

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу

Назва теми

КвРІСТ 200186 20.01.10 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Виконав: студент IV курсу, група ІСТ-20-1

Підпис

А.В.Смірнов

Ініціали, прізвище

Керівник

Підпис, дата

Є.Г. Гнатчук

Ініціали, прізвище

Нормоконтролер

Підпис, дата

І.О. Засорнова

Ініціали, прізвище

До захисту допускаю:
Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

Підпис

Т.О. Говорущенко

Ініціали, прізвище

«12» червня 2024 р.

Хмельницький 2024

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
Освітній рівень БАКАЛАВР
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ
Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О. Говорушенко

" 10 " 01 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Смірнову Андрію Валдимовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу

Керівник проекту (роботи) Є.Г. Гнатчук., к.т.н., доцент.

Прізвище, ім'я, по батькові, науковий ступінь, місце заняття

Затверджена наказом ректора університету від 15.02.2024 р. № 8

2. Строк подання студентом проекту (роботи) на кафедру 07.06.2024р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу
та постановка задачі щодо її удосконалення

Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу.

Постановка задачі щодо удосконалення інформаційної системи підтримки інтернет-магазину чоловічого одягу.

Проектування системи обробки інформації для підтримки роботи інтернет-магазину чоловічого одягу.

Програмна реалізація інформаційної системи підтримки інтернет-магазину чоловічого одягу.

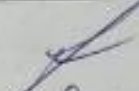
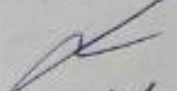

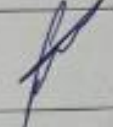
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень):

Захист інформації в програмному забезпеченні

Опис реалізації програмного забезпечення інформаційної системи

Структурні схеми додатків «Goods», «Users», «Orders»

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Засорнова І.О., доцент кафедри КПС		
Антиплагиат	Нічепорук А.О., доцент кафедри КПС		

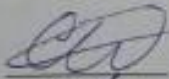

7. Дата видачі завдання « 17 » _____ 01 _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2024	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2024	виконано
3	Робота над розділом 1 – характеристика предметної області та постановка задачі	01.03.2024	виконано
4	Робота над розділом 2 – Проектування інформаційної системи	01.04.2024	виконано
5	Робота над розділом 3 – Програмна реалізація інформаційної системи підтримки інтернет-магазину чоловічого одягу	29.04.2024	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2024	виконано
7	Попередній захист ВКР	30.05.2024	виконано
	Захист ВКР на засіданні ЕК	Червень 2024 року	

Студент

Керівник роботи


Підпис

Підпис

Смірнов А.В.
Ініціали, прізвище
Є.Г.Гнатчук
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу».

Автор роботи: Смірнов Андрій Вадимович.

Керівник роботи: Гнатчук Єлизавета Геннадіївна.

Пояснювальна записка: 66 с., 48 рис., 3 табл., 3 дод., 58 джерел.

ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРНЕТ МАГАЗИН, ЗАМОВЛЕННЯ ТОВАРУ, РЕАЛІЗАЦІЯ, КОРИСТУВАЧ

Метою дипломної роботи є організація підтримки роботи інтернет-магазину чоловічого одягу визначення умов шляхом створення інформаційної системи.

Об'єктом дослідження є принципи побудови інформаційної системи підтримки роботи інтернет-магазину чоловічого одягу.

Предметом дослідження є інформаційна система підтримки роботи інтернет-магазину чоловічого одягу.

Практичне значення має спроектована та реалізована система підтримки роботи інтернет-магазину чоловічого одягу, яка дозволяє організувати зручну та ефективну взаємодію для кінцевих користувачів.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.






Підпис студента

30.05.2024

Дата

ЗМІСТ

ВСТУП.....	4
1 ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ.....	5
1.1 Аналіз предметної області.....	5
1.2 Огляд інформаційного забезпечення предметної області.....	8
1.3 Постановка задачі.....	16
2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	21
2.1 Комплексна інформаційна система для інтернет-магазину чоловічого одягу: функції та переваги.....	21
2.2 Архітектура.....	23
2.3 Функційні вимоги до ІС.....	25
2.4 Нефункційні вимоги.....	27
2.5 Користувацькі вимоги.....	29
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ІНТЕРНЕТ-МАГАЗИНУ ЧОЛОВІЧОГО ОДЯГУ.....	31
3.1 Опис реалізації програмного забезпечення інформаційної системи підтримки інтернет магазину чоловічого одягу.....	31
3.2 Архітектура та налаштування середовища розробки.....	33
3.3 Створення головного додатку «Manage.py».....	35
3.4 Створення додатку «Main».....	36
3.5 Створення додатку «Goods».....	38
3.6 Створення додатку «Users».....	43
3.7 Створення додатку «Carts».....	50
3.8 Підключення PostgreSQL.....	56
3.9 Додавання Ajax до інформаційної системи.....	59
3.10 Додаток «Orders».....	61

КвРІСТ 200186 20.01.10 ПЗ					
Зм.	Арк.	№локум.	Підпис	Дата	
Виконав		Смирнов А.В.			Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу Пояснювальна записка
Перевір.		Гаштук С.Г.			
Н.контр.		Васильова І.О.		12.01	Літера у
Затвер.		Гончаренко Т.О.			Аркуш 2
					Аркушів 66
					ХНУ ІСТ-20-1

3.11 Висновки.....	64
ВИСНОВКИ.....	65
ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАНЬ.....	67
ДОДАТОК А Копія креслення «Структура захисту інформації в програмному забезпеченні».....	72
ДОДАТОК Б Копія креслення «Схема реалізації програмного забезпечення інформаційної системи».....	73
ДОДАТОК В Копія креслення «Структурні схеми додатків «Goods», «Users», «Orders».....».....	74

					КВРІСТ 200186 20.01.10 ПЗ	Арк. 3
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У сучасному цифровому світі інтернет-магазини стали невід'ємною складовою електронної комерції, забезпечуючи зручність і доступність покупок для мільйонів користувачів по всьому світу. Зростання популярності інтернет-торгівлі, особливо в сфері одягу, підкреслює значущість оптимізації та покращення інформаційних систем, що підтримують роботу інтернет-магазинів.

Ця дипломна робота присвячена розробці та впровадженню інформаційної системи, спрямованої на підтримку ефективної роботи інтернет-магазину чоловічого одягу. Метою даного дослідження є аналіз поточних проблем та недоліків у функціонуванні інтернет-магазинів в цьому сегменті ринку, а також розробка та реалізація інформаційної системи, спрямованої на їх вирішення та покращення.

В рамках цієї роботи буде проведений аналіз сучасних тенденцій у розвитку електронної комерції та особливостей функціонування інтернет-магазинів одягу. На основі отриманих даних буде розроблено концепцію інформаційної системи, яка враховує потреби користувачів та сприяє підвищенню ефективності управління магазином, оптимізації процесів замовлення та обробки товарів, а також покращенню обслуговування клієнтів.

Результати дослідження та розробки інформаційної системи мають на меті забезпечити конкурентні переваги інтернет-магазину чоловічого одягу на ринку електронної комерції, забезпечуючи зручність, швидкість та надійність обслуговування для клієнтів та оптимізуючи внутрішні процеси управління.

У цьому контексті, дослідження та реалізація інформаційної системи підтримки роботи інтернет-магазину чоловічого одягу має великий практичний і науковий інтерес, сприяючи розвитку як сфери електронної комерції в цілому, так і окремих підприємств у цьому сегменті ринку.

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 ХАРАКТЕРИСТИКА ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної області

Інформаційна система підтримки роботи інтернет-магазину є ключовим елементом для успішного функціонування сучасного електронного бізнесу. Ця предметна область охоплює широкий спектр технологій та методів, спрямованих на автоматизацію та оптимізацію процесів продажу товарів та послуг через інтернет.

Одним із основних компонентів інформаційної системи для інтернет-магазину є електронна торговельна платформа, яка забезпечує можливість розміщення товарів, приймання замовлень, обробку платежів та взаємодію з клієнтами. Ці платформи можуть бути розроблені відповідно до власних потреб компанії: Magento(<https://account.magento.com/customer/account/login>), Shopify(<https://www.shopify.com/>), WooCommerce(<https://woocommerce.com/>) тощо.

Для забезпечення ефективної роботи інтернет-магазину необхідно використовувати бази даних для збереження інформації про товари, клієнтів, замовлення, а також для відстеження і аналізу даних про продажі та поведінку покупців. Для оптимізації роботи з базою даних можуть використовуватися різноманітні технології, такі як MySQL, PostgreSQL, MongoDB тощо.

Одним із важливих аспектів інформаційної системи підтримки інтернет-магазину є інтеграція з системами управління запасами (ERP) (рисунком 1.1), що дозволяє автоматизувати процеси складського обліку, відстеження руху товарів та управління постачанням. Це допомагає забезпечити належний рівень наявності товарів і уникнути ситуацій з нестачею або перевищенням запасів. Посилання на джерела

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.1 – Система управління запасами (ERP) [1]

Ще одним важливим аспектом є інтеграція з платіжними системами, що дозволяє забезпечити зручні та безпечні способи оплати для клієнтів. Це може включати підтримку кредитних карт, електронних гаманців, онлайн-платежів тощо.

Для підтримки взаємодії з клієнтами важливо використовувати системи управління відносинами з клієнтами (CRM) (рисунок 1.2), що дозволяють вести базу даних клієнтів, аналізувати їхню покупчу поведінку, надавати персоналізовані пропозиції та підтримку [1].

Узагальнюючи, інформаційна система підтримки роботи інтернет-магазину є складною інтегрованою системою, яка об'єднує різноманітні технології та методи для забезпечення ефективного функціонування електронного бізнесу. Вона забезпечує автоматизацію ключових процесів, оптимізацію управління та підтримку взаємодії з клієнтами, що є важливим для досягнення успіху в онлайн-торгівлі.

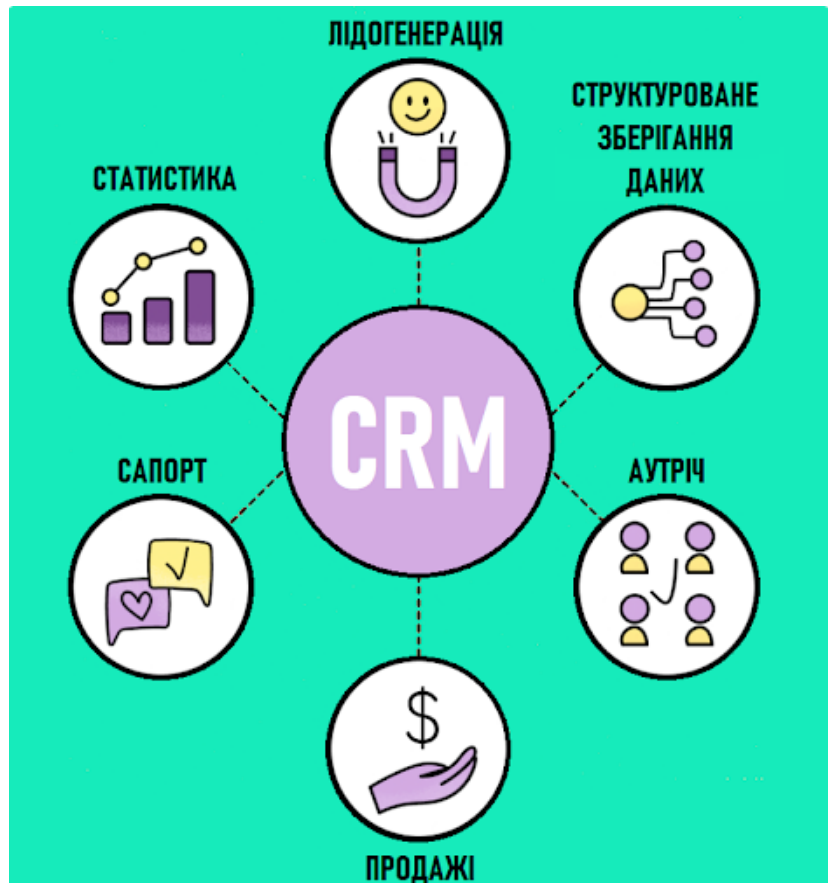


Рисунок 1.2 – Управління відносинами з клієнтами (CRM) [2]

Крім того, важливо враховувати мобільність та адаптивність інформаційної системи. З урахуванням зростання використання мобільних пристроїв для онлайн-покупок, інтерфейс магазину повинен бути оптимізований для різних пристроїв та розмірів екранів, забезпечуючи зручний та ефективний досвід користувача.

Також важливо враховувати розвиток технологій штучного інтелекту та машинного навчання. Використання таких рішень може полегшити персоналізацію пропозицій для клієнтів, автоматизувати процеси обробки замовлень та виявлення шахраїв, а також підвищити точність та ефективність аналітики даних.

Не останнім аспектом є постійне оновлення та підтримка інформаційної системи. Швидкий розвиток технологій та зміни вимог клієнтів вимагають постійного удосконалення та адаптації системи, а також швидкого вирішення будь-яких технічних проблем чи інцидентів, що можуть виникнути.

1.2 Огляд інформаційного забезпечення предметної області

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу є важливим компонентом для успішного функціонування бізнесу в цій сфері. Аналіз інформаційного забезпечення предметної області дозволяє виявити ключові аспекти, які необхідно враховувати для ефективної підтримки роботи такого інтернет-магазину. Важливо забезпечити зручний та ефективний спосіб розміщення товарів у каталозі (рисунок 1.3). Це включає можливість додавання фотографій, детальних описів, розмірів, цін та інших характеристик товарів, що дозволяє покупцям отримати вичерпну інформацію про кожен товар. Крім того, важливо мати можливість категоризувати товари за різними критеріями, такими як тип, бренд, колекція тощо, щоб спростити пошук та навігацію для клієнтів.

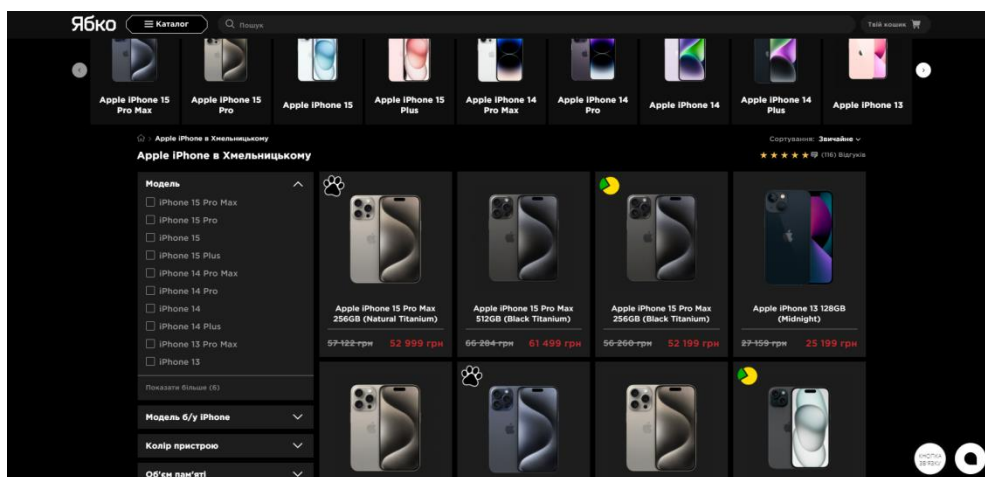


Рисунок 1.3 – Приклад каталогу товару мережі магазинів Ябло [56]

Інформаційна система повинна забезпечувати можливість приймання та обробки замовлень від клієнтів, що включає формування кошика покупок, обробку платежів та зв'язок з системами платіжних шлюзів для забезпечення безпечних та зручних транзакцій. Це дозволяє забезпечити безперебійну роботу магазину, від моменту додавання товару в кошик до завершення покупки.

Важливим є також впровадження системи управління клієнтськими відносинами (CRM), яка дозволяє збирати та аналізувати дані про клієнтів, їхні покупки та взаємодію з магазином. Це допомагає підтримувати довгострокові

відносини з клієнтами, надавати персоналізовані пропозиції, інформувати про новинки та спеціальні акції, а також оперативно реагувати на запити та скарги.

Аналітика та звітність також є важливою складовою інформаційної системи. Система повинна забезпечувати можливість ведення аналізу даних про продажі, відвідуваність сайту, конверсію, середній чек та інші важливі показники. Це дозволяє керівництву магазину розуміти ефективність бізнесу, виявляти тенденції, коригувати маркетингові стратегії та приймати обґрунтовані рішення щодо подальшого розвитку бізнесу. Зокрема, аналіз поведінки користувачів на сайті може допомогти виявити, які товари найбільш популярні, які сторінки викликають найбільше інтересу та на яких етапах покупці найчастіше залишають процес покупки.

Таблиця 1.1 – Мобільність та Адаптивність інтернет магазину

Мобільність	
Мобільні додатки	Розробка мобільних додатків для платформ Android та iOS дозволяє користувачам зручно переглядати товари, робити покупки та отримувати сповіщення про акції та новини магазину.
Мобільний веб-сайт	Оптимізація веб-сайту для мобільних пристроїв, щоб забезпечити швидкий та зручний доступ до інформації про товари та послуги.
Відповідність мобільним стандартам	Впевненість, що мобільні додатки та веб-сайт магазину відповідають всім вимогам та стандартам, що дозволяють їм працювати ефективно та безпечно на різних пристроях.

Кінець таблиці 1.1 – Мобільність та Адаптивність інтернет магазину

Адаптивність	
Responsive design	Використання адаптивного дизайну для того, щоб забезпечити, що веб-сайт адаптується до різних розмірів екрану та пристроїв, включаючи комп'ютери, планшети та смартфони.
Гнучкість контенту	Створення контенту, який легко читається та взаємодіється з користувачем на будь-якому пристрої, незалежно від його розміру екрану чи специфікацій.
Швидкість завантаження	Оптимізація веб-сайту для швидкого завантаження на різних пристроях, що дозволяє забезпечити позитивний користувацький досвід та зменшити відскок користувачів.

Безпека інформації є критично важливим аспектом для будь-якого інтернет-магазину. Інформаційна система повинна мати вбудовані заходи безпеки, такі як шифрування даних, захист від кібератак та вразливостей, а також регулярні оновлення системи безпеки (рисунок 1.4).

Це дозволяє захистити конфіденційну інформацію клієнтів, зокрема особисті дані та дані про платежі, від несанкціонованого доступу та крадіжок. Крім того, система повинна мати механізми резервного копіювання даних та відновлення у разі виникнення непередбачуваних ситуацій, щоб мінімізувати ризики втрати даних та забезпечити безперервну роботу магазину.

Важливим аспектом є інструктування персоналу магазину з питань кібербезпеки та поведінки в разі можливих інцидентів. Це включає усвідомлення загроз і правильні дії для запобігання витоку інформації або доступу до неї осіб, не маючих на це дозволу.

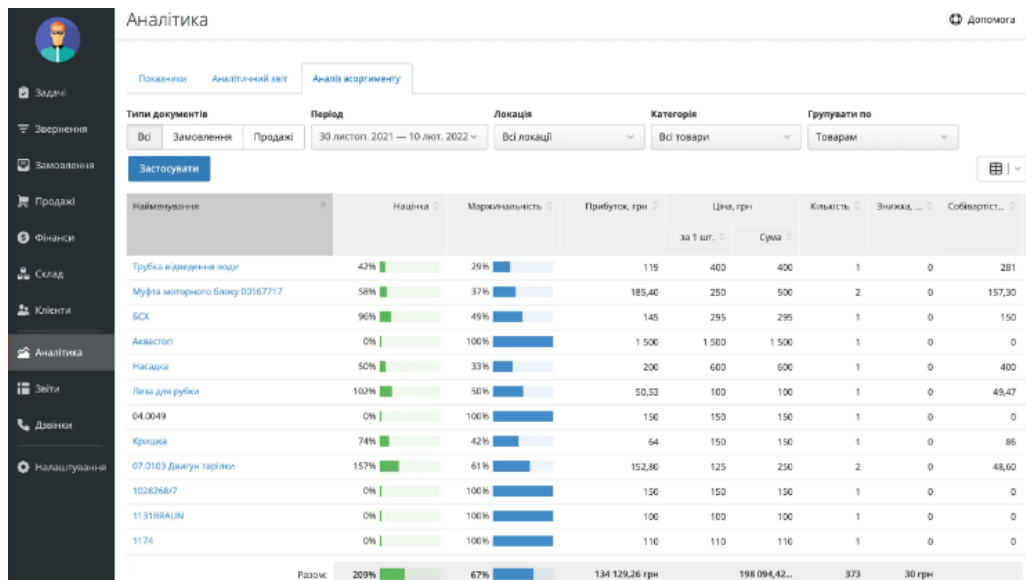


Рисунок 1.4 – Аналітика та звітності, вбудовані в базу даних

Це дозволяє захистити конфіденційну інформацію клієнтів, зокрема особисті дані та дані про платежі, від несанкціонованого доступу та крадіжок. Крім того, система повинна мати механізми резервного копіювання даних та відновлення у разі виникнення непередбачуваних ситуацій, щоб мінімізувати ризики втрати даних та забезпечити безперервну роботу магазину.

З урахуванням зростання використання мобільних пристроїв для онлайн-покупок, інтерфейс інтернет-магазину повинен бути оптимізований для різних пристроїв та розмірів екранів. Це включає адаптивний дизайн, який автоматично підлаштовується під екран пристрою користувача, забезпечуючи зручний та ефективний досвід користувача незалежно від того, чи використовується смартфон, планшет чи комп'ютер. Також важливо враховувати швидкість завантаження сайту та зручність навігації, щоб користувачі могли швидко та легко знаходити потрібну інформацію та здійснювати покупки.

У підсумку, інформаційна система підтримки роботи інтернет-магазину чоловічого одягу повинна забезпечувати зручність та ефективність розміщення товарів, приймання та обробки замовлень, управління клієнтськими відносинами, аналітики та звітності, а також забезпечення безпеки інформації та оптимізації для

мобільних пристроїв. Це дозволяє створити надійну та конкурентоспроможну платформу для розвитку бізнесу та задоволення потреб клієнтів.

Окрім цього, важливим аспектом є інтеграція зі сторонніми платформами та майданчиками електронної комерції. Це може включати підключення до онлайн-майданчиків, таких як Amazon або eBay, а також інтеграцію з соціальними мережами та маркетплейсами для розширення аудиторії та збільшення обсягів продажів.

Необхідно також враховувати можливості персоналізації та рекомендаційного маркетингу. Інформаційна система може аналізувати покупки та перегляди клієнтів, щоб надавати індивідуально підібрані рекомендації щодо товарів, що можуть їх зацікавити. Це допомагає збільшити конверсію та підвищити середній чек.

Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу є важливим компонентом для успішного функціонування бізнесу в цій сфері. Аналіз інформаційного забезпечення предметної області дозволяє виявити ключові аспекти, які потрібно враховувати для ефективної підтримки роботи інтернет-магазину чоловічого одягу.

Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу є важливим компонентом для успішного функціонування бізнесу в цій сфері. Вона повинна забезпечувати зручний та ефективний спосіб розміщення товарів у каталозі, включаючи можливість додавання фотографій, описів, розмірів, цін та інших характеристик товарів. Важливо мати можливість категоризувати товари за типом, брендом, колекцією тощо, щоб полегшити клієнтам пошук необхідних речей. Крім того, система повинна мати функціонал для приймання та обробки замовлень, що включає формування кошика покупок, обробку платежів та зв'язок з платіжними шлюзами для забезпечення безпечних та зручних транзакцій.

Для підтримки взаємодії з клієнтами необхідно мати систему управління клієнтськими відносинами (CRM), яка дозволяє збирати та аналізувати дані про клієнтів, їхні покупки та взаємодію з магазином. Це допомагає підтримувати

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

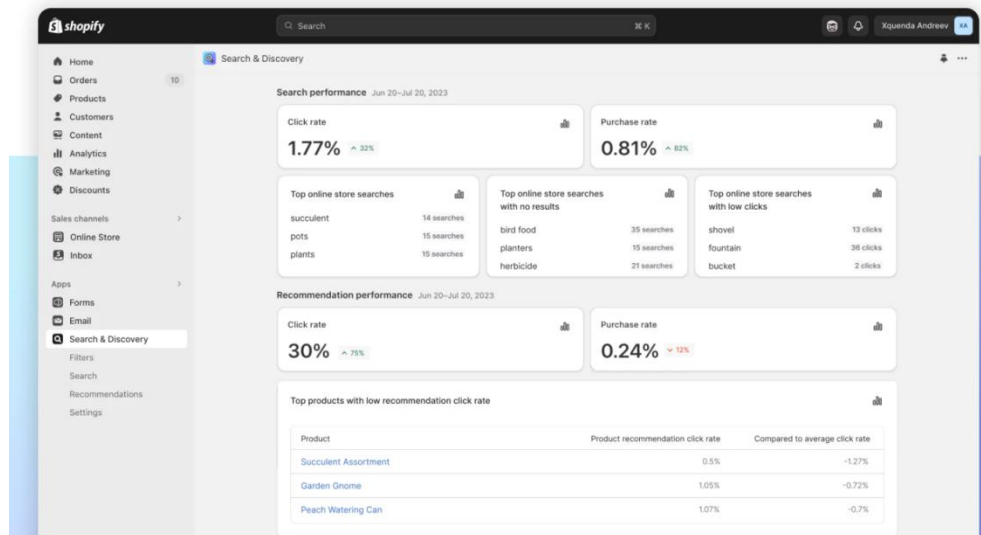
довгострокові відносини з клієнтами, надавати персоналізовані пропозиції та забезпечувати високий рівень обслуговування. Важливою складовою інформаційної системи є можливість ведення аналізу даних про продажі, відвідуваність сайту, конверсію тощо. Це допомагає розуміти ефективність бізнесу, виявляти тренди та приймати обґрунтовані рішення щодо стратегії розвитку.

Забезпечення захисту конфіденційної інформації клієнтів та даних про платежі є критично важливим аспектом. Інформаційна система повинна мати вбудовані заходи безпеки, такі як шифрування даних, захист від кібератак та вразливостей, щоб забезпечити захист конфіденційної інформації клієнтів та даних про платежі. Враховуючи зростання використання мобільних пристроїв для онлайн-покупок, інтерфейс магазину повинен бути оптимізований для різних пристроїв та розмірів екранів. Це забезпечить зручний та ефективний досвід користувача незалежно від того, чи користується клієнт смартфоном, планшетом чи комп'ютером. Оптимізація включає адаптивний дизайн, швидкість завантаження сайту та зручність навігації, що дозволить клієнтам швидко та легко знаходити необхідну інформацію та здійснювати покупки.

Окрім цього сьогоднішній день існує багато сучасних засобів для створення інформаційних систем в предметній області інтернет-магазинів чоловічого одягу. Ось деякі з найпопулярніших та ефективних засобів:

Magento(<https://account.magento.com/customer/account/login>): Magento є однією з найпоширеніших платформ для створення інтернет-магазинів. Вона має велику спільноту розробників та багатий набір функцій, включаючи розширення для управління товарами, замовленнями, платежами та доставкою. Magento також має потужну систему аналітики та звітності.

Shopify (<https://www.shopify.com/>): Shopify - це хмарна платформа для створення інтернет-магазинів, яка дозволяє швидко запускати та керувати електронним бізнесом. Вона має простий у використанні інтерфейс та багато готових шаблонів дизайну, а також інструменти для управління товарами, замовленнями та платежами (рисунок 1.5):



Рисинок 1.5 – Керування інтернет магазином за допомогою сервісу Shopify

WooCommerce (<https://woocommerce.com/>): WooCommerce - це безкоштовний плагін для WordPress, який перетворює ваш сайт на повноцінний інтернет-магазин. Він має велику кількість розширень та тем, які дозволяють налаштовувати магазин під індивідуальні потреби.

BigCommerce(<https://www.bigcommerce.com/>): BigCommerce - це інша популярна хмарна платформа для створення інтернет-магазинів, яка має великий набір інструментів для управління бізнесом, включаючи замовлення, оплату, доставку та аналітику.

OpenCart(<https://opencart.ua/>): OpenCart - це відкрита система управління контентом (CMS), спеціально призначена для створення інтернет-магазинів. Вона має простий у використанні інтерфейс та багато функцій для налаштування магазину.

Ці засоби надають широкий набір функцій і можливостей для створення, налаштування та управління інтернет-магазином чоловічого одягу, включаючи управління товарами, замовленнями, платежами, доставкою та аналітикою. Вибір конкретного інструменту залежить від потреб вашого бізнесу, ресурсів та експертизи управління.

Зм.	Арк.	№ докум.	Підпис	Дата

Окрім цього, інформаційні системи в області інтернет-магазинів одягу можуть бути створені за допомогою різних мов програмування та технологій. Ось деякі з найпоширеніших з них:

PHP: PHP є однією з найпопулярніших мов програмування для створення інтернет-магазинів. Вона часто використовується у поєднанні з веб-фреймворками, такими як Laravel або Symfony, для швидкого та ефективного розроблення інтерактивних веб-додатків.

JavaScript (зокрема фронтендні фреймворки): JavaScript використовується для створення динамічних інтерфейсів користувача на стороні клієнта. Фронтендні фреймворки, такі як React, Vue.js або Angular, дозволяють створювати складні та відзвучиві веб-інтерфейси для інтернет-магазинів. (<https://www.w3schools.com/js/>)

HTML/CSS: HTML та CSS використовуються для створення структури та візуального оформлення веб-сторінок відповідно. Вони є основою будь-якого веб-сайту, включаючи інтернет-магазини, і використовуються разом з іншими технологіями для створення користувацького інтерфейсу та його дизайну[22].

SQL (Structured Query Language): SQL використовується для роботи з базами даних. Багато інтернет-магазинів використовують реляційні бази даних, такі як MySQL або PostgreSQL, для зберігання та управління даними про товари, замовлення, клієнтів тощо.

Окремо хочеться зазначити мову Python, оскільки дана мова програмування за останні кілька років здобула популярність серед веб-розробників, завдяки своїм потужним фреймворкам, зручному синтаксису та широким можливостям.

Ось декілька популярних інтернет-магазинів, створених за допомогою Python, включають:

Instagram(<https://www.instagram.com/>):

Хоча Instagram не є прямою торговельною платформою, він надає можливість підприємствам продавати свої товари через інтегровані рекламні інструменти. Instagram використовує Django, популярний веб-фреймворк Python, для свого веб-застосунку.

Reddit Gifts(<https://www.reddit.com/r/Gifts/>): Reddit Gifts - це платформа для обміну подарунками, де користувачі можуть створювати, відправляти та отримувати подарунки від інших користувачів. Цей сайт також побудований на Django.

Pinterest(<https://ru.pinterest.com/>): Pinterest, популярний соціальний медіа-сайт для зберігання та обміну ідеями, також використовує Python у своєму стеку технологій для створення та підтримки своєї платформи.

Dropbox(https://www.dropbox.com/uk_UA/): Хоча Dropbox - це не інтернет-магазин в звичайному розумінні, він використовується для зберігання та обміну файлами, у тому числі для комерційних цілей. Багато з його функцій інтегровано за допомогою Python.

Ці інтернет-платформи побудовані на Python, а саме на його фреймворках, таких як Django або Flask, які надають потужні можливості для розробки веб-додатків швидко та ефективно

1.3 Постановка задачі

Розробити інформаційну систему підтримки роботи інтернет-магазину чоловічого одягу

Основні завдання:

1. Розробити функціонал для реєстрації та авторизації користувачів, включаючи систему керування ролями (рисунок 1.6).
2. Створити інтерфейс для перегляду каталогу товарів з можливістю пошуку, фільтрації за категоріями та сортуванням за різними параметрами.
3. Реалізувати можливість додавання товарів до кошика, редагування та видалення з кошика, а також оформлення замовлення з вибором методу оплати та доставки (рисунок 1.7).

2. Безпечне зберігання паролів: Зберігання паролів користувачів у вигляді хешів з соллю для запобігання простого витягування оригінальних паролів у разі витоку бази даних.

3. Захист від SQL-ін'єкцій: Використання параметризованих запитів та інших заходів для запобігання SQL-ін'єкціям, які можуть призвести до несанкціонованого доступу до бази даних. (рисунок 1.8).

4. Захист від міжсайтового скриптіngu (XSS): Фільтрація та екранування вхідних даних перед відображенням на веб-сторінках для запобігання XSS-атак, які можуть використовуватися для отримання доступу до даних користувачів. (рис.1.9)

5. Захист від міжсайтового підбору паролів (CSRF): Використання токенів CSRF та інших заходів для перевірки автентичності запитів і запобігання виконання небажаних дій від імені користувачів.

6. Моніторинг та журналювання подій: Ведення журналів подій для виявлення потенційних атак та невдалих спроб втручання, що дозволяє вчасно реагувати на можливі загрози безпеці.

7. Регулярні аудити безпеки: Проведення регулярних аудитів безпеки системи для виявлення і виправлення потенційних вразливостей та слабких місць в інфраструктурі.

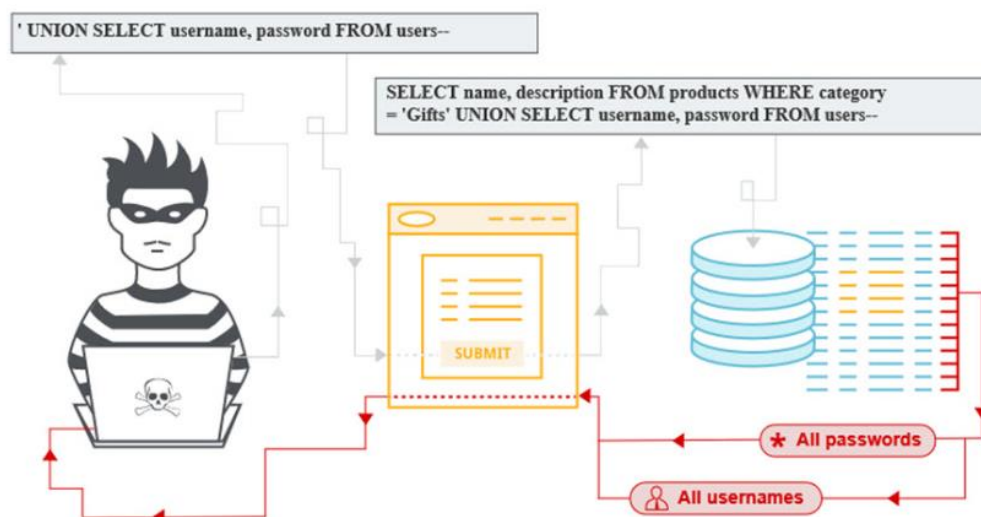


Рисунок 1.8 – Захист від SQL-ін'єкцій

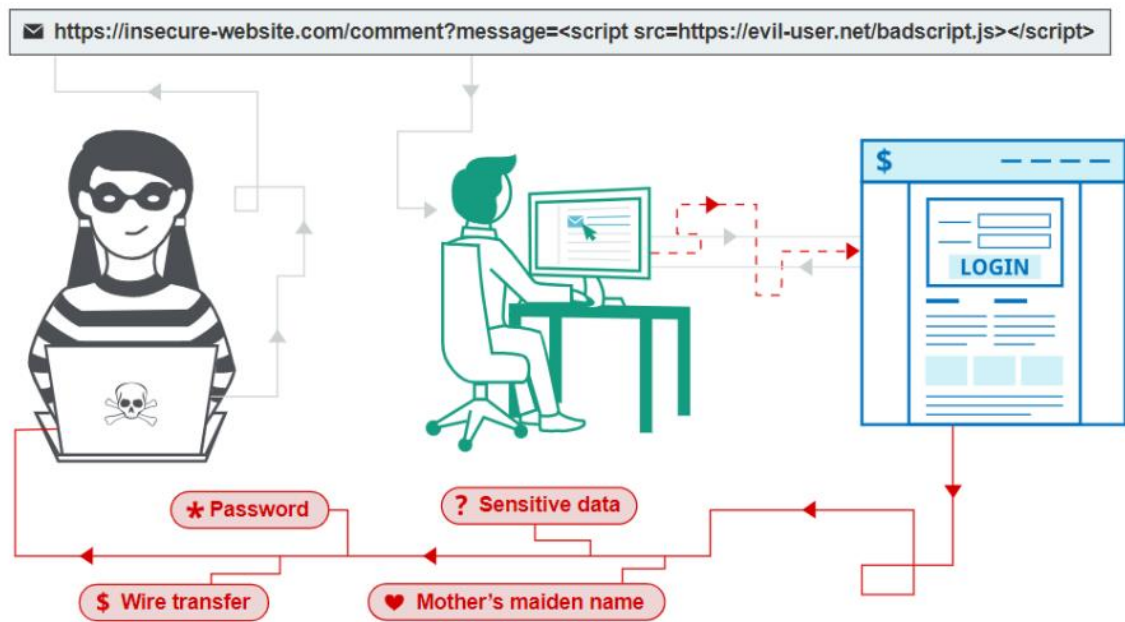


Рисунок 1.9 – Захист від міжсайтового скриптингу (XSS)

8. Оптимізувати інтерфейс для відображення на різних типах пристроїв (респонсивний дизайн) (рисунок 1.10) , забезпечивши зручну роботу на комп'ютерах, планшетах та смартфонах.

Очікуваний результат:

Розроблена та впроваджена Інформаційна система підтримки роботи Інтернет-магазину чоловічого одягу, яка надає користувачам зручний інтерфейс для перегляду, вибору та покупки товарів, а також адміністраторам - інструменти для ефективного управління магазином. Система повинна бути стабільною, безпечною та відповідати вимогам сучасних стандартів електронної комерції.

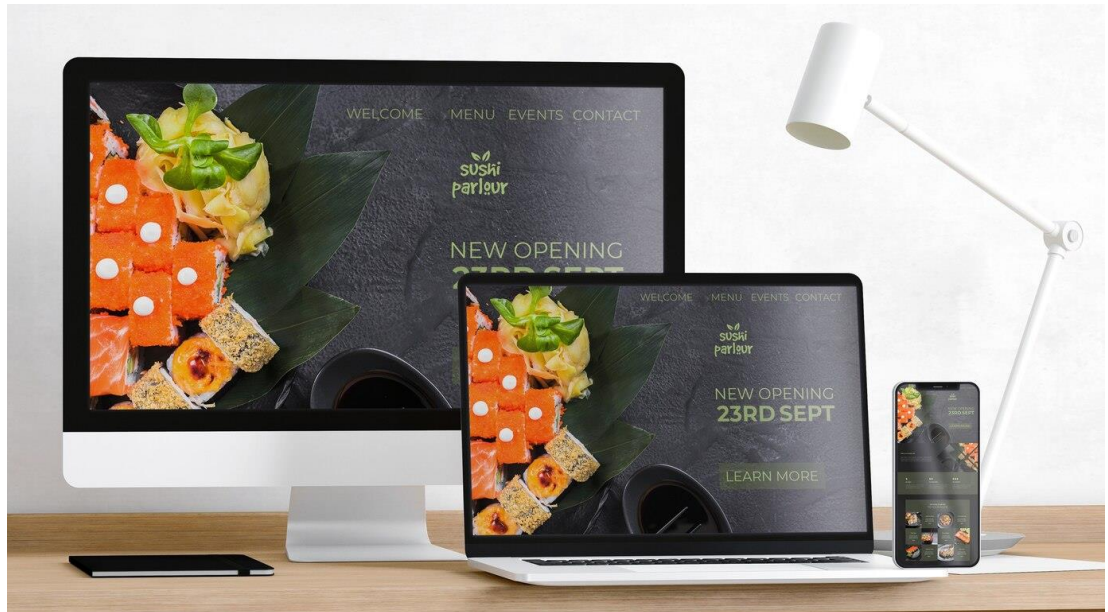


Рисунок 1.10 – Респонсивний дизайн [54]

2 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Комплексна інформаційна система для інтернет-магазину чоловічого одягу: функції та переваги

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу являє собою комплексне рішення, спрямоване на автоматизацію та оптимізацію всіх процесів, пов'язаних з управлінням та функціонуванням магазину в онлайн просторі. Основна мета такої системи – забезпечити зручну та ефективну роботу як для адміністраторів магазину, так і для його клієнтів.

Перш за все, інформаційна система надає можливість створення та керування каталогом товарів, де адміністратор може додавати нові товари, редагувати існуючі, встановлювати ціни, зображення та описи. Крім того, система дозволяє впорядковувати товари за категоріями, що полегшує навігацію для покупців. Для зручності покупців передбачено створення облікового запису, де вони можуть переглядати історію своїх покупок, керувати адресами доставки та здійснювати оплату онлайн. Інформаційна система забезпечує зберігання даних користувачів у захищеному середовищі, яке виключає несанкціонований доступ.

Для збільшення продажів та привернення уваги клієнтів система може включати інструменти електронного маркетингу, такі як розсилка електронних листів з акційними пропозиціями, створення персональних пропозицій на основі попередніх покупок або переглядів, інтеграція з соціальними медіа для просування товарів тощо. Управління замовленнями та доставкою також автоматизовано: адміністратор може відстежувати статус замовлення, призначати їх для доставки, відправляти сповіщення клієнтам про стан їхніх покупок. Інтеграція з транспортними службами дозволяє отримувати реальні дані про доставку та оцінювати час прибуття товару.

Безпека є одним з ключових аспектів інформаційної системи. Вона має високий рівень захисту, що включає шифрування даних, заходи аутентифікації та авторизації, а також постійний моніторинг потенційних загроз безпеці. Щоб забезпечити стабільну та безперебійну роботу магазину, система може бути розгорнута на надійних серверах з можливістю масштабування та резервним копіюванням даних для запобігання втраті інформації у разі аварійних ситуацій.

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

Рисунок 2.2 – Інтегровані аналітичні інструменти

2.2 Архітектура

Архітектура інформаційної системи для інтернет-магазину чоловічого одягу базується на розподіленій моделі, що включає в себе як фронтенд, так і бекенд компоненти. Бекенд реалізований за допомогою Python фреймворка Django (рисунок 2.3) , який забезпечує швидку розробку та підтримку веб-додатків [16]

Основною складовою бекенду є база даних, де зберігається вся інформація про товари, користувачів, замовлення та інші аспекти функціонування магазину. Django забезпечує ORM (Object-Relational Mapping), що дозволяє зручно взаємодіяти з базою даних і використовувати об'єктно-орієнтований підхід при роботі з даними [17].

Фронтенд реалізований з використанням CSS, JavaScript та HTML. Він відповідає за представлення інформації користувачам через веб-інтерфейс. Фронтенд включає в себе сторінки для перегляду каталогу товарів, процесу оформлення замовлення, реєстрації та авторизації користувачів, а також інші компоненти інтерфейсу, такі як кнопки, форми та меню [45].

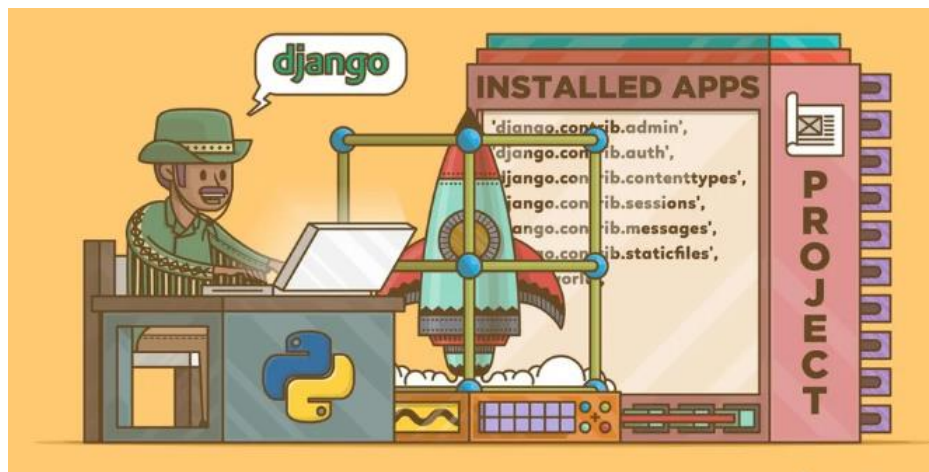


Рисунок 2.3 – Мова програмування Django [55]

Для забезпечення взаємодії між фронтендом і бекендом використовується механізм AJAX (Asynchronous JavaScript and XML), що дозволяє асинхронно взаємодіяти з сервером без необхідності перезавантаження сторінки.

Архітектура інформаційної системи інтернет-магазину чоловічого одягу повинна бути ретельно задокументованою для полегшення підтримки та подальшого розвитку. Документація має включати опис структури бази даних, API для взаємодії між фронтендом і бекендом, а також інструкції щодо розгортання та налаштування системи. Використання шаблонів проектування, таких як MVC (Model-View-Controller) або MVVM (Model-View-ViewModel), дозволяє відокремити логіку додатку від його представлення та взаємодії з даними, що спрощує розробку та підтримку.

Однією з ключових складових бекенду є система автоматизованої обробки замовлень. Ця система включає логіку обробки покупок, перевірку наявності товару на складі, розрахунок вартості замовлення, обробку оплати та формування замовлення для відправлення. Для забезпечення ефективної обробки великої кількості запитів використовується асинхронна обробка даних та завдань за допомогою інструментів на кшталт Celery або Redis. Це дозволяє обробляти запити паралельно та зменшує час очікування для користувачів.

Для забезпечення стабільності та відновлення роботи системи після можливих відмов впроваджується моніторинг системи та резервне копіювання даних. Моніторинг дозволяє вчасно виявляти проблеми та оперативно їх вирішувати, тоді як резервне копіювання забезпечує можливість відновлення даних у разі їх втрати або пошкодження.

На фронтенді, крім базових HTML, CSS та JavaScript, використовуються інші технології для покращення користувацького досвіду, такі як фреймворки для реактивного програмування, анімації та взаємодії з користувачем. Окрім стандартного інтерфейсу веб-додатку, можуть бути розроблені мобільні додатки для iOS та Android, які надають додаткові можливості для користувачів, такі як

сповіщення про статус замовлення, перегляд акцій та зручний доступ до каталогу товарів.

При розробці архітектури системи важливо враховувати міжнародність та локалізацію, забезпечуючи можливість відображення інформації на різних мовах та адаптацію до різних культурних особливостей. Загалом, архітектура інформаційної системи інтернет-магазину чоловічого одягу повинна бути добре збалансованою та гнучкою, щоб забезпечити ефективне функціонування та масштабування системи у відповідності до зростаючих потреб бізнесу та користувачів.

2.2 Функційні вимоги до ІС

Інформаційна система для сайту чоловічого одягу розробляється на мові Python з використанням фреймворку Django для забезпечення ефективного та стабільного функціонування. Для досягнення максимальної функціональності та задоволення потреб користувачів система повинна включати ряд ключових функцій [38].

1. Реєстрація та аутентифікація користувачів (рисунок 2.4): система має надавати можливість реєстрації нових користувачів та аутентифікації існуючих через імейл або соціальні мережі.

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

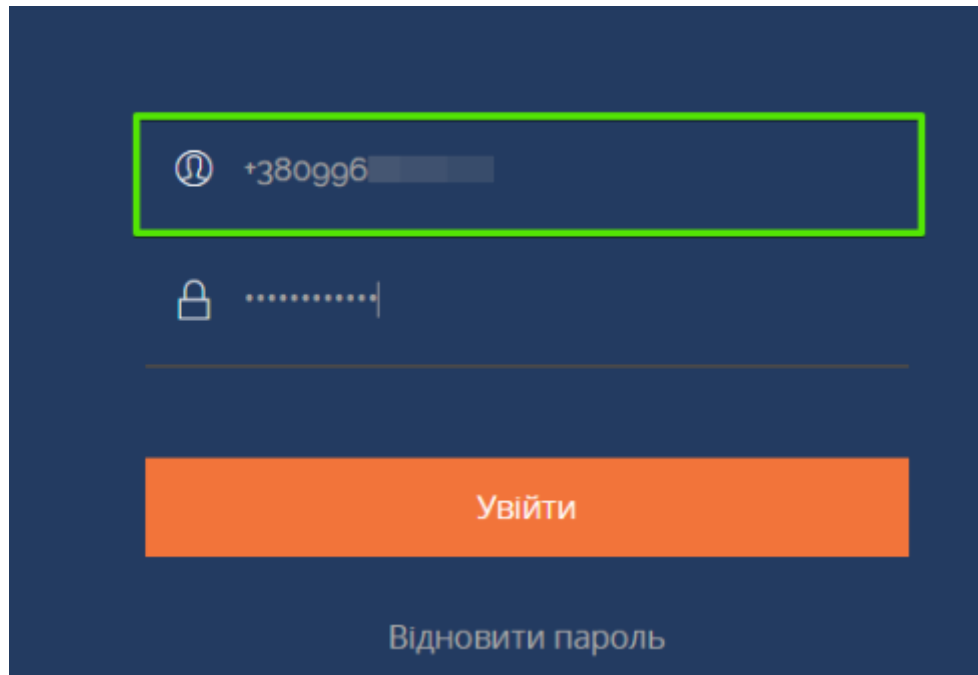


Рисунок 2.4 – Реєстрація та аутентифікація користувачів

2. Профіль користувача: кожен користувач має мати особистий профіль, де він може змінювати особисту інформацію, адресу доставки та переглядати свої замовлення.

3. Пошук та фільтрація товарів: система повинна забезпечувати зручний пошук та фільтрацію товарів за різними параметрами, такими як розмір, категорія, колір, ціновий діапазон тощо.

4. Каталог товарів (рисунок 2.5): наявність повного та актуального каталогу товарів з детальним описом, фотографіями та цінами для зручного перегляду користувачами.

5. Кошик покупок: можливість додавання товарів у кошик, перегляд вмісту кошика, зміна кількості товарів та оформлення замовлення.

6. Оплата та доставка: інтеграція з різними платіжними системами для прийому оплати за замовлення, а також вибір різних методів доставки.

7. Система оцінок та відгуків: можливість користувачів залишати відгуки та оцінки товарів, що допомагає іншим користувачам зробити правильний вибір.

2. Надійність: мінімізація відмов системи та забезпечення стабільної роботи, а також відновлення системи після аварійних ситуацій.

3. Безпека: захист конфіденційної інформації користувачів, включаючи особисті дані та платіжну інформацію, за допомогою шифрування даних та заходів аутентифікації та авторизації.

4. Сумісність: забезпечення сумісності системи з різними веб-браузерами (Chrome, Firefox, Safari, Edge тощо) та операційними системами (Windows, MacOS, Linux) для максимальної доступності користувачам.

5. Масштабованість: можливість горизонтального та вертикального масштабування системи для підтримки зростаючої кількості користувачів та обсягу даних.

6. Доступність: забезпечення доступності системи для людей з обмеженими можливостями за допомогою відповідних технологічних рішень та відповідності веб-стандартам доступності (наприклад, WCAG).

7. Локалізація та міжнародні стандарти: підтримка локалізації для різних мов та культур, а також відповідність міжнародним стандартам та законодавству щодо електронної торгівлі.

8. Ефективне управління пам'яттю та ресурсами: мінімізація використання пам'яті та ресурсів сервера для оптимізації продуктивності та зниження витрат.

9. Документація та підтримка: надання зрозумілої та повної документації для розробників та адміністраторів системи, а також надання ефективної підтримки користувачам.

10. Ергономіка та дизайн: забезпечення зручного та інтуїтивно зрозумілого інтерфейсу користувача з урахуванням принципів ергономіки та модерного дизайну.

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4 Користувацькі вимоги

Зважаючи на індивідуальні потреби та очікування користувачів, сайт чоловічого одягу повинен бути спроектований з урахуванням таких аспектів, як інтуїтивність використання, естетика та зручність. Користувачі очікують, що при перебуванні на сайті вони будуть відчувати зручність і комфорт, що сприятиме приємному досвіду покупок.

1. Зручний пошук продукції: користувачі хочуть легко та швидко знаходити потрібні товари, використовуючи пошук за категоріями, розміром, кольором тощо(рисунк 2.8).

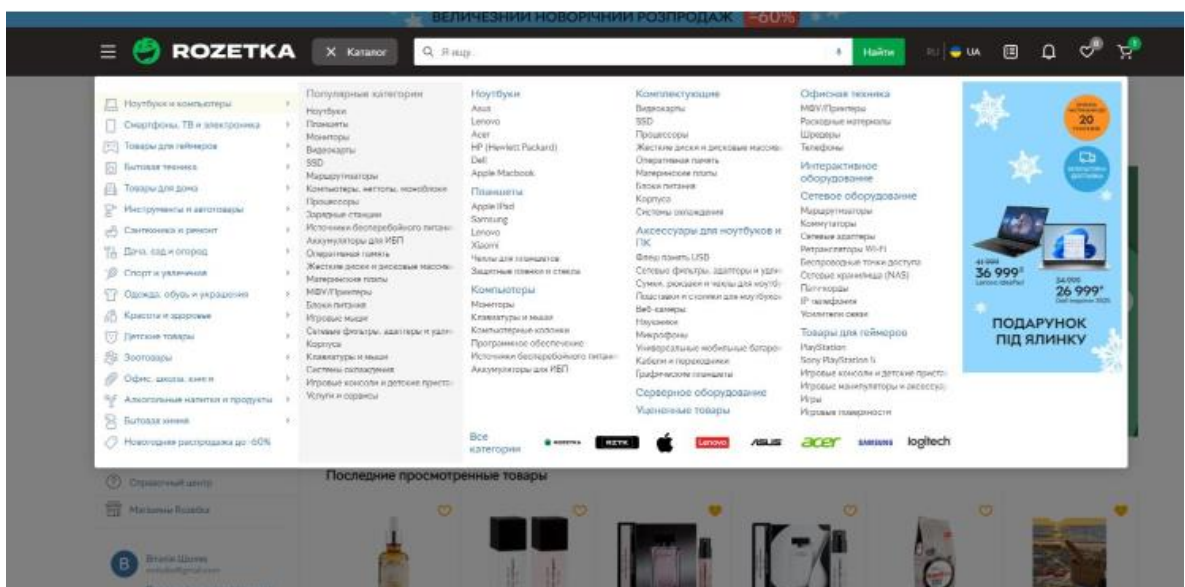


Рисунок 2.8 – Зручний пошук продукції [57]

2. Привабливий та зручний дизайн: користувачі очікують на привабливий та сучасний дизайн сайту, який не лише привертає увагу, а й забезпечує зручну навігацію.

3. Інформативний контент: потенційні покупці хочуть отримати вичерпну інформацію про кожен товар, включаючи фотографії з різних ракурсів, опис, склад матеріалу, рекомендації по догляду тощо.

4. Легкий процес покупки: користувачі очікують на зручну систему додавання товарів до кошика, швидку обробку замовлення та простий процес оплати.

5. Мобільна сумісність: враховуючи зростання використання мобільних пристроїв, сайт повинен бути адаптивним та добре відображатися на різних розмірах екрану.

6. Безпека та конфіденційність: користувачі мають право на захист своїх особистих даних та платіжної інформації, тому очікують, що сайт забезпечить високий рівень безпеки.

7. Система зворотнього зв'язку: наявність форми зворотнього зв'язку (рисунок 2.9) або можливості звернутися до служби підтримки для вирішення питань чи непорозумінь.

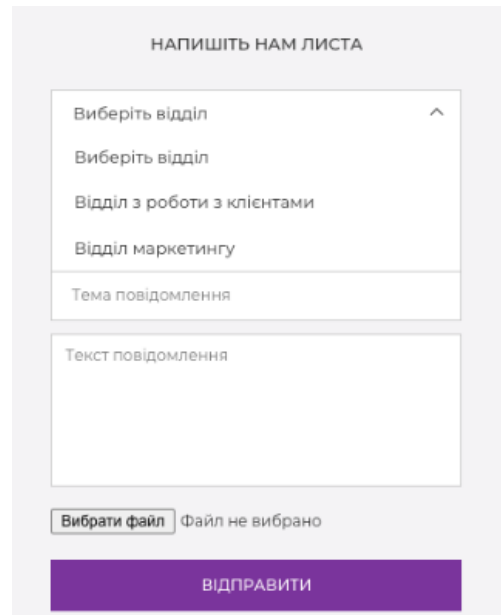


Рисунок 2.9 – Форма зворотнього зв'язку

8. Зручний процес повернення та обміну товару: важливо мати можливість зручного повернення або обміну товару у випадку, якщо він не відповідає очікуванням.

9. Ефективна клієнтська підтримка: користувачі цінують швидку та ефективну підтримку у випадку виникнення проблем або питання

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПІДТРИМКИ ІНТЕРНЕТ – МАГАЗИНУ ЧОЛОВІЧОГО ОДЯГУ

3.1 Опис реалізації програмного забезпечення інформаційної системи підтримки інтернет магазину чоловічого одягу

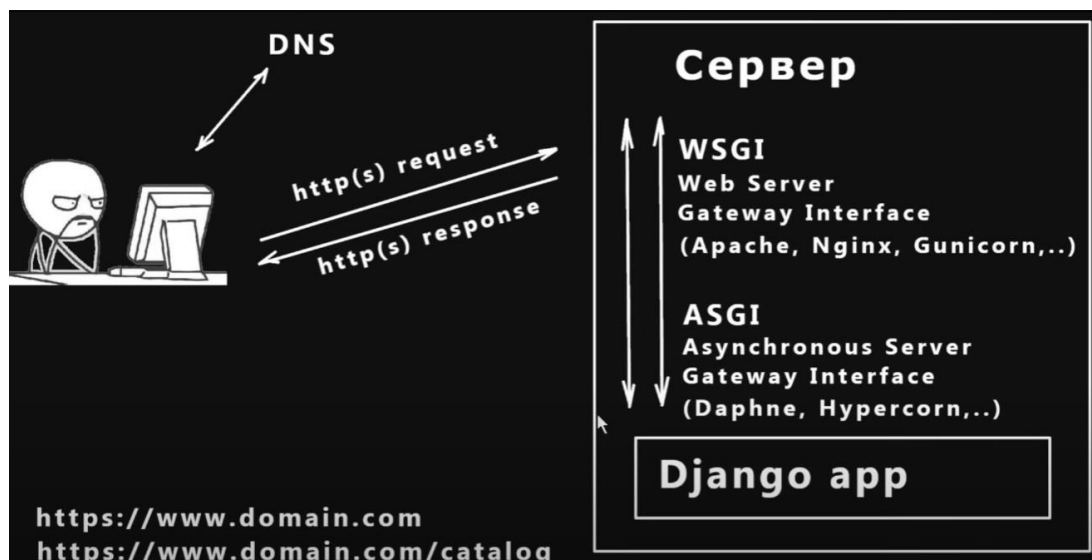


Рисунок 3.1 – Схема роботи Backend

На схемі (рисунок 3.1) показано, як працює інтернет-магазин, реалізований на Django. Користувач на своєму комп'ютері вводить URL-адресу інтернет-магазину, наприклад, <https://www.domain.com> або <https://www.domain.com/catalog>, і цей запит надсилається через Інтернет до DNS-сервера. DNS-сервер відповідає за перетворення доменного імені у IP-адресу сервера, на якому розміщено інтернет-магазин, після чого запит переспрямовується на відповідний сервер [14].

Сервер приймає HTTP(S) запит від клієнта і, залежно від конфігурації, використовує WSGI або ASGI для його обробки. WSGI (Web Server Gateway Interface) використовується для синхронних запитів і зазвичай застосовується з веб-серверами такими, як Apache, Nginx, Gunicorn, тоді як ASGI (Asynchronous Server Gateway Interface) підтримує асинхронні запити і працює з серверними інтерфейсами, такими як Daphne, Hypercorn. Django-додаток обробляє отриманий

запит, взаємодіє з базою даних, бізнес-логікою, формує відповідь (наприклад, HTML-сторінку) і повертає її назад через сервер клієнту. Відповідь від сервера повертається клієнту у вигляді HTTP(S) відповіді, і користувач бачить результат свого запиту в браузері, веб-сторінку інтернет-магазину.

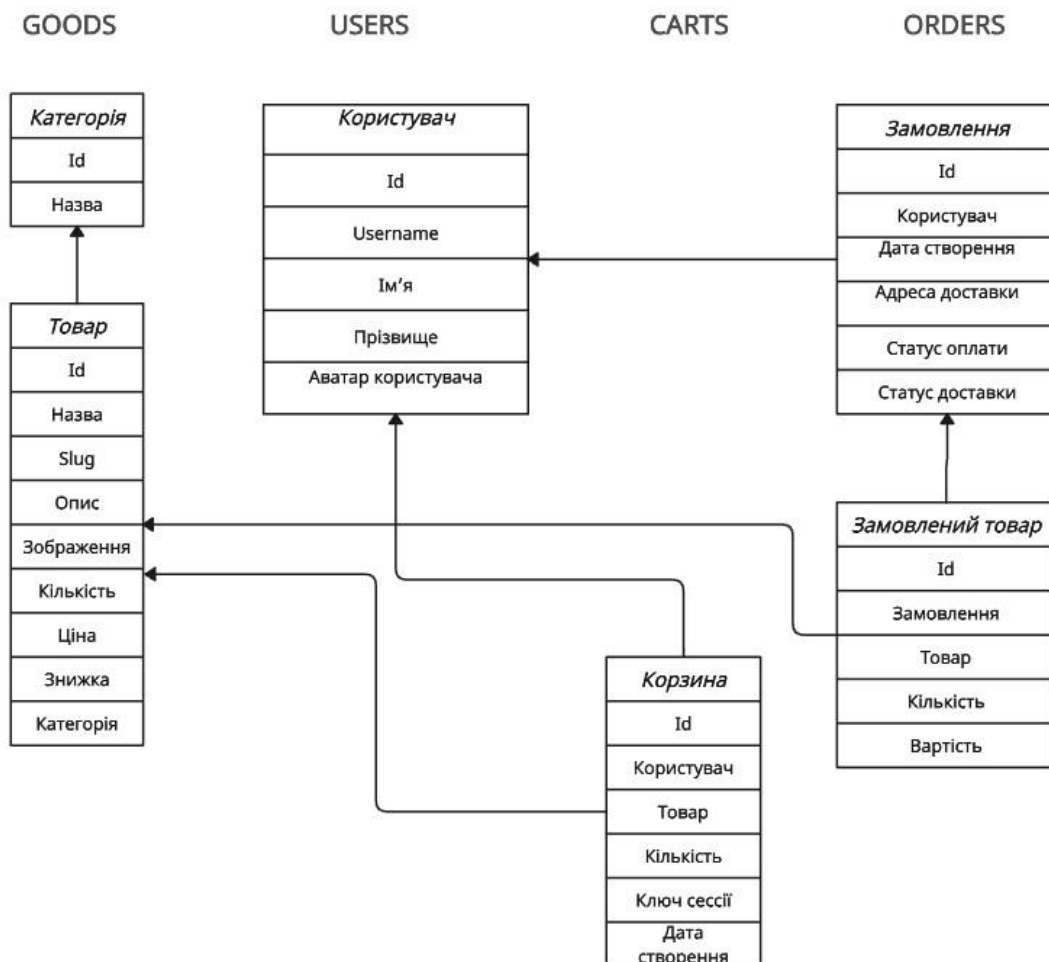


Рисунок 3.2 – Структурна схема інформаційної системи

На схемі (рисунок 3.2) зображена база даних інтернет-магазину, яка складається з чотирьох основних частин: товари, користувачі, кошики та замовлення. Товари організовані в категорії, кожна з яких має унікальний ідентифікатор і назву. Товари містять інформацію про себе, таку як назва, короткий текстовий ідентифікатор для URL, опис, зображення, кількість, ціна, знижка та

посилання на відповідну категорію. Користувачі мають унікальний ідентифікатор, ім'я користувача, реальні ім'я та прізвище, а також посилання на аватар користувача. Кошик містить товари, які користувач хоче придбати, і включає унікальний ідентифікатор, посилання на користувача, товар, кількість товару, ідентифікатор сесії для незареєстрованих користувачів та дату створення кошика. Замовлення зберігають інформацію про покупку, маючи унікальний ідентифікатор, посилання на користувача, який зробив замовлення, дату створення, адресу доставки, статус оплати та статус доставки. Крім того, кожне замовлення включає замовлені товари з інформацією про відповідне замовлення, товар, кількість та вартість. Це дозволяє відстежувати, які товари були замовлені і в якій кількості, а також зберігати інформацію про користувачів і стан їхніх замовлень.

3.2 Архітектура та налаштування середовища розробки

У проєкті використовується Django версії 4.2 та Python версії 3.11.8. Вибір цих версій був здійснений з урахуванням їхньої стабільності, підтримки та нововведень, які вони пропонують [35].

Для забезпечення зручного та ефективного розробницького середовища налаштування середовища включає в себе використання віртуальних середовищ Python, а саме «venv», що дозволяє ізолювати залежності проєкту від інших програмних пакетів, зберігаючи чистоту середовища.

Для керування залежностями та конфігурацією проєкту використовується «pip.», стандартний менеджер пакетів Python, який дозволяє легко встановлювати, оновлювати та видаляти необхідні пакети.

Крім того, для розробки та налагодження використовуються інтегровані середовища розробки (IDE) або текстові редактори з підтримкою Django, такі як PyCharm, VS Code, Sublime Text чи інші, що спрощують написання коду, налагодження та управління проєктом.

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 33
Зм.	Арк.	№ докум.	Підпис	Дата		

З метою оптимізації та підтримки найкращих практик у розробці веб-додатків, також використовуються додаткові інструменти для тестування, автоматизації процесів розробки та контролю версій коду, а саме Git, так як це необхідно для розгортання та управління інфраструктурою проекту.

Всі ці елементи разом дозволяють забезпечити ефективну розробку, тестування та підтримку проекту на базі Django 4.2 та Python 3.11.8, забезпечуючи високу якість та надійність веб-додатку.

Також у процесі розробки інформаційної системи був впроваджений Git client, який є незамінним інструментом для керування версіями коду та спільної роботи розробників. Git дозволяє зберігати весь історичний контекст змін в коді, що робить його ідеальним для колаборативної розробки, особливо коли розробники працюють в команді або над різними функціональними аспектами проекту.

Одним з основних переваг Git є його розподілена структура, що дозволяє кожному розробнику мати повну копію репозиторію з історією змін. Це робить Git дуже гнучким і надійним для спільної роботи, оскільки зміни можуть бути вноситись локально та потім синхронізуватись з центральним репозиторієм.

Крім того, Git надає можливість створювати гілки (branches), що дозволяє розробникам відокремлювати різні функціональні можливості чи експериментувати з кодом без впливу на основну гілку розробки. Це сприяє ефективному керуванню розробкою нових функцій та виправленнями помилок, оскільки кожна гілка може мати свою власну історію змін.

Однією з ключових можливостей Git є також можливість легко відновлювати попередні версії коду або відкочуватись до попередніх станів проекту (рисунок 3.3) та (рисунок 3.4), що робить його надзвичайно корисним інструментом для управління ризиками та підтримки стабільності програмного продукту.

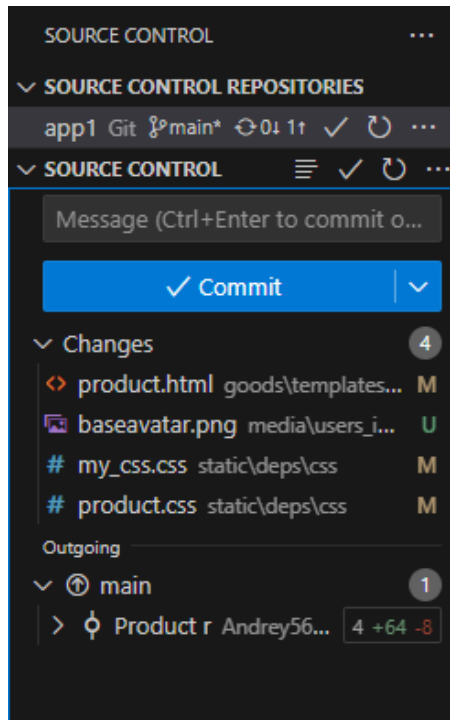


Рисунок 3.3 – Відображення змін у проекті через commit

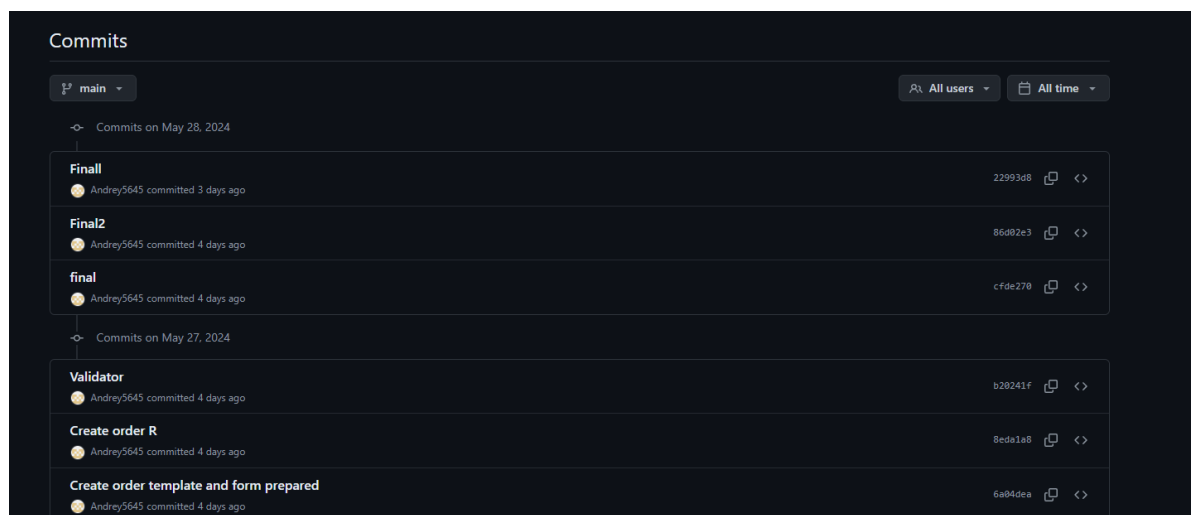


Рисунок 3.4 – Збережені «Комміти» у GitHub

3.3 Створення головного додатку “Manage.py”

Файл manage.py є важливим інструментом для виконання різних адміністративних завдань у Django проекті через командний рядок. Його основна функція полягає в запуску різних управлінських команд Django, таких як запуск

локального сервера, створення бази даних, міграції моделей, керування суперкористувачами та іншими адміністративними завданнями.

У файлі `manage.py` визначена функція `main()`, яка налаштовує потрібні середовищні змінні, такі як `DJANGO_SETTINGS_MODULE`, що вказує Django на поточні налаштування проекту. Він також імпортує необхідну функцію `execute_from_command_line` з пакету `django.core.management`, яка реалізує виконання команд з командного рядка.

Файл містить обробку винятків для випадку, якщо Django не може бути імпортований, що може виникнути через неправильну установку Django або неактивоване віртуальне середовище Python.

Останній рядок у файлі `manage.py`, `if __name__ == '__main__':`, гарантує, що функція `main()` буде викликана тільки при прямому запуску файлу, що є стандартним підходом для запуску скриптів у Python.

Всі ці елементи разом роблять файл `manage.py` невід'ємною частиною розробки і управління Django додатками, забезпечуючи зручний та ефективний інтерфейс для адміністративних завдань.

3.4 Створення додатку «Main»

Проект "main" включає в себе ключові компоненти для створення веб-додатку з магазину одягу. Файл `views.py` містить два види: `"index"` та `"about"`. Вид `"index"` відповідає за головну сторінку магазину (рисунок 3.5), де відображається привітальний заголовок та функціональні блоки з інформацією про якість продукції та інноваційні технології. Кожен блок містить відповідні зображення товарів та короткий опис. Вид `"about"` створений для сторінки "Про нас", де представлено докладний опис магазину, його пропозицій та контактних даних.

Файл `urls.py` містить конфігурацію маршрутів для обробки запитів користувачів. Цей файл визначає шляхи, за якими можна звертатися до різних

сторінок додатку, таких як головна сторінка ("/main") та сторінка "Про нас" ("/main/about").

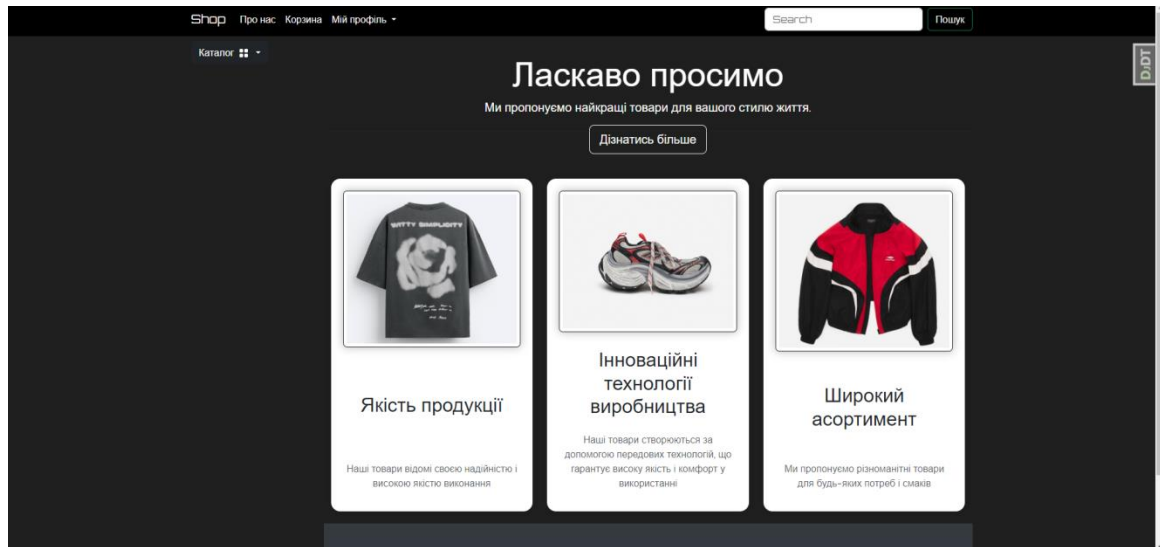


Рисунок 3.5 – Головна сторінка інтернет магазину

Крім того, в папці також зберігається HTML-шаблон "about.html", який використовує базовий шаблон "base.html". На сторінці "Про нас"(рисунок 3.6) відображається детальний опис магазину, його асортименту та послуг, а також контактна інформація для зв'язку з клієнтами. Файл "index.html" використовується для головної сторінки магазину, де розміщені заголовки, опис продуктів та функціональні блоки з фотографіями товарів.

Клас конфігурації "MainConfig" у файлі "apps.py" налаштовує основні параметри додатку "main", включаючи використання автоматичного поля для моделей бази даних.

Не менш важливим є надання користувачам зручного інтерфейсу, що відповідає сучасним веб-стандартам та вимогам доступності. Розробка дизайну інтерфейсу користувача, який є інтуїтивно зрозумілим та привабливим, сприяє підвищенню задоволення користувачів від взаємодії з додатком.

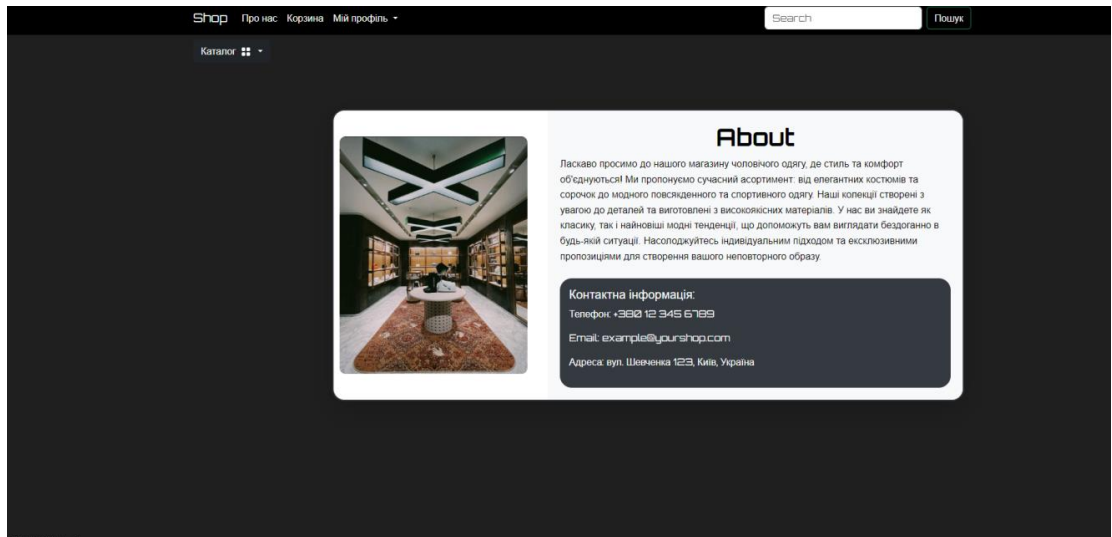


Рисунок 3.6 – Сторінка «Про нас»

3.5 Створення додатку “Goods”

У додатку "Goods" використовується Django для організації каталогу товарів із можливістю детального перегляду кожного товару. Основними компонентами цього додатку є файли models.py, views.py, urls.py, а також шаблони HTML для відображення сторінок (рисунок 3.7)

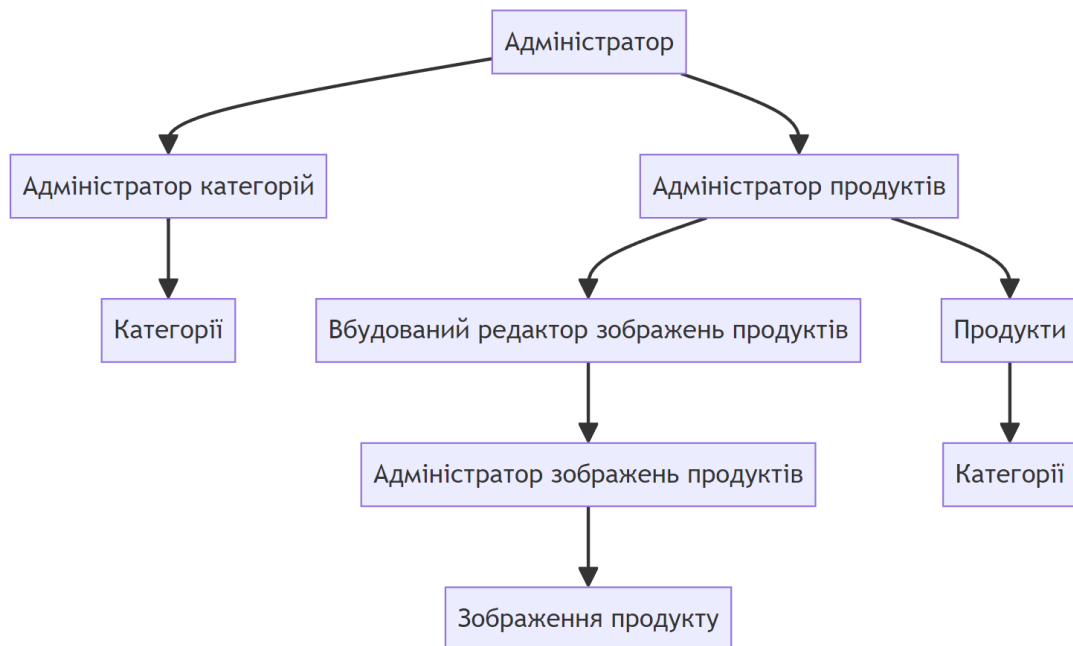


Рисунок 3.7 – Структурна схема роботи додатку «Goods»

Файл `models.py` в додатку "Goods" визначає структуру бази даних для категорій продуктів та самих продуктів. Модель `Categories` (рисунок 3.8) містить поля `name`, що зберігає назву категорії, і `slug`, яке генерує унікальний URL-ідентифікатор на основі назви категорії.

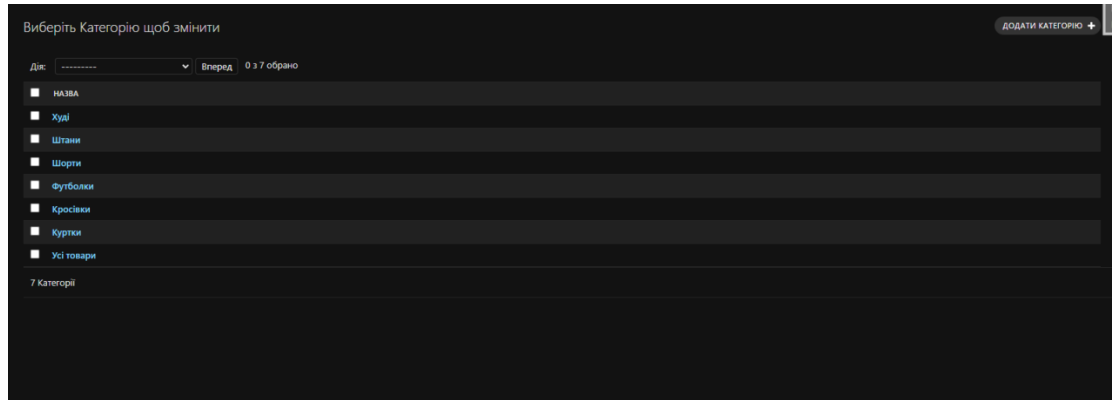


Рисунок 3.8 – Відображення таблиці категорій в інформаційній системі

Модель `Products` (рисунок 3.9) та (рисунок 3.10) включає поля для назви, опису, ціни, знижки, кількості та посилається на категорію за допомогою зовнішнього ключа. Кожен продукт також має можливість мати кілька зображень завдяки моделі `ProductImage`, яка зв'язана з продуктом через зовнішній ключ і містить поле для зображення товару. Взаємодія між моделями дозволяє структурувати і організовувати товари в каталозі, забезпечуючи зручну навігацію та пошук користувачам.

Файл `views.py` в додатку "Goods" містить логіку обробки запитів користувачів для каталогу товарів та окремих продуктів. У функції `catalog` обробляються запити, пов'язані з відображенням списку товарів за певною категорією або всіх товарів взагалі. Запити можуть містити параметри, такі як сторінка, фільтр за знижкою та сортування. Функція використовує об'єкт `Paginator` для розділення товарів на сторінки та підготовки їх для відображення. У функції `product` обробляється запит на конкретний продукт за його унікальним ідентифікатором. Для кожної сторінки використовуються відповідні шаблони для

візуалізації даних користувачам, що дозволяє забезпечити зручне навігаційне середовище для покупців

НАЗВА	КІЛЬКІСТЬ	ЦІНА	ЗНИЖКА В %	КАТЕГОРІЯ
ФУТБОЛКА З НАПИСОМ	39	949,00	10,00	Футболки
ФУТБОЛКА З ПРИНТОМ	39	1099,00	0,00	Футболки
ХУДІ ВОХУ-FIT	40	1699,00	3,00	Худі
СКЕЙТЕРСЬКІ КРОСИВКИ ЗІ ШНУРКАМИ-МОТУЗКАМИ	20	2399,00	10,00	Кросівки
ШОРТИ BAGGY-FIT	10	1449,00	5,00	Шорти
СИРЕ ХУДІ	40	1699,00	20,00	Худі
ХУДІ З НАШИВКАМИ	150	1699,00	20,00	Худі
ДЖИНСИ WIDE-LEG	22	1899,00	30,00	Штани
ДЖИНСОВА КУРТКА	139	1699,00	10,00	Куртки
БАЗОВА ФУТБОЛКА	89	749,00	20,00	Футболки
ШОРТИ-ДЖОГЕРИ	50	1099,00	20,00	Шорти
СНИКЕРСИ З ТОВСТОЮ ПІДОШВОЮ	70	1699,00	20,00	Кросівки
СТЬОБАНА КУРТКА ЗІ ШТУЧНОЇ ШКИРИ	30	2899,00	0,00	Куртки

Рисунок 3.9 – Відображення таблиці товарів в інформаційній системі

Рисунок 3.10 – Вікно додавання товару в каталог через таблицю «Products»

Файл `urls.py` містить маршрутизацію URL для кожної сторінки додатку. Він визначає, які функції `views.py` відповідають за обробку кожного URL, що дозволяє користувачеві навігувати між сторінками каталогу і товарів.

Шаблони HTML, такі як catalog.html та product.html, використовуються для відображення каталогу товарів і окремих продуктів у веб-додатку на основі Django. Вони включають в себе HTML-структури разом з Django Template Language (DTL), яка дозволяє динамічно генерувати контент на сторінці.

У файлі catalog.html (рисунок 3.11) використовується функціональність фільтрації та сортування товарів за допомогою HTML-форми, яка включає в себе елементи вибору та прапорці для фільтрації за наявністю знижок та сортування за ціною (рисунок 3.11). Результати запитів відображаються у вигляді карток продуктів з використанням Bootstrap CSS для стилізації та розмітки [13].

Шаблон також містить логіку пагінації для зручності користувачів, які шукають багатосторінкові результати.

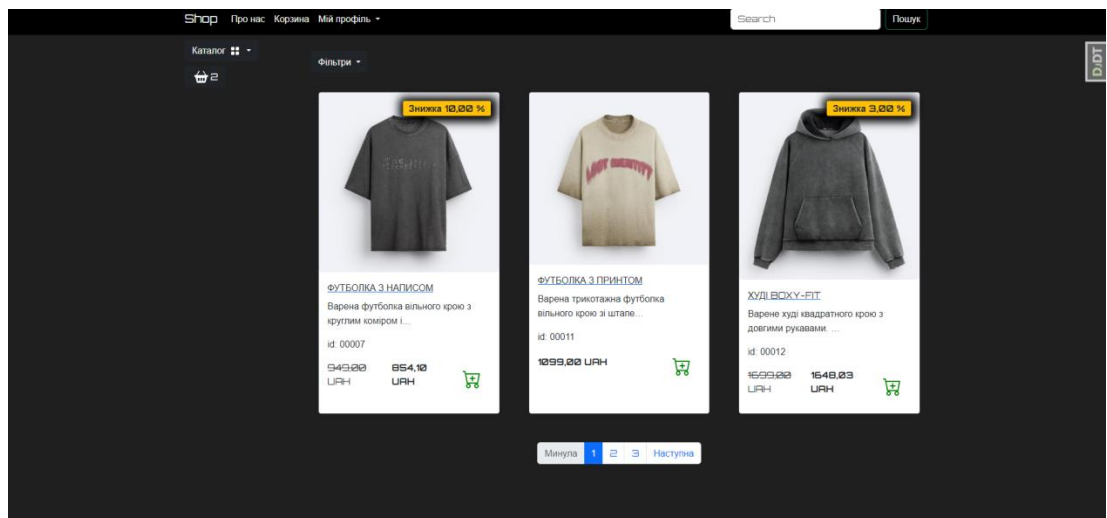


Рисунок 3.11 – Відображення каталогу товарів інформаційної системи

Файл product.html відображає окремий продукт разом з усією його інформацією, включаючи назву, опис, ціну та доступні зображення (рисунок 3.13). Кожен продукт може мати свої унікальні зображення, які показуються у вигляді слайдеру або галереї, щоб користувачі могли отримати повний огляд товару перед покупкою. Додаткові стилі та елементи керування, такі як кнопка "Додати до кошика", інтегруються для полегшення взаємодії з користувачем і забезпечення покупцям зручності. Обидва шаблони використовуються для створення приємного

інтерфейсу для користувачів, що дозволяє їм з легкістю оглядати та покупати продукти в інтернет-магазині.

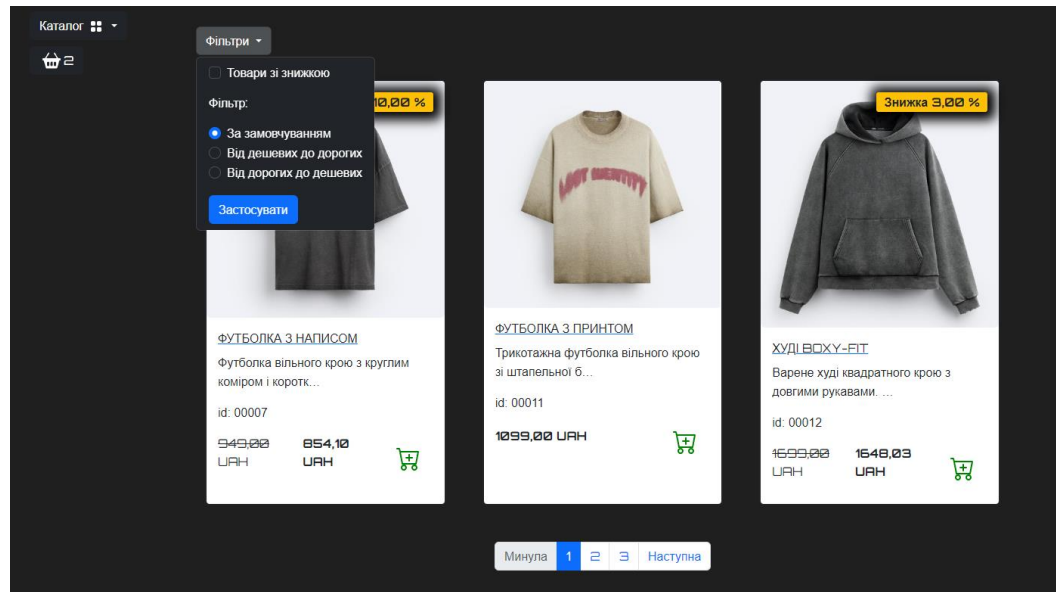


Рисунок 3.12 – Відображення функціоналу фільтрації в каталозі

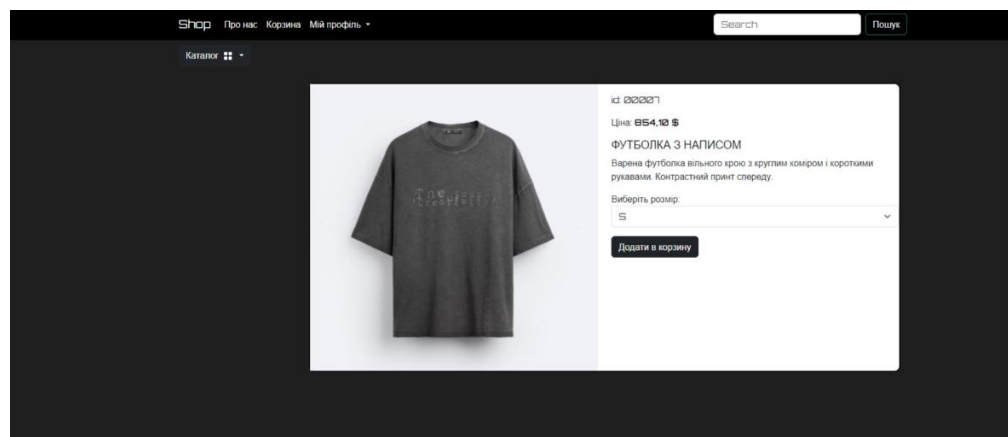


Рисунок 3.13 – Відображення картки товару

Усі ці компоненти взаємодіють між собою, де моделі Categories, Products і ProductImage забезпечують зберігання та доступ до даних про категорії, товари і їх зображення. Функції в views.py обробляють HTTP-запити, обробляють дані з моделей та передають їх в шаблони HTML для відображення. Маршрути в urls.py визначають, які views будуть викликані для кожного URL, забезпечуючи навігацію із однієї сторінки на іншу в додатку.

У системі управління інформаційною(системою товари представлені через моделі даних Django, що включають категорії (Categories) і продукти (Products). Модель Categories(забезпечує унікальні назви категорій та URL-шляхи для них.

Модель Products містить дані про кожен товар, включаючи назву, опис, ціну, наявність знижки, кількість і належність до певної категорії. Відповідні візуалізації цих даних здійснюються через HTML-шаблони, такі як catalog.html та product.html, які дозволяють користувачам переглядати і шукати товари, адміністраторам - управляти товарами через адміністративний інтерфейс Django.

3.6 Створення додатку “Users”

Додаток користувачів в інформаційній системі інтернет-магазину чоловічого одягу, розроблений на мові програмування Python з використанням фреймворку Django, відповідає за керування обліковими записами користувачів, процесами авторизації, реєстрації та управління профілем. Він складається з декількох основних компонентів, що взаємодіють між собою для забезпечення цілісної роботи системи (таблиця 3.1).

Таблиця 3.1 – Основні компоненти додатку “Users.py”

Компонент	Опис	Основні функції
1	2	3
Форма авторизації	Сторінка для входу в систему	Введення облікових даних, перенаправлення після успішної авторизації
Форма профілю	Сторінка для перегляду та редагування особистих даних	Оновлення даних користувача, завантаження аватара

Кінець таблиці 3.1 – Основні компоненти додатку “Users.py”

1	2	3
Адміністративна панель	Керування користувачами через адміністративну панель	Перегляд, редагування, видалення користувачів, перегляд даних корзин та замовлень
Налаштування додатке	Файлконфігурації додатка	Вказання основних налаштувань додатка
Модель користувача	Структура таблиці користувачів у базі даних	Збереження та взаємодія з даними користувачів у базі даних
Форма реєстрації	Сторінка для створення нового облікового запису	Заповнення форми реєстрації, перевірка даних, реєстрація нового користувача

Форма авторизації (login.html) (рисунок 3.14) надає користувачам можливість увійти в систему, ввівши свої облікові дані. Користувач заповнює поля для введення імені користувача та паролю, після чого форма відправляється на сервер для перевірки правильності введених даних. Якщо дані коректні, користувач перенаправляється на головну сторінку або на сторінку, яку він намагався відкрити до авторизації.

Файл profile.html відповідає за відображення профілю користувача, включаючи форму для редагування особистих даних, а також розділи для перегляду корзини та замовлень. Він складається з кількох ключових розділів, кожен з яких виконує свою специфічну функцію.

Авторизація

Ім'я користувача

Пароль

Увійти

[Забули пароль?](#) | [Створити аккаунт](#)

Або авторизуйтеся через:



 Google
 Facebook

Рисунок 3.14 – Форма авторизації

Відображення профілю користувача (рисунок 3.15).

Основна частина profile.html зосереджена на відображенні особистих даних користувача. Це включає ім'я, прізвище, ім'я користувача, електронну пошту та аватар. Якщо користувач завантажив аватар, він відображається; в іншому випадку показується стандартне зображення. Користувач може змінювати свої дані, завантажувати новий аватар та зберігати зміни. Всі ці дії обробляються через форму, яка забезпечує валідацію даних і відображення помилок у разі некоректного вводу.

Відображення корзини

Користувач може переглядати свій поточний кошик, де показані всі товари, які він додав для покупки. Цей розділ використовує спеціальні теги шаблонів для динамічного отримання даних про корзину користувача. Тут користувач може бачити назви товарів, їх кількість та ціну, а також мати можливість видаляти товари з корзини або змінювати їх кількість.

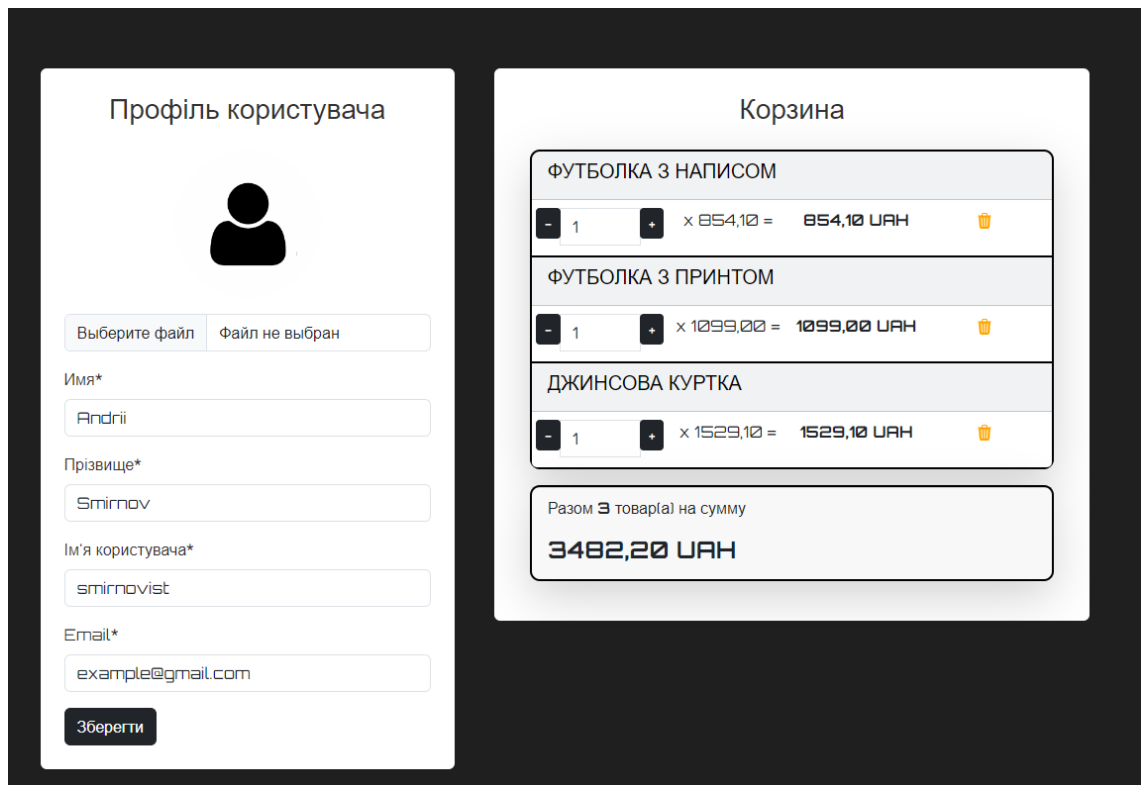


Рисунок 3.15 – Профіль користувача та його корзини

Вікно попередніх замовлень користувача

Окрім поточного кошика, користувач також має можливість переглядати свої попередні замовлення (рисунок 3.16). Кожне замовлення відображається у вигляді меню, яке розкривається для показу деталей замовлення, таких як товари, кількість та ціна. Це дозволяє користувачу легко знаходити інформацію про свої минулі покупки і контролювати виконання замовлень.

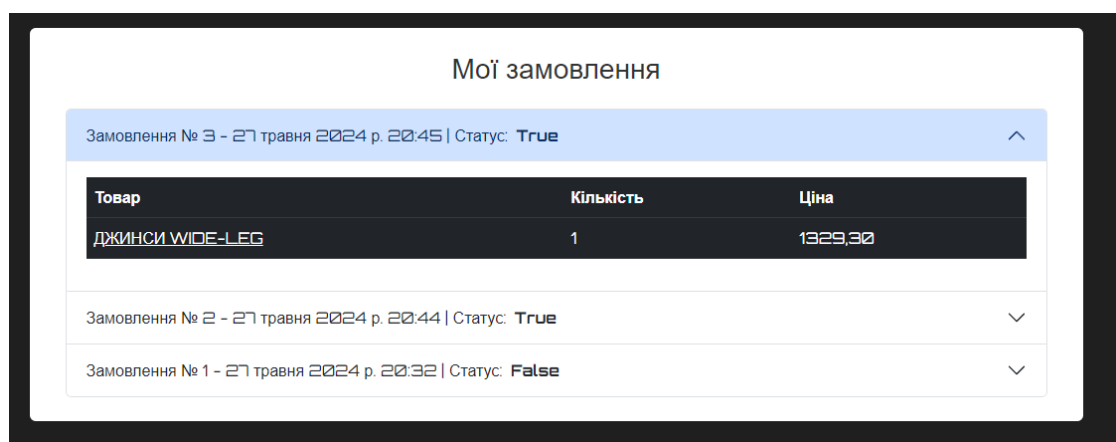


Рисунок 3.16 – Історія замовлень користувача

Форма реєстрації (registration.html) (рисунок 3.17) використовується новими користувачами для створення облікового запису. Вона містить поля для введення імені, прізвища, імені користувача, електронної пошти, паролю та підтвердження паролю. Після заповнення форми і натискання кнопки реєстрації дані перевіряються на наявність помилок, і, якщо все заповнено правильно, новий користувач додається в систему.

Реєстрація

Ім'я* Прізвище*

Ім'я користувача* Email*

Пароль* Підтвердження паролю*

Рисунок 3.17 – Форма реєстрації користувача

Адміністративна панель (admin.py) надає адміністраторам можливість керувати обліковими записами користувачів через зручний інтерфейс. Адміністратор може переглядати, редагувати та видаляти користувачів, а також переглядати пов'язані дані корзин та замовлень. Це дозволяє ефективно управляти всією базою користувачів інтернет-магазину.

Налаштування додатка (apps.py) містять основні налаштування для додатка користувачів, зокрема назву додатка та тип поля для автоматичного додавання унікальних ідентифікаторів. Цей файл забезпечує коректну реєстрацію додатка в системі Django, що дозволяє використовувати його в інших частинах проекту.

Дата приєднання: Дата: 27.05.2024 Сьогодні | 📅

Час: 19:57:59 Зараз | 🕒

Примітка: Ви на 3 години попереду серверного часу.

Аватар: Наразі: users_images/baseavatar_fkylmO5.png Очистити

Змінити: Файл не вибран

Phone number:

Рисунок 3.20 – Відображення полів image (аватар користувача) та phone_number

Взаємодія цих компонентів забезпечує цілісну роботу додатка користувачів. Коли користувач створює обліковий запис через форму реєстрації, його дані зберігаються в базі даних за допомогою моделі користувача. Після цього він може авторизуватися через форму авторизації, і його дані будуть використовуватися для входу в систему. Після авторизації користувач може переглядати та редагувати свої дані через форму профілю. Адміністратор має можливість керувати всіма користувачами через адміністративну панель, забезпечуючи ефективне управління системою. Таким чином, всі компоненти додатка працюють разом для забезпечення безперервного та зручного користування системою (рисунок 3.21).

Важливою частиною системи є моделі, які визначають структуру даних та їх взаємодію. Кожна модель має свої атрибути та методи, що дозволяють виконувати різні операції з даними. Форми служать для збору та валідації даних, що вводяться користувачем. Види (views) відповідають за обробку запитів користувачів та генерацію відповідей. Вони використовують форми та моделі для забезпечення необхідного функціоналу.

Крім того, адміністративна панель дозволяє адміністратору здійснювати контроль за даними та виконувати необхідні операції з ними. Це може включати додавання, редагування або видалення користувачів, а також перегляд статистики

використання системи. Усі ці компоненти разом формують потужний інструмент для управління користувачами та їхніми даними. Завдяки цьому, система залишається гнучкою та легкою у використанні, що робить її придатною для широкого кола застосувань.

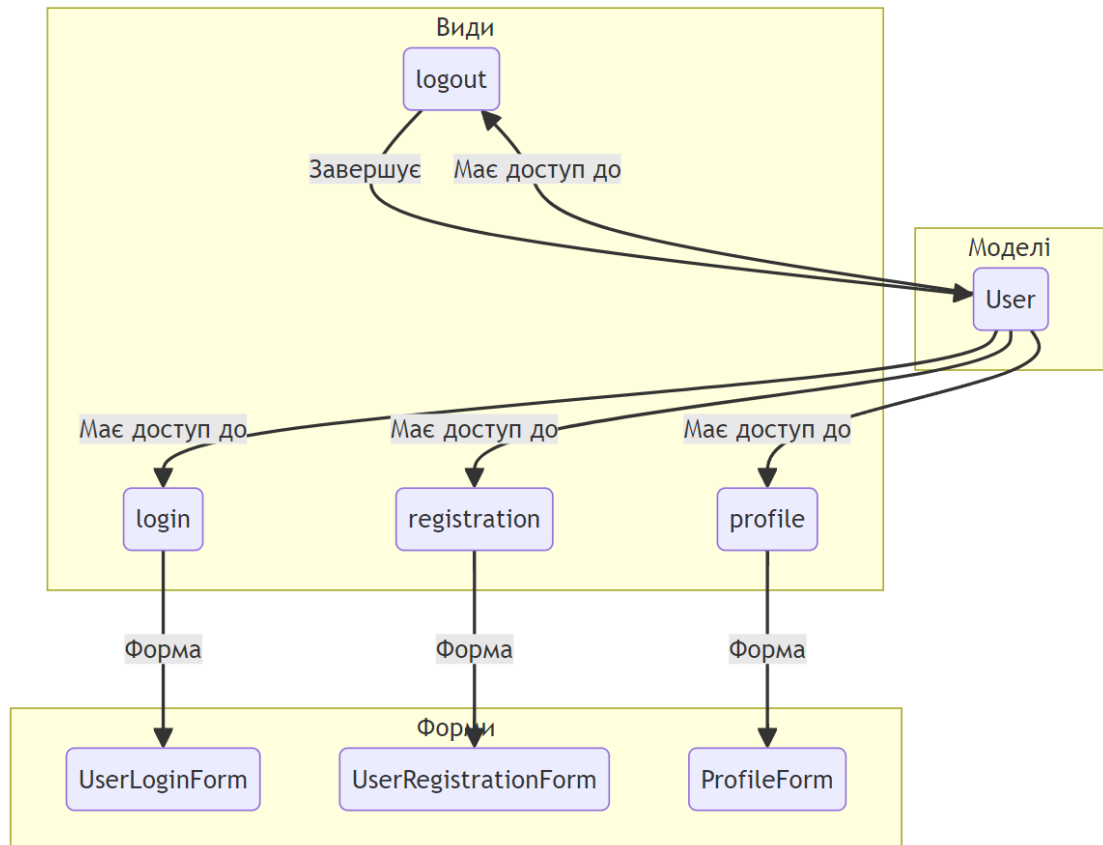


Рисунок 3.21 – Структурна схема додатку «Users»

3.7 Створення додатку «Carts»

Додаток корзини є важливим компонентом системи інтернет-магазину, що дозволяє користувачам зручно управляти своїми покупками. Основні файли та функції цього додатку працюють у тісній взаємодії, забезпечуючи безшовний користувацький досвід.

Файл `included_cart.html` відповідає за відображення вмісту корзини користувача у вигляді окремих елементів на веб-сторінці (рисунок 3.22). Цей файл є частиною шаблону Django, що використовується для динамічного формування

HTML-коду, який відображається у браузері користувача. Він завантажується та оновлюється асинхронно за допомогою JavaScript, що забезпечує миттєву реакцію на дії користувача без необхідності перезавантаження сторінки. Ось детальний опис структури та функціональності цього файлу:

Файл `included_cart.html` починається з завантаження статичних файлів Django за допомогою тегу `{% load static %}`. Це дозволяє використовувати статичні ресурси, такі як зображення або стилі, у шаблоні.

Для кожного товару у корзині створюється окремий блок. Цей блок включає заголовок з назвою продукту та елементи керування для зміни кількості товару або його видалення з корзини. Елементи керування включають кнопки з класами `decrement` та `increment`, які зменшують або збільшують кількість товару відповідно. Ці кнопки містять атрибути `data-cart-id` та `data-cart-change-url`, що використовуються для передачі ідентифікатора товару та URL для зміни кількості через AJAX-запити.

Ціна за одиницю товару та загальна вартість товару у корзині також відображаються у відповідних блоках. Вартість обчислюється на основі кількості товару та його ціни за одиницю, а результати виводяться за допомогою тегів Django, таких як `{{ cart.product.sell_price }}` та `{{ cart.products_price }}`.

Після відображення всіх товарів у корзині, файл завершується блоком, що підсумовує загальну кількість товарів та їх загальну вартість. Це відображається у нижній частині картки за допомогою тегів `{{ carts.total_quantity }}` та `{{ carts.total_price }}`, які обчислюються на сервері та передаються у шаблон.

Також передбачено кнопку для оформлення замовлення, яка відображається лише у тому випадку, якщо корзина не порожня і немає активних замовлень. Ця кнопка містить посилання на URL для створення замовлення і дозволяє користувачу перейти до завершального етапу покупки.

Таким чином, `included_cart.html` забезпечує зручне та інтуїтивно зрозуміле відображення вмісту корзини користувача, дозволяючи легко керувати товарами у ній. Використання тегів Django та елементів Bootstrap [33] забезпечує динамічне та

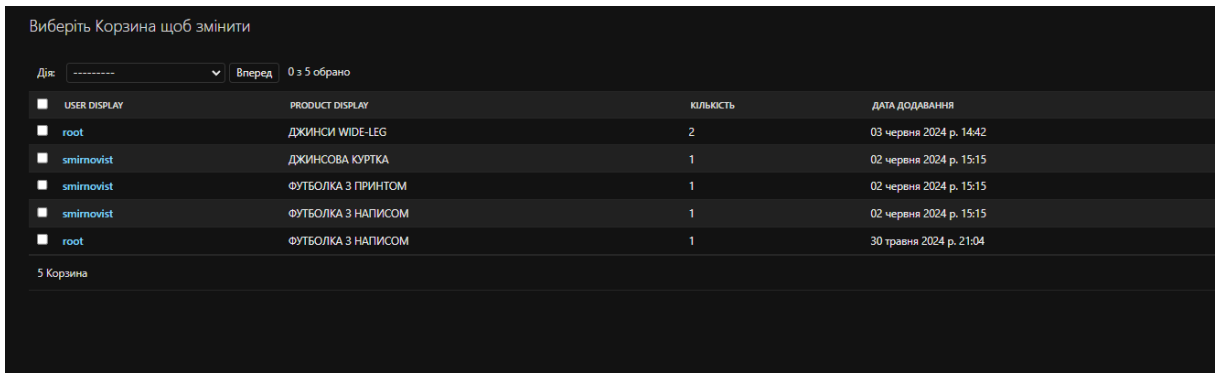


Рисунок 3.23 – Вигляд корзини в адмін панелі

Модель Cart також містить клас CartQueryset, який визначає користувацькі методи запитів для обчислення загальної ціни та кількості товарів у корзині. Метод total_price підраховує сумарну вартість всіх товарів у корзині, викликаючи метод products_price для кожного об'єкта корзини і складаючи результати. Метод total_quantity підраховує загальну кількість одиниць товару у корзині.

У самому класі Cart визначені методи і властивості, що полегшують роботу з даними. Наприклад, метод products_price обчислює вартість товарів у корзині, помножуючи ціну за одиницю товару на кількість одиниць. Метод __str__ повертає зручне для читання рядкове представлення об'єкта корзини, яке включає ім'я користувача (якщо він автентифікований), назву товару та кількість.

Метаклас Meta задає деякі додаткові параметри моделі, такі як ім'я таблиці в базі даних (db_table) і людське представлення назви моделі в однині та множині (verbose_name і verbose_name_plural).

Модель Cart є основою для управління корзиною покупок у додатку, забезпечуючи збереження, обробку та відображення даних про товари, додані користувачем до корзини. Вона дозволяє легко отримувати інформацію про корзину, обчислювати загальну вартість і кількість товарів, а також працювати як з автентифікованими, так і з неавтентифікованими користувачами.

Файл admin.py у Django-додатку є ключовим для налаштування адміністративного інтерфейсу, який дозволяє адміністраторам переглядати, редагувати та керувати даними моделей, зокрема моделлю Cart. Завдяки цьому

файлу можна зручно переглядати статус замовлень та здійснювати інші адміністративні дії.

Спершу, файл імпортує необхідні модулі: `admin` з пакету `django.contrib` та модель `Cart` з додатку `carts`. Потім створюється клас `CartTabAdmin`, який визначає, як відображати модель `Cart` в табличному вигляді в інтерфейсі адміністратора. В цьому класі задаються поля, які будуть відображатися: `product`, `quantity` та `created_timestamp`. Також зазначаються поля для пошуку (`search_fields`) і поля тільки для читання (`readonly_fields`). Властивість `extra` встановлює кількість додаткових пустих рядків для введення нових записів.

Далі йде декоратор `@admin.register(Cart)`, який реєструє модель `Cart` в адміністративному інтерфейсі. Клас `CartAdmin` успадковує `admin.ModelAdmin` і налаштовує відображення моделі `Cart` у вигляді списку. Параметр `list_display` визначає, які поля будуть відображені в списку записів: `user_display`, `product_display`, `quantity`, `created_timestamp`. Це дозволяє адміністратору швидко бачити основну інформацію про кожну корзину.

Параметр `list_filter` додає можливість фільтрування записів за певними полями: `created_timestamp`, `user`, `product__name`. Це полегшує пошук і сортування записів за конкретними критеріями, такими як дата створення, користувач або назва продукту.

У класі `CartAdmin` визначені два методи: `user_display` та `product_display`. Метод `user_display` повертає ім'я користувача, пов'язаного з корзиною, або "Анонімний користувач", якщо корзина належить неавторизованому користувачу. Метод `product_display` повертає назву продукту, пов'язаного з корзиною.

Загалом, файл `admin.py` налаштовує адміністративний інтерфейс для моделі `Cart`, що дозволяє адміністраторам ефективно переглядати та керувати замовленнями. Завдяки цьому файлу, адміністратори можуть бачити деталі замовлень, здійснювати пошук і фільтрацію даних, а також легко орієнтуватися в адміністративному інтерфейсі (рисунок 3.24).

					КВРІСТ 200186 20.01.10 ПЗ	Арк. 54
Зм.	Арк.	№ докум.	Підпис	Дата		

Виберіть Заовлення щоб змінити

Пошук

Дія: ----- Вперед 0 з 3 обрано

ID	КОРИСТУВАЧ	ПОТРІБНА ДОСТАВКА	СТАТУС ЗАОВЛЕННЯ	ОПЛАТА ПРИ ОТРИМАННІ	ОПЛАЧЕНО	ДАТА СТВОРЕННЯ ЗАОВЛЕННЯ
3	smirnovist	●	В обробці	●	●	27 травня 2024 р. 20:45
2	smirnovist	●	В обробці	●	●	27 травня 2024 р. 20:44
1	smirnovist	●	В обробці	●	●	27 травня 2024 р. 20:32

3 Заовлення

Рисунок 3.24 – Вигляд заовлень в адмін панелі

Файл конфігурації додатку `apps.py` містить налаштування додатку корзини, включаючи визначення імені додатку та налаштування його поведінки в системі.

Маршрутизація додатку визначається у файлі `urls.py`, де вказані маршрути для взаємодії з корзиною. Це включає маршрути для додавання товару до корзини, зміни кількості товарів та видалення товарів з корзини.

Файл `views.py` є ключовим компонентом у Django-додатку для управління корзиною покупок. Він містить логіку обробки запитів, взаємодії з моделями та підготовки даних для відображення у шаблонах. Основними функціями цього файлу є додавання товарів до корзини, зміна кількості товарів та видалення товарів з корзини. Функція `cart_add` обробляє запити на додавання товарів до корзини. Коли користувач натискає кнопку для додавання товару до корзини, відправляється POST-запит із ідентифікатором товару. Функція отримує цей ідентифікатор та знаходить відповідний товар у базі даних за допомогою моделі `Products`. Далі функція перевіряє, чи користувач автентифікований. Якщо так, товар додається до корзини цього користувача. Якщо корзина з цим товаром вже існує, кількість товару збільшується на одиницю. Якщо ж користувач не автентифікований, товар додається до корзини, асоційованої з сесією користувача. Після додавання або оновлення товару, функція оновлює HTML-код корзини за допомогою шаблону `included_cart.html` та повертає відповідь у форматі JSON, яка містить оновлену корзину.

Функція `cart_change` відповідає за зміну кількості товарів у корзині. Вона отримує POST-запит із ідентифікатором корзини та новою кількістю товару. Після

знаходження відповідного запису в базі даних, функція оновлює кількість товару та зберігає зміни. Потім вона знову генерує оновлений HTML-код корзини і повертає відповідь у форматі JSON з оновленими даними [14].

Функція `cart_remove` обробляє видалення товарів з корзини. Вона отримує POST-запит із ідентифікатором корзини, знаходить відповідний запис у базі даних і видаляє його. Після цього функція оновлює HTML-код корзини та повертає відповідь у форматі JSON з повідомленням про видалення товару та оновленою корзиною. Усі ці функції використовують допоміжну функцію `get_user_carts`, яка повертає поточну корзину користувача або сесії, що дозволяє зручно отримувати актуальні дані про стан корзини [20].

Таким чином, файл `views.py` забезпечує повний цикл взаємодії користувача з корзиною покупок: від додавання товарів до зміни їх кількості та видалення, підтримуючи актуальність даних та інтерфейсу в режимі реального часу.

Допоміжні утиліти, що містяться у файлі `utils.py`, включають функції для отримання даних про корзину користувача, що спрощує доступ до цих даних у різних частинах додатку.

Таким чином, всі ці компоненти працюють разом, щоб забезпечити зручне та ефективне управління корзиною користувача в інтернет-магазині. Користувачі можуть легко додавати товари до корзини, змінювати їх кількість або видаляти, а також переглядати загальну вартість своїх покупок. Адміністратори мають інструменти для ефективного управління корзинами, що допомагає забезпечувати високу якість обслуговування клієнтів.

3.8 Підключення PostgreSQL

До інформаційної системи підтримки інтернет-магазину було додано PostgreSQL (рисунок 3.25) за допомогою налаштування параметрів у файлі `settings.py` Django. Введення PostgreSQL надає значні переваги для ефективності та надійності системи.

PostgreSQL забезпечує значні переваги для оптимальної роботи з таблицями у системі підтримки інтернет-магазину (таблиця 3.2). Вона відзначається високою продуктивністю та масштабованістю для обробки великих обсягів даних, забезпечує цілісність даних за допомогою механізмів транзакцій і обмежень, має гнучкість для розширення функціональності за рахунок користувацьких функцій та збережених процедур, а також високий рівень безпеки з можливостями аутентифікації та авторизації користувачів.

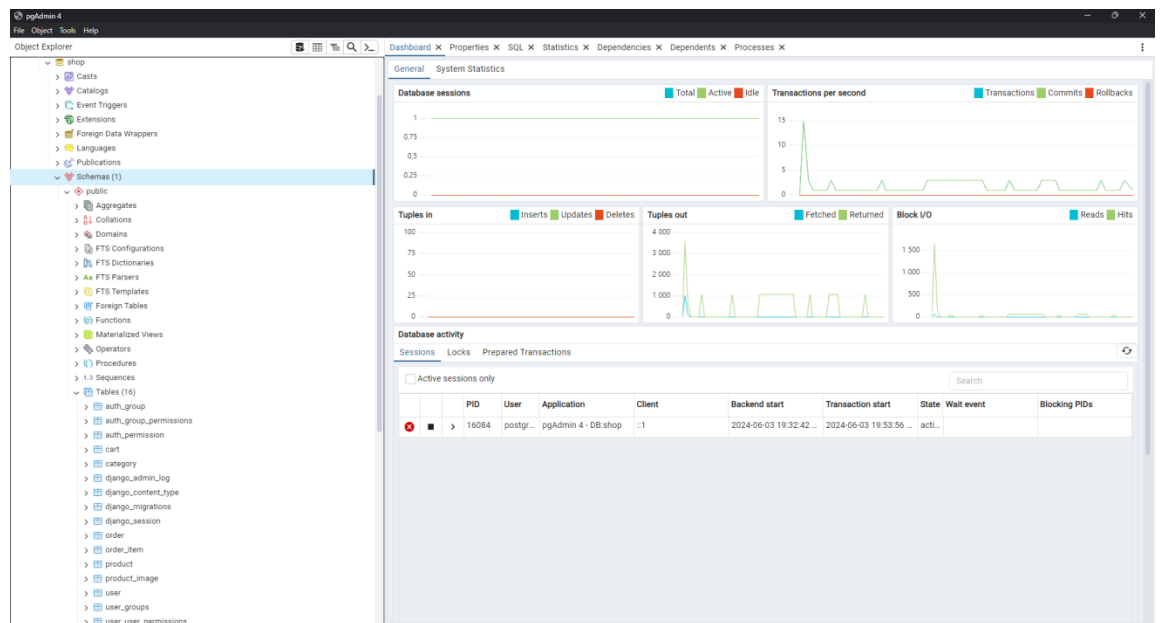


Рисунок 3.25 – Вигляд таблиць в базі даних PostgreSQL

Крім того, PostgreSQL підтримує реплікацію та резервне копіювання, що забезпечує надійність і доступність даних у випадку відмови системи. Вона має потужні інструменти для аналітики і звітності, що дозволяє проводити глибокий аналіз даних. PostgreSQL також підтримує різноманітні типи даних і індексів, що підвищує ефективність запитів.

Таблиця 3.2 – Опис переваг PostgreSQL

Перевага	Опис
----------	------

Продуктивність та масштабованість	PostgreSQL здатна обробляти великі обсяги даних та підтримувати велику кількість одночасних запитів. Це забезпечує швидке виконання операцій з додавання нових продуктів, обробки замовлень та керування категоріями товарів.
Цілісність даних	Використання механізмів транзакцій та обмежень дозволяє гарантувати, що дані в таблицях users, product, orders та category завжди будуть коректними та узгодженими.
Гнучкість та розширюваність	PostgreSQL пропонує інструменти для реалізації складної бізнес-логіки на рівні бази даних, такі як користувацькі функції, тригери та збережені процедури. Це зменшує навантаження на сервер застосунку та покращує загальну продуктивність системи.
Безпека	Завдяки розширеним можливостям аутентифікації та авторизації PostgreSQL дозволяє точно контролювати доступ до даних та запобігати несанкціонованому доступу клієнтів та фінансові операції.

Підключення PostgreSQL до інтернет-магазину, що працює на Django, здійснюється через файл налаштувань settings.py, а саме через поля:

1. 'ENGINE': 'django.db.backends.postgresql': Вказує Django використовувати движок PostgreSQL для взаємодії з базою даних.
2. 'NAME': 'shop': Це назва вашої бази даних PostgreSQL, яку створили для вашого інтернет-магазину.
3. 'USER': 'shop': Це користувач бази даних PostgreSQL, який має права доступу до цієї бази даних.
4. 'PASSWORD': 'shop': Пароль для входу до бази даних PostgreSQL в якості користувача 'shop'.
5. 'HOST': 'localhost': Вказує на те, що база даних знаходиться на тому ж комп'ютері, де виконується Django (локальний сервер).
6. 'PORT': '5432': Порт, на якому працює PostgreSQL. Зазвичай, 5432 - це стандартний порт для PostgreSQL, якщо ви його не змінювали.

Цей конфігураційний файл settings.py інтегрує Django з PostgreSQL, дозволяючи Django ORM взаємодіяти з базою даних для зберігання та витягування даних з таблиць, таких як users, product, orders, category та інші, які ви визначили у вашій базі даних. Таке підключення забезпечує ефективну та надійну роботу інтернет-магазину, забезпечуючи необхідну інфраструктуру для обробки та зберігання даних користувачів, замовлень та товарів.

3.9 Додавання Ажах в інформаційну систему інтернет магазину

Окрім цього, в інформаційну систему управління інтернет-магазином чоловічого одягу було додано Ажах. Ажах (Asynchronous JavaScript and XML) є технологією, яка дозволяє здійснювати асинхронні запити до сервера без перезавантаження сторінки. У контексті коду, використаного у даній дипломній роботі, Ажах використовується для реалізації динамічного оновлення веб-сторінки без її повного перезавантаження при взаємодії з корзиною товарів (таблиця 3.3).

Таблиця 3.3 – Основні аспекти використання технології Ажах в інформаційній системі

Аспект коду	Характеристика
1	2
Додавання в корзину	<p>При кліканні на кнопку «Додати в корзину» (add-to-cart), відбувається Ажах запит POST на сервер для додавання вибраного товару в корзину.</p> <p>Після успішного додавання, виводиться повідомлення про успішну операцію (рисунок 3.23), оновлюється лічильник товарів у корзині та відображається актуальний вміст корзини [11].</p>
Видалення з корзини	<p>При кліканні на кнопку «Видалити з корзини» (remove-from-cart), відбувається Ажах запит на видалення товару з корзини. Після успішного видалення також виводиться повідомлення, оновлюється лічильник товарів у корзині та відображається актуальний вміст корзини.</p>
Зміна кількості товару в корзині	<p>Кнопки «Збільшити» (increment) і «Зменшити» (decrement) дозволяють змінювати кількість товару в корзині за допомогою Ажах запитів. При зміні кількості товару, оновлюється корзина, лічильник товарів та відображається відповідне повідомлення [30].</p>

Кінець таблиці 3.2 – Основні аспекти використання технології Ажах в інформаційні системи

				КВРІСТ 200186 20.01.10 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис		Дата

1	2
Сповіщення	Повідомлення про успішні дії (додавання, видалення, зміна кількості) відображаються на сторінці для користувача за допомогою фейдінгу і витухання.

3.10 Додаток «Orders»

Додаток orders у Django для інтернет-магазину чоловічого одягу є важливою складовою, що забезпечує обробку замовлень. Він складається з декількох ключових файлів: admin.py, views.py, models.py і create_order.html.

У файлі admin.py налаштовується адміністративний інтерфейс для моделей Order та OrderItem. Клас OrderItemTabulareAdmin налаштовує відображення замовлених товарів у табличному вигляді з полями product, name, price, quantity, додаючи можливість пошуку за цими полями. Клас OrderTabulareAdmin налаштовує відображення замовлень у табличному вигляді з полями requires_delivery, status, payment_on_get, is_paid, created_timestamp, з можливістю пошуку та фільтрації. Клас OrderAdmin визначає відображення замовлень у вигляді списку, додаючи можливість фільтрації та пошуку за різними параметрами (рисунок 3.26 та рисунок 3.27).

Замовлення	Продукт	Назва	Ціна	Кількість
Замовлення № 3 Покупець Andrii Smitnov	ДЖИНСИ WIDE-LEG Кількість - 22	ДЖИНСИ WIDE-LEG	1329,30	1
Замовлення № 2 Покупець Andrii Smitnov	ДЖИНСОВА КУРТКА Кількість - 139	ДЖИНСОВА КУРТКА	1529,10	1
Замовлення № 1 Покупець Andrii Smitnov	ФУТБОЛКА З НАПИСОМ Кількість - 39	ФУТБОЛКА З НАПИСОМ	854,10	1
Замовлення № 1 Покупець Andrii Smitnov	ФУТБОЛКА З ПРИНТОМ Кількість - 39	ФУТБОЛКА З ПРИНТОМ	1099,00	1

Рисунок 3.26 – Відображення проданих товарів в адмін панелі

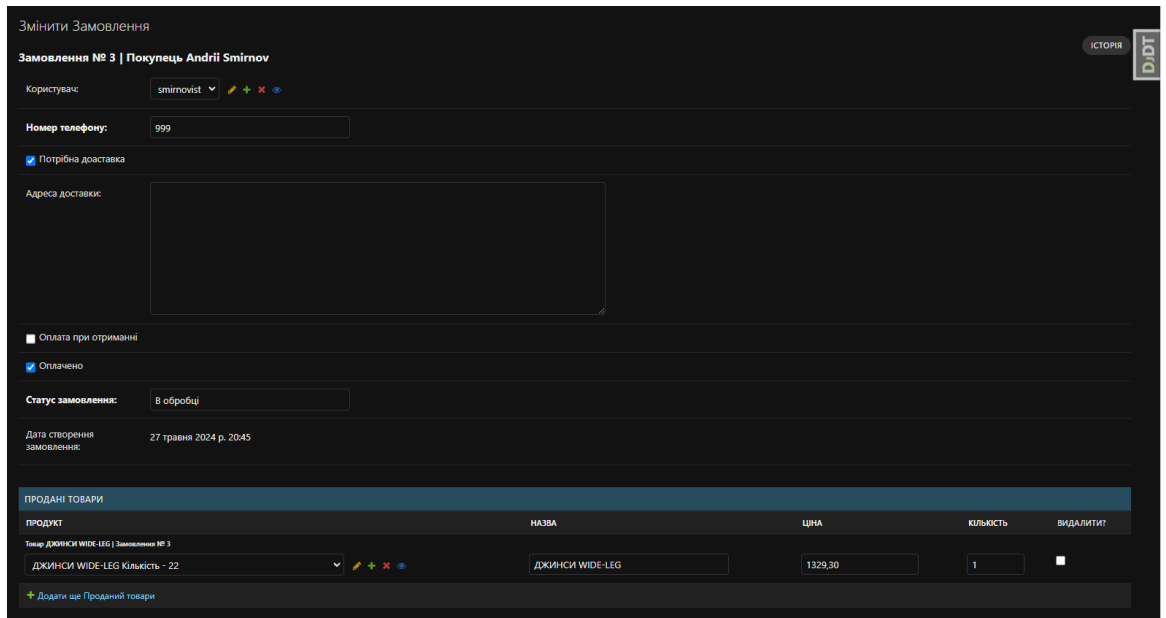


Рисунок 3.27 – Вигляд замовлення в адмін панелі

Файл `views.py` містить логіку для створення замовлення. Функція `create_order` обробляє запит на створення замовлення, перевіряючи правильність форми і створюючи запис у базі даних. Якщо всі перевірки успішні, створюється об'єкт `Order`, а потім для кожного елемента корзини створюється об'єкт `OrderItem`.

Після цього корзина користувача очищується, і користувача перенаправляють на сторінку профілю з повідомленням про успішне створення замовлення.

Файл `models.py` містить визначення моделей `Order` і `OrderItem`. Модель `Order` містить інформацію про замовлення, включаючи користувача, дату створення, номер телефону, необхідність доставки, адресу доставки, спосіб оплати, статус оплати та статус замовлення. Модель `OrderItem` містить інформацію про замовлені товари, включаючи посилання на замовлення, продукт, назву, ціну, кількість і дату створення.

Шаблон `create_order.html` містить HTML-розмітку для сторінки оформлення замовлення. Він включає форми для введення деталей замовлення, таких як ім'я, прізвище, номер телефону, адреса доставки та спосіб оплати. Також на цій сторінці відображаються вибрані товари з корзини (Рисунок 3.28).

Таким чином, додаток orders забезпечує повний цикл обробки замовлень, включаючи їх створення, збереження в базі даних і відображення в адміністративному інтерфейсі (рисунки 3.28-3.29).

Вибрані товари

ФУТБОЛКА З НАПИСОМ			
-	1	+	x 654,10 = 654,10 UAH 🗑️
ДЖИНСИ WIDE-LEG			
-	1	+	x 1329,30 = 1329,30 UAH 🗑️
Разом 2 товар(а) на сумму			
2183,40 UAH			

Деталі замовлення

Ім'я*: Прізвище*:

Номер телефону*:

Спосіб доставки: Потрібна доставка Сомовивіз

Адреса доставки*:

Спосіб оплати: Оплата картою Готівкою/картою при отриманні

Оформити замовлення

Рисунок 3.28 – Вікно оформлення замовлення

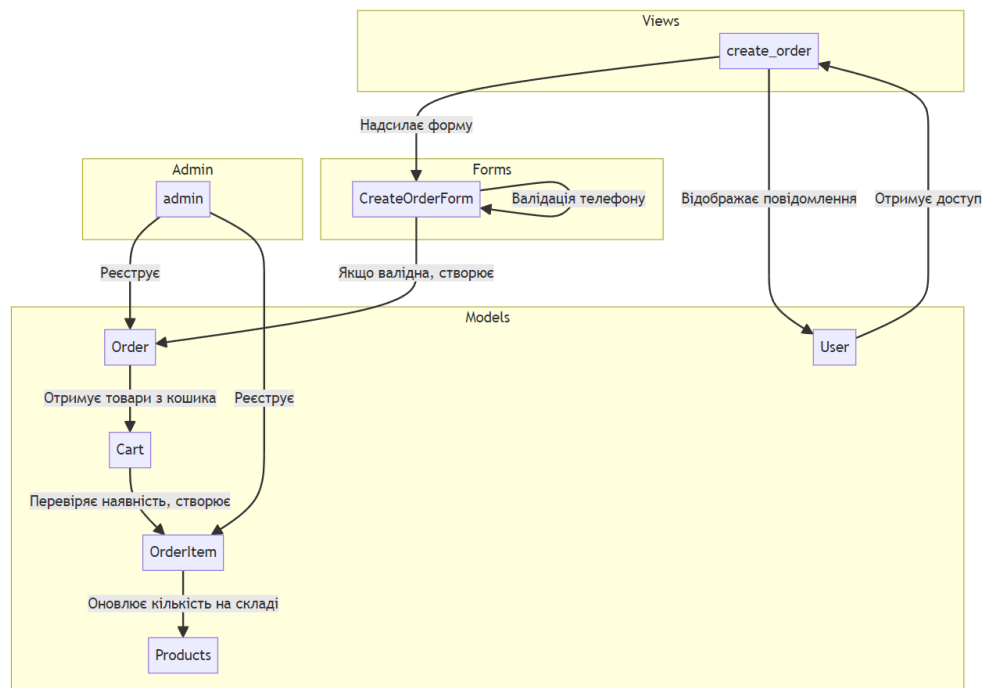


Рисунок 3.29 – Структурна схема роботи додатку «Orders»

3.11 Висновки

У даному розділі було розроблено основні компоненти інформаційної системи інтернет магазину чоловічого одягу, а саме такі додатки як: «Users», «Manage.py», «Goods», «Orders», «Carts» та «Main», які тісно взаємодіють між собою, задля того, щоб користувачу було зручно та швидко замовляти товари та шукати їх на сайті.

Окрім цього в даному розділі було розглянуто підключення до інформаційної системи таких технологій як: Ajax та базу даних PostgreSQL задля автоматичного оновлення даних на сторінці та швидкого пошуку товарів на сайті.

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Розробка інформаційної системи підтримки роботи інтернет-магазину чоловічого одягу на основі фреймворку Django стала ключовим проектом, що продемонстрував можливості сучасних веб-технологій у сфері електронної комерції. Основна мета проекту полягала у створенні функціонального, зручного та надійного інструменту для управління інтернет-магазином, який би задовольняв як потреби користувачів, так і адміністративні вимоги власників бізнесу.

Створений веб-додаток включає в себе комплексний набір функцій, необхідних для успішної роботи інтернет-магазину. Однією з головних функцій є можливість замовлення товару, що реалізується через зручний та інтуїтивно зрозумілий інтерфейс користувача. Клієнти можуть легко додавати обрані товари до кошика, оформлювати замовлення, вводити необхідну інформацію для доставки та оплати. Усі ці дії здійснюються з мінімальними витратами часу та зусиль з боку користувача, що значно підвищує зручність та ефективність взаємодії з сайтом.

Каталог товарів є ще однією важливою складовою проекту. Він структурований таким чином, щоб користувачі могли легко знайти необхідні товари за різними категоріями та параметрами. Функція фільтрації товарів дозволяє користувачам звужити вибір відповідно до певних критеріїв, таких як ціна, розмір, колір, бренд тощо. Це забезпечує швидкий доступ до потрібної інформації та спрощує процес вибору товарів.

Реєстрація користувачів та можливість входу до особистого кабінету є ще однією важливою функцією, що підвищує зручність для постійних клієнтів. У особистому кабінеті користувачі можуть переглядати історію своїх замовлень, зберігати улюблені товари, оновлювати особисту інформацію та налаштування. Це створює більш персоналізований досвід взаємодії з інтернет-магазином та сприяє підвищенню лояльності клієнтів.

Функція пошуку товарів дозволяє користувачам швидко знаходити потрібні позиції за допомогою ключових слів. Це особливо корисно для великих каталогів,

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 65
Зм.	Арк.	№ докум.	Підпис	Дата		

де ручне переглядання всіх товарів може бути занадто трудомістким. Пошуковий механізм оптимізовано для швидкого та точного відображення результатів, що задовольняють запити користувачів.

Адміністрування сайту включає обробку замовлень та управління контентом. Адміністратори мають доступ до зручної панелі керування, де вони можуть відстежувати нові замовлення, змінювати статуси замовлень, керувати асортиментом товарів, додавати нові позиції, редагувати описи та ціни. Це дозволяє ефективно організувати роботу магазину та оперативно реагувати на зміни в потребах клієнтів.

Проект демонструє, як за допомогою сучасних технологій можна створити зручний і функціональний інструмент для електронної комерції, що відповідає вимогам користувачів та адміністраторів. Використання фреймворку Django дозволило швидко та ефективно реалізувати всі необхідні функції, забезпечити високу продуктивність та безпеку системи. Завдяки модульній архітектурі проекту, можливе його подальше розширення та адаптація під нові вимоги бізнесу.

У підсумку, розроблена інформаційна система є успішним прикладом застосування сучасних веб-технологій у створенні інтернет-магазину. Вона забезпечує високу зручність та ефективність взаємодії для кінцевих користувачів, а також надає потужні інструменти для адміністрування та управління бізнесом. Цей проект може слугувати основою для подальших розробок у сфері електронної комерції, демонструючи можливості та переваги використання Django у таких проектах.

					КвРІСТ 200186 20.01.10 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ТА ПОСИЛАНЬ

1. ERP система – що це? URL: <https://spar.ua/blogs/erp-sistema-shcho-tse> (дата звернення: 30.05.2024).
2. Що таке CRM система. URL: <https://snov.io/glossary/ua/customer-relationship-management-crm-ua/> (дата звернення: 08.06.2024).
3. Краус К.М., Краус Н.М., Манжура О.В. Електронна комерція та Інтернет-торгівля: навчально-методичний посібник. Київ: Аграр Медіа Груп. 2021. 454 с.
4. Zhang H., Cardoza, A., Chen, P. B., Angel, S., & Liu, V. Fault-tolerant and transactional stateful serverless workflows. In 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20). 2020. 1204 p.
5. Deshpande T. DynamoDB Cookbook. Packt Publishing Ltd. 2015. P.120–140.
6. Saini, R., & Behl, R. An Introduction to AWS-EC2 (Elastic Compute Cloud). In ICRMAT. 2020. 102 с.
7. Liu, J., Zhang, S., Wang, Q., & Wei, J. Coordinating Fast Concurrency Adapting with Autoscaling for SLO-Oriented Web Applications. *IEEE Transactions on Parallel and Distributed Systems*. 2022. P. 1-12
8. Qu, C., Calheiros, R. N., & Buyya, R. A reliable and cost-efficient auto-scaling system for web applications using heterogeneous spot instances. *Journal of Network and Computer Applications*. 2016. P. 167–180.
9. Виклюк Я. І., Камінський Р. М., Пасічник В. В. Моделювання складних систем: посібник. 2000. 322 с.
10. Петренко Д. М., Коваленко І. О. Розробка інтернет-магазину за допомогою фреймворку Django. Інформаційні технології та комп'ютерні системи. 2021. 120 с.
11. Технологія AJAX: основи та переваги використання. URL: <https://developer.mozilla.org/uk/docs/Web/Guide/AJAX> (дата звернення: 08.06.2024).
12. Марченко І. В., Сидоренко Ю. В. Застосування JavaScript у веб-розробці. Програмні системи і комплекси. 2019. 200 с.

					КВРІСТ 200186 20.01.10 ПЗ	Арк. 67
Зм.	Арк.	№ докум.	Підпис	Дата		

13. Bootstrap: сучасний фреймворк для розробки адаптивних інтерфейсів. URL: <https://getbootstrap.com/docs/5.0/getting-started/introduction/> (дата звернення: 30.06.2024).

14. Django Documentation: The Web Framework for Perfectionists with. URL: <https://www.djangoproject.com/start/overview/> (дата звернення: 03.06.2024).

15. Савченко М. П. Інформаційні системи для підтримки електронної комерції. Київ: Наукова думка. 2017. 174 с.

16. Переваги використання Django у розробці веб-застосунків. URL: <https://www.djangoproject.com/start/overview/> (дата звернення: 08.06.2024).

17. Василенко Ю. О. Архітектура сучасних веб-застосунків на базі Django. *Вісник Київського національного університету. Серія: Інформатика та кібернетика*. 2020. № 3. С. 75–82.

18. Литвиненко О. В. Інтернет-магазини: тенденції розвитку та технологічні аспекти. *Журнал інформаційних технологій та мереж*. 2019. С. 58–65.

19. Кириленко С. М. Використання фреймворку Django для створення веб-додатків. *Вісник Одеського національного університету. Серія: Інформаційні технології*. 2021. №4. С. 98–105.

20. Interactive JavaScript: Creating Dynamic Web Applications. URL: <https://www.w3schools.com/js/> (дата звернення: 08.06.2024).

21. Гаврилюк Ю. В., Кузьменко М. П. Адаптивна розробка веб-додатків на базі Bootstrap. *Вісник Львівського національного університету. Серія: Інформатика*. 2020. №1. С. 114–121.

22. Основи HTML та CSS для веб-дизайну. URL: <https://www.w3schools.com/html/> (дата звернення: 03.06.2024).

23. Романчук І. В. Принципи роботи та переваги AJAX у сучасних веб-додатках. *Науковий вісник Чернівецького університету. Серія: Комп'ютерні науки*. 2019. №2. С. 67–73.

24. Орлов Д. С. Моделювання інформаційних систем для інтернет-магазинів. *Інформаційні технології та системи*. 2018. 209 с.

					КВРІСТ 200186 20.01.10 ПЗ	Арк. 68
Зм.	Арк.	№ докум.	Підпис	Дата		

25. Петров В. В. Технології та інструменти веб-розробки на основі Django. *Вісник Харківського національного університету. Серія: Інформатика та кібернетика*. 2021.№2. С. 43–50.

26. Гнатенко М. О. Розробка веб-додатків з використанням JavaScript та AJAX. *Журнал "Інформаційні технології та засоби"*. 2019.№1. С. 56–62.

27. Лебедев В. О. Використання HTML5 у сучасних веб-додатках. *Вісник Вінницького національного технічного університету*. 2018.№3. С. 103–110.

28. Кравчук М. І. Особливості розробки інтернет-магазинів за допомогою Django. *Вісник Національного технічного університету "ХПИ". Серія: Інформатика та кібернетика*. 2020.№1. С. 77–83.

29. Переваги та можливості використання AJAX у веб-розробці. URL: <https://www.ibm.com/developerworks/library/wa-ajaxintro1/> (дата звернення: 04.06.2024).

30. Козак І. Г. Моделювання та розробка веб-додатків з використанням Django. *Вісник Тернопільського національного технічного університету*. 2019.№3. С. 88–94.

31. Шевченко І. О. Застосування JavaScript у створенні інтерактивних веб-інтерфейсів. *Науковий вісник Херсонського державного університету. Серія: Інформатика та програмування*. 2020.№4. С. 101–108.

32. Використання Bootstrap для створення адаптивних веб-сторінок. URL: (дата звернення: 04.06.2024).

33. Захарченко М. В. HTML5 та його роль у сучасних веб-технологіях. *Вісник Дніпровського національного університету. Серія: Комп'ютерні науки*. 2018.№2. С. 69–75.

34. E-commerce Website Development Using Django. URL: <https://realpython.com/get-started-with-django-1/> (дата звернення: 08.06.2024).

35. Бондаренко О. В. Технологічні аспекти створення веб-додатків з використанням JavaScript та AJAX. *Вісник Полтавського національного технічного університету. Серія: Інформаційні технології*. 2021.№1. С. 53–60.

					КВРІСТ 200186 20.01.10 ПЗ	Арк. 69
Зм.	Арк.	№ докум.	Підпис	Дата		

36. Коваленко В. П. Інформаційні системи для підтримки електронної комерції: сучасні тенденції та перспективи. Київ: Наукова думка. 2018. 240 с.
37. Використання Django у розробці веб-додатків.URL: <https://djangobook.com/> (дата звернення: 05.06.2024).
38. Соколенко І. А. Розробка адаптивних веб-сайтів з використанням Bootstrap. *Вісник Національного університету "Львівська політехніка". Серія: Інформаційні технології.* 2020.№1. С. 91–98.
39. Bootstrap for Responsive Web Design. Web Technologies and Systems. 2019. 121 p.
40. Grigorik, I. High Performance Browser Networking. O'Reilly Media. 2013. 408 p.
41. Osmani, A. Learning JavaScript Design Patterns. O'Reilly Media. 2012. 254 p.
42. Haverbeke, M. Eloquent JavaScript: A Modern Introduction to Programming. No Starch Press. 2018. 240 p.
43. Forcier, J., Bissex, P., & Chun, W. J. Python Web Development with Django. Addison-Wesley Professional. 2008.456 p.
44. AJAX: A New Approach to Web Applications.URL: <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX> (дата звернення: 04.06.2024).
45. MacDonald, M. HTML5: The Missing Manual. O'Reilly Media. 2020.702 p.
46. Vincent, W. S. Django for Beginners: Build websites with Python and Django. 2018. 335 p.
47. Flanagan, D. JavaScript: The Definitive Guide. O'Reilly Media. 2020. 510 p.
48. Pinkham, A. Django Unleashed. Addison-Wesley Professional. 2015. 840 p.
49. Horstmann, C. S. Modern JavaScript for the Impatient. Addison-Wesley Professional. 2020. 384 p.
50. Crockford, D. JavaScript: The Good Parts. O'Reilly Media. 2008. 172 p.
51. Matthes, E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press. 2019. 160 p.

					КВРІСТ 200186 20.01.10 ПЗ	Арк. 70
Зм.	Арк.	№ докум.	Підпис	Дата		

53. Robbins, J. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. O'Reilly Media. 2018. 619 p.

54. Шоляк В. Що таке респонсивний дизайн.URL: <https://wizeclub.education/blog/shho-take-responsivnij-dizajn/> (дата звернення: 08.06.2024).

55. З'явлася нова версія фреймворку Django для Python: що нового.URL: <https://highload.today/wp-content/uploads/2023/04/Bezumyannuj-3-1200x900.jpg> (дата звернення: 08.06.2024).

56. Каталог товарів на сайті Ябко.URL: <https://jabko.ua/> (дата звернення: 08.06.2024).

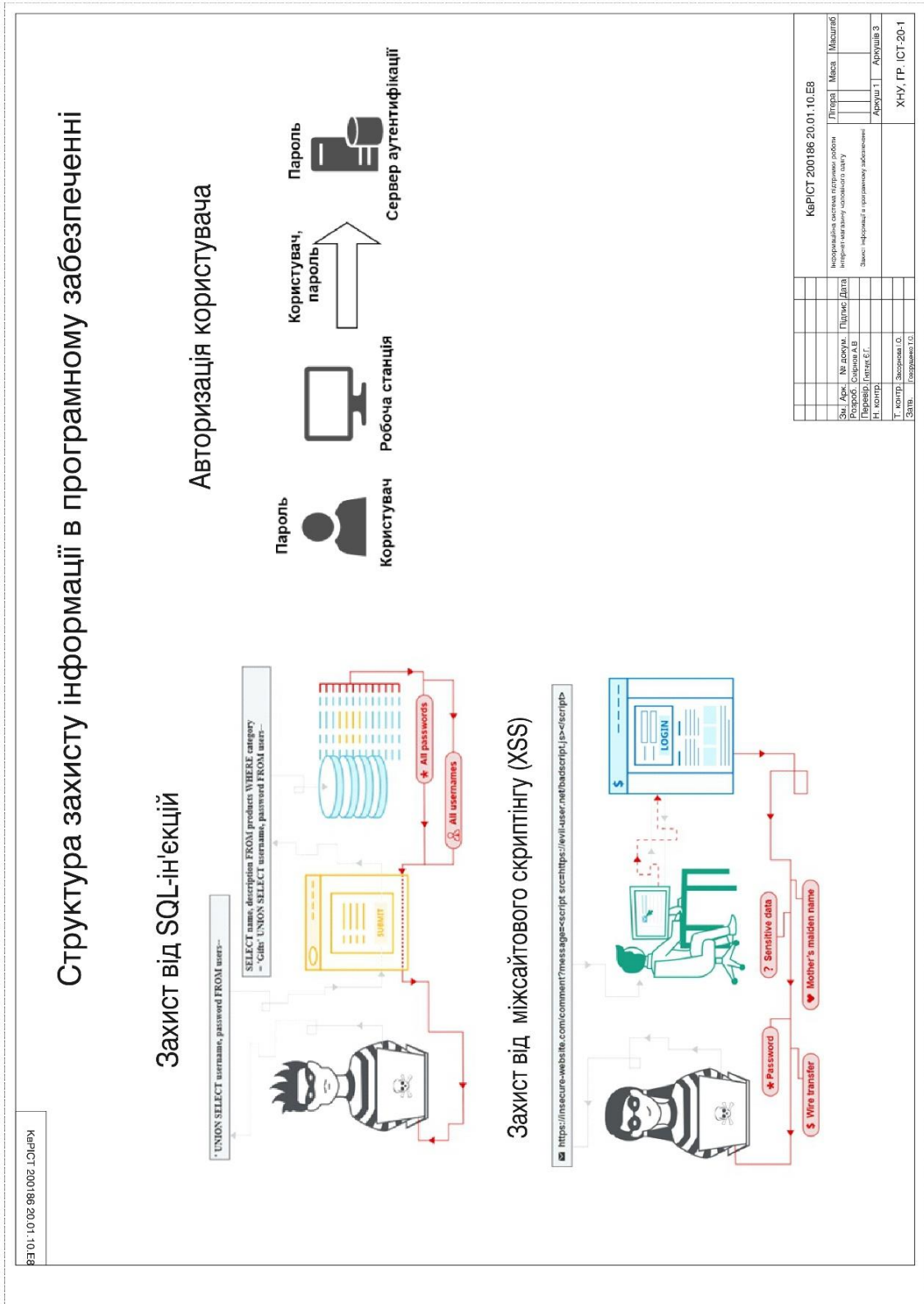
57. Пошуку продукції на сайті Rozetka.URL: <https://rozetka.com.ua/> (дата звернення: 08.06.2024).

58. Каталогів товарів на сайті Adidas.URL: <https://www.adidas.ua/> (дата звернення: 08.06.2024).

					КВРІСТ 200186 20.01.10 ПЗ	Арк. 71
Зм.	Арк.	№ докум.	Підпис	Дата		

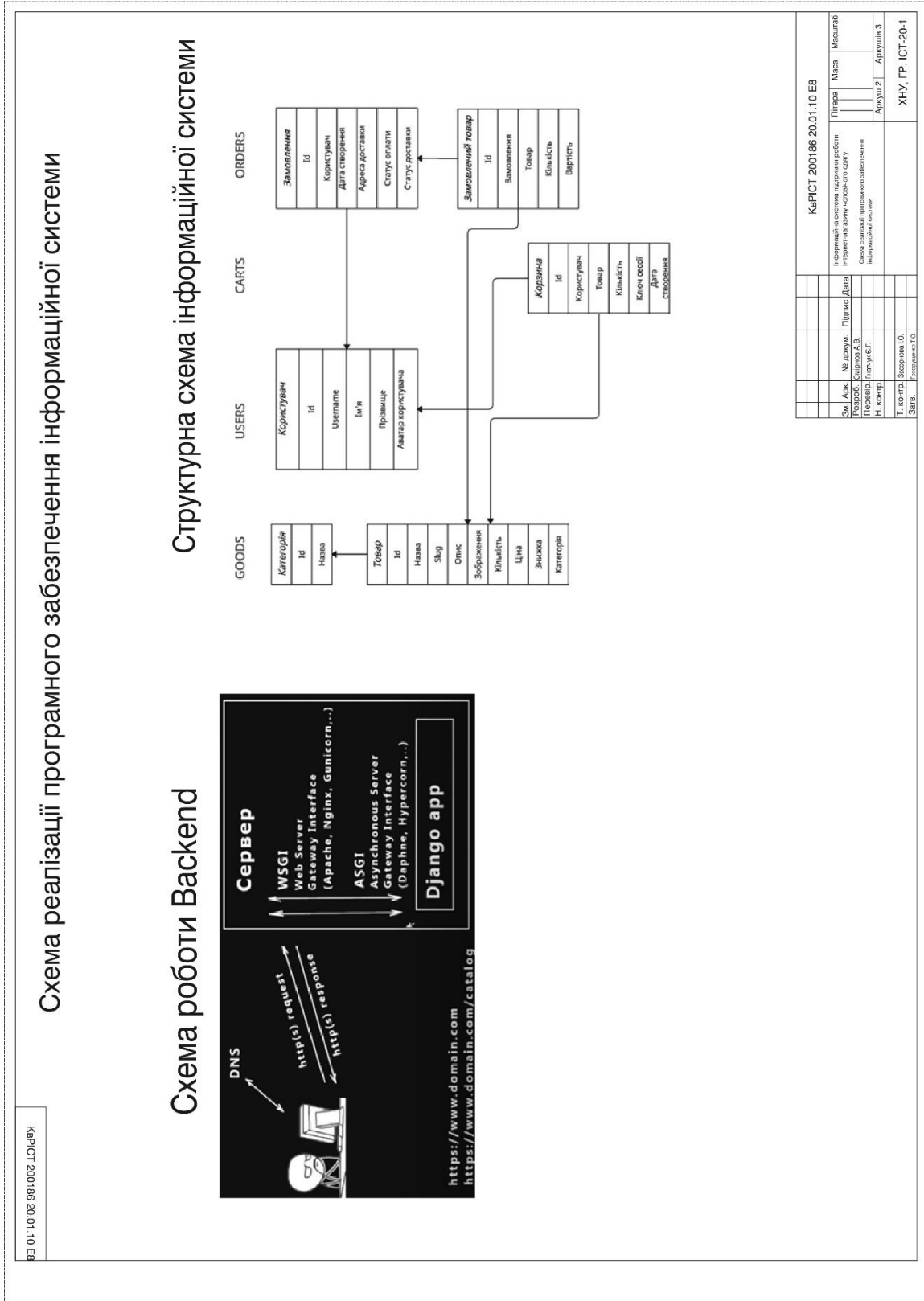
Додаток А (обов'язковий)

Копія креслення «Структура захисту інформації в програмному забезпеченні»



Додаток Б (обов'язковий)

Копія креслення «Схема реалізації програмного забезпечення інформаційної системи»



Ім'я користувача:
Кафедра КІ

ID перевірки:
1016343836

Дата перевірки:
10.06.2024 19:02:42 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.06.2024 20:52:18 EEST

ID користувача:
100005591

Назва документа: Смірнов_Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу

Кількість сторінок: 76 Кількість слів: 11160 Кількість символів: 87120 Розмір файлу: 4.56 MB ID файлу: 1016145314

5.76% Схожість

Найбільша схожість: 0.83% з Інтернет-джерелом (<https://science.lpnu.ua/sites/default/files/journal-paper/2023/jan/2975>).

5.42% Джерела з Інтернету 756 Сторінка 78

1.92% Джерела з Бібліотеки 107 Сторінка 81

0.35% Цитат

Цитати 2 Сторінка 82

Не знайдено жодних посилань

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 2

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en_US, ru_RU, ua_UA Помилки в документах: 9%

ID: 129517 Назва: БКР Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу Додано в БД: 2024-06-10 Автора: А.В.Смірнов Керівники: С.Г.Гнатчук Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	79385	612	1768 (2%)	28 (5%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РЕЦЕНЗІЯ НА ДИПЛОМНУ РОБОТУ

Дипломник: Смірнов Андрій Валдимович

Тема: Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг дипломної роботи:

Кількість листів креслень 3 ; кількість сторінок записки 72

1. Короткий зміст роботи та прийнятих рішень: Розроблена інформаційна система підтримки роботи інтернет-магазину чоловічого одягу надає користувачам зручний інтерфейс для перегляду, вибору та покупки товарів, а адміністраторам - інструменти для ефективного управління магазином.

2. Висновок про відповідність роботи дипломному завданню Дипломна робота відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі було проаналізовано предметну область. Основною метою такої систем є реалізація набору функцій, необхідних для успішної роботи інтернет-магазину. Однією з головних функцій є можливість замовлення товару, що реалізується через зручний та інтуїтивно зрозумілий інтерфейс користувача. Клієнти можуть легко додавати обрані товари до кошика, оформлювати замовлення, вводити необхідну інформацію для доставки та оплати. Усі ці дії здійснюються з мінімальними витратами часу та зусиль з боку користувача, що значно підвищує зручність та ефективність взаємодії з сайтом. Опираючись на аналіз, було сформовано перелік основних вимог до інформаційної системи підтримки роботи інтернет-магазину чоловічого одягу.

В другому розділі було проведено проектування та розроблена архітектура інформаційної системи. Розглянуті функційні, нефункційні, а також користувацькі вимоги.

В третьому розділі розроблено інтерфейс представленої інформаційної системи та представлена структура, що забезпечує ефективне зберігання та управління даними.

4. Позитивні сторони роботи: розроблена інформаційна система є успішним прикладом застосування сучасних вебтехнологій у створенні інтернет-магазину. Вона забезпечує високу зручність та ефективність взаємодії для кінцевих користувачів, а також надає потужні інструменти для адміністрування та управління бізнесом. Цей проєкт може слугувати основою для подальших розробок у сфері електронної комерції, демонструючи можливості та переваги використання Django у таких проєктах.

5. Негативні сторони роботи: В роботі не приділено достатньої уваги можливостям використання розробленої системи на мобільних пристроях.

6. Оцінка графічного оформлення та пояснювальної записки роботи: пояснювальна записка та листи креслення виконані згідно діючих вимог

7. Відгук про роботу в цілому: В загальному робота виконана на достатньому професійному рівні

8. Інші зауваження: -

9. Оцінка дипломної роботи:

Розглянувши позитивні та негативні сторони представленої дипломної роботи вважаю, що робота заслуговує оцінки «добре» 4,00 (С)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи)
Шинько О.М., доцент кафедри МБ(ХЗЛУ)

" " 2024р.



Завідувачу кафедри КПС
д-р.техн.наук, проф. Говорущенко Т. О.

Смірнова Андрія Вадимовича

ІІІІ здобувач вищої освіти

ФІТ, 4 курсу, групи ІСТ-20-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

30 травня 2024 року

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Інформаційна система підтримки роботи інтернет-магазину чоловічого одягу

Автор: Смірнов Андрій Вадимович

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Гнатчук Є.Г., к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

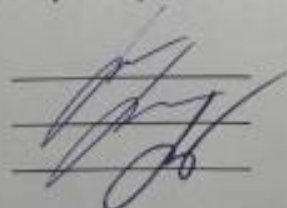
- 1) запозичення, розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформлені посилання;
- 3) окремі виявлені збіги є загальнознаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості Unicheck, складає 5.76% і адресується до 756 першоджерела; та системою Anti-Plagiarism складає 1%.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Є. Г. Гнатчук

Є. Г. Гнатчук

Т. О. Говорушенко