% extends 'product/base.html' %}
3
4 {% block content %}
5 <center>
6     {% if form.errors %}
7         {{ form.errors }}
8     {% endif %}
9 </center>
10
11 <div class="comment-form-wrap pt-5">
12     <center><h3 class="mb-5">Organic Food | Реєстрація </h3></center>
13     <form action="{% url 'register' %}" method="post" class="p-5 bg-light">
14         {% csrf_token %}
15         <div class="form-group">
16             <label for="{ form.username.id_for_label }">Ім'я</label>
17             {{ form.first_name }}
18         </div>
19         <div class="form-group">
20             <label for="{ form.last_name.id_for_label }">Прізвище *</label>
21             {{ form.last_name }}
22         </div>
23         <div class="form-group">
24             <label for="{ form.username.id_for_label }">Номер телефону</label>
25             {{ form.username }}

```

Рисунок 3.5 - Захист від атак за допомогою токена

Безпека з використанням токенів CSRF - це стандартний і ефективний спосіб захисту від атак зловмисників, які намагаються зловживати довірою автентифікованих користувачів. Токени CSRF можна використовувати для захисту веб-додатків від потенційних атак.

CSRF-Token – повинен бути вбудованим в верхню частину коду для того, щоб бути доступним для всіх форм та запитів на сторінці, так він гарантовано буде включений у запит і захистить дані користувача. Токен використовується у дужках зверху документу, для безпеки користувачів, при спробі поставити його внизу, програма видасть помилку, яка повідомляє про те, що токен не використовується.

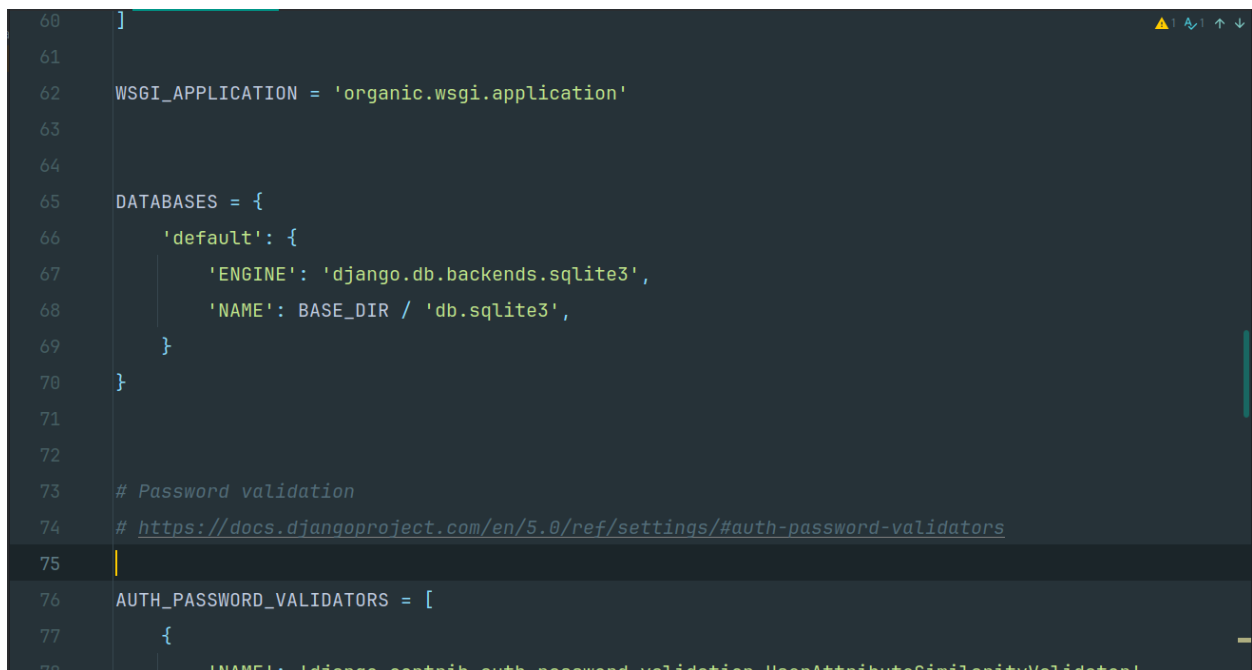
3.3 Налагодження та впровадження бази даних інтернет-магазину

Створення ефективної та надійної бази даних - важливий етап розробки інтернет-магазину органічних продуктів на фреймворку Django. Налаштування та впровадження бази даних передбачає створення моделей, міграцію даних та забезпечення ефективної взаємодії з ними. Це складний процес, але при правильному плануванні та використанні можливостей фреймворку, він може забезпечити надійну та продуктивну основу для інтернет-магазину.

Завдяки технології Django ми можемо спростити зв'язок з Базою даних в моєму проекті. Серед баз даних є багато варіантів вибору, але я зупинився на SQLite, вона представлена у Django за замовчуванням, що облегує нам роботу з нею, а ще вона проста у використанні, оскільки весь двигун бази даних інтегрований безпосередньо в додаток. Це означає, що не потрібен окремий сервер або конфігурація. Файли баз даних SQLite компактні і не потребують зовнішньої конфігурації. Це робить їх ідеальними для використання в мобільних та інших додатках з обмеженою пам'яттю та ресурсами, що потрібно для даного проекту, адже користувачі можуть замовляти товари з мобільних пристроїв, також потребує мінімальних

ресурсів для роботи. SQLite відома своєю високою продуктивністю та швидкістю відгуку, особливо при виконанні операцій читання. Вона ідеально підходить для різноманітних завдань, таких як зберігання конфігурацій, кешування даних та обробка локальних баз даних.

Один з найбільших плюсів це - транзакції, оскільки вони забезпечують атомарне зберігання даних шляхом об'єднання декількох транзакцій бази даних в одну логічну одиницю, для сайту з продажу продуктів це пункт номер один, тому що кількість користувачів може бути великою, щоб не відлякати покупців, потрібно швидше з'єднання.

A screenshot of a code editor with a dark background and light-colored text. The code is Python, specifically Django settings. It shows the configuration for the database engine and name. The code is as follows:

```
60 ]
61
62 WSGI_APPLICATION = 'organic.wsgi.application'
63
64
65 DATABASES = {
66     'default': {
67         'ENGINE': 'django.db.backends.sqlite3',
68         'NAME': BASE_DIR / 'db.sqlite3',
69     }
70 }
71
72
73 # Password validation
74 # https://docs.djangoproject.com/en/5.0/ref/settings/#auth-password-validators
75
76 AUTH_PASSWORD_VALIDATORS = [
77     {
78         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator'
```

Рисунок 3.6 - Зв'язок з базою.

Для початку створення моделей слід правильно обдумати які саме будуть поля в таблицях бази, а саме для таблиці Product з полями:

1. Ім'я продукту.
2. Ціна.
3. Вага.
4. Про товар.

5. Термін придатності.
6. Склад товару.
7. Наявність.
8. Публікація.
9. Фото.

Товар мусить мати виробника. Це необхідно вказати для того, щоб користувачі могли орієнтуватися на якість товару через рейтинги і популярність бренду. У структурі бази даних інтернет-магазину наявність цієї таблиці необхідна.

```

74 class CompanyProducingCategory(models.Model):
75     name = models.CharField(max_length=100, db_index=True, verbose_name='Компанія виробник')
76     company_slug = models.SlugField(max_length=255, blank=True, null=True, unique=True, db_index=True, verbose_name='URL')
77
78     def __str__(self):
79         return self.name
80
81     def get_absolute_url(self):
82         return reverse( viewname: 'company', kwargs={'company_slug': self.company_slug})
83
84     class Meta:
85         verbose_name = 'Компанію Виробник'
86         verbose_name_plural = 'Компанії Виробники'
87         ordering = ['id']
88
89

```

Рисунок 3.7 - Таблиця компанії виробника продукту.

Основне призначення категорії сутності в базі даних інтернет-магазину - групувати товари. Категорії мають ієрархію (наприклад, категорія-підкатегорія) і повинні розглядатися як атрибути. В даній роботі категорій 4 це - Фрукти, Овочі, Соки, Сушені(Гриби, горіхи, овочі тощо)

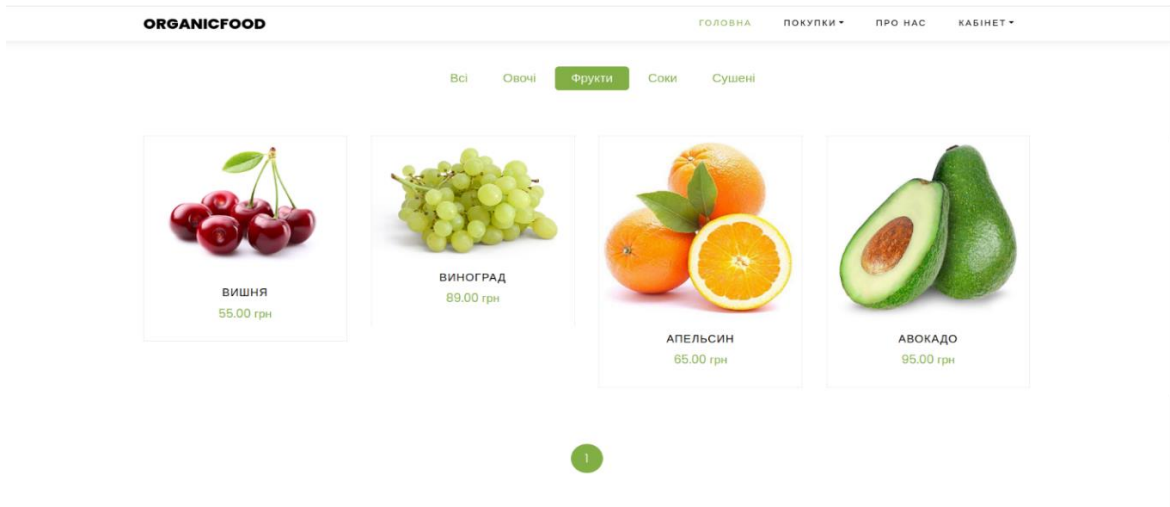


Рисунок 3.8 - Зображення категорій на сайті.

В сучасному світі продукти вирощують у всіх точках нашої планети, тому користувач потрібен знати звідки приїхав даний товар, таблицю Країни Виробника слід пов'язати з таблицею Product для зручності відображення на сторінці продукту.

Однією з ключових таблиць в базі даних є користувач, в ході виконання дипломної роботи я використав модель User, додавши до неї поле "ImageField", яке дозволить користувачу додати свою фотографію у власному кабінеті.

Хорошою практикою є додавання на сайт логіки "Корзина Користувача". Це дає змогу збільшити дохід сайту, тим що користувач не відправляється робити покупку зразу після вибраного, а "ходить" по сайті в пошуках інших товарів, які йому слід купити. Натиснувши кнопку "Додати в кошик" користувач додає продукцію в свій електронний возик для подальшої його покупки, всі елементи корзини користувач може переглянути коли перейде на сторінку "Кабінет", на ньому він побачить власні дані, фото та елементи корзини, які може оформити для замовлення.

Коли поля готові, ми починаємо робити моделі та зв'язки між ними. Після успішного написання моделі для Баз Даних слід прописати 2 прості команди для створення таблиць, це команди :

```
python manage.py makemigrations
```

коли міграції успішні, та пройшли без помилок пишемо:

```
python manage.py migrate.
```

3.4 Оцінка ефективності проекту

Інтернет-магазини сьогодні це невід'ємна складова покупок, людям в рази легше та швидше замовити продукти, техніку чи одяг з доставкою додому враховуючи ці факти, було поставлено завдання розробити веб-сайт, який дасть змогу відвідувачам сайту оформляти замовлення онлайн будь-де та будь-коли. Завдяки таким можливостям сайтів, вони стали частиною нас, адже завдяки веб-сайтам можна реалізовувати будь-яку ідею, їх було описано в Розділі 1 (1.1).

Процес проектування здійснюється для того, щоб краще зрозуміти архітектури різних компонентів і систем, які будуть розроблятися, системи, що розробляється, та її робочих частин. Оцінка проекту інтернет-магазину включає в себе ряд факторів, які визначають складність, обсяг робіт і вартість проекту. Під час оцінювання проекту інтернет магазину з продажу органічної продукції можна вказати наступу інформацію.

Конкурентоспроможність розробленого інтернет-магазину має унікальну пропозицію, яка виділяє його серед інших подібних магазинів, це товари власного виробництва, веб-сайт має привабливий вигляд та швидку взаємодію користувача з сервером.

Зручний інтерфейс сайту OrganicFood дозволяє легко переглядати товари, швидко знаходити потрібний товар і зручно замовляти замовлення користувача.

На сторінки "Про Нас" користувачі можуть з'єднатись з менеджерами сайту підтримки клієнтів для отримання додаткової інформації про продукти, допомоги з оформленням замовлення чи вирішення проблем.

Органічна продукція магазину OrganicFood є високої якості, а сертифікати і стандарти якості можуть підтверджувати її екологічність та безпеку для споживачів.

Магазин може представляти франшизу для клієнтів, що хочуть відкрити власну точку товарів, також OrganicFood ставить перед собою ціль розвитку та підтримки партнерських відносин з постачальниками органічних продуктів та іншими зацікавленими сторонами.

Інтернет магазин OrganicFood є потужним магазином серед конкурентів, та забезпечує швидке завантаження сторінок та товарів на сайті, завдяки технологіям які були застосовані при створенні веб-сайту.

Вище було наведено незначна частка плюсів ефективності такого інтернет магазину. Також не варто забувати потреби користувачів, та прислухатись до їх порад щодо покращення магазину. Завдяки введенню в роботу такого сайту, прибуток починає збільшуватись близько 30% завдяки швидкому та простому замовленню продуктів для клієнтів.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Аналіз небезпеки під час роботи за комп'ютером

Комп'ютери та інші цифрові пристрої стали настільки невід'ємною частиною нашого життя, що без них важко уявити роботу чи дозвілля. Однак їхня зручність і безмежні можливості приховують у собі підступну небезпеку: ризики для нашого здоров'я.

Занадто багато часу, проведеного перед монітором, може призвести до серйозних фізичних і психічних проблем. Неправильна постава, повторювані рухи, концентрація на екрані - все це може спричинити широкий спектр проблем.

Екрани є основним джерелом небезпеки через свої випромінювання у вигляді: рентгенівського, ультрафіолетового, інфрачервоного та електромагнітного випромінювань. Для кожного з цих видів випромінювання встановлені гранично допустимі норми.

До основних факторів, які наносять шкоду організму людини можна віднести:

Для зору характерні сухість, втома, затуманення зору – всі ці неприємні симптоми можуть стати супутниками, якщо не давати очам відпочивати від екрану.

Біль у спині через тривале сидіння чи неправильно обраній поставі можуть початись проблеми з спинними м'язами.

Ожиріння через сидячий спосіб життя, який веде більшість офісних працівників, сприяє набору зайвої ваги та ожирінню.

Оскільки робота за комп'ютером вимагає обробки великих обсягів інформації та постійної концентрації уваги, тривала робота за комп'ютером часто призводить до розумової втоми та розсіювання уваги.

Щоб уникнути проблем при роботі за комп'ютером і зменшити ризик виникнення негативних наслідків для здоров'я, можна вжити наступні заходи для збереження свого здоров'я.

Використовувати комфортне та ергономічне крісло, налаштувати монітор на правильну висоту так, щоб верхня частина екрана була на рівні очей, та скористатись ергономічною клавіатурою та мишею.

Робити короткі перерви кожні 30-60 хвилин, щоб зняти напругу з очей та м'язів, виконувати вправи для очей та розтяжки м'язів.

Сидіти прямо і не спирайтеся на руки, регулярно робити паузи для розслаблення та розтяжки.

Робити періодичні провітрювання в приміщенні, та прибирати робоче місце від пилу.

4.2 Безпека в надзвичайних ситуаціях

Загальною метою врахування безпеки в надзвичайних ситуаціях під час розробки вебсайту є захист як самого вебсайту, так і даних користувачів. Незалежно від того, чи є це потенційні кібератаки, втрата даних або інші непередбачувані події, важливо приділити належну увагу заходам безпеки для захисту як вебсайту, так і користувачів. Наприклад:

1. Захист від кібератак: Забезпечення безпеки вебсайту передбачає використання надійних методів аутентифікації, шифрування даних, захисту від вірусів та зловмисного програмного забезпечення. Важливо забезпечити постійне оновлення програмного забезпечення та регулярні аудити безпеки для виявлення потенційних загроз.

Резервне копіювання даних копій вебсайту та бази даних є важливим кроком у забезпеченні безпеки. Це допоможе уникнути втрати даних внаслідок випадкових помилок, хакерських атак або технічних проблем.

Захист особистих даних користувачі, важливо дотримуватись норм та законодавства, що стосуються захисту особистих даних користувачів. Забезпечення шифрування передачі особистої інформації, обмеження доступу до цих даних і використання безпечних методів зберігання може запобігти можливим проблемам і порушенням приватності.

Автоматичні оновлення та патчі потрібні автоматичні оновлення для вебсайту та всього використовуваного програмного забезпечення, включаючи системи управління контентом та плагіни. Це дозволить отримувати оновлення безпеки та виправлення помилок для запобігання вразливості перед новими загрозами.

Рекомендується розробити план дій для відновлення роботи вебсайту в разі надзвичайної ситуації, такої як кібератака, втрата даних або технічний збій. Це може включати наявність резервних серверів, контрольних точок відновлення та процедури відновлення даних.

4.3 Аналіз травмонебезпечних ситуацій під час виконання робіт

При аналізі травмонебезпечних ситуацій під час виконання робіт для розробки вебсайту інтернет магазину, можна переглянути наступні аспекти:

Проводиться детальний аналіз, потенційних травмонебезпечних ситуацій, які можуть статись в процесі розробки. Сюди можна включити такі пункти, як робота з апаратним забезпеченням, підключення до мережі, робота з електрикою і тому подібне.

Встановлення можливих ризиків травматичних ситуацій або загроз, пов'язаних з різними етапами розробки вебсайту. Для прикладу: ризики пов'язані з програмним забезпеченням, неправильні версії програм тощо.

Розробка стратегій та методів з метою попередження можливих ризиків травмування. Сюди входить навчання персоналу, правильне розташування мережевих та електричних кабелів і звісно використання безпечних методів роботи які значно знижують ризик травмонебезпечних ситуацій.

Розробка безпечних вказівок, цей аспект стосується визначення чітких правил безпеки, яких слід дотримуватись під час розробки вебсайту. Сюди входять регулярні перерви для відпочинку, контроль робочого місця на наявність небезпек, а також дотримання правил безпеки при роботі з технікою.

Документування та звітність стосується формулювання записів щодо будь-яких травмонебезпечних ситуацій, які виникають під час розробки. Це допомагає виявити тенденції, та прийняти відповідні заходи для запобігання подібним ситуаціям у майбутньому

ВИСНОВКИ

Веб-розробка стає важливою складовою сучасного світу, вона є потрібною для бізнесу який прагне добиватись більших успіхів для себе. Сучасні технології для розробки таких проектів набирають популярності з кожним днем, в цій кваліфікаційній роботі було використано фреймворк Django, який займає лідируючі позиції в списку світових технологій, він швидкий, простий у використанні, та легкозрозумілий для всіх, адже в мові програмування Python використовується велика кількість слів з англійської мови. Також завдяки технології проект не лише приваблює увагу користувачів дизайном чи зручним функціоналом, але й забезпечує високу швидкість завантаження веб-сайту, високий рівень безпеки як сайту так і користувача.

У даній кваліфікаційній роботі було розроблено сайт на тему продажу органічної продукції, який дає змогу користувачам здійснювати покупки продуктів онлайн, давши їм змогу додавати асортимент в корзину, міняти особисті дані у власному кабінеті та мати зворотній зв'язок з менеджерами сайту. Важливо додати, що сайт OrganicFood був розробленим з урахуванням усіх новітніх технологій та нового дизайну, з можливістю у майбутньому додати новий функціонал, що забезпечить комерційний успіх для компанії.

У першому розділі ми розглянули основні поняття у веб-розробці, зрозуміли для чого і як створюють сайти, які технології в цьому зачеплені та звернули увагу на підходи створення таких продуктів в інтернеті. Торкнулись особливостей функціонування їх, давши розуміння без чого сайт не буде працювати, та не буде приносити прибуток. Також було проведено огляд на існуючі сайти, щоб мати розуміння, що слід покращити в даному проекті, побачити помилки інших розробників, та не повторити їх на власному досвіді.

Другий розділ дав розуміння опису та моделювання систем баз даних, ми переглянули як створюються бази даних для необхідно правильної роботи додатку, яка в них схема та об'єднування між собою, переглянули конкурентів у технологіях, та знайшли саме ту, яка дасть змогу легко та швидко користуватись чи керувати сайтом.

У третьому розділі ми реалізували проект. Розробили правильну частину BackEnd та створили приємний дизайн на сайті, підключили базу даних, яка веде усі потрібні записи сайту, а саме: дані користувачів, кількість продукції, фотографії тощо. Для правильної роботи проекту було проведено оцінку ефективності, яка дала розуміння, що сайт OrganicFood є конкурентоспроможним на ринку.

Розділ під номером чотири охопив питання безпеки при роботі під час розробки сайту, чи роботи за комп'ютером в загальному. Для цього рекомендується розробити власний графік провітрювання приміщення, фізичні розвантаження для організму в цілому, та пильний нагляд за технікою і чистотою на робочому місці.

Також під час реалізації проекту, я на здобув нових знань у створенні сайтів на мові програмування Python з допомогою фреймворку Django, використання стилів за допомогою CSS, та мови розмітки сайту HTML, в результаті получивши привабливий сайт, який налаштований для зручних покупок користувачам.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Офіційна Django документація, Registration. URL: <https://docs.djangoproject.com/en/5.0/topics/auth/default/> (дата звернення 10.03.24р)
2. Етапи створення веб сайтів: які є основні кроки розробки. URL: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennya-veb-sajtiv/> (дата звернення 15.03.2024 р)
3. Коровін М., Коровіна В. Електронна комерція: технології та розробки. С.132-139.
4. Django Girls, Django ORM і QuerySets. URL: https://tutorial.djangogirls.org/uk/django_orm/ (дата звернення 18.04.2024 р)
5. Лутц М. Програмування на Python. 2011. 992 с.
6. Офіційна документація SQLite. URL: <https://www.sqlite.org/docs.html> (дата звернення 20.04.2024 р)
7. Форс Д. Django. Розробка веб-додатків на Python. URL: http://nbuv.gov.ua/UJRN/Pekon_2015_2_11 (дата звернення 23.04.2024 р)
8. Маловичко С. В. Аналіз сучасних тенденцій та динаміки розвитку електронної торгівлі на підприємствах України. *Економіка та управління народним господарством*. №2. 2015. С. 65-69.
9. Андрій Пінчук. Будівництво успішного бренду. *Академічна та університетська наука: результати та перспективи*. 2020. С.272-285.
10. Панченко М. Впровадження інтернет-магазину на Django: кращі практики та рекомендації.
11. Мельник В. Проектування та реалізація інтернет-магазину на Django.
12. Steve Krug. Don't Make Me Think: A Common Sense Approach to Web Usability. 2013. 216 р.

13. Джон Дакетт. HTML and CSS: Design and Build Websites. – URL: <http://surl.li/rvgjq> (дата звернення 28.04.2024 р)
14. Корнієнко В. Дослідження та розробка інтернет-магазину на базі фреймворку Django.
15. Jeremy Keith. HTML5 for Web Designers. Second Edition, 2015.
16. William S. Vincent. Django for Beginners. URL:<https://djangoforbeginners.com/introduction/> (дата звернення 03.05.2024 р)
17. Вдовиченко О. Електронна комерція: від ідеї до успіху.
18. Robert C. Martin .Clean Code: A Handbook of Agile Software Craftsmanship. 464р.
19. Трофименко О.Г., Козін О.Б., Задерейко О.В., Плачінда О.Є. Веб-технології та веб-дизайн : навч. посібник. Одеса: Фенікс, 2019. 284 с.
20. Інструкція з охорони праці при роботі з персональним комп'ютером. URL: <https://pro-op.com.ua/article/485-nstruktsya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter> (дата звернення 18.05.2024 р)

ДОДАТКИ

Додаток А

Код сторінки Реєстрація

```

{% extends 'product/base.html' %}
{% block content %}
<center>
  {% if form.errors %}
    {{ form.errors }}
  {% endif %}
</center>
  <div class="comment-form-wrap pt-5">
    <center><h3 class="mb-5">Organic Food | Реєстрація </h3></center>
    <form action="{% url 'register' %}" method="post" class="p-5 bg-light">
      {% csrf_token %}
      <div class="form-group">
        <label for="{{ form.username.id_for_label }}">Ім'я</label>
        {{ form.first_name }}
      </div>
      <div class="form-group">
        <label for="{{ form.last_name.id_for_label }}">Прізвище *</label>
        {{ form.last_name }}
      </div>
      <div class="form-group">
        <label for="{{ form.username.id_for_label }}">Номер Теле-
фону</label>
        {{ form.username }}
      </div>
      <div class="form-group">
        <label for="{{ form.email.id_for_label }}">Email</label>
        {{ form.email }}
      </div>
      <div class="form-group">
        <label for="{{ form.password1.id_for_label }}">Пароль</label>
        {{ form.password1 }}
      </div>
      <div class="form-group">
        <label for="{{ form.password2.id_for_label }}">Повтор па-
ролю</label>
        {{ form.password2 }}
      </div>
    </center>
  </div>

```

```
<div class="form-group">
  <input type="submit" value="Рєєстрація" class="btn py-3 px-4 btn-
primary">
</div>
</center>
<center> <p>Якщо у вас є аккаунт ви можете - <a href="{% url 'login'
%}">Увійти </a></p> </center>
</form>
</div>
{% endblock content %}
```


Додаток Б

Код файлу views.py (Users)

```
from .models import User
from .forms import UserLoginForm, RegisterUserForm, UserProfileForm
from product.models import Cart

def register(request):
    if request.method == "POST":
        form = RegisterUserForm(data=request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Вітаємо, реєстрація успішна!')
            return HttpResponseRedirect(reverse('login'))
    else:
        form = RegisterUserForm()

    context = {'form': form, 'title': 'Organic Food | Реєстрація'}
    return render(request, 'users/register.html', context)

def login(request):
    if request.method == 'POST':
        form = UserLoginForm(data=request.POST)
        if form.is_valid():
            username = request.POST['username']
            password = request.POST['password']
            user = auth.authenticate(username=username, password=password)
            if user:
                auth.login(request, user)
                return HttpResponseRedirect(reverse('home'))
    else:
        form = UserLoginForm()
    context = {'form': form, 'title': 'Organic Food | Вхід'}

    return render(request, 'users/login.html', context)

def logout(request):
    auth.logout(request)
    return HttpResponseRedirect(reverse('home'))
```

```
@login_required
def profile(request):
    form = UserProfileForm(instance=request.user)

    context = {'form': form,
               'title': 'Особистий кабінет',
               'carts': Cart.objects.filter(user=request.user),
               }

    return render(request, 'users/profile.html', context)
```

Додаток В**Код файлу admin.py**

```

from django.contrib import admin

from .models import Product, CountryProducingCategory, CompanyProducingCategory, ChoiceProductCategory, CompleteCart, Cart

class ProductAdmin(admin.ModelAdmin):
    list_display = ('product_name', 'price', 'id', 'photo', 'publication')
    list_display_links = ('id', 'product_name')
    search_fields = ('about', 'product_name', 'id')
    list_editable = ('publication',)
    list_filter = ('publication',)
    prepopulated_fields = {'slug': ('product_name', 'weight', 'price')}

class CompleteCartAdmin(admin.ModelAdmin):
    list_display = ('name_cart', 'price_cart', 'id',)

class AdminCart(admin.ModelAdmin):
    list_display = ('product', 'user')

class CompanyAdmin(admin.ModelAdmin):
    list_display = ('name', 'id')
    list_display_links = ('id', 'name')
    search_fields = ('name', 'id')
    prepopulated_fields = {"company_slug": ("name", )}

class CountryAdmin(admin.ModelAdmin):
    list_display = ('name', 'id')
    list_display_links = ('id', 'name')
    search_fields = ('name', 'id')
    prepopulated_fields = {"country_slug": ("name", )}

```

```
class ChoiceAdmin(admin.ModelAdmin):  
    list_display = ('name', 'id')  
    list_display_links = ('id', 'name')  
    search_fields = ('name', 'id')  
    prepopulated_fields = {"choice_slug": ("name", )}  
  
admin.site.register(Product, ProductAdmin)  
admin.site.register(CompleteCart, CompleteCartAdmin)  
admin.site.register(CountryProducingCategory, CountryAdmin)  
admin.site.register(CompanyProducingCategory, CompanyAdmin)  
admin.site.register(ChoiceProductCategory, ChoiceAdmin)  
admin.site.register(Cart, AdminCart)
```