

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка Web-застосунку інтернет-магазин "Anime Store" на Python з використанням Django»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

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

_____ Вадим БОГУТА
(підпис)

Виконав: здобувач вищої освіти групи ПД-43

_____ Вадим БОГУТА

Керівник: _____ Тимур ДОВЖЕНКО
к.т.н.

Рецензент: _____

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2024 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Богуті Вадиму Сергійовичу _____

1. Тема кваліфікаційної роботи: «Розробка Web-застосунку інтернет-магазин "Anime Store" на python з використанням Django»
керівник кваліфікаційної роботи к.т.н., доцент кафедри ІІЗ Тимур ДОВЖЕНКО,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2024 р. № 36.
2. Строк подання кваліфікаційної роботи «28» травня 2024 р.
3. Вихідні дані до кваліфікаційної роботи: перегляд аніме-товарів та фільтрація товарів, покласти товар у кошик, редагування профілю та редагування кошику, оформлення замовлення та спостереження за статусом замовлення
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 1. Проаналізувати вже існуючі застосунки що дозволяють купляти і продавати аніме-товари, виявити їх переваги та недоліки, скласти таблицю порівнянь.
 2. На основі проаналізованих застосунків пов'язаних з аніме-товарами розробити функціональні та нефункціональні вимоги.
 3. Розробити архітектуру та дизайн застосунку.

4. Розробити застосунок на основі обраних інструментів.
5. Провести тестування Web-застосунку та проаналізувати результати тестування.
6. Пройти апробацію роботи на науково-технічних конференціях.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.
3. Програмні засоби реалізацій.
4. Діаграма карта сайту.
5. Діаграма варіантів використання.
6. Діаграма бази даних.
7. Екранні форми.
8. Демонстрація екрану.
9. Апробація результатів дослідження.

6. Дата видачі завдання «28» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	28.02.-06.03.2024	
2	Аналіз та дослідження існуючих аналогів	07.03-13.03.2024	
3	Розробка архітектури та загальної структури системи	14.03-28.03.2024	
4	Розробка дизайну веб-додатку	29.03-07.04.2024	
5	Програмна реалізація застосунку	08.04-17.04.2024	
6	Тестування застосунку	18.04-28.04.2024	
7	Оформлення роботи: вступ, висновки, реферат	29.04-05.05.2024	
8	Розробка демонстраційних матеріалів	06.05-12.05.2024	
9	Попередній захист роботи	13.05-31.05.2024	

Здобувач вищої освіти

(підпис)

Вадим БОГУТА

Керівник

кваліфікаційної роботи

(підпис)

Тимур ДОВЖЕНКО

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 62 стор., 1 табл., 39 рис., 18 джерел.

Мета роботи – спрощення процесу покупки аніме-товарів в мережі інтернет за рахунок розробки інтернет-магазину "Anime Store". Зменшення кількості операцій користувача для вирішення задачі покупки аніме-товарів.

Об'єкт дослідження – процес покупки аніме-товарів в інтернет-магазині.

Предмет дослідження – інтернет-магазин для покупки аніме-товарів.

Короткий зміст роботи: В роботі було продемонстровано можливість використання фреймворку Django для розробки складних веб-застосунків, таких як інтернет-магазини. Django надає широкий спектр інструментів та можливостей, які значно спрощують процес розробки та дозволяють створювати якісні та масштабовані веб-застосунки.

Розроблено застосунок інтернет-магазин програмно реалізовані ключові функціональні можливості, зокрема: Реєстрація в системі, використовуючи свою адресу електронної пошти та пароль, або через соціальні мережі, створювати та редагувати свій профіль, фільтрація та пошук, можливість додавати продукти до кошика, переглядати вміст кошика та видаляти продукти з кошика, оформити замовлення, надавши свою адресу доставки та інформацію про оплату, відстежувати статус своїх замовлень. В роботі використано мову програмування Python, фреймворк Django для створення веб застосунку, PostgreSQL для бази даних та Stripe для прийому та обробки електронних платежів.

Сферою використання застосунку є ефективне розповсюдження та продаж товарів з мультиплікаційного жанрів аніме , що є унікальним для українського ринку та користується попитом.

КЛЮЧОВІ СЛОВА: Інтернет-магазин аніме, Автоматизація покупки аніме-товарів, Django, Python, PostgreSQL, Stripe

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
1 ЗАДАЧА СТВОРЕННЯ WEB-ЗАСТОСУНКУ ІНТЕРНЕТ-МАГАЗИНУ "Anime Store" НА PYTHON З ВИКОРИСТАННЯМ DJANGO	9
1.1 Постановка задачі.....	9
1.2 Переваги та недоліки інтернет-магазинів.....	10
1.3 Висновки до розділу.....	11
2 АНАЛІЗ АНАЛОГІВ ІНТЕРНЕТ-МАГАЗИНІВ З АНІМЕ-ТОВАРАМИ.....	12
2.1 Розгляд аналогів.....	13
2.2 Розгляд інтернет-магазину «Yorokobi».....	14
2.3 Розгляд інтернет-магазину «lolibuy».....	16
2.4. Розгляд інтернет магазину «ITEMSHOP».....	18
2.5 Висновки до розділу.....	21
3 ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЙ.....	22
3.1 Python.....	22
3.2 Django.....	23
3.3 PostgerSQL.....	24
3.4 Stripe.....	25
3.5 Vootstrap.....	26
3.6 Середовище розробки PyCharm.....	28
3.7 Вибір архітектурного шаблону.....	30
3.8 Мова розмітки HTML.....	31
3.9 Висновки до розділу.....	32
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ.....	33
4.1 Опис функціональності системи.....	39
4.2Опис бази даних.....	40

5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ.....	44
5.1 Системні вимоги.....	44
5.2 Робота з програмним модулем.....	44
ВИСНОВКИ.....	45
ПЕРЕЛІК ПОСИЛАНЬ.....	46
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	48
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ.....	54

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення

IDE – інтегральне середовище розробки

SQL – мова програмування для зберігання та обробки інформації в реляційній базі даних

MVC – це архітектурний шаблон, для розробки програмного забезпечення(англ. Model-View-Controller)

MVT – це архітектурний шаблон, для розробки програмного забезпечення(англ. Model-View-Template)

HTML – стандартизована мова розмітки документів для перегляду веб сторінок (англ. Hyper Text Markup Language)

CSS – мова стилю сторінок (англ. Cascading Style Sheets)

JSON – текстовий формат обміну даними на основі JavaScript

JS – мова програмування JavaScript

XML – мова розмітки, яка представляє правила для визначення будь-яких даних

ACID – набір властивостей транзакцій бази даних, покликаних гарантувати достовірність даних, незважаючи на помилки, збої живлення та інші збої

Sass – метамова на основі CSS, призначена для збільшення рівня абстракції CSS-коду

CDN – географічно розподілена мережева інфраструктура

БД – база даних

Django - фреймворк мови програмування Python

API – програмний інтерфейс додатку

ВСТУП

В останні роки в Україні спостерігається стрімке зростання популярності аніме. Цей жанр анімації, який виник у Японії, зачаровує глядачів своїми оригінальними сюжетами, емоційними персонажами та візуально вражаючими стилями. Зростання популярності аніме призвело до зростання попиту на товари та послуги, пов'язані з ним. Це, в свою чергу, створило сприятливі умови для розвитку онлайн-магазинів, що спеціалізуються на продажі аніме-продукції. Однак, багато існуючих українських онлайн-магазинів аніме не відповідають сучасним вимогам користувачів. Їхні веб-сайти та мобільні додатки часто незручні у використанні, мають застарілий дизайн та обмежений функціонал. Це робить процес пошуку та покупки аніме-товарів для користувачів складним та некомфортним.

Втілення цих ідей дозволить українським онлайн-магазинам аніме стати більш конкурентоспроможними та задовольнити зростаючі потреби шанувальників цього жанру в Україні.

Згідно з дослідженням JustWatch, у 2023 році кількість українців, які дивляться аніме, зросла на 30% у порівнянні з 2022 роком.

Netflix повідомив, що в Україні аніме-контент у 2023 році зріс на 50% у порівнянні з 2022 роком.

Метою розробки є спрощення процесу покупки аніме-товарів в мережі інтернет за рахунок розробки інтернет-магазину "Anime Store". Зменшення кількості операцій користувача для вирішення задачі покупки аніме-товарів.

Завданням роботи є:

-Проаналізувати вже існуючі застосунки що дозволяють купляти і продавати аніме-товари, виявити їх переваги та недоліки, скласти таблицю порівнянь.

-На основі проаналізованих застосунків пов'язаних з аніме-товарами розробити функціональні та нефункціональні вимоги.

-Розробити архітектуру та дизайн застосунку.

-Розробити застосунок на основі обраних інструментів.

-Провести тестування Web-застосунку та проаналізувати результати тестування.

-Пройти апробацію роботи на науково-технічних конференціях.

1 ЗАДАЧА СТВОРЕННЯ WEB-ЗАСТОСУНКУ ІНТЕРНЕТ-МАГАЗИНУ "Anime Store" НА PYTHON З ВИКОРИСТАННЯМ DJANGO

Метою даної роботи є створення веб-додатку, який автоматизує процес купівлі аніме-товарів в мережі Інтернет шляхом розробки інтернет-магазину «Anime Store».

Це зменшує кількість операцій користувача та вирішує проблему покупки аніме-товарів.

1.1 Постановка задачі

Розроблений програмний продукт, має відповідати вимогам, а саме:

- Можливість реєстрації в системі, використовуючи свою адресу електронної пошти та пароль, або через соціальні мережі.
- Можливість створювати та редагувати свій профіль.
- Можливість робити фільтрацію та пошук товарів по інтернет магазину з аніме-товарами.
- Можливість додавати продукти до кошика, переглядати вміст кошика та видаляти продукти з кошика.
- Система повинна підтримувати різні способи оплати.
- Можливість оформити замовлення, надавши свою адресу доставки та інформацію про оплату.
- Можливість відстежувати статус своїх замовлень

Веб-застосунок по інтернет магазину з аніме товарами має містити в собі наступний функціонал:

1. Перегляд товарів на сторінці каталог.
2. Можливість перейти за посиланням на конкретну сторінку.
3. Можливість переходити між вкладками по інтернет-магазину.
4. Можливість змінювати користувачем інформацію про себе у профілі користувача.
5. Можливість керувати адмін-панелью, а саме додати, видаляти та змінювати інформацію для адміністратора.

Виконання наступних задач є ключовим фактором для успішної реалізації програмного забезпечення та досягнення поставлених цілей:

1. Проаналізувати вже існуючі застосунки що дозволяють купляти і продавати аніме-товари, виявити їх переваги та недоліки, скласти таблицю порівнянь.

2. На основі проаналізованих застосунків пов'язаних з аніме-товарами розробити функціональні та нефункціональні вимоги.

3. Розробити архітектуру та дизайн застосунку.
4. Розробити застосунок на основі обраних інструментів.
5. Провести тестування Web-застосунку та проаналізувати результати тестування.
6. Пройти апробацію роботи на науково-технічних конференціях.

1.2 Переваги та недоліки інтернет-магазинів

Інтернет-магазин (електронний магазин, інтернет-магазин) - це програмний комплекс, який дозволяє продавати товари та послуги в мережі Інтернет і автоматизувати управління бізнес-процесами.

Електронні магазини поєднують елементи прямого маркетингу і традиційної роздрібною торгівлі.

Основна відмінність інтернет-магазину від звичайного – це інтерактивність, великий обсяг інформації та асортименту, а також індивідуальний підхід до кожного відвідувача.

Можна перелічити переваги інтернет-магазину найбільш очевидною перевагою електронної комерції є можливість робити покупки в Інтернеті, не виходячи з дому. Якими мотивами керуються ті, хто головною перевагою інтернет-торгівлі вважає можливість замовлення в магазині, не виходячи з дому. Цим мотивом виявилось небажання витратити час на відвідування звичайних магазинів, тобто економія часу і це ще не все.

По-перше все це гарантія отримання сертифікованої продукції дуже високої якості. По-друге, це можливість уникнути дослідницької втоми. По-третє, ви можете заощадити, оскільки ціни на більшість товарів в інтернет-магазинах зазвичай трохи нижчі, ніж зазвичай. І, нарешті, по-четверте, ви можете знайти все, що завгодно, тому що в Інтернеті можна купити майже все.

Далі поговоримо про нестабільність вибору товарів, що є недоліком інтернет-магазинів. Тому магазини, які забезпечують користувачів усіма необхідними товарами, можуть отримати конкурентну перевагу. Крім того, вибір деяких магазинів нестабільний. Тобто після вибору та покупки товару може виявитися, що потрібного товару немає в наявності, або магазин просто не виходить на зв'язок з покупцем. Реєстрація обов'язкова. Деякі магазини вимагають від клієнтів пройти тривалу покрокову реєстрацію. Це відлякує потенційних клієнтів від покупок у цьому магазині, особливо якщо клієнт цінує свій час там. Питання віртуальності. Неможливість помацати і взяти в руки куплений товар (на фото він може виглядати добре, але насправді все зовсім по-іншому).

1.3 Висновки до розділу

У даному розділі було описано мету дипломної роботи, задачі та вимоги до розробленого програмного забезпечення(ПЗ) і ще було описано необхідний функціонал котрий повинен містити себе у ПЗ та кроки для досягнення поставленої мети, а також було проаналізовано переваги та недоліки сучасних інтернет-магазинів для подальшої розробки веб-застосунку інтернет-магазину для аніме-товарів.

2. АНАЛІЗ АНАЛОГІВ ІНТЕРНЕТ-МАГАЗИНІВ З АНІМЕ-ТОВАРАМИ

Аніме, японська анімація, вже давно завоювала серця мільйонів людей по всьому світу, і Україна не стала винятком. Хоча й популярність аніме в нашій країні з'явилася пізніше, ніж в інших, вона стрімко зростає, що призвело до виникнення нових магазинів, як фізичних, так і онлайн, які пропонують широкий спектр товарів для фанатів.

Аніме стало не просто видом розваги для багатьох українців. Це також спосіб спілкування з друзями, знайомства з новою культурою та самовираження. Існує безліч аніме-клубів та спільнот в Україні, де фанати можуть збиратися, щоб ділитися своєю любов'ю до аніме.

Існує декілька факторів, які сприяли цьому:

- **Різноманіття жанрів та тем:** Аніме пропонує широкий спектр історій, які можуть зацікавити людей різного віку та уподобань. Від захоплюючих бойовиків та фентезі до зворушливих драм та комедій, кожен може знайти аніме, яке йому сподобається.
- **Яскраві та емоційні персонажі:** Персонажі аніме часто бувають добре опрацьованими та багатограними, що робить їх легко впізнаваними та емоційно близькими глядачам.
- **Унікальний стиль анімації:** Візуальний стиль аніме відрізняється від західної анімації, що робить його свіжим та цікавим для багатьох людей.
- **Доступність:** Завдяки розвитку Інтернету аніме стало легко доступним для людей з усього світу, включаючи Україну. Існує безліч сайтів та онлайн-сервісів, де можна дивитися аніме безкоштовно або за доступною ціною.

З ростом популярності аніме в Україні з'явилося й більше магазинів, де можна купити аніме-товари. Ось декілька варіантів:

- **Фізичні магазини:** У деяких великих містах України є фізичні магазини, які спеціалізуються на продажу аніме-товарів. В таких магазинах можна знайти широкий спектр товарів, включаючи мангу, DVD, Blu-ray, фігурки, одяг та багато іншого.
- **Інтернет-магазини:** В Україні є безліч інтернет-магазинів, які пропонують аніме-товари. Це зручний варіант для людей, які живуть у маленьких містах або не мають можливості відвідати фізичний магазин.
- **Соціальні мережі:** Деякі українські продавці аніме-товарів використовують соціальні мережі, такі як Facebook та Instagram, для продажу своєї продукції. Це може бути гарним варіантом для пошуку унікальних товарів або для підтримки місцевих продавців.

2.1 Розгляд аналогів

Зараз ми проведемо комплексну оцінку існуючих веб-застосунків з веб-застосунком мого виробництва котрий червоного кольору для продажу аніме-товарів, створивши порівняльну таблицю з їхніми перевагами та недоліками.(Таблиця 2.1)

Назва	Yorokobi	lolibuy	ITEMSHOP	AnimeStore
Реєстрація користувачів	-	+	-	+
Стеження за замовленнями	+	+	-	+
Система оплати	Післяоплата, оплата на рахунок, Онлайн-оплата	Післяоплата, предоплата, оплата за реквізитами	Післяплата, Оплата Liqpay	Післяоплата, Оплата Stripe
Фільтрація по категоріям	+	+	+	+
Реєстрація через соціальні мережі	-	-	-	+
Кросбраузерність	+	+	-	+

Таблиця 2.1 — Порівняльна таблиця з перевагами та недоліками

Порівняльна таблиця 3 обраних аніме-магазинів в Україні демонструє, що Yorokobi та Lolibuy мають найширший асортимент товарів, тоді як ITEMSHOP та AnimeStore пропонують більш зручні системи оплати, але загалом, всі магазини мають свої переваги та недоліки, тому вибір найкращого залежить від індивідуальних потреб та вподобань покупця.

Але все одно вибір найкращого аніме-магазину в Україні залежить від індивідуальних потреб та вподобань покупця і не лише від функціоналу а ще від товару який є на цих інтернет-магазинів з аніме-товарами.

2.2 Розгляд інтернет-магазину «Yorokobi»

На даному етапі ми більш детально розглянемо один з трьох вибраних веб-застосунків котрий було проаналізовано для таблиці порівнянь а саме «Yorokobi». Головна сторінка при вході на даний інтернет-магазин зображено на рисунку 2.1

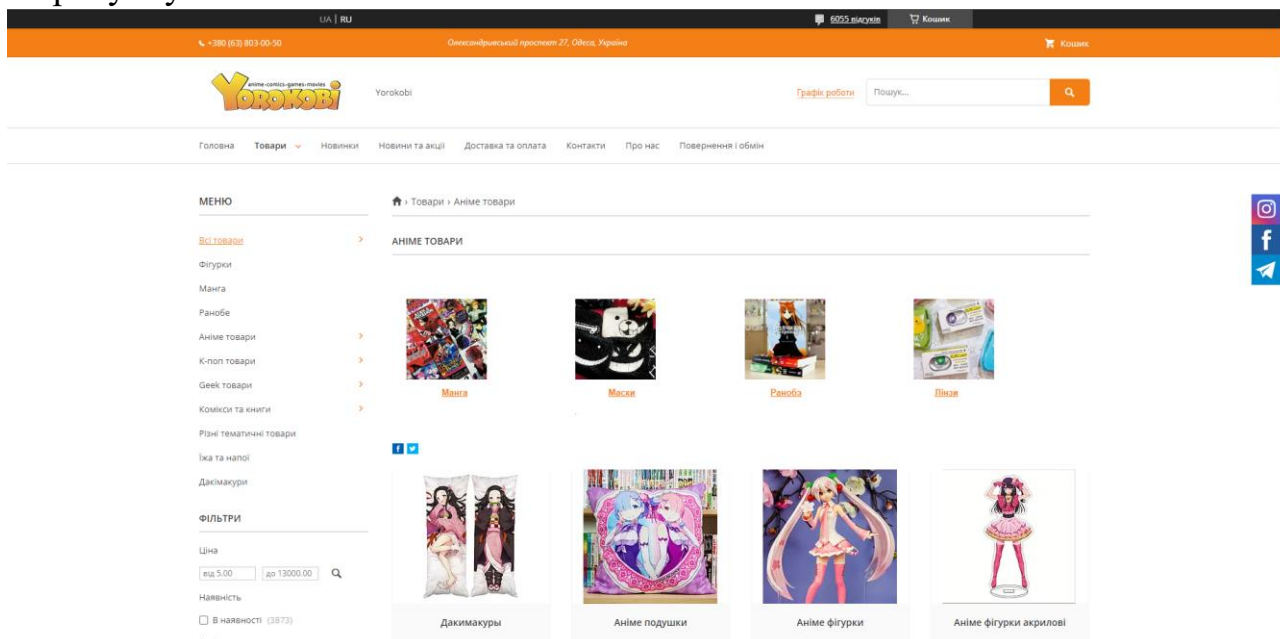


Рисунок 2.1 –Головна сторінка веб-застосунку з аніме-товарами «Yorokobi».

Yorokobi - це аніме-магазин в Одесі, який пропонує широкий вибір товарів, включаючи мангу, фігурки, одяг, аксесуари та багато іншого, а також зручний інтернет-магазин з пошуком, фільтрами, кошиком та швидкою доставкою по всій Україні, але деякі покупці відзначають незручний інтерфейс та відсутність деяких функцій, тому загалом, Yorokobi - це хороший вибір для шанувальників аніме, які шукають широкий вибір товарів та швидку доставку, але рекомендується порівняти його з іншими магазинами перед покупкою.

Деталізація:

- Широкий вибір товарів: Yorokobi пропонує один з найширших виборів аніме-товарів в Україні, включаючи мангу, фігурки, одяг, аксесуари та багато іншого.
- Зручний інтернет-магазин: Yorokobi має зручний інтернет-магазин з пошуком, фільтрами, кошиком та швидкою доставкою по всій Україні.
- Деякі недоліки: Деякі покупці відзначають незручний інтерфейс та відсутність деяких функцій, наприклад, мобільного додатку та можливості попереднього замовлення.

- Загалом, хороший вибір: Yorokobi - це хороший вибір для шанувальників аніме, які шукають широкий вибір товарів та швидку доставку, але рекомендується порівняти його з іншими магазинами перед покупкою.

Далі в нас йде вигляд оформлення замовлення у цьому інтернет магазині котре зображено на рисунку 2.2

Рисунок 2.2 – Сторінка з оформленням замовлення веб-застосунку «Yorokobi»

Сторінка оформлення замовлення в інтернет-магазині Yorokobi для аніме-товарів проста та зручна, вона містить поля для введення контактної інформації, адреси доставки, способу оплати, а також кнопку для підтвердження замовлення.

Деталізація:

- **Контактна інформація:** На сторінці є поля для введення імені, прізвища, номера телефону та електронної адреси покупця.
- **Адреса доставки:** На сторінці є поля для введення адреси доставки, включаючи країну, місто, вулицю, номер будинку та квартиру.
- **Спосіб оплати:** На сторінці є можливість вибрати один з доступних способів оплати, таких як післяплата, оплата на рахунок, онлайн-оплата.
- **Підтвердження замовлення:** Після введення всієї необхідної інформації покупець може натиснути кнопку "Оформити замовлення", щоб підтвердити своє замовлення.

Загалом, сторінка оформлення замовлення в Yorokobi проста та зручна, вона полегшує покупцям оформлення замовлення та отримання товарів.

2.3 Розгляд інтернет-магазину «lollibuy»

Тепер ми розглянемо наступний аналог веб-застосунку інтернет-магазину з аніме-товарами котрий теж було проаналізовано, а саме «lollibuy». Далі ми можемо побачити на рисунку 2.3 головну сторінку цього інтернет-магазину.



Рисунок 2.3 – Головна сторінка веб-застосунку інтернет-магазину «lollibuy»

Сторінка веб-застосунку інтернет-магазину з аніме товарами Lollibuy, що складається з секцій заголовка, основного контенту, бічної панелі та підвалу, де користувачі можуть переглядати логотип та назву магазину, меню навігації, зображення та опис товару, ціну та кнопку "Додати до кошика", посилання на інші розділи магазину, інформацію про доставку та повернення, контактну інформацію та авторські права, а також читати описи товарів, переглядати ціни, додавати товари до кошика, переходити до інших розділів магазину та отримувати інформацію про доставку та повернення, вирізняється зручним та стильним дизайном, оптимізованим для мобільних пристроїв, пропонує широкий вибір аніме товарів за доступними цінами, швидку доставку по всій Україні та зручне обслуговування клієнтів, що робить процес пошуку та купівлі аніме товарів приємним та легким.

Далі ми переглянемо кошик у даному веб-застосунку lollibuy проаналізуємо його та опишемо переваги та недоліки в ньому. На рисунку 2.4 можемо побачити кошик веб-застосунку інтернет-магазину з аніме-товарами

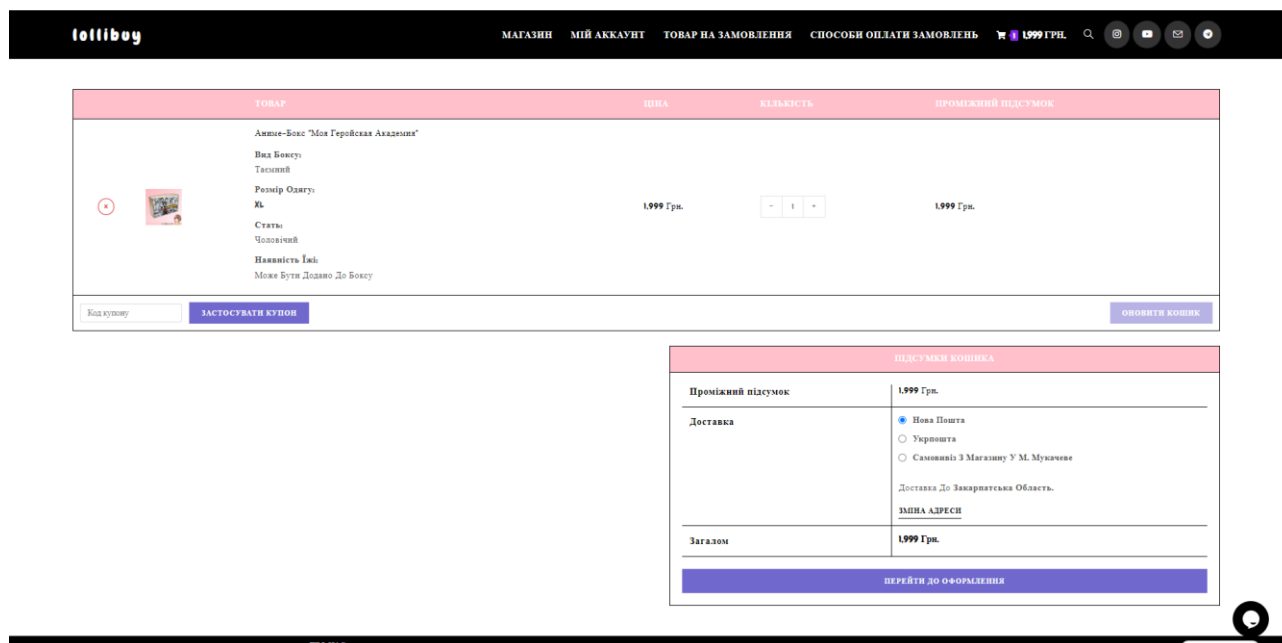


Рисунок 2.4 – Сторінка кошику у веб-застосунку інтернет-магазину «lollibuy»

Кошик Lollibuy - це простий, інформативний та зручний кошик з лаконічним дизайном, який надає всю необхідну інформацію про замовлення, включаючи вартість доставки та загальну суму, та дозволяє користувачам перейти до сторінки оформлення замовлення.

Переваги:

- **Простота:** Кошик Lollibuy простий і зрозумілий у використанні.
- **Інформативність:** Кошик надає всю необхідну інформацію про замовлення.
- **Функціональність:** Кошик дозволяє користувачам переходити до сторінки оформлення замовлення.

Недоліки:

- **Відсутність можливості збереження:** Кошик Lollibuy не дозволяє користувачам зберігати свої замовлення для подальшого використання.
- **Відсутність можливості редагування кількості:** Кошик Lollibuy не дозволяє користувачам редагувати кількість замовлених товарів.

2.4. Розгляд інтернет магазину «ITEMSHOP»

Тепер ми розглянемо та проаналізуємо останній аналог веб-застосунків інтернет-магазинів з аніме-товарами, а саме ITEMSHOP. Далі ми можемо побачити сторінку даного веб-застосунку котра з'являється при вході в цей інтернет-магазин з аніме-товарами на рисинку 2.5

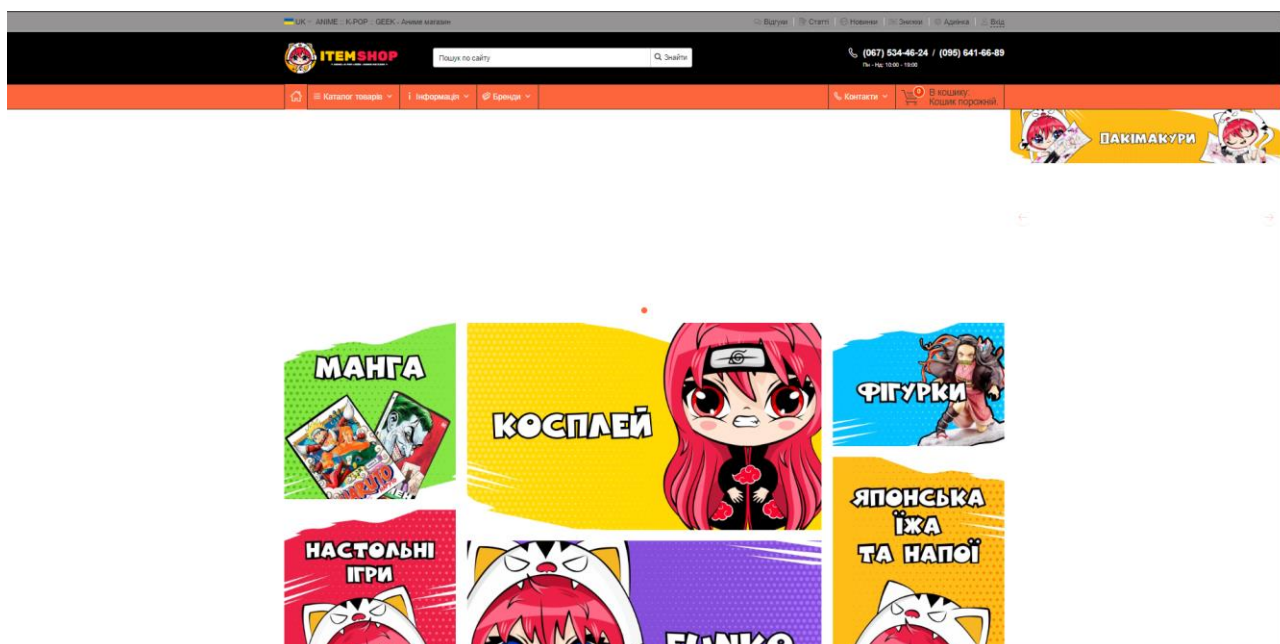


Рисунок 2.5 – Головна сторінка веб-застосунку інтернет-магазину «ITEMSHOP»

Переваги дизайну:

- **Простий та інтуїтивно зрозумілий:** інтерфейс сайту простий та інтуїтивно зрозумілий, що полегшує навігацію та пошук потрібних товарів.
- **Чистий та елегантний:** дизайн сайту чистий та елегантний, що робить його приємним для очей.
- **Високоякісні зображення:** зображення товарів на сайті високоякісні, що дозволяє чітко бачити деталі.
- **Детальні описи продуктів:** описи продуктів на сайті детальні та інформативні, що допомагає приймати обґрунтовані рішення про покупку.
- **Багато варіантів сортування та фільтрації:** користувачі можуть сортувати та фільтрувати продукти за різними критеріями, що полегшує пошук того, що вони шукають.
- **Зручний кошик покупок:** кошик покупок на сайті зручний у використанні, що полегшує оформлення замовлення.

Недоліки дизайну:

- **Відсутність багатомовності:** сайт не доступний іншими мовами, крім української, що може ускладнити його використання для іноземних клієнтів.
- **Відсутність відгуків про продукти:** на сайті немає відгуків про продукти, що може ускладнити прийняття рішення про покупку.
- **Відсутність інформації про доставку та повернення:** на сайті немає інформації про доставку та повернення, що може ускладнити оформлення замовлення.
- **Незручні банери:** на сайті є кілька банерів, які можуть бути нав'язливими та заважати користувачам навігуватися по сайту.

Веб-застосунок інтернет-магазину з аніме товарами «ITEMSHOP» є гарним місцем для покупки аніме-товарів. Дизайн сайту простий, чистий та інтуїтивно зрозумілий, а на сайті представлений широкий вибір товарів. Однак є кілька недоліків, які можна покращити, наприклад, додати багатомовність, відгуки про продукти та інформацію про доставку та повернення. Далі ми розглянемо та проаналізуємо сторінку кошика даного інтернет-магазину і його можемо побачити на рисунку 2.6

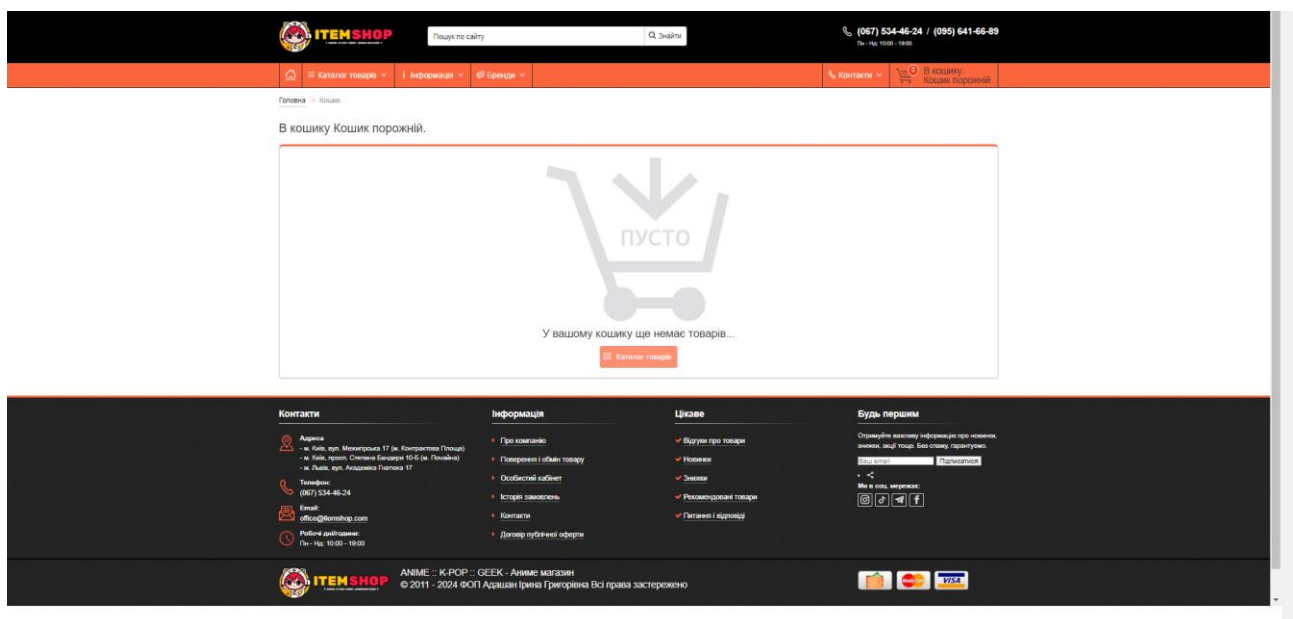


Рисунок 2.6 – Сторінка кошику у веб-застосунку інтернет-магазину «ITEMSHOP»

Сторінка кошика веб-застосунку інтернет-магазину з аніме товарами «ITEMSHOP» має простий і лаконічний дизайн. Вона поділена на два блоки:

Лівий блок містить інформацію про замовлення, включаючи список товарів, їхню кількість, ціну та загальну суму замовлення. Також тут розміщені кнопки для видалення товарів з кошика та оновлення інформації про замовлення.

Правий блок містить інформацію про доставку та оплату. Тут можна вибрати спосіб доставки та оплати, а також вказати адресу доставки.

Переваги дизайну:

- **Простий і лаконічний:** Дизайн сторінки кошика зрозумілий і простий у використанні. Користувачеві не потрібно шукати інформацію, адже вона розміщена на видному місці.
- **Інформативний:** На сторінці кошика представлена вся необхідна інформація про замовлення, включаючи список товарів, їхню кількість, ціну, загальну суму замовлення, спосіб доставки та оплати, а також адресу доставки.
- **Функціональний:** На сторінці кошика є всі необхідні кнопки та поля для оформлення замовлення. Користувач може легко додати або видалити товари з кошика, оновити інформацію про замовлення, вибрати спосіб доставки та оплати, а також вказати адресу доставки.

Недоліки дизайну:

- **Відсутність візуальних елементів:** На сторінці кошика немає візуальних елементів, таких як зображення товарів. Це може зробити сторінку менш привабливою для користувачів.
- **Незручна панель прокрутки:** У правому блоці сторінки кошика є дві панелі прокрутки. Одна з них не працює, що може ускладнити користувачам доступ до інформації про доставку та оплату.

Також на сторінці був помічен баг у правому блоці сторінки кошика є дві панелі прокрутки. Одна з них не працює, що може ускладнити користувачам доступ до інформації про доставку та оплату. Це є багом, який слід виправити.

Загалом, дизайн сторінки кошика веб-застосунку інтернет-магазину з аніме товарами «ITEMSHOP» є простим, лаконічним та інформативним. Однак він має деякі недоліки, такі як відсутність візуальних елементів та незручна панель прокрутки. Також на сторінці кошика є баг, який слід виправити.

2.5 Висновки до розділу

Yorokobi і Lollibuy пропонують найбільший асортимент товарів, а ITEMSHOP і AnimeStore пропонують більш зручну систему оплати, але в цілому всі магазини мають свої плюси і мінуси, тому вибір товару хороший в залежності від потреб і вподобань кожного покупця.

Також цьому розділі ще більш детальний аналіз зі скріншотами деяких сторінок веб-додатків інтернет-магазинів, що пропонують аніме-товари, де ми запишемо плюси і мінуси цих сторінок.

3 ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЙ

3.1 Python

Python — це динамічно інтерпретована об'єктно-орієнтована сценарна мова програмування зі строгою динамічною типізацією, був розроблений у 1990 році голландським програмістом Гвідо ван Россумом.

Python — це універсальна мова програмування, яка дозволяє писати код, який легко читається. Мова Python відносно коротка, що дозволяє писати програми, які набагато коротші, ніж програми, написані іншими мовами.

Python — це кросплатформна мова програмування. Це означає, що програми Python можуть працювати без змін у різних операційних системах. Ще однією перевагою Python є стандартна бібліотека, яка встановлюється разом з Python і містить готові інструменти для роботи з операційною системою, веб-сторінками, базами даних, різними форматами даних, створенням графічних інтерфейсів користувача тощо. Програми, написані мовою програмування Python, можуть бути невеликими сценаріями або складними системами. Одним із потенційних недоліків Python є швидкість, з якою виконується код.

Python не є скомпільованою мовою. Код Python спочатку компілюється у внутрішній байт-код, а потім виконується інтерпретатором Python. У більшості випадків використання Python зробить ваші програми повільнішими порівняно з мовами, такими як C. Однак сучасні комп'ютери мають таку обчислювальну потужність, що для більшості програм швидкість розробки важливіша за швидкість виконання, а написання програм на Python зазвичай займає багато часу. Крім того, Python можна легко розширити за допомогою модулів, написаних на C або C++. Такі модулі можуть бути використані для запуску частин програм, які інтенсивно використовують процесор.

Тепер про переваги і недоліки Python. Простота вивчення простий і лаконічний синтаксис, що робить його легко доступним для початківців. Читабельність, код читабельний і зрозумілий, що робить його зручним для співпраці та обслуговування. Універсальність, можна використовувати для широкого кола завдань, від розробки веб-сайтів до машинного навчання. Велике співтовариство, має велике та активне співтовариство користувачів і розробників, що означає, що ви можете легко знайти допомогу та ресурси. Багато бібліотек, існує безліч бібліотек та фреймворків для Python, які можуть допомогти вам виконати практично будь-яке завдання.

Недоліки мови програмування Python, а саме швидкість, може бути повільнішим за деякі інші мови програмування, такі як C або C++. Є інтерпретованою мовою, що означає, що код спочатку перекладається в інструкції, які може зрозуміти комп'ютер, перш ніж його можна виконати. Це може зробити Python трохи повільнішим за компільовані мови, такі як C++. Має динамічну систему типізації, що означає, що типи змінних не перевіряються під час компіляції. Це може призвести до помилок під час виконання.

Загалом, Python є потужною та універсальною мовою програмування, яка підходить для широкого кола завдань. Її простота вивчення, читабельність та велике співтовариство користувачів роблять її чудовим вибором як для початківців, так і для досвідчених розробників.

3.2 Django

Django — це високорівневий веб-фреймворк Python, який дозволяє швидко створювати безпечні веб-сайти, які зручно підтримувати. Створений досвідченими розробниками, Django усуває більшість проблем із веб-розробкою, дозволяючи вам зосередитися на створенні веб-додатків, не винаходячи велосипеда. Він безкоштовний із відкритим вихідним кодом, із зростаючою та активною спільнотою, чудовою документацією та багатьма варіантами безкоштовної та платної підтримки.

Django допоможе вам створити повне програмне забезпечення. Django дотримується філософії «все включено», забезпечуючи майже все, що потрібно розробнику з коробки. Усе, що вам потрібно, є частиною одного «продукту», тому все бездоганно працює разом, дотримується послідовних принципів проектування та забезпечує вичерпну актуальну документацію. Універсальний За допомогою Django ви можете (і створювали) майже будь-який тип веб-сайту, від систем керування контентом і вікі до соціальних мереж і сайтів новин. Він працює в будь-якому клієнтському середовищі та може надавати вміст майже в будь-якому форматі (HTML, RSS-канали, JSON, XML тощо). Сторінка, яку ви зараз читаете, створена за допомогою Django!

Django надає рішення майже для будь-якої необхідної вам функції (наприклад, для деяких популярних баз даних, механізмів шаблонів тощо), але ви також можете розширити його внутрішньо за допомогою компонентів сторонніх розробників. Захищений Django допомагає розробникам уникнути багатьох поширених пасток безпеки, надаючи структуру, призначену для автоматичного захисту вашого сайту. Наприклад, Django забезпечує безпечний спосіб керування обліковими записами та паролями користувачів, зберігаючи інформацію про сесії в уразливих файлах cookie (натомість файли cookie

містять лише ключ, а фактичні дані зберігаються в базі даних. Уникайте поширених помилок, таких як: Використовуйте прямі паролі замість хешів паролів. Хеш пароля — це значення фіксованої довжини, створене шляхом обробки пароля за допомогою криптографічної хеш-функції.

Django може перевірити правильність введеного вами пароля, запустивши хеш-функцію та порівнявши результат із збереженим хеш-значенням. Односторонній характер цієї функції ускладнює зловмиснику визначення оригінального пароля, навіть якщо збережене хеш-значення скомпрометовано. Django за замовчуванням забезпечує захист від багатьох вразливостей, включаючи впровадження SQL, міжсайтовий скриптинг, підробку міжсайтового запиту та клікджекінг. Масштабований Django використовує компонентну архітектуру "спільного використання нічого" (кожна частина є незалежною від інших, тому її можна замінити або змінити за потреби). Чітко розділяючи частини, Django може масштабуватися зі збільшенням трафіку, додаючи апаратне забезпечення на кожному рівні (кеш-сервер, сервер бази даних, сервер додатків). Деякі з найбільш відвідуваних веб-сайтів успішно розширили Django (наприклад, Instagram і Disqus, якщо назвати лише два). Підтримуваний Код Django написаний з використанням принципів і шаблонів проектування, які сприяють створенню підтримуваного та багаторазового коду. Зокрема, він використовує принцип «Не відповідай собі» (DRY), який усуває непотрібне дублювання та зменшує кількість коду.

Django також рекомендує групувати пов'язані функції в повторно використовувані «додатки» і, на нижчому рівні, групувати пов'язаний код у модулі (дотримуючись шаблону Model View Controller (MVC)). Portable Django написано мовою Python і працює на багатьох платформах. Це означає, що ви не прив'язані до певної серверної платформи та можете запускати свої програми на багатьох версіях Linux, Windows і Mac OS.

3.3 PostgerSQL

Чому це безкоштовна система керування базами даних (СУБД). Це одна з найпопулярніших СУБД, тому нею користуються тисячі організацій по всьому світу. Система базується на некомерційній системі керування базами даних Postgres, яка розпочалася як проект із відкритим кодом в Каліфорнійському університеті в Берклі. Він був започаткований у 1986 році. Майклом Стоунбрейкером, директором проекту Engres, і називався «Post Ingres». Згодом він поширився за межі університетів і став відомий як PostgreSQL, і продовжує розвиватися та вдосконалюватися сьогодні. PostgreSQL базується на мові SQL (ви можете вивчити SQL у дії за допомогою курсу FoxmindED QA Automation).

PostgreSQL належить до такого типу СУБД: об'єктно-реляційна (тобто поєднує в собі переваги реляційних баз даних і об'єктно-орієнтованого програмування). Основні функції PostgreSQL підтримує низку функцій, які роблять його потужною та гнучкою СУБД. Ось деякі з основних функцій: PostgreSQL забезпечує транзакції з властивостями ACID для забезпечення цілісності даних. Підтримує автоматичне оновлення перегляду, що дозволяє користувачам переглядати дані в базі даних у реальному часі. Підтримує матеріалізовані подання, які зберігають копію даних подання в базі даних. Підтримує тригери, які можуть виконувати дії, коли дані в таблиці змінюються. Підтримує зовнішні ключі для створення зв'язків між таблицями. Підтримує збережені процедури, які виконують складні обчислення на стороні сервера.

Застосування в різних сферах Бази даних PostgreSQL успішно використовуються в різних галузях. У веб-розробці він служить основою для динамічних веб-сайтів і програм. Що стосується аналізу даних, то він має високу продуктивність при обробці великих обсягів інформації. Масштабні програми, такі як соціальні мережі, покладаються на масштабованість і надійність цієї системи. Наприклад, у веб-розробці (наприклад, програми електронної комерції) PostgreSQL може ефективно обробляти мільйони транзакцій, записувати замовлення, зберігати профілі клієнтів, керувати запасами та забезпечувати стабільність і масштабованість.

Аналітика даних (додатки аналізу маркетингових даних) – зберігайте та обробляйте дані клієнтів, ефективно запускайте складні запити для аналізу купівельної поведінки та забезпечуйте надійну основу для прийняття стратегічних рішень. Він також служить основою для великомасштабних програм (соціальних мереж), де мільйони користувачів можуть взаємодіяти, ділитися вмістом і отримувати персоналізовані рекомендації.

3.4 Stripe

Stripe - це популярна платіжна система, що використовується для обробки онлайн-платежів. Її використовують як малі підприємства, так і великі корпорації. Stripe пропонує широкий спектр функцій, які роблять її привабливим вибором для багатьох підприємців.

Переваги Stripe:

- **Простота використання:** Stripe має простий та інтуїтивно зрозумілий інтерфейс, що робить його легким для використання навіть для початківців.
- **Гнучкість:** Stripe підтримує широкий спектр платіжних методів, включаючи кредитні картки, дебетові картки, банківські перекази та електронні гаманці.

- **Безпека:** Stripe використовує передові технології безпеки для захисту транзакцій та інформації клієнтів.
- **Масштабованість:** Stripe може масштабуватися разом з вашим бізнесом, тому вам не доведеться турбуватися про те, що система не зможе впоратися з зростанням вашого обсягу продажів.
- **Глобальний охоплення:** Stripe доступний у багатьох країнах світу, що робить його чудовим вибором для підприємств, які продають міжнародно.
- **Підтримка:** Stripe пропонує чудову підтримку клієнтів, включаючи цілодобову телефонну лінію, електронну пошту та чат.

Недоліки Stripe:

- **Вартість:** Stripe стягує комісію за кожну транзакцію. Ці комісії можуть бути значними для підприємств, які обробляють багато транзакцій.
- **Відсутність деяких функцій:** Stripe не пропонує деякі функції, які є доступними в інших платіжних системах, наприклад, можливість виставляти рахунки та створювати підписки.
- **Складність налаштування:** Налаштування Stripe може бути складним для деяких підприємців.

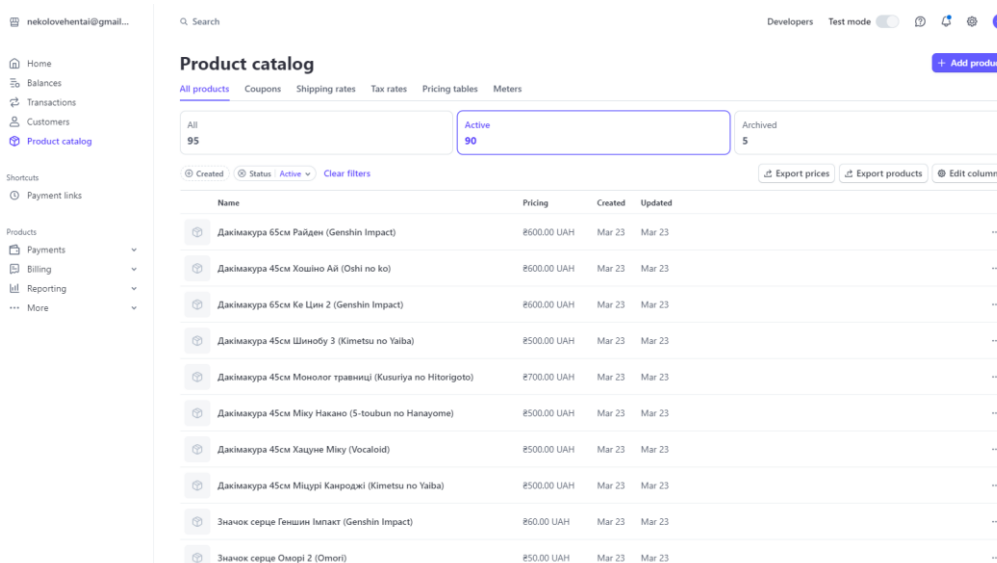


Рисунок 3.1 – використання Stripe для інтернет-магазину

3.5 Bootstrap

Bootstrap — це відкритий і безкоштовний фреймворк HTML, CSS і JS для швидкого створення адаптивного дизайну веб-сайтів і веб-додатків. Найчастіше використовується для розробки інтерфейсів веб-сайтів та панелі адміністратора. Цей фреймворк значно полегшує розробку веб-сайтів за допомогою шаблонів дизайну на основі HTML і CSS для типографіки, форм,

кнопок, таблиць, навігації, плагінів JavaScript тощо. Все це дозволяє легко створювати адаптивний дизайн. Bootstrap є відкритим кодом і доступний на Github. Ліцензія MIT. Це означає, що його можна безкоштовно використовувати як у відкритих, так і в комерційних програмах. Bootstrap дуже популярний серед розробників. Це тому, що ви можете створювати веб-сайти у багато разів швидше, ніж за допомогою чистого CSS або JavaScript. Це добре розроблений інтерфейсний фреймворк, який можна легко налаштувати шляхом редагування змінних Sass і використання міксинів. Навіть новачки (без глибоких знань і достатньої практики) можуть швидко створити дуже якісні макети. Фреймворк Bootstrap складається з набору файлів CSS і JavaScript. Щоб використовувати ці файли, просто підключіть їх до свого сайту або ввімкніть подібні функції у своєму CDN (мережі доставки вмісту).

Переваги цього фреймворку це - простота використання, будь-хто з базовими знаннями HTML і CSS може почати використовувати його. Швидше створюйте високоякісні адаптивні макети. Можливість налаштування відповідно до вашого проекту. Усі компоненти створені в одному стилі. Багатофункціональний: Adaptive CSS адаптується до телефонів, планшетів і настільних комп'ютерів. Сумісний з усіма сучасними браузерами (Chrome, Firefox, Edge, Safari, Opera). Ваш веб-сайт відобразатиметься та функціонуватиме належним чином у всіх браузерах та операційних системах, які підтримуються цим фреймворком.

І мінус цього фреймворку Bootstrap це - кінцеві файли CSS і JS вашого проекту будуть великими. Складність використання Bootstrap для створення веб-сайту з власним дизайном. Bootstrap складається з: Інструменти компоновання. Базовий дизайн контенту: Текст, зображення, код, таблиці, діаграми. Готові компоненти: кнопки, форми, горизонтальні та вертикальні панелі навігації, повзунки, списки, гармошки, модальні вікна, підказки тощо. Допоміжні класи для вирішення традиційних завдань, з якими найчастіше стикаються веб-розробники: вирівнювання тексту, показ і приховання елементів, завдання кольору, фону, полів і відступів тощо. Якщо ви вирішили використовувати Bootstrap у своєму веб-проекті, вам спочатку потрібно зрозуміти, як інтегрувати Bootstrap. Існує кілька способів зробити це, і ми розглянемо три основні з них:

Використання CDN (мережі доставки вмісту): Це фактично те, як ви «орендуєте» Bootstrap. Просто додайте посилання на стилі та сценарії Bootstrap з Інтернету до свого HTML. Це просто, зручно та отримує ресурси з віддалених серверів, завдяки чому ваші проекти завантажуються швидше.

Встановлення за допомогою менеджера пакетів: Якщо у вас є досвід роботи з npm (менеджер пакетів Node), RubyGems або Composer (інструмент для керування залежностями в проекті), ви можете використовувати їх для

встановлення Bootstrap. Просто скажіть їм, що вам потрібен Bootstrap, і ці інструменти завантажуть і встановлять його для вас.

Завантажте вихідний код : якщо ви хочете повністю контролювати код, ви можете завантажити вихідний код фреймворку з офіційного веб-сайту та реалізувати його у своєму проєкті. Це дає вам повну свободу налаштувати та змінювати код за потреби.

Загалом це потужний інструмент, фреймворк, яким одразу можуть скористатися багато людей. Окрім компонентів і стилів, ви також можете легко налаштувати його відповідно до своїх потреб. Якщо ви тільки знайомитеся з Boots Trap, не зупиняйтеся на цьому; продовжуйте читати документацію та спробуйте розробити власні проєкти.

3.6 Середовище розробки PyCharm

PyCharm: Потужне середовище розробки для Python PyCharm — це інтегроване середовище розробки (IDE), спеціально розроблене для мови програмування Python. Він був розроблений компанією JetBrains, яка також відома розробкою інших популярних IDE, таких як IntelliJ IDEA. PyCharm доступний у двох основних версіях:

Community Edition: Безкоштовна версія з відкритим кодом, яка пропонує широкий спектр функцій для розробки на Python.

Professional Edition: Платна версія, яка включає додаткові функції, такі як підтримка веб-розробки за допомогою Django та Flask, наукові обчислення за допомогою Anaconda та професійні інструменти аналізу коду.

Основні функції PyCharm: Редактор коду з підсвічуванням синтаксису, автозаповненням і перевіркою коду: PyCharm надає простий у використанні редактор коду, який полегшує написання та читання коду Python. Він включає підсвічування синтаксису, автозавершення коду, перевірку помилок і попередження, які допоможуть вам писати чистий і правильний код. Вбудований налагоджувач: PyCharm має потужний графічний налагоджувач, який дозволяє вам покроково переглядати ваш код, встановлювати контрольні точки та переглядати значення змінних і вирази. Це полегшує пошук і виправлення помилок у вашому коді. Інструменти аналізу коду: PyCharm включає різноманітні інструменти аналізу коду, які допоможуть вам визначити потенційні проблеми у вашому коді, зокрема:

- Неєфективна конфігурація, дублювання коду та можливі помилки, це покращує якість вашого коду та полегшує його читання.

- Підтримка наукових обчислень : PyCharm Professional Edition забезпечує вбудовану підтримку наукових обчислень за допомогою Anaconda, це спрощує імпорт і використання наукових обчислювальних бібліотек, таких як NumPy, Pandas і Matplotlib, безпосередньо в PyCharm.
- Підтримка веб-розробки: PyCharm Professional Edition також забезпечує підтримку веб-розробки Django та Flask, це включає такі функції, як автозаповнення HTML і CSS, візуальний редактор шаблонів Django та інтеграцію веб-сервера.
- Контроль версій: PyCharm інтегрується з популярними системами контролю версій, такими як Git і Mercurial, це дозволяє легко відстежувати зміни у вашому коді, співпрацювати з іншими розробниками та повернутися до попередніх версій коду.
- Розширюваність: PyCharm можна розширити за допомогою плагінів, які додають нові функції.

PyCharm має багато безкоштовних і платних плагінів, які дозволяють налаштувати IDE відповідно до ваших потреб. Переваги використання PyCharm:

- **Покращення якості коду:** інструменти аналізу коду та інші функції PyCharm допомагають писати чистий, правильний і читабельний код.
- **Легке налагодження:** потужний налагоджувач PyCharm робить налагодження вашого коду Python швидшим і простішим.
- **Підтримує різні стилі розробки:** PyCharm можна налаштувати відповідно до ваших уподобань і стилю розробки.
- **Велика спільнота:** PyCharm має велику та активну спільноту користувачів, що дозволяє легко знайти допомогу та підтримку, коли вона вам потрібна.

PyCharm – хороший вибір для розробників Python усіх рівнів. У безкоштовній версії Community Edition є все, що вам потрібно для початку роботи, тоді як професійна версія пропонує додаткові функції, корисні для досвідчених розробників. На рисунку 3.1 зображено інтерфейс середовища розробки PyCharm

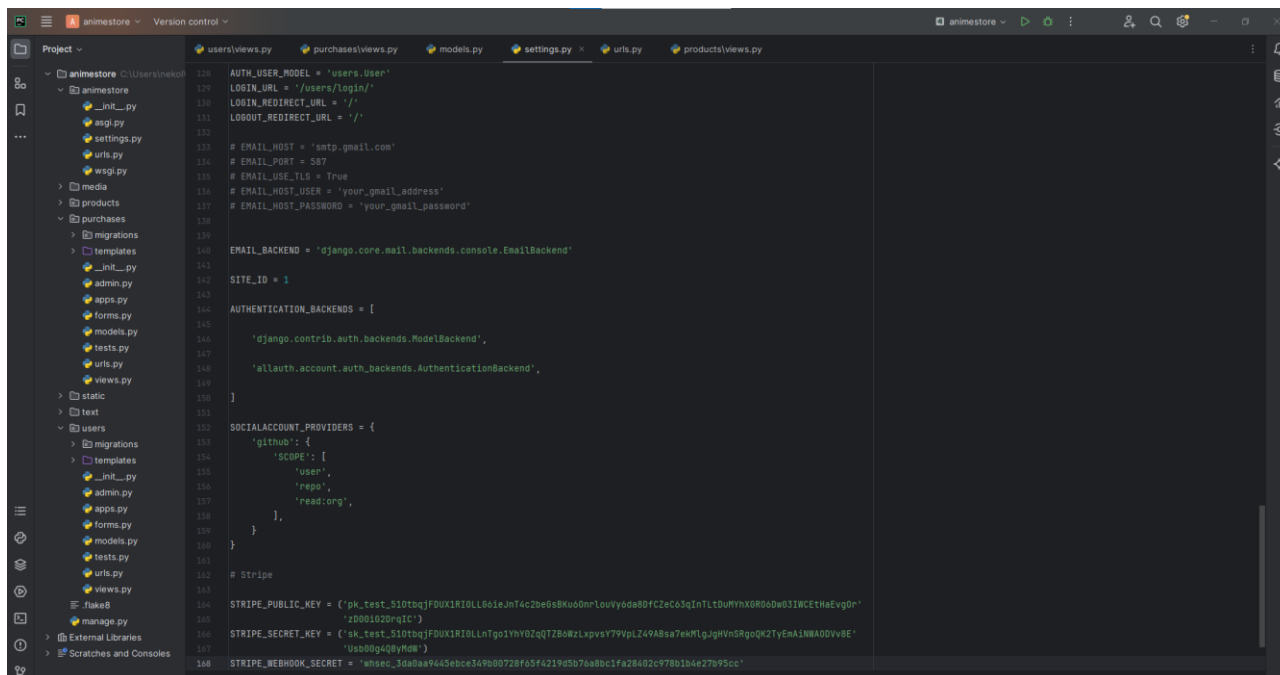


Рисунок 3.2 – середовища розробки PyCharm

3.7 Вибір архітектурного шаблону

Дослідивши велику кількість шаблонів, було обрано шаблон проектування MVT. (Рис.3.2) MVT(Model-View-Template) - це шаблон проектування програмного забезпечення, який використовується для розробки веб-застосунків. Він ґрунтується на шаблоні MVC (Model-View-Controller), але з деякими ключовими відмінностями. MVT складається з трьох компонентів:

Модель (Model): Представляє дані вашого веб-застосунку. Містить бізнес-логіку, пов'язану з даними, та відповідає за їх зберігання, маніпулювання та отримання. Зазвичай реалізується за допомогою класів Python та баз даних.

Представлення (View): Відповідає за візуальний інтерфейс користувача та взаємодію з ним. Отримує запити від користувачів, обробляє дані з моделі та генерує HTML-відповіді. Зазвичай реалізується за допомогою функцій Python, які повертають шаблони HTML.

Шаблон (Template): Визначає структуру та макет HTML-сторінок веб-застосунку. Містить змінні шаблону, які заповнюються даними з представлення. Зазвичай пишуться мовою шаблонів Django, яка є розширенням HTML.

Model View Template

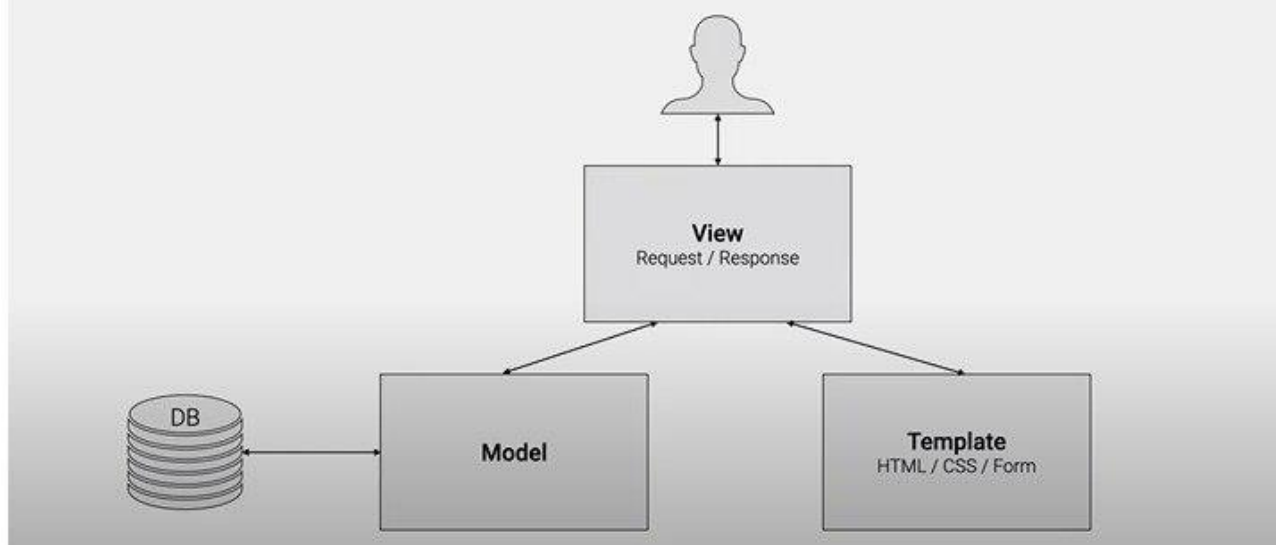


Рисунок 3.2

Загалом MVT –це потужний і гнучкий шаблон проектування, який підходить для розробки швидких, простих і економічно ефективних веб-додатків. Це особливо корисно для проектів, де важливі швидкість розробки та простота коду.

3.8 Мова розмітки HTML

HTML – це стандартизована мова розмітки документів для відображення веб-сторінок у браузері. Веб-браузер отримує HTML-документ із сервера через протокол HTTP/HTTPS або відкриває HTML-документ із локального жорсткого диска та інтерпретує код для створення інтерфейсу, який відображається на екрані монітора. Елементи HTML є будівельними блоками сторінок HTML. Структури HTML дозволяють вставляти зображення та інші об'єкти (наприклад, інтерактивні форми) у відтворені сторінки. HTML надає можливість створювати структуровані документи, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML розділені тегами, укладеними в кутові дужки. Браузери не відображають теги HTML, але використовують їх для інтерпретації вмісту сторінки. HTML може містити вбудовані програми, написані мовами сценаріїв, такими як JavaScript, які впливають на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і макет вашого вмісту. Консорціум Всесвітньої павутини (W3C), який контролює стандарти HTML і CSS, з 1997 року заохочує використання CSS замість явного представлення HTML. HTML представляє такі інструменти:

- Створюйте структуровані документи, позначаючи структурну організацію тексту: заголовки, абзаци, списки, таблиці, цитати тощо.
- Отримувати інформацію з всесвітньої павутини за допомогою гіперпосилань.
- Створення інтерактивних форм.
- Вбудовування зображень, аудіо, відео та інших об'єктів у текст

HTML використовується для створення різноманітних веб-сторінок, зокрема:

- **Статичні веб-сайти:** Це веб-сайти з фіксованим вмістом, який змінюється, а не динамічний.
- **Динамічні веб-сайти:** Ці веб-сайти використовують JavaScript та інші технології для створення динамічного вмісту, який може змінюватися залежно від дій користувача.
- **Веб-програми:** Це веб-сайти, які надають користувачам багато специфічних функцій, подібних до тих, що є в настільних програмах.
- **Мобільний веб-сайт:** Веб-сайт, оптимізований для перегляду на мобільних пристроях, таких як смартфони та планшети.

3.9 Висновки до розділу

Для реалізації веб-застосунку було використано широкий спектр програмних засобів , а саме: мова програмування Python, фреймворк Django, база даних PostgreSQL, платіжна система Stripe, фреймворк Bootstrap, мова розмітки HTML, середовище розробки PyCharm та архітектурний шаблон MVT(Model-View-Template) можемо побачити їх логотипи на рисунку 3.3. Завдяки використанню цих програмних засобів було створено веб-застосунок інтернет-магазин для аніме-товарів, який є модульним, масштабованим і відповідає вимогам.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Відкрийте для себе безмежний світ аніме-товарів за допомогою програмного продукту, розробленого спеціально для шанувальників аніме. Цей продукт дозволить вам легко знаходити, купувати та замовляти омріяні аніме-товари, не виходячи з дому, щоб розпочати потрібно зайти в веб-застосунок інтернет-магазин де ми потрапимо на головну сторінку котра показана на рисунку 4.1 і звідти ми можемо перейти у сам каталог з аніме товарами або авторизуватись або зареєструватись

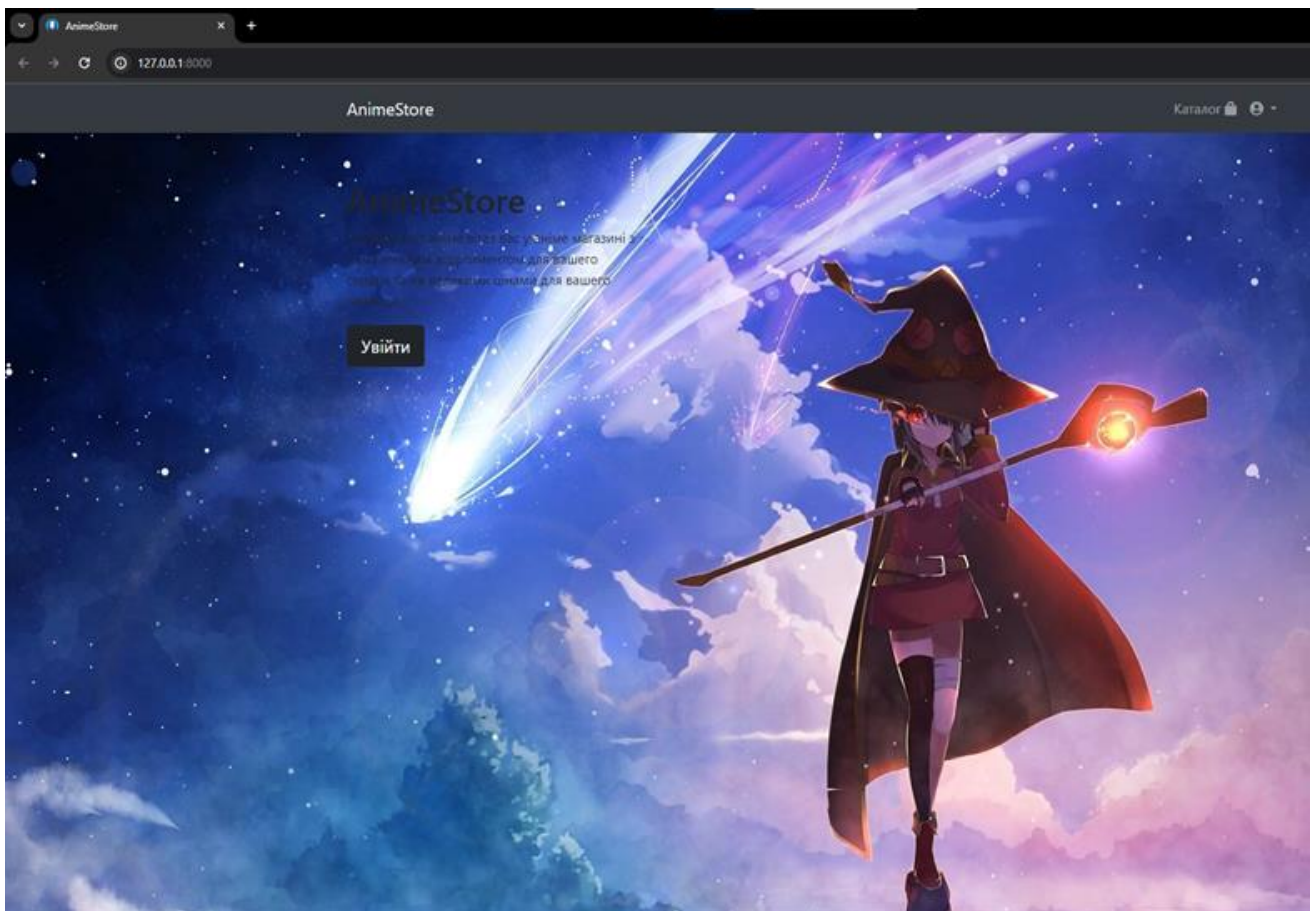


Рисунок 4.1 – Головна сторінка при вході у інтернет-магазин

Після того, як ви потрапите на головну сторінку нашого інтернет-магазину, вам відкриється доступ до безмежного світу аніме-товарів. Щоб розпочати свою подорож, просто натисніть кнопку "Увійти" у верхній частині екрану.

Перед вами постане розкішний каталог, де ви знайдете все, що душа забажає: від дакімакури та аксесуарів до фігурок, костюмів для косплею, патріотичної атрибутики та манги. Зручні категорії допоможуть вам швидко знайти те, що вас цікавить, а завдяки функції пагінації ви зможете переглянути всі доступні товари, не обмежуючись однією сторінкою. На сторінці можна

побачити категорії, пагінацію, фільтрація, додати до кошика товар але це тільки для авторизованих користувачів, на рисунку 4.2 ми можемо побачити сторінку каталогу з аніме-товарами, категоріями і всім іншим

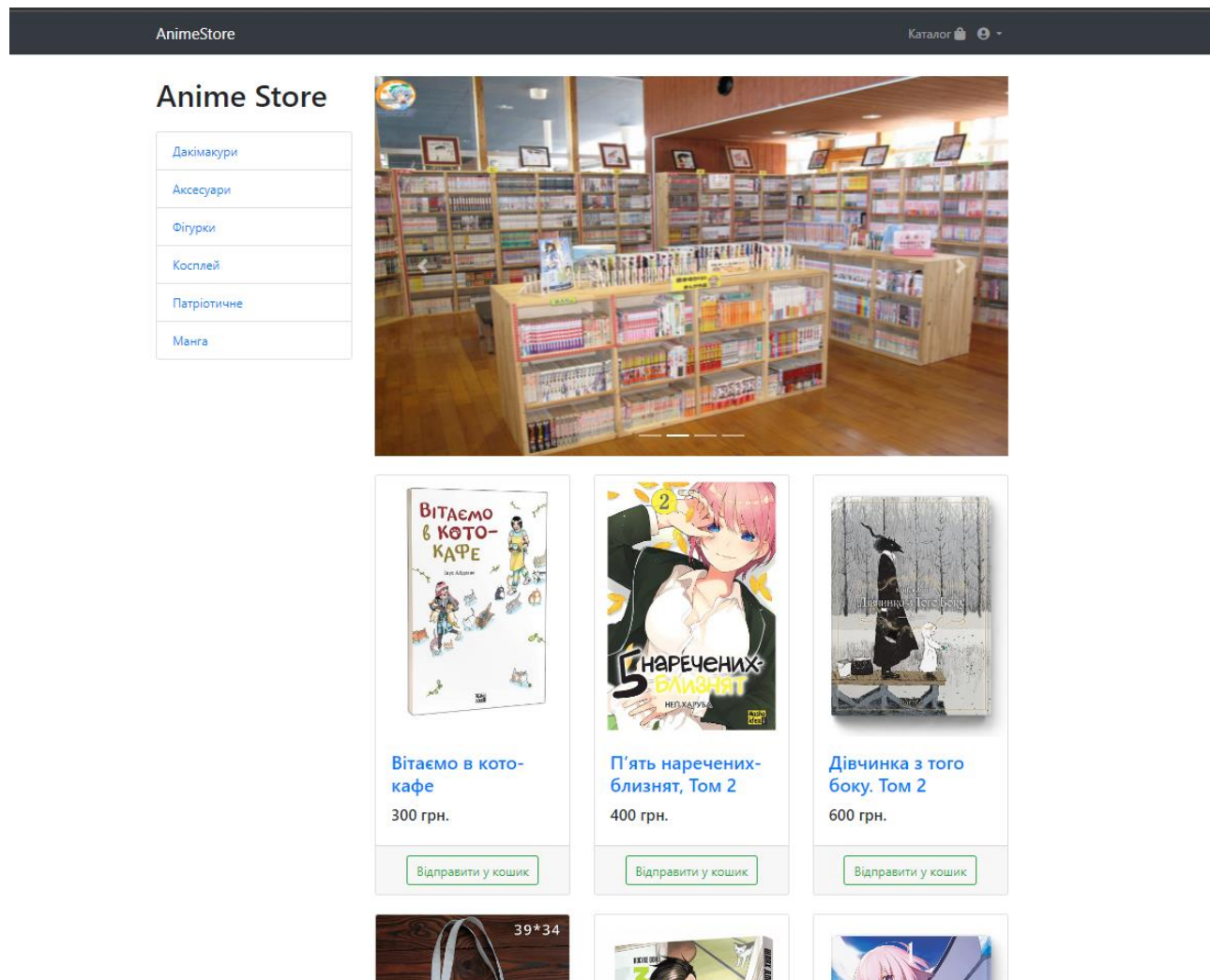


Рисунок 4.2 – Сторінка каталогу

Після того, як ви авторизувалися або зареєструвалися на рисунку 4.4 зображена панель авторизації де ми можемо авторизуватись або зареєструватись завдяки вводу електронної пошти або через соціальні мережі, створивши свій обліковий запис, ви можете вибрати товар, який вам найбільше сподобався: Прогуляйтеся по нашому широкому асортименту аніме-товарів, від фігурок та одягу до манги. Використовуйте зручні фільтри та категорії, щоб швидко знайти те, що вас цікавить, щоб отримати точні результати. Після цього перейти до профілю у лівій стороні екрана ви знайдете свій профіль. Тут ви можете:

Редагувати інформацію про себе: Змініть фотографію профілю, ім'я і прізвище але не можна змінювати ім'я користувача та адрес електронної пошти

Переглянути історію замовлень: Перевірте, які товари ви замовляли раніше, та відстежуйте статус своїх поточних замовлень.

Перевірити кошик: Праворуч від профілю ви знайдете кошик з обраними вами аніме-товарами.

Під час перевірки кошику тут ви можете:

Переглянути список товарів: Переконайтеся, що всі товари, які ви хочете купити, додано до кошика.

Змінити кількість: Змініть кількість кожного товару у вашому кошику.

Видалити товар: Видаліть будь-який товар з кошика, якщо ви передумали його купувати.

Перейти до оформлення замовлення: Коли ви будете готові до покупки, натисніть кнопку "Оформити замовлення", щоб завершити процес.

Переглянути загальну ціну: Нижче кошика ви знайдете загальну ціну за всі обрані вами аніме-товари.

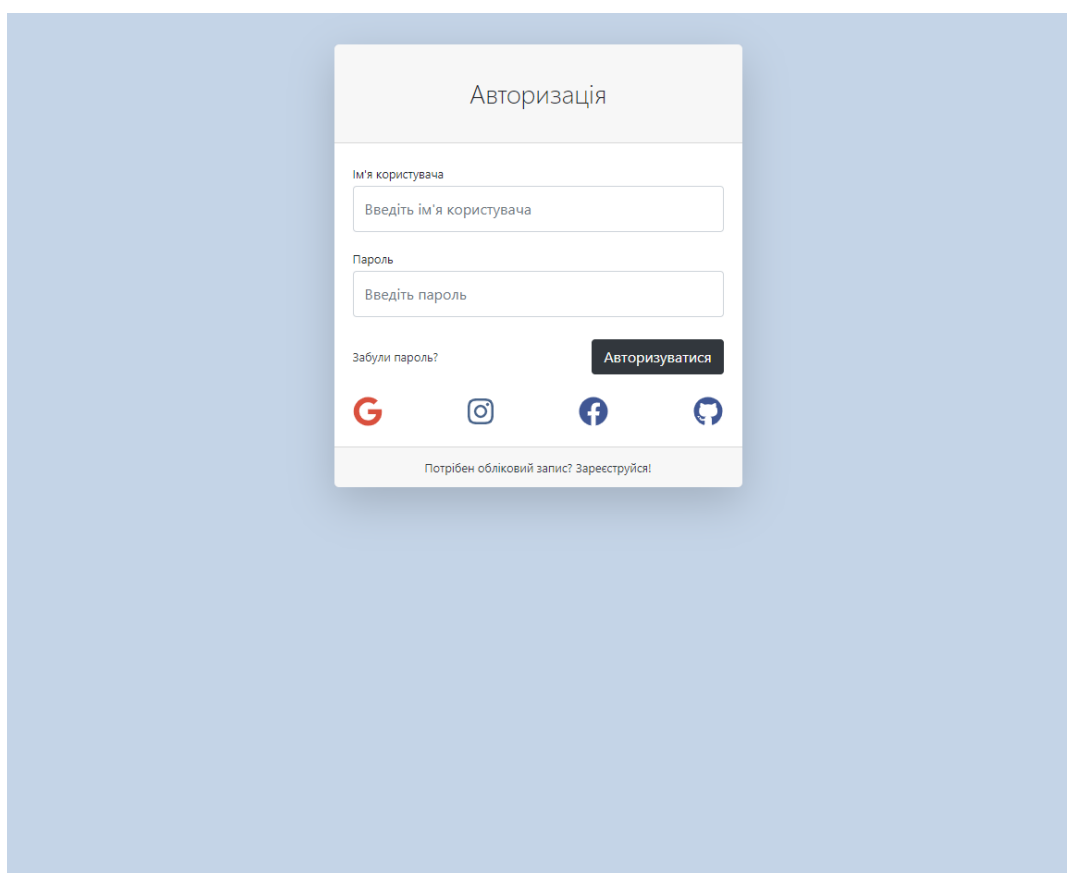


Рисунок 4.4 – панель авторизації

Сторінку профілю ми можемо побачити на рисунку 4.3 де буде профіль з обраним товаром у кошику

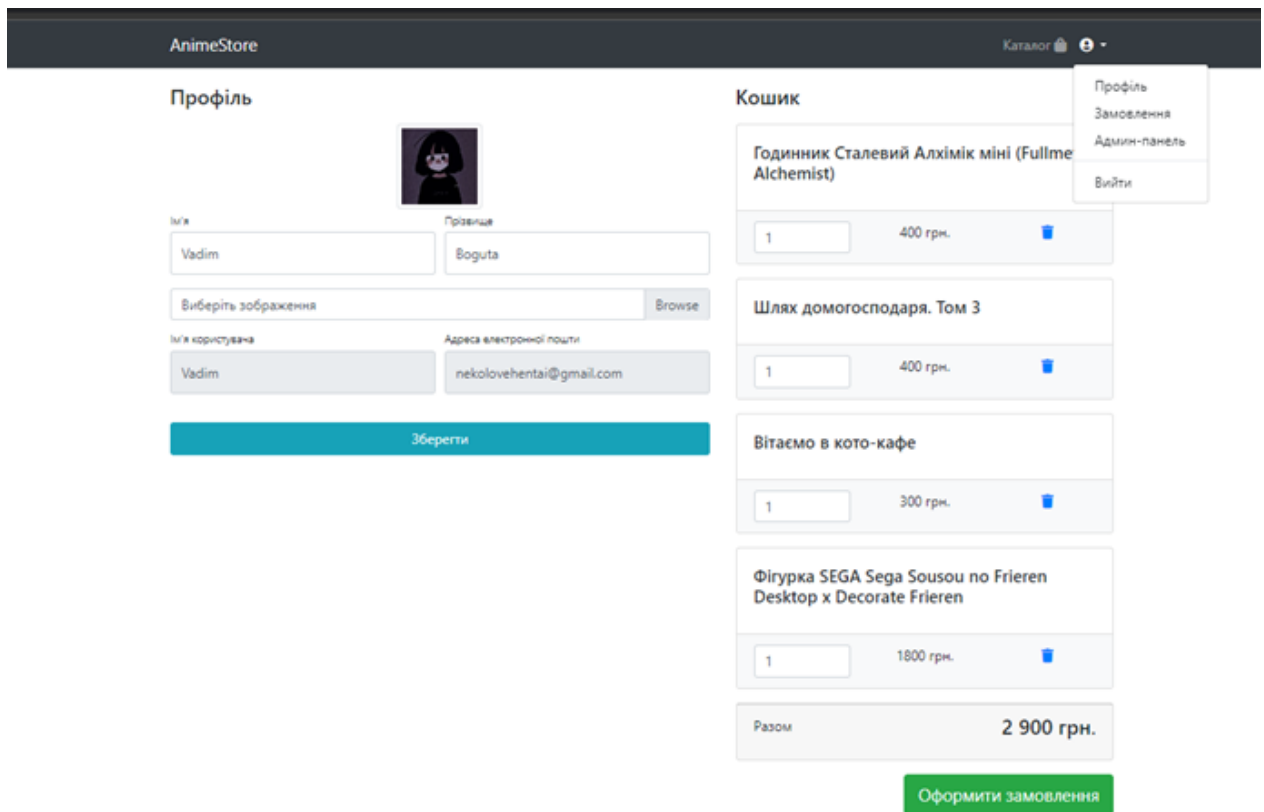


Рисунок 4.4 – Сторінка профілю з товаром у кошику

Сторінку каталогу було реалізовано завдяки коду на рисунку 4.5

```

10 class StartProject(Title, TemplateView):
11     template_name = 'products/start.html'
12     title = 'AnimeStore'
13
14
15 class ProductsListView(Title, ListView):
16     model = Product
17     template_name = 'products/products.html'
18     paginate_by = 6
19     title = 'Katanor'
20
21     def get_context_data(self, *, object_list=None, **kwargs):
22         context = super(ProductsListView, self).get_context_data()
23         context['categories'] = ProductPredicament.objects.all()
24         return context
25
26     def get_queryset(self):
27         queryset = super(ProductsListView, self).get_queryset()
28         category_id = self.kwargs.get('category_id')
29         return queryset.filter(category_id=category_id) if category_id else queryset
30
31
32 @login_required
33 def basket_add(request, product_id):
34     product = Product.objects.get(id=product_id)
35     baskets = Basket.objects.filter(user=request.user, product=product)
36
37     if not baskets.exists():
38         Basket.objects.create(user=request.user, product=product, quantity=1)
39     else:
40         basket = baskets.first()
41         basket.quantity += 1
42         basket.save()
43
44     return HttpResponseRedirect(request.META['HTTP_REFERER'])
45
46
47 @login_required
48 def basket_remove(request, basket_id):
49     basket = Basket.objects.get(id=basket_id)
50     basket.delete()
51     return HttpResponseRedirect(request.META['HTTP_REFERER'])
52

```

Рисунок 4.5 – Частина коду для реалізації сторінки каталогу

Класи:

- **StartProject**: Цей клас описує головну сторінку магазину. Він використовує шаблон `products/start.html` та задає заголовок "AnimeStore".
- **ProductsListView**: Цей клас описує список продуктів. Він використовує модель `Product` та шаблон `products/products.html`. Він також пагінує список по 6 продуктів на сторінку та задає заголовок "Каталог".
- **get_context_data**: Цей метод використовується для додавання додаткового контексту до шаблону. Він додає список категорій продуктів до контексту.
- **get_queryset**: Цей метод використовується для фільтрації списку продуктів за категорією. Він отримує ID категорії з URL-адреси та фільтрує список продуктів за цим ID.

Функції:

- **basket_add**: Ця функція використовується для додавання продукту до кошика. Вона отримує ID продукту з URL-адреси та шукає кошик користувача з цим продуктом. Якщо кошика не існує, вона створює новий кошик з кількістю 1. Інакше вона збільшує кількість продукту в кошику на 1.

- **basket_remove:** Ця функція використовується для видалення продукту з кошика. Вона отримує ID кошика з URL-адреси та видаляє кошик.

Інші функції:

- **@login_required:** Цей декоратор використовується для захисту функцій, щоб до них могли отримати доступ лише зареєстровані користувачі.
- **HttpResponseRedirect:** Цей об'єкт використовується для перенаправлення користувача на іншу сторінку.

А авторизація, реєстрація, підтвердження електронної пошти та сам профіль користувача був реалізований завдяки коду на рисунку 4.6

```
class UserLog(Title, LoginView):
    template_name = 'users/login.html'
    form_class = UserLoginForm
    title = 'Авторизація'

class UserReg(Title, SuccessMessageMixin, CreateView):
    model = User
    form_class = UserRegForm
    template_name = 'users/reg.html'
    success_url = reverse_lazy('users:login')
    success_message = 'Ви успішно зареєструвалися '
    title = 'Реєстрація'

class UserProfile(Title, UpdateView):
    model = User
    form_class = ProfileForm
    template_name = 'users/profile.html'
    title = 'Профіль'

    def get_success_url(self):
        return reverse_lazy('users:profile', args=(self.object.id,))

    def get_context_data(self, **kwargs):
        context = super(UserProfile, self).get_context_data()
        context['baskets'] = Basket.objects.filter(user=self.object)
        return context

class EmailVerificationView(Title, TemplateView):
    title = 'Ваш обліковий запис підтверджено'
    template_name = 'users/verification.html'

    def get(self, request, *args, **kwargs):
        code = kwargs['code']
        user = User.objects.get(email=kwargs['email'])
        email_verifications = Verification.objects.filter(user=user, code=code)
        if email_verifications.exists() and not email_verifications.first().is_expired():
            user.is_verified_email = True
            user.save()
            return super(EmailVerificationView, self).get(request, *args, **kwargs)
        else:
            return HttpResponseRedirect(reverse('start'))
```

Рисунок 4.6 – Частина коду для реалізації сторінок:авторизація, реєстрація,профіль

Детальніше ми можемо побачити інтернет магазин на мапі сайту котра зображена на рисунку 4.7

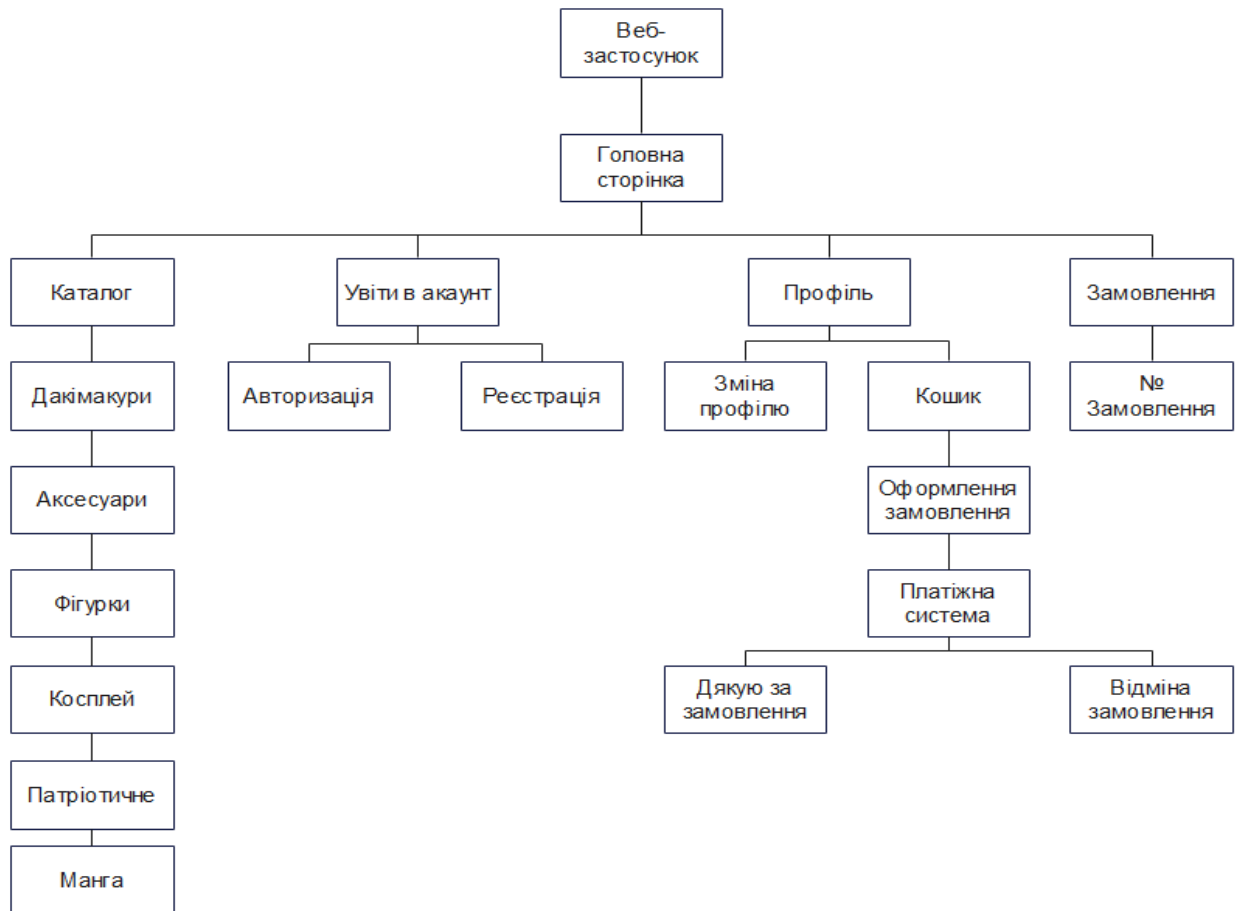


Рисунок 4.5 – Мапа сайту веб-застосунку

4.1 Опис функціональності системи

Веб-застосунок для покупки аніме-товарів містить у собі актора – користувача системи. На рисунку 4.5 можемо побачити діаграму варіантів використання

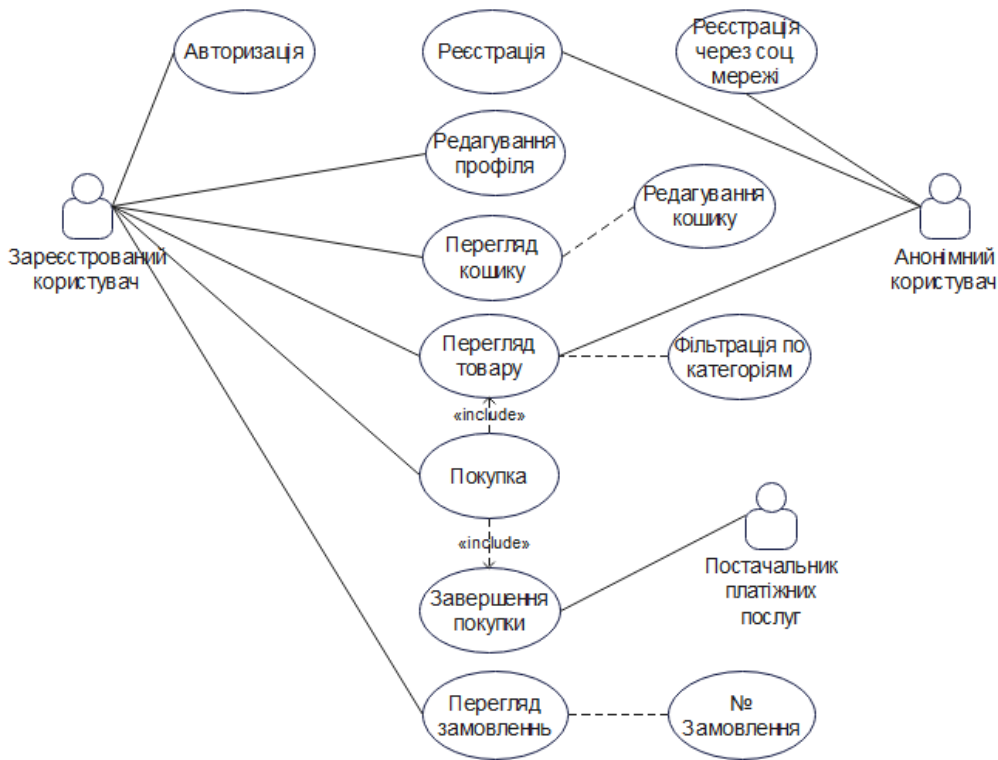


Рисунок 4.6 – Діаграма варіантів використання

Система онлайн-магазину дозволяє користувачам:

- Реєструватися та входити в систему.
- Переглядати та фільтрувати товари за категоріями.
- Додавати товари до кошика та редагувати його вміст.
- Купувати товари та безпечно оплачувати їх за допомогою кредитної картки або іншого способу оплати.
- Переглядати свої замовлення та відстежувати їх статус.

Система інтегрується з постачальником платіжних послуг для обробки платежів. Діаграма показує, що система онлайн-магазину пропонує широкий спектр функцій для своїх користувачів. Система проста у використанні та дозволяє користувачам легко знаходити та купувати товари.

4.2 Опис бази даних

В даному інтернет-магазині з аніме-товарами існує панель адміністратора, в якій можна вносити зміни в наповнені сайту. На рисунку 4.7 зображено вхід до панелі адміністратора, що здійснюється звичайною авторизацією через логін та пароль

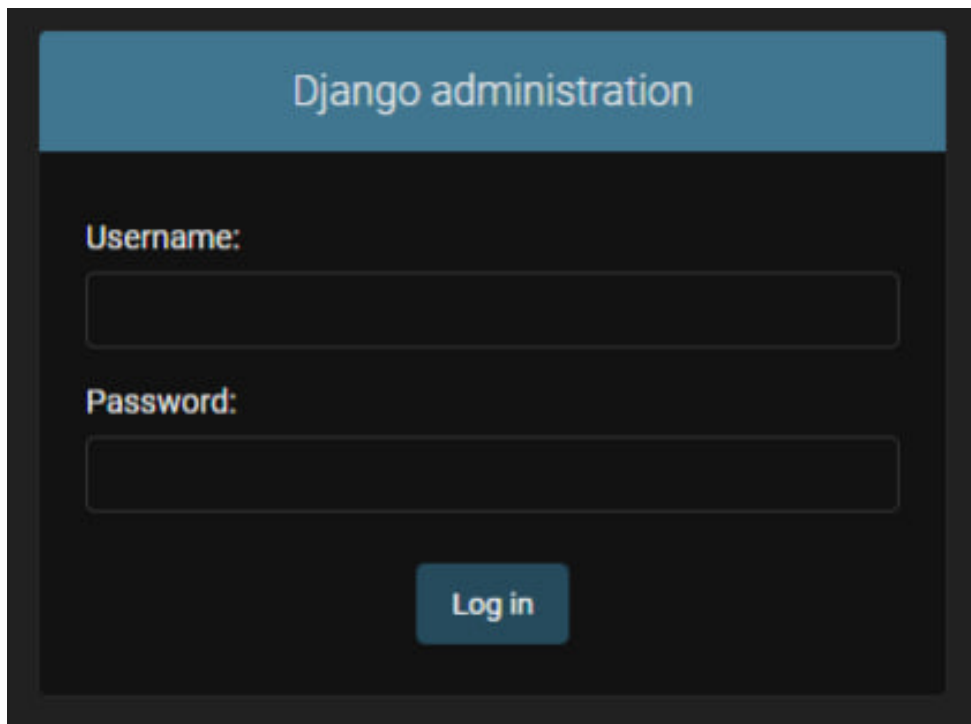


Рисунок 4.7 – Панель для авторизації

В БД існують таблиці, що зберігають всі дані про користувачів, товари, категорії, замовлення, товари у кошику, інформацію котрі можуть змінювати адміністратори

На сервері існують моделі, що описують таблиці з БД:

- Модель категорій і модель товару, що відобразатиметься в карточці зображено на рисунку 4.8
- Модель користувача і модель верифікації, що відобразатимуться на вкладках зображено на рисунку 4.9
- Модель замовлення, на якій відобразатимуться карточки зображено на рисунку 4.10

```

class ProductPredicament(models.Model):
    name = models.CharField(max_length=256, unique=True)
    description = models.TextField(null=True, blank=True)

    class Meta:
        verbose_name = 'predicament'
        verbose_name_plural = 'predicaments'

    def __str__(self):
        return self.name

class Product(models.Model):
    name = models.CharField(max_length=256)
    description = models.CharField(max_length=256)
    price = models.DecimalField(max_digits=10, decimal_places=0)
    quantity = models.PositiveSmallIntegerField(default=0)
    image = models.ImageField(upload_to='products_images', null=True, blank=True)
    stripe_product_price_id = models.CharField(max_length=128, null=True, blank=True)
    category = models.ForeignKey(to=ProductPredicament, on_delete=models.PROTECT)

```

Рисунок 4.8 – Модель сутності категорій і товару

```

class User(AbstractUser):
    image = models.ImageField(upload_to='users_images', null=True, blank=True)
    is_verified_email = models.BooleanField(default=False)

class Verification(models.Model):
    code = models.UUIDField(unique=True)
    user = models.ForeignKey(to=User, on_delete=models.CASCADE)
    created = models.DateTimeField(auto_now_add=True)
    expiration = models.DateTimeField()

```

Рисунок 4.9 – Модель сутності профілю і верифікації

```

class Purchase(models.Model):

    CREATED = 0
    PAID = 1
    ON_WAY = 2
    DELIVERED = 3
    STATUSES = (
        (CREATED, 'Створен'),
        (PAID, 'Оплачений'),
        (ON_WAY, 'В дорозі'),
        (DELIVERED, 'Доставлено'),
    )

    first_name = models.CharField(max_length=128)
    last_name = models.CharField(max_length=128)
    email = models.EmailField(max_length=256)
    address = models.CharField(max_length=256)
    basket_history = models.JSONField(default=dict)
    created = models.DateTimeField(auto_now_add=True)
    initiator = models.ForeignKey(to=User, on_delete=models.CASCADE)
    status = models.SmallIntegerField(default=CREATED, choices=STATUSES)

```

Рисунок 4.9 – Модель сутності замовлення

Ми розглянули реалізовану базу даних для даного веб-застосунку інтернет-магазину, можемо побачити схему бази даних на рисунку 4.10, ми побачили моделі категорій, товару, профілю, верифікації, замовлення і за що вони відповідають:

Категорії: Ця модель описує структуру каталогу товарів. Вона може містити такі атрибути, як ID категорії, назва категорії, опис категорії, категорія-батько (для ієрархічної структури) та інші релевантні поля.

Товари: Ця модель зберігає інформацію про окремі продукти, що продаються в магазині. Її атрибути можуть включати ID товару, назву товару, опис товару, ціну, зображення, категорію, характеристики, доступність та інші необхідні дані.

Профілі: Ця модель зберігає інформацію про зареєстрованих користувачів магазину. Її атрибути можуть включати ID користувача, ім'я користувача, пароль, адресу електронної пошти, адресу доставки, історію замовлень та інші релевантні дані.

Перевірка: Ця модель використовується для зберігання інформації про процес верифікації користувачів, наприклад, підтвердження електронної пошти або активацію облікового запису.

Замовлення: Ця модель зберігає дані про замовлення, зроблені користувачами. Її атрибути можуть включати ID замовлення, ID користувача, дату замовлення, статус замовлення, загальну суму, адресу доставки, список замовлених товарів та інші релевантні дані.

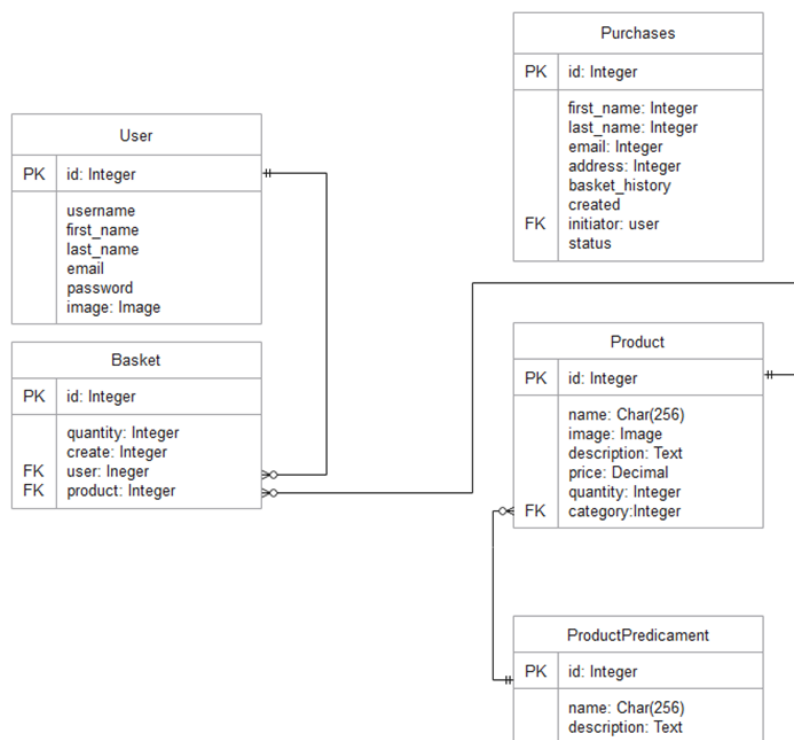


Рисунок 4.10 – Схема бази даних

5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

5.1 Системні вимоги

Для забезпечення стабільної роботи програми для купівлі аніме-товарів у інтернет-магазині має відповідати наступним вимогамЖ:

1. Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari або Opera.
2. Операційна система: Windows 7 або новіша, Android 5.0 або новіша
3. Обладнання: Процесор 1 GHz або новіший, 2 ГБ оперативної пам'яті, 5 ГБ вільного місця на диску, стійке підключення до Інтернету

5.2 Робота з програмним модулем

Після запуску клієнтської частини в браузері користувач побачить головну сторінку(рис. 5.1) та зможе з неї перейти до каталогу і передивлятися товар або перейти у вкладку з авторизацією у закріпленому меню що розміщений праворуч зверху

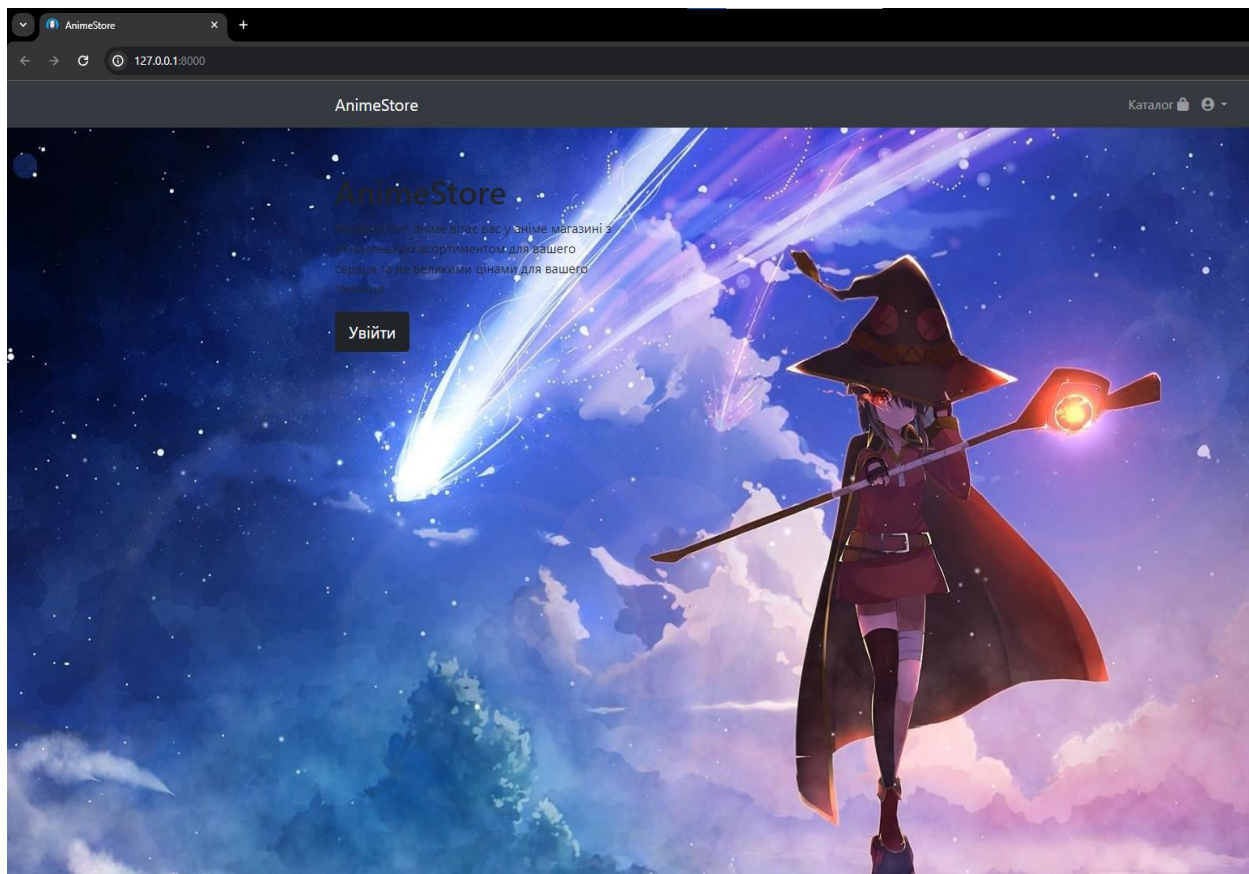


Рисунок 5.1 – головна сторінка інтернет-магазину

ВИСНОВКИ

Дипломна практика надає професійної підготовки майбутнім спеціалістам, вчить самостійно знаходити рішення для реалізації програмного продукту, ініціативності, набуттю організаційних, дослідницьких навичок, які, забезпечують повноцінну роботу випускників

Практика відіграє важливу роль у підготовці спеціалістів, які володіють сучасними технологіями, вміло використовують методологію та методики проектування різних програмних систем на різних мовах програмувань. Метою переддипломної роботи є не лише закріплення теоретичних знань, отриманих під час навчання, а й набуття, вдосконалення та розширення практичних навичок та компетенцій у вибраній галузі.

Розроблено і створено програмний продукт, а саме веб-застосунок інтернет-магазин для спрощення процесу покупки аніме-товарів в мережі інтернет, який відображає можливість реєструватись та створювати свій профіль у інтернет-магазині та змінювати цей профіль де будуть відображені аніме-товари котрі ви покладете собі у кошик на сторонці каталогу з товарами та можливістю їх фільтрувати по категоріям, та після купівлю реалізована можливість передивлюватись статус свого замовлення та саме замовлення також

Набуто досвіду розробки і створення програмних засобів. Отримано навички застосування сучасних, інформаційних технологій при розробці й створенні програмних продуктів, в підборі інформаційних матеріалів для подальшої роботи над дипломним проектом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційна документація Django [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.djangoproject.com/en/5.0/>.
2. Офіційна документація по Django Debug Toolbar [Електронний ресурс] – Режим доступу до ресурсу: <https://django-debug-toolbar.readthedocs.io/en/latest/>.
3. Офіційна документація по Stripe [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.stripe.com/>.
4. Офіційна документація по Bootstrap [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/>.
5. Офіційний сайт YoroKobi [Електронний ресурс] – Режим доступу до ресурсу: <https://yorokobi.com.ua/ua/>.
6. Офіційний сайт ITEMSHOP [Електронний ресурс] – Режим доступу до ресурсу: <https://itemshop.com.ua/>.
7. Офіційний сайт lollibuy [Електронний ресурс] – Режим доступу до ресурсу: <https://lolibuy.com/>.
8. Jeff F. Django Web Application Development in Python / F. Jeff, B. Paul, C. Wesley., 2009. – 40 с.
9. Vincent. Django for Beginners: Build Websites with Python and Django / William S. Vincent., 2020.
10. Офіційний сайт foxmind [Електронний ресурс] – Режим доступу до ресурсу: <https://foxminded.ua/bootstrap/>.
11. Сайт W3schoolsUA [Електронний ресурс] – Режим доступу до ресурсу: <https://w3schoolsua.github.io/hyperskill/pycharm-basics.html#gsc.tab=0>.
12. Офіційний сайт DesdeLinux про середовище розробки PyCharm [Електронний ресурс] – Режим доступу до ресурсу: <https://blog.desdelinux.net/uk/pycharm-%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%BE%D0%B2%D0%B8%D1%89%D0%B5-%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B8-%D0%B4%D0%BB%D1%8F-python/>.

13. Сайт МІЙКЛАС про систему програмування PyCharm [Електронний ресурс] – Режим доступу до ресурсу: <https://www.miyklas.com.ua/p/informatica/8-klas/algorithmi-ta-programi-394917/suchasni-movi-programuvannia-osnovi-roboti-v-isr-394274/re-1c2610f7-fbec-428a-8b2c-e011c30a05d3>.
14. Сайт Cybertec, пол PostgreSQL [Електронний ресурс] – Режим доступу до ресурсу: https://www.cybertec-postgresql.com/en/services/postgresql-support/?gad_source=1&gclid=Cj0KCQjw3tCyBhDBARIsAEY0XNk4IDcDQLJsibRoQmSouXmXkiLtqiqHEB-705lm1BNrzyg-f6EUWGMaAs_EALw_wcB.
15. Mastering PostgreSQL 13: Build, administer, and maintain database applications efficiently with PostgreSQL 13, Fourth Edition / Hans-Jürgen Schönig., 2020. – 476 с.
16. Сайт IT Рейтинг України про типи сайтів [Електронний ресурс] – Режим доступу до ресурсу: <https://it-rating.ua/tipi-saytiv-yaki-buvayut-vidi-saytiv-ta-ih-riznitsya-klasifikatsiya>.
17. Introduction to Django and Model-View-Template (MVT) Applications in Django [Електронний ресурс] – Режим доступу до ресурсу: <https://pbp-fasilkom-ui.github.io/ganjil-2023/en/assignments/tutorial/tutorial-1>.
18. Basics of Django: Model-View-Template (MVT) Architecture [Електронний ресурс] – Режим доступу до ресурсу: <https://angelogentileiii.medium.com/basics-of-django-model-view-template-mvt-architecture-8585aecffb6>.

ДОДАТОК А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка Web-застосунку інтернет-магазин "Anime Store" на python з використанням Django

Виконав студент 4 курсу
групи ПД-43

Богута Вадим Сергійович
Керівник роботи

Довженко Тимур Павлович кандидат технічних наук,
доцент кафедри Інженерії програмного забезпечення

Київ – 2024

Мета, об'єкт та предмет дослідження

- **Мета роботи:** Спрощення процесу покупки аніме-товарів в мережі інтернет за рахунок розробки інтернет-магазину "Anime Store".
- **Об'єкт дослідження:** Процес покупки аніме-товарів в інтернет-магазині.
- **Предмет дослідження:** інтернет-магазин для покупки аніме-товарів.

2

Задачі дипломної роботи

1. Проаналізувати вже існуючі застосунки що дозволяють купляти і продавати аніме-товари, виявити їх переваги та недоліки, скласти таблицю порівнянь.
2. На основі проаналізованих застосунків пов'язаних з аніме-товарами розробити функціональні та нефункціональні вимоги.
3. Розробити дизайн застосунку.
4. Розробити застосунок на основі обраних інструментів.
5. Провести тестування Web-застосунку та проаналізувати результати тестування.
6. Пройти апробацію роботи на науково-технічних конференціях.

3

Аналіз аналогів

Назва	Yorokobi	lolibuy	ITEMSHOP	AnimeStore
Реєстрація користувачів	-	+	-	+
Стеження за замовленнями	+	+	-	+
Система оплати	Післяоплата, оплата на рахунок, Онлайн-оплата	Післяоплата, предоплата, оплата за реквізитами	Післяплата, Оплата Liqpay	Післяоплата, Оплата Stripe
Фільтрація по категоріям	+	+	+	+
Реєстрація через соціальні мережі	-	-	-	+
Кросбраузерність	+	+	-	+

4

Вимоги до програмного забезпечення

Функціональні вимоги:

1. Реєстрація в системі, використовуючи свою адресу електронної пошти та пароль, або через соціальні мережі.
2. Створювати та редагувати свій профіль.
3. Фільтрація та пошук.
4. Можливість додавати продукти до кошика, переглядати вміст кошика та видаляти продукти з кошика.
5. Система повинна підтримувати різні способи оплати.
6. Оформити замовлення, надавши свою адресу доставки та інформацію про оплату.
7. Відстежувати статус своїх замовлень та переглядати вміст замовлення.

Нефункціональні вимоги:

1. Підтримка різних видів браузерів: Google, Microsoft Edge, Exploer

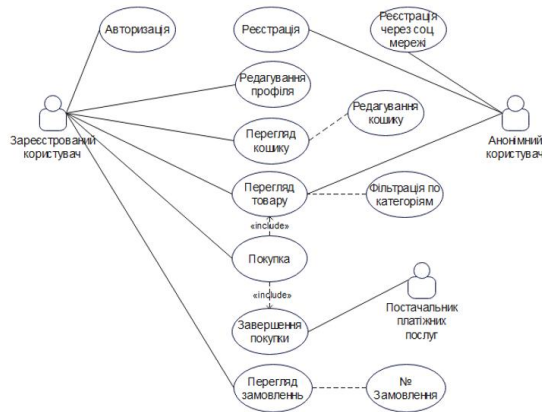
5

Програмні засоби реалізації



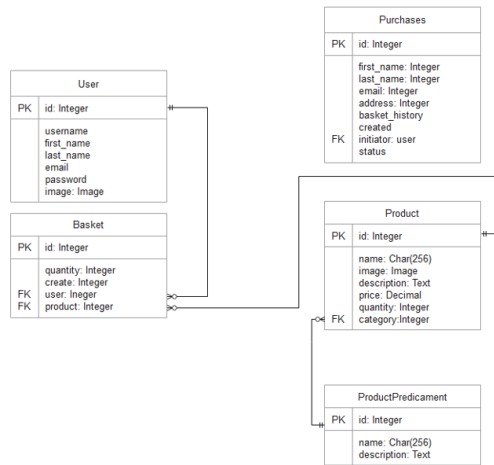
6

Діаграма варіантів використання



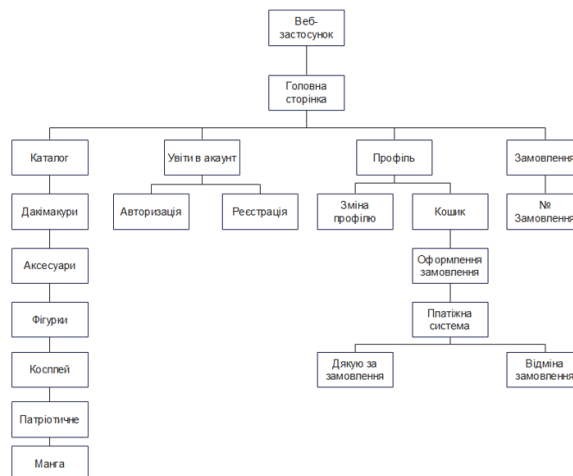
7

Схема бази даних



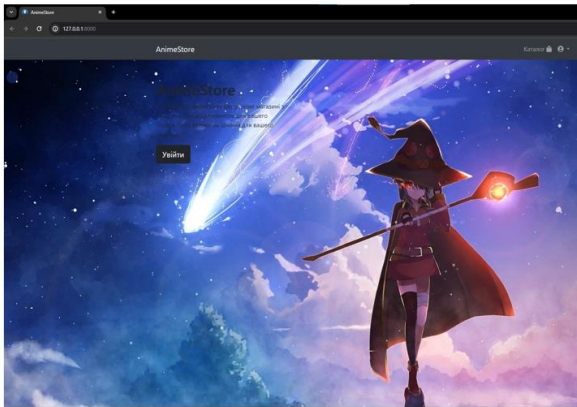
8

Мапа сайту

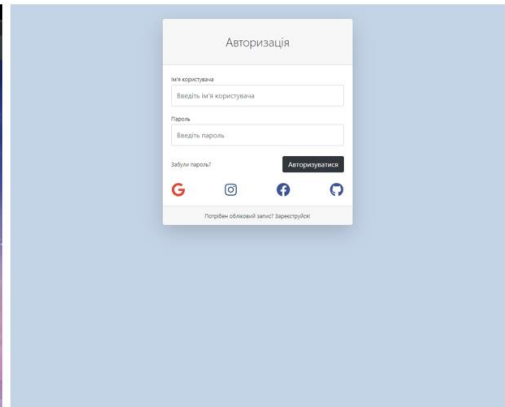


9

Екранні форми



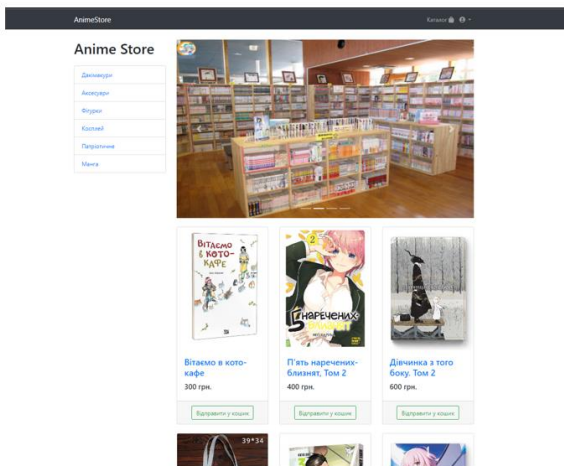
Вхід



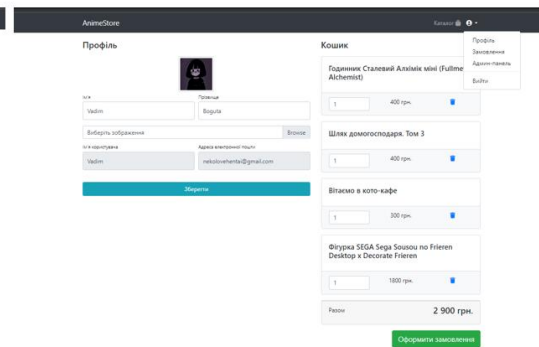
Авторизація

10

Екранні форми



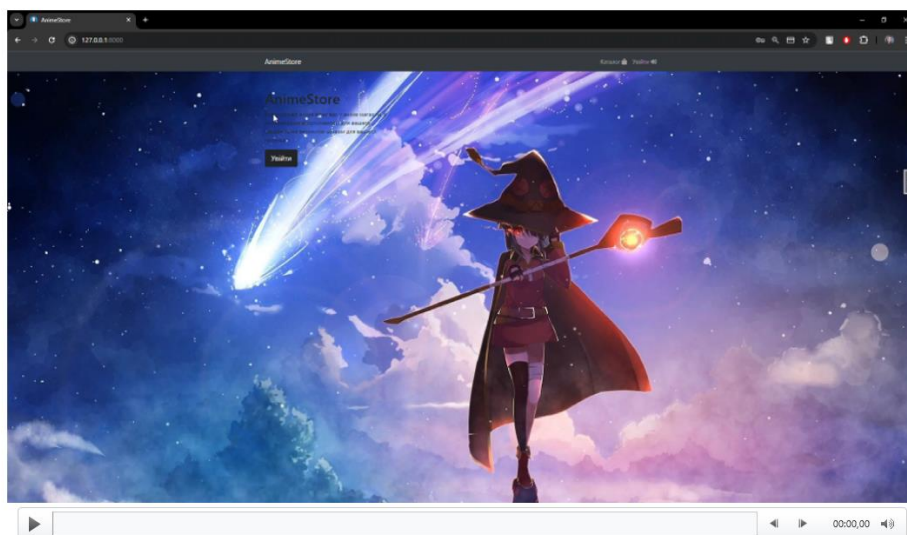
Каталог



Профіль

11

Демонстрація роботи програми



12

Апробація результатів дослідження

1. Зінченко О. В. Богута В. С. Розробка Web-застосунку на python з використанням Django // IV Всеукраїнська Науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті» - подано до друку. Київ: ДУТ, 2024;
2. Зінченко О. В. Богута В. С. Розробка на python з використанням Django // IV Всеукраїнська Науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті» - подано до друку. Київ: ДУТ, 2024;

13

Висновки

1. Проаналізовано вже існуючі застосунки, а саме lolibuy, ITEMSHOP, Yorokobi, виявлено їх переваги та недоліки та складено таблицю порівнянь.
2. Розроблені технічне завдання і визначені функціональні та нефункціональні вимоги до застосунку.
3. Досліджені та обрані інструменти розробки: Python, Django, Stripe, PostgreSQL, Bootstrap.
4. Розроблено дизайн застосунку і застосунок інтернет-магазин для купівлі аніме-товарів на основі обраних інструментів розробки.
5. Протестовано Web-застосунок на відповідність вимогам. За результатами тестування Web-застосунок працює відповідно вимогам
6. Пройдена апробація робіт на науково-технічних конференціях.

14

ДОДАТОК Б

ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ

settings.py

```
from pathlib import

BASE_DIR =

SECRET_KEY = 'django-insecure-w&1tmv1n81h3v-uo=!_z4cfu+p(*1$f-a-z9

DEBUG = True

ALLOWED_HOSTS =

DOMAIN_NAME =

INSTALLED_APPS = [
    'django.contrib.admin'
    'django.contrib.auth'
    'django.contrib.contenttypes'
    'django.contrib.sessions'
    'django.contrib.messages'
    'django.contrib.staticfiles'
    'django.contrib.sites'
    'django.contrib.humanize'

    'allauth',
    'allauth.account',
    'allauth.socialaccount'
    'allauth.socialaccount.providers.github
#
#
    "debug_toolbar",

    'products',
    'users',
    'purchases',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware
    'django.contrib.sessions.middleware.SessionMiddlewar
    'django.middleware.common.CommonMiddleware
    'django.middleware.csrf.CsrfViewMiddleware
    'django.contrib.auth.middleware.AuthenticationMiddlewar
    'django.contrib.messages.middleware.MessageMiddleware
    'django.middleware.clickjacking.XFrameOptionsMiddlewar
    'allauth.account.middleware.AccountMiddleware
    'debug_toolbar.middleware.DebugToolbarMiddleware
]
```

```

ROOT_URLCONF = 'animestore.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                'products.context.baskets',
            ],
        },
    },
]

WSGI_APPLICATION = 'animestore.wsgi.application'

INTERNAL_IPS = [
    "127.0.0.1",
    "localhost",
]

DATABASES = {
    "default": {
        "ENGINE": "django.db.backends.postgresql_psycopg2",
        "NAME": "animestore_db",
        "USER": "postgres",
        "PASSWORD": "vaallfaall",
        "HOST": "localhost",
        "PORT": "5432",
    }
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
]

```

```

    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'uk-ua'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

STATIC_URL = '/static/'
if DEBUG:
    STATICFILES_DIRS = (
        BASE_DIR / 'static',
    )
else:
    STATIC_ROOT = BASE_DIR / 'static'

MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

AUTH_USER_MODEL = 'users.User'
LOGIN_URL = '/users/login/'
LOGIN_REDIRECT_URL = '/'
LOGOUT_REDIRECT_URL = '/'

EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'

SITE_ID = 1

AUTHENTICATION_BACKENDS = [

    'django.contrib.auth.backends.ModelBackend',

    'allauth.account.auth_backends.AuthenticationBackend',

]

SOCIALACCOUNT_PROVIDERS = {
    'github': {
        'SCOPE': [

```



```

        'user',
        'repo',
        'read:org',
    ],
}
}

# Stripe

STRIPE_PUBLIC_KEY =
('pk_test_510tbqjFDUX1RI0LLG6ieJnT4c2beGsBKu60nr1ouVy6da8DfCZeC63qInTLtDuMYhXGR06Dw
03IWCEtHaEvgOr'
    'zD00iG2DrqIC')

STRIPE_SECRET_KEY =
('sk_test_510tbqjFDUX1RI0LLnTgo1YhY0ZqQTZB6WzLxpvsY79VpLZ49ABsa7ekMlgJgHVnSRgoQK2Ty
EmAiNWAODVv8E'
    'Usb00g4Q8yMdW')

STRIPE_WEBHOOK_SECRET =
'whsec_3da0aa9445ebce349b00728f65f4219d5b76a8bc1fa28402c978b1b4e27b95cc'

```

products/views.py

```

from django.contrib.auth.decorators import login_required
from django.shortcuts import HttpResponseRedirect
from django.views.generic.base import TemplateView
from django.views.generic.list import ListView

from products.models import Basket, Product, ProductPredicament
from text.title import Title

class StartProject(Title, TemplateView):
    template_name = 'products/start.html'
    title = 'AnimeStore'

class ProductsListView(Title, ListView):
    model = Product
    template_name = 'products/products.html'
    paginate_by = 6
    title = 'Каталог'

    def get_context_data(self, *, object_list=None, **kwargs):
        context = super(ProductsListView, self).get_context_data()
        context['categories'] = ProductPredicament.objects.all()
        return context

    def get_queryset(self):
        queryset = super(ProductsListView, self).get_queryset()

```

```

        category_id = self.kwargs.get('category_id')
        return queryset.filter(category_id=category_id) if category_id else
queryset

@login_required
def basket_add(request, product_id):
    product = Product.objects.get(id=product_id)
    baskets = Basket.objects.filter(user=request.user, product=product)

    if not baskets.exists():
        Basket.objects.create(user=request.user, product=product, quantity=1)
    else:
        basket = baskets.first()
        basket.quantity += 1
        basket.save()

    return HttpResponseRedirect(request.META['HTTP_REFERER'])

@login_required
def basket_remove(request, basket_id):
    basket = Basket.objects.get(id=basket_id)
    basket.delete()
    return HttpResponseRedirect(request.META['HTTP_REFERER'])

```

products/model.py

```

import stripe
from django.conf import settings
from django.db import models

from users.models import User

stripe.api_key = settings.STRIPE_SECRET_KEY

class ProductPredicament(models.Model):
    name = models.CharField(max_length=256, unique=True)
    description = models.TextField(null=True, blank=True)

    class Meta:
        verbose_name = 'predicament'
        verbose_name_plural = 'predicaments'

    def __str__(self):
        return self.name

class Product(models.Model):
    name = models.CharField(max_length=256)

```

```

description = models.CharField(max_length=256)
price = models.DecimalField(max_digits=10, decimal_places=0)
quantity = models.PositiveSmallIntegerField(default=0)
image = models.ImageField(upload_to='products_images', null=True, blank=True)
stripe_product_price_id = models.CharField(max_length=128, null=True,
blank=True)
category = models.ForeignKey(to=ProductPredicament, on_delete=models.PROTECT)

class Meta:
    verbose_name = 'product'
    verbose_name_plural = 'products'

def __str__(self):
    return f'Продукт: {self.name} | Категорія: {self.category.name}'

def add_stripe_product_price(self):
    stripe_product = stripe.Product.create(name=self.name)
    stripe_product_price = stripe.Price.create(
        product=stripe_product['id'],
        unit_amount=round(self.price * 100),
        currency='uah'
    )
    return stripe_product_price

def save(self, force_insert=False, force_update=False, using=None,
update_fields=None):
    if not self.stripe_product_price_id:
        stripe_product_price = self.add_stripe_product_price()
        self.stripe_product_price_id = stripe_product_price['id']
    super(Product, self).save(force_insert=False, force_update=False,
using=None, update_fields=None)

class BasketQuerySet(models.QuerySet):
    def total_sum(self):
        return sum(basket.sum() for basket in self)

    def total_quantity(self):
        return sum(basket.quantity for basket in self)

    def stripe_products(self):
        line_items = []
        for basket in self:
            item = {
                'price': basket.product.stripe_product_price_id,
                'quantity': basket.quantity
            }
            line_items.append(item)
        return line_items

```

```

class Basket(models.Model):
    user = models.ForeignKey(to=User, on_delete=models.CASCADE)
    product = models.ForeignKey(to=Product, on_delete=models.CASCADE)
    quantity = models.PositiveSmallIntegerField(default=0)
    created_timestamp = models.DateTimeField(auto_now_add=True)

    objects = BasketQuerySet.as_manager()

    def __str__(self):
        return f'Кошик для {self.user.username} | Продукт: {self.product.name}'

    def sum(self):
        return self.product.price * self.quantity

    def basket_json(self):
        basket_item = {
            'product_name': self.product.name,
            'quantity': self.quantity,
            'price': float(self.product.price),
            'sum': float(self.sum()),
        }
        return basket_item

```

products/admin.py

```

from django.contrib import admin

from products.models import Basket, Product, ProductPredicament

admin.site.register(ProductPredicament)

@admin.register(Product)
class PrAdmin(admin.ModelAdmin):
    list_display = ('name', 'price', 'quantity', 'category',)
    fields = ('image', 'name', 'description', ('price', 'quantity'),
'stripe_product_price_id', 'category')
    readonly_fields = ('description',)
    search_fields = ('name',)
    ordering = ('-name',)

class BasketAdmin(admin.TabularInline):
    model = Basket
    fields = ('product', 'quantity', 'created_timestamp')
    readonly_fields = ('created_timestamp',)
    extra = 0

```

products/urls.py

```
from django.urls import path

from products.views import ProductsListView, basket_add, basket_remove

app_name = 'products'

urlpatterns = [
    path('', ProductsListView.as_view(), name='start'),
    path('category/<int:category_id>', ProductsListView.as_view(),
name='category'),
    path('page/<int:page>/', ProductsListView.as_view(), name='paginator'),
    path('baskets/add/<int:product_id>/', basket_add, name='basket_add'),
    path('baskets/remove/<int:basket_id>/', basket_remove, name='basket_remove'),
]
```