

Київський національний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

*«Розробка web-сайту інтернет-магазину продажу мотоциклів
2WHEELS»*

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

Кравченко Антона
Юрійовича

(підпис студента)

Науковий керівник
кандидат технічних наук,
професор

Пашорін Валерій
Іванович

(підпис керівника)

Гарант освітньої програми
кандидат технічних наук,
доцент

Цензура Микола
Олександрович

(підпис керівника)

Київ – 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення та кібербезпеки

Освітній ступінь бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Затверджую

Зав. кафедри інженерії
програмного забезпечення
та кібербезпеки
Криворучко О.В.
"20" жовтня 2020 р.

**Завдання
на випускню кваліфікаційну роботу студентові**

Кравченко Антону Юрійовичу

(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи «Розробка web-сайту інтернет-магазину продажу мотоциклів «2WHEELS»»

Затверджена наказом ректора від "30" жовтня 2020 р. № 3225

2. Строк здачі студентом закінченої роботи

3. Цільова установка та вихідні дані до роботи

Мета роботи розробка інтернет-магазину для розширення та підвищення прибутковості компанії з продажу мотоциклів «2WHEELS»

Об'єкт дослідження інтернет-магазину з продажу мотоциклів «2WHEELS»

Предмет дослідження розробка інтернет-магазину з продажу мотоциклів «2WHEELS»

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант	Підпис, дата	
№	(прізвище, ініціали)	Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ WEB-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS»

1.1 Концепція використання web-сайтів інтернет-магазинів

1.2 Веб-застосунки у сфері роздрібної торгівлі

1.3 Огляд існуючих інтернет-магазинів у сфері роздрібної торгівлі

1.4 Висновки до розділу 1

РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

2.1 Характеристика предметної області інтернет-магазину з продажу мотоциклів «2WHEELS»

2.2 Інфологічна та даталогічна модель даних

2.3 Вибір системи керування базою даних

2.4 Створення таблиць

2.5 Висновки до розділу 2

РОЗДІЛ 3. РЕАЛІЗАЦІЯ WEB-САЙТУ ІНТЕРНЕТ МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS»

3.1 Вибір середовища програмування

3.2 Моделювання веб-сайту

3.3 Розробка серверної частини веб-сайту

3.4 Розробка інтерфейсу

3.5 Висновки до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	21.09.2020	21.09.2020
2.	<i>Вступ та перелік літературних джерел</i>	14.12.2020	14.12.2020
3.	<i>Розділ 1. «Аналіз предметної області веб-сайту інтернет-магазину з продажу мотоциклів «2WHEELS»»</i>	19.02.2021	19.02.2021
4.	<i>Розділ 2. «Проектування та реалізація бази даних»</i>	05.03.2021	05.03.2021
5.	<i>Розділ 3. «Реалізація веб-сайту інтернет-магазину з продажу мотоциклів «2WHEELS»»</i>	10.04.2021	10.04.2021
6.	<i>Висновки</i>	24.04.2021	24.04.2021
7.	<i>Здача випускної кваліфікаційної роботи на кафедру (перша перевірка)</i>	14.05.2021	14.05.2021
8.	<i>Підготовка автореферату та презентації доповіді</i>	17.05.2021	17.05.2021
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>	18.05.2021 – 21.05.2021	21.05.2021
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.06.2021	01.06.2021
11.	<i>Здача прожитої випускної кваліфікаційної роботи на кафедру</i>	02.06.2021	02.06.2021
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «20» жовтня 2020 р.

8. Науковий керівник випускної кваліфікаційної роботи Пашорін В. І.
(прізвище, ініціали, підпис)

9. Гарант освітньої програми Цензура М. О
(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Кравченко А. Ю.
(прізвище, ініціали, підпис)

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There is no text or other markings on the paper.

Відмітка про попередній захист _____
(ПІБ, підпис, дата)

Випускна кваліфікаційна робота студента Кравченко А. Ю.
(прізвище, ініціали)

Гарант освітньої програми _____ Цензура М. О.
(прізвище, ініціали, підпис)

Завідувач кафедри _____ Криворучко О. В.
(підпис, прізвище, ініціали)

« _____ » 2021 г.

АНОТАЦІЯ

Дипломна робота на тему «Розробка web-сайту інтернет-магазину з продажу мотоциклів «2WHEELS»» містить 61 сторінку, 42 рисунки, та 6 додатків. Перелік посилань налічує 20 найменувань.

Метою випускної кваліфікаційної роботи є розробка інтернет-магазину для розширення та підвищення прибутковості компанії з продажу мотоциклів «2WHEELS».

Об'єкт дослідження: інтернет-магазин з продажу мотоциклів «2WHEELS».

Предмет дослідження: розробка інтернет магазину з продажу мотоциклів «2WHEELS».

У першому розділі проаналізовано існуючі у сфері роздрібної торгівлі веб-застосунки з продажу мотоциклів, де було визначено їх «сильні» та «слабкі» сторони, на основі зібраної інформації про сайти-аналоги було складено технічне завдання, описано основні вимоги до майбутнього проекту.

У другому розділі проаналізовано та обрано платформу для роботи з базою даних згідно сучасних вимог до подібних веб-застосунків, вибір обґрунтовано. Також було розроблено логічну та фізичну моделі, де зображено сутності, атрибути та відносини між ними в майбутній БД. На основі побудованих моделей було реалізовано та наповнено базу даних.

У третьому розділі обрано мову, середовище програмування, а також додаткові інструменти. За допомогою обраних технологій створено серверну частину сайту, додано динамічні подання до БД, а також розроблено адаптивний інтерфейс користувача.

ANNOTATION

The thesis on the topic "Development of the web-site of the online store for the sale of motorcycles" 2WHEELS "" contains 61 pages, 42 drawings, and 6 appendices. The list of links includes 20 items.

The purpose of the study: qualifying work is to develop an online store to expand and increase the profitability of the company selling motorcycles "2WHEELS".

Object of research: online store selling motorcycles "2WHEELS".

Subject of research: development of an web-site of online store selling motorcycles "2WHEELS".

The first section analyzes the existing web applications for the sale of motorcycles in the field of retail, where their "strengths" and "weaknesses" were identified, based on the collected information about similar sites, a technical task was prepared, the main requirements for the future project are described there.

The second section analyzes and selects a platform for working with the database in accordance with modern requirements for such web applications, the choice is justified. A logical and physical model was also developed, which depicts the entities, attributes and relationships between them in the future database. Based on the built models, a DB was implemented and filled.

The third section selects the language, programming environment and additional tools. With the help of selected technologies, the server part of the site was created, dynamic views were added to the database, and an adaptive user interface was developed.

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ WEB-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS».....	5
1.1 Концепція використання web-сайтів інтернет-магазинів.....	5
1.2 Веб-застосунки у сфері роздрібної торгівлі	8
1.3 Огляд існуючих інтернет-магазинів у сфері роздрібної торгівлі	13
1.4 Висновки до розділу 1.....	19
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ БАЗИ ДАНИХ	20
2.1. Характеристика предметної області інтернет-магазину з продажу мотоциклів «2WHEELS».....	20
2.2. Інфологічна та даталогічна модель даних	22
2.3. Вибір системи керування базою даних	27
2.4. Створення таблиць	29
2.5. Висновки до розділу 2	34
РОЗДІЛ 3 РЕАЛІЗАЦІЯ WEB-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS».....	35
3.1. Вибір середовища програмування.....	35
3.2. Моделювання веб-сайту	36
3.3. Розробка серверної частини веб-сайту.....	41
3.4. Розробка інтерфейсу	48
3.5. Висновки до розділу 3	57
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	60
ДОДАТКИ.....	62
Додаток А.....	62
Додаток Б	63
Додаток В	64
Додаток Г	65
Додаток Д.....	66
Додаток Ж.....	67

					КНТЕУ 121 07-11.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка web-сайту інтернет-магазину з продажу мотоциклів «2WHEELS»	Стадія	Аркуш	Аркушів
Зав. кафедри		Криворучко О.В.				Зміст	2	61
Керівник		Пашорін В.І.				Факультет інформаційних технологій, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розроб.		Кравченко А.Ю.						
					ЗМІСТ			

ВСТУП

Актуальність

Сьогодні жодна організація не може обійтися у своїй роботі без використання комп'ютерів, які з успіхом замінюють рутинну роботу, яка раніше виконувалася вручну, підвищуючи ефективність роботи будь-якої організації. Сфера використання ПК в даний час настільки широка, що немає такої області, де їх застосування було б недоцільним. Особливо важлива роль ПК для розвитку науки, зростання промислового виробництва, підвищення ефективності управління та розвитку підприємництва. В сучасних умовах комп'ютерні програми вирішують найрізноманітніші завдання за змістом і за народногосподарським значенням.

Є чимало шляхів, що дозволяють підвищити прибутковість магазину. Один з найбільш перспективних – створити власний інтернет-магазин.

Наявність власного веб-ресурсу з продажу дозволяє:

- автоматизувати основні процеси, що відбуваються на торговому підприємстві: облік товару, облік покупців, оформлення продажів, проведення платежів;
- отримувати своєчасні і точні звіти про поточні на підприємстві процесах;
- розширити охоплення аудиторії поза межі міст, в яких наявні торгові точки;
- створення власного впізнаваного «бренду», що саме-по-собі привертає більше потенційних клієнтів;
- підвищити якість та швидкість обслуговування клієнтів;
- посилити конкурентні переваги магазину;

					КНТЕУ 121 07-11.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка web-сайту інтернет-магазину продажу мотоциклів «2WHEELS»	Стадія	Аркуш	Аркушів
Зав. кафедри		Криворучко О.В.				В	3	61
Керівник		Пашорін В.І.				Факультет інформаційних технологій, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розроб.		Кравченко А.Ю.						
					ВСТУП			

- можливість відмовитись від зайвих витрат, таких як оренда магазину та супутні цьому витрати, наймання додаткового персоналу та ін.;
- можливість представляти ширший асортимент, ніж це могло б бути можливо при офлайн-торгівлі;

Метою випускної кваліфікаційної роботи є розробка інтернет-магазину для розширення та підвищення прибутковості компанії з продажу мотоциклів «2WHEELS».

Об'єкт дослідження: інтернет-магазин з продажу мотоциклів «2WHEELS»

Предмет дослідження: розробка інтернет магазину з продажу мотоциклів «2WHEELS»

Поставлена мета передбачає виконання наступних **завдань**:

- ознайомитися із загальною концепцією розробки веб-додатку з продажу мотоциклів компанії «2WHEELS»;
- провести аналіз аналогічних веб-ресурсів, сформувати ТЗ майбутнього проекту;
- здійснити постановку задачі на розробку веб-сайту з даної предметної області;
- побудувати моделі даних, для подальшої розробки БД, обрати програмні засоби для реалізації бази даних;
- обрати програмні засоби для розробки веб-сайту та розробити його згідно вимогами.

Методи дослідження: спостереження, порівняння та абстрагування інформації літературних джерел, метод моделювання.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		4

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ WEB-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS»

1.1 Концепція використання web-сайтів інтернет-магазинів.

З початком широкого застосування всесвітньої мережі інтернет у промислових галузях, одним з перших напрямків стала інтернет-торгівля. Почали з'являтися автоматизовані системи обліку товарів, інтернет майданчики з оголошеннями про продаж чи купівлю та інтернет-магазини як всесвітньо відомих компаній, так і представників малого та середнього бізнесу.

Сьогодні, в умовах ринкової економіки та вільного доступу до мережі інтернет, відбулося збільшення масштабів торгівлі, асортименту, потоку покупців, що значно перевищує ці показники за ті, що були ще декілька десятків років тому.

Без інтернет-магазину торгівельне підприємство позбавляється широкого ринку покупців, що в умовах сучасного високого рівня конкуренції може призвести до відсутності прибутків, і навпаки – навіть мале підприємство, за умови вдало налаштованого web-сайту зможе охоплювати більшу аудиторію потенційних покупців та показувати високу прибутковість.

Варто зазначити, що такі всесвітньо відомі корпорації як Amazon inc. та AliBaba Group починали свій шлях саме з створення інтернет-майданчиків, що надало їм можливість охоплювати максимальну кількість споживачів по всьому світові.

					КНТЕУ 121 07-11.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка web-сайту інтернет-магазину продажу мотоциклів «2WHEELS» АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ WEB-САЙТУ ІНТЕРНЕТ- МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS»	Стадія	Аркуш	Аркушів
Зав. кафедри		Криворучко О.В.				P1	5	61
Керівник		Пашорін В.І.				Факультет інформаційних технологій, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розроб.		Кравченко А.Ю.						

Інтернет-магазин (онлайн магазин) – це застосунок, який дає змогу продавати товари чи надавати послуги за допомогою всесвітньої мережі інтернет. Ключовою відмінністю онлайн торгівлі від традиційної є можливість представити велике різноманіття товарів та послуг без потреби значних витрат на трогове приміщення.

Серед основних функцій інтернет-магазину можна відзначити:

- надання інформації про представлені на ресурсі товари чи послуги;
- управління замовленнями;
- авторизація користувачів;
- проведення платежів (за умови підключення до платіжної системи);
- обробка статистичної інформації.

Вирішення проблеми створення якісного інтернет-застосунку онлайн-магазину неможливе без вивчення споживачів. Саме тому важливо інтегрувати можливість авторизації користувача, для збору їх даних для подальшого ведення внутрішньої статистики.

За основними сегментами діяльності сучасна торгівля має структуру, показану на рисунку 1.1.

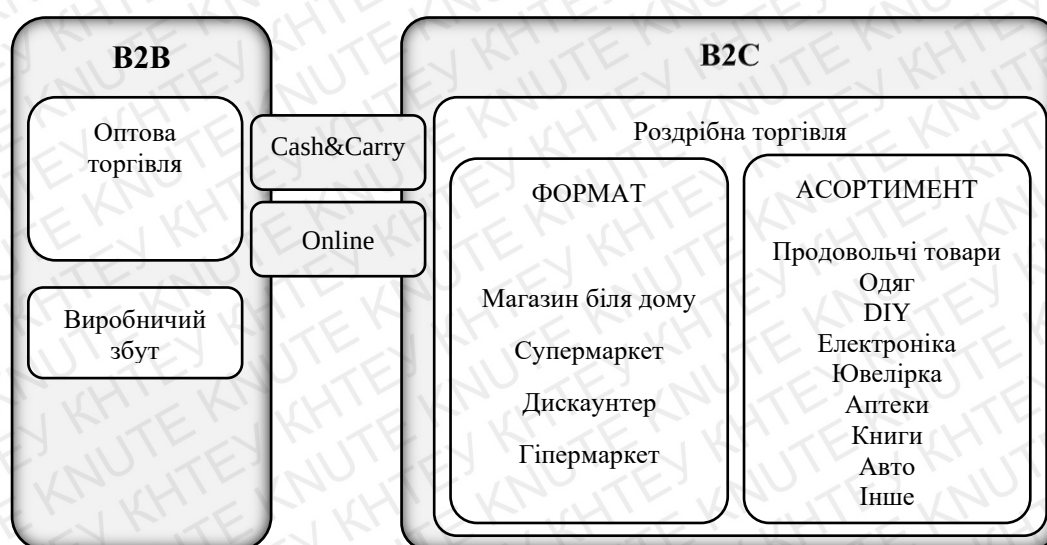


Рис. 1.1. Сегментування торгової галузі

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		6

Враховуючи, що інтернет магазин з продажу мотоциклів «2WHEELS» працює в сегменті роздрібної торгівлі, розглянемо особливості використання веб-застосунків саме в цьому аспекті.

Роздрібна торгівля є «пляшковим горлечком» у ланцюжку постачань, найвужчим місцем. У жодного магазину немає можливості представити покупцям весь асортимент поставлених або вироблених товарів. Причина полягає у фізичному обмеженні розмірів торгових приміщень. Саме тому, в умовах сучасного ринку, підприємству з продажу або виробництва необхідно мати власний веб-сайт.

Розробка веб-застосунку відбувається з урахуванням ряду специфічних вимог, які диктує асортимент роздрібної торгівлі. Зокрема, система роздрібної торгівлі засобами пересування (мотоциклами) має враховувати те, що більшість покупок на ринку авто там мотоциклів відбувається з використанням іноземної валюти, адже більша частина цих товарів імпортована – ввезена з закордону.

З тим же, в цілому, протягом останніх двадцяти років не відбувається значних змін у функціональному наповненні веб-застосунків з онлайн-торгівлі, адже незмінним залишається головне – її суть та принципи. Слід зазначити, що на сьогодні, західні аналоги, які були створені ще в 90-х роках минулого століття, перенесли деякі зміни в їх юзабіліті (зручності використання) та пройшли деяку оптимізацію внутрішніх процесів, включаючи безпеку інтернет-платежів, більш грамотні алгоритми реклами власних товарів та деякі косметичні зміни Front-end (клієнтська сторона веб-застосунку, інтерфейс) частини сайтів. Але в цілому, тенденції в розробці подібних ресурсів залишились аналогічними тим, що були ще близько двадцяти років тому і навряд понесуть кардинальні зміни у своїй суті.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		7

1.2 Веб-застосунки у сфері роздрібної торгівлі

Інтернет-магазин – це сайт, що торгує товарами за допомогою всесвітньої мережі інтернет. Насьогодні, вже сформовані базові вимоги до інтернет-ресурсу з продажів. В даній предметній області фундаментальні зміни в структурі не притаманні, тому всі веб-застосунки у сфері роздрібної торгівлі подібні або аналогічні за функціоналом та зовнішнім виглядом, що дає змогу більш ефективно підтримувати їх роботу та не вимагає вузько направлених розробників для обслуговування чи розробки подібних ресурсів. «Серцем» застосунку з роздрібною торгівлі у мережі інтернет є саме каталог товарів. Основні вимоги до структури сформовані у таблиці 1.1.

Таблиця 1.1

Структура каталогу товарів веб-застосунку з роздрібною торгівлі

Структура каталогу товарів	Групування	Товари мають бути згруповані за їх призначеннями, технічними особливостями, виробником та іншими характеристиками.
	Поля карток товарів	Кожна картка товару має містити назву товару, ціну, артикул, детальну інформацію про характеристики, властивості та виробника.
	Фільтрація та сортування	В каталозі товарів має бути перелік фільтрів та можливість сортування товарів.
	Візуалізація товарів	Товар має містити фото, бажано декілька; детальний добре оформлений опис; відео-огляд дуже бажаний, за його наявності.
	Пошук	Має бути реалізована можливість пошуку товарів за: • артикулом чи кодом (якщо система кодів товару або артикулів реалізована на сайті) • іменуванням; • виробником.

					КНТЕУ 121.07-11.БР	Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

Каталог товарів на веб-ресурсі з роздрібною торгівлі є його найголовнішою частиною, на етапі його розробки слід приділити увагу до дотримання базових структурних рекомендацій приведених у таблиці вище. Каталог має містити картки з товарами в яких буде зазначена основна і найбільш важлива потенційному покупцеві інформація до того, як він захоче переглянути детальний опис товару. Серед основних полів, які мають міститися на картці попереднього перегляду в каталозі обов'язково мають міститись фото товару, його іменування, виробник, ціна та найбільш важливі характеристики. Зважаючи на предметну область інтернет-магазину з продажу мотоциклів «2WHEELS» в картці товару варто зазначити основні характеристики двигуна: потужність (у кінських силах), час розгону від 0 до 100 км/год та середню витрату пального на 100 кілометрів пройденого шляху.

Другою, але не менш важливою частиною веб-застосунку з роздрібною торгівлі є сам процес покупки товарів. Для того, щоб у користувача склалось гарне враження про сайт, на якому він здійснює покупку, останній крок його взаємодії з ресурсом, тобто оформлення покупки, має бути максимально простим та логічним. Основні вимоги до структури процесу покупки наведені у таблиці 1.2.

Таблиця 1.2

Структура процесу покупки веб-застосунку з роздрібною торгівлі

Процес покупки	Облік клієнтів	Панель адміністратора має містити інформацію про покупців, їхні персональні дані та замовлення.
	Кошик покупця	Кошик має містити весь список товарів з зображеннями та іменуваннями, вказаною кількістю, ціною за одиницю та суму.
	Сторінка оформлення замовлення	Має бути логічно скомпонованим, з мінімально можливою кількістю input-елементів. Якщо всі замовлення оформлено вірно, клієнт має отримати про це сповіщення.

					КНТЕУ 121.07-11.БР	Аркуш
						9
Зм.	Аркуш	№ докум	Підпис	Дата		

Для того, щоб сайт був ефективний і міг підвищувати показники ефективності з точки зору комерції, він повинен мати інструменти для його адміністрування – admin-панель (панель адміністратора), для збору статистичних даних та частковим управлінням вмісту сайту. Адміністратор повинен мати доступ до всіх зроблених на сайті замовлень, інформації про клієнтів: ім'я клієнта, його контактні дані, дата народження та, за наявності, інформацію про не завершені покупки, якщо товар залишився в корзині без подальшого оформлення замовлення.

Сайт обов'язково має містити інформаційну сторінку та сторінку з контактною інформацією. Ці дані зазвичай прийнято виносити на окрему сторінку, особливо якщо існує декілька точок збуту товарів, але часто можна побачити, що подібна інформація зазначається в футері або хедері сайту. Оскільки інтернет-магазин з продажу мотоциклів «2WHEELS» має декілька центрів дистрибуції, то всі контактні дані будуть винесені на окрему сторінку. Основні вимоги до структури сторінки з контактною інформацією винесено до таблиці 1.3.

Таблиця 1.3

Структура інформаційної сторінки веб застосунку з роздрібною торгівлі

Структура інформаційної сторінки	Контактна інформація	Сторінка з контактною інформацією має містити номери телефонів, адреси та графік роботи служби підтримки і центрів точок дистрибуції.
	Інформація про доставку	Сторінка з інформацією про доставку має містити перелік логістичних компаній, з якими працює даний застосунок та умови, за яких доставка може здійснюватися.
	Інформація про оплату	Сторінка з інформацією про оплату має містити перелік всіх можливих способів сплати за товари.
	Інформація про гарантію	Сторінка з інформацією про гарантію має містити основні гарантійні умови, які надає компанія.

					КНТЕУ 121.07-11.БР	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

Розділ з інформацією є «точкою опори» для потенційного клієнта, адже так він може запевнитись у тому, що підприємство реальне і він дійсно отримає товар, за який сплатив.

Головна сторінка веб-додатку з роздрібною торгівлі – це, зазвичай, перше що бачить потенційний клієнт, який щойно перейшов на сайт. Тому вона має містити саме ті елементи та інформацію які можуть мотивувати користувача залишитись саме на цьому веб-сайті та здійснити покупку. В описі структури даної сторінки я буду опиратись на специфіку інтернет магазину з продажу мотоциклів. Серед основних та найбільш бажаних елементів головної сторінки можна відзначити:

- інформація про наявні акційні пропозиції та поставки нових товарів;
- інформація про компанію та привітання клієнта;
- інформація про переваги даного ресурсу над конкурентами;
- інформація про послуги, які надає компанія;
- інформація про бренди, дистрибуцією товарів яких займається компанія.

Також, дуже важливим, а іноді і основним для користувача аспектом інтернет-магазину є візуальна частина сайту – front-end. Основні вимоги до візуальної частини веб-застосунку з продажу мотоциклів зазначені в таблиці 1.4.

Таблиця 1.4

Вимоги до візуальної частини інтернет-магазину з продажу мотоциклів «2WHEELS»

Візуальна частина сайту	Загальні вимоги	Шапка сайту (header)	Повинен містити логотип компанії, навігацію з посиланнями на конкретні сторінки, кнопки з посиланням на корзину та на авторизацію або кабінет користувача, якщо авторизація виконана. Має бути виконаний в чорному або темно-сірому кольорі з синім відтінком, всі написи – білого кольору.
-------------------------	-----------------	----------------------	---

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		11

		Підвал сайту (footer)	Має містити дату створення ресурсу та його назву. Має бути такого ж кольору як і шапка сайту. Колір написів – сірий.
		Тіло сайту (body)	Має бути білого кольору. Всі елементи мають бути згруповані, для покращення адаптивності під різні типи дисплеїв.
Специфічні вимоги до сторінок	Головна сторінка	Має містити привітання та елементи за анімованим слайд-шоу, в якому буде відображена інформація про акційні пропозиції та інше.	
	Каталог	Має містити від чотирьох до п’яти рядів карток з товарами. Кожна з них має містити фото товару, ціну, основні характеристики, кнопки «Придбати» та «Детальніше». Мають бути виконані в тому ж кольорі що і тіло сайту, можливо з незначними змінами відтінку. Список фільтрів має бути розташований у горизонтальній орієнтації над картками за каталогу.	
	Інформаційна сторінка	Має містити інтерактивну карту, яка вказує на місцезнаходження конкретної точки дистрибуції або головного офісу компанії.	

Сучасний зовнішній вигляд є важливою частиною будь-якого сайту, але потрібно пам'ятати й про специфіку веб-сайтів з роздрібною торгівлі, основною задачею яких є утримання уваги клієнта для подальшої продажі. Важливо зазначити, що така комбінація кольорів була обрана не випадково, а тому, що дослідження у сфері маркетингу показали, що мінімалістичний дизайн з використанням білого кольору у тілі сайту та темних у футері та хедері сприяє утриманню уваги клієнта саме на товарах чи послугах представлених на сайті.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		12

За даними веб-аналітичної компанії SimilarWeb близько 47,46% користувачів заходять на веб-сайти з мобільних пристроїв, а в період з 2017 по 2019 роки трафік з мобільних пристроїв зріс на 30,6%. Тому дуже важливо, щоб веб-сайт був адаптивним, тобто коректно відображався на всіх сучасних типах дисплеїв – від смартфонів та планшетів до персональних комп'ютерів.

1.3 Огляд існуючих інтернет-магазинів у сфері роздрібної торгівлі

У сфері торгівлі наявність у підприємства інтернет-магазину вже давно не є перевагою над конкурентами, як це було ще пару десятиліть тому, на сьогодні це необхідність, без якої майже неможливий розвиток підприємства з точки зору комерції.

У сегменті інтернет-магазинів роздрібної торгівлі мотоциклами та іншими транспортними засобами достатньо велика конкуренція, деякі компанії навіть не мають достатньо великих точок дистрибуції, щоб в повному обсязі представляти такі специфічні та габаритні товари як транспортні засоби, а мають лише невеликі шоу-руми чи точки видачі, і спрямовані цілком на розвиток саме своїх веб-сайтів. Тому розглянемо деякі веб-застосунки у даній сфері.

Більшість подібних інтернет-магазинів є монобрендовими. Це означає, що на цих ресурсах представлені товари лише від одного виробника, найчастіше ці сайти є офіційними філіалами цих брендів.

Серед найбільш поширених та популярних брендів в Україні можна відзначити Yamaha, Ducati, Suzuki, Kawasaki, BMW. Сайти деяких з цих брендів ми і розглянемо.

1. Інтернет-магазин «Ducati» (рис. 1.2.) – це офіційний веб-сайт дилерської компанії з роздрібної торгівлі мотоциклами бренду Ducati. На сайті представлені всі актуальні моделі мотоциклів від виробника.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		13

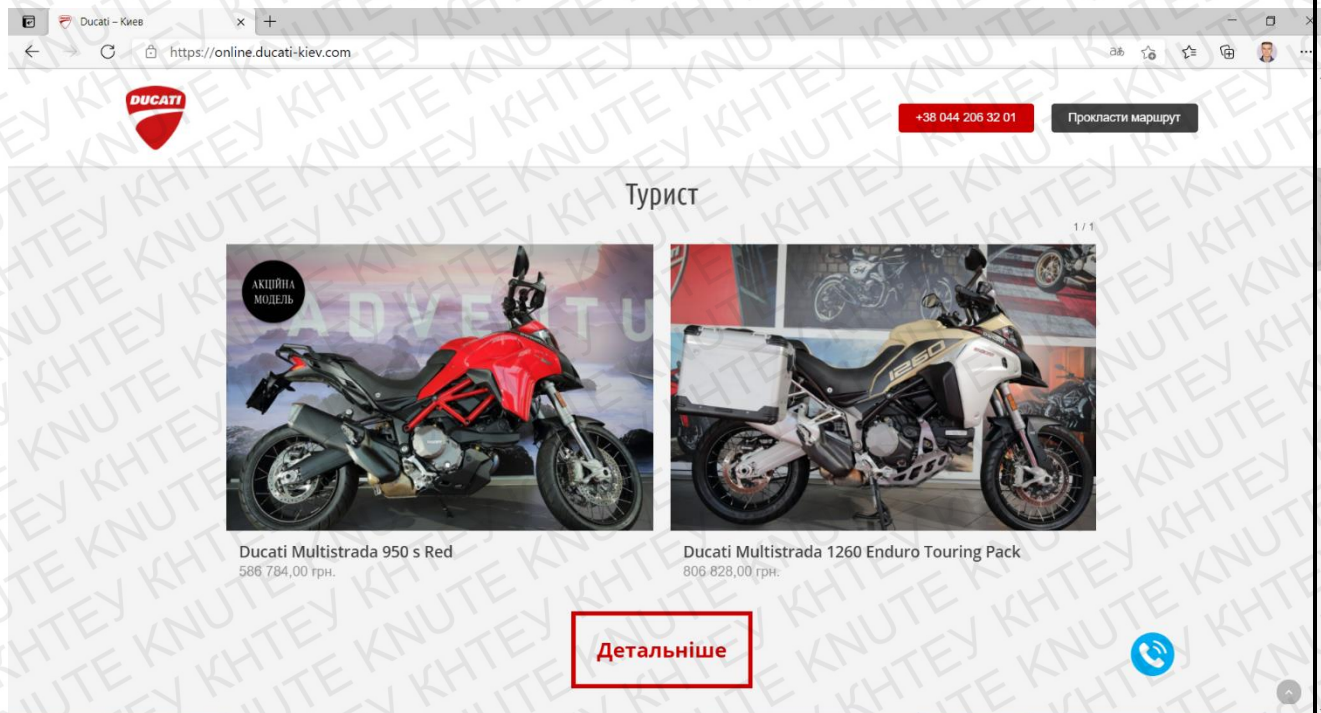


Рис. 1.2. Інтернет-магазин «Ducati»

Одразу можна відзначити мінімалістичний дизайн веб-сайту з мінімальним використанням кольорових елементів. Головна сторінка одночасно є і сторінкою-каталогом. Шапка сайту не має нагромадження елементів, на перший погляд може здатися, що сайт пустий або недопрацьований.

Серед сильних сторін даного ресурсу можна відзначити:

- дружній до користувача інтерфейс;
- мінімалістичний дизайн, використані шрифти читабельні;
- товари груповані по категоріям;
- простота процесу оформлення замовлення (рис. 1.3.)

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		14

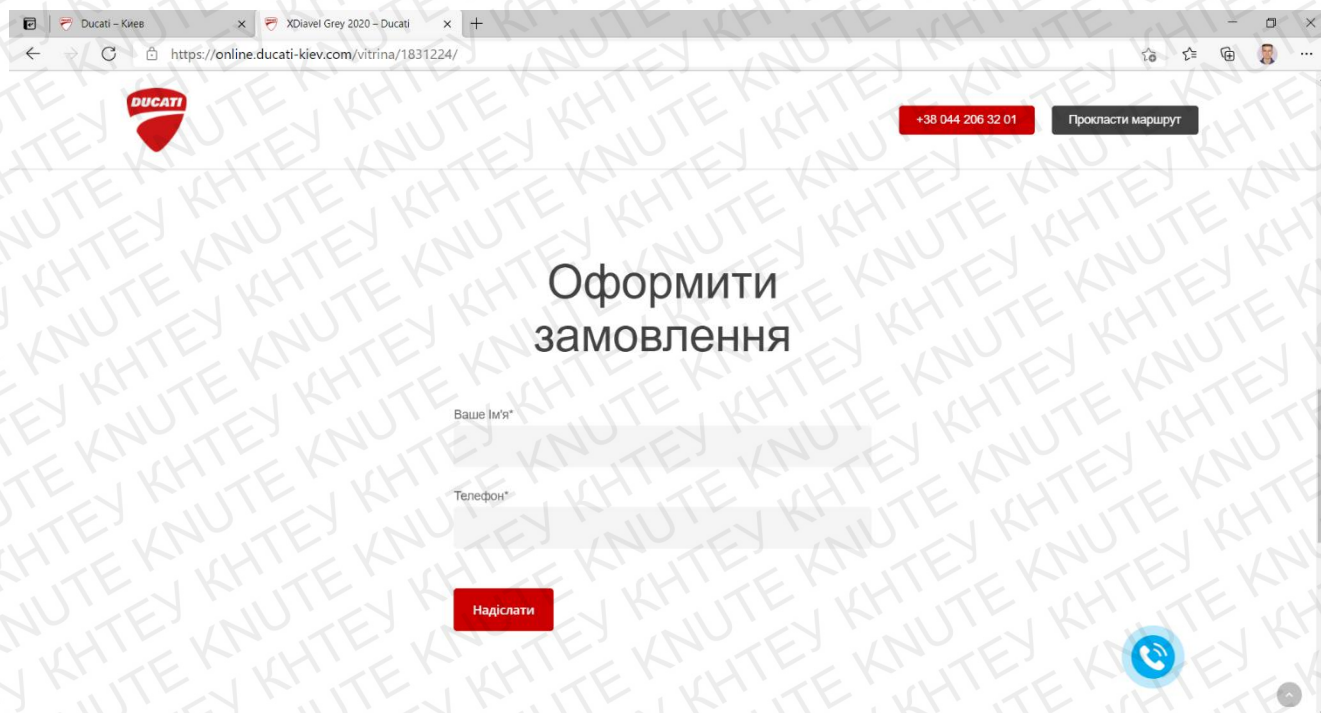


Рис. 1.3. Процес оформлення замовлення на сайті «Ducati».

Як можна помітити, велика частина функціональних та структурних «рекомендацій» дотримана в процесі розробки даного веб-сайту. Але важливо відзначити, що відсутність деяких функціональних аспектів сильно виділяє сайт «Ducati» на фоні конкурентів в негативну сторону. Серед основних недоліків можна зазначити:

- відсутність кошика товарів, що робить процес покупки більш ніж одного товару довшим, аніж він може бути;
- немає можливості авторизуватись на сайті, що ускладнює процес збору статистичних даних про цільову аудиторію сайту;
- відсутність особистого кабінету користувача;
- відсутня можливість пошуку по веб-сайту;
- немає великого каталогу товарів з можливістю фільтрувати позиції за потрібними користувачеві критеріями;
- невеликий асортимент;

					КНТЕУ 121.07-11.БР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

Ці недоліки роблять користування даним веб-сайтом не лише менш комфортним для користувача, але і для власників компанії, адже повністю відсутня можливість збору статистичних даних про покупців і можливість сповіщати клієнтів про акції та нові пропозиції шляхом рекламних листів на електронну пошту.

2. Інтернет-магазин «Yamaha» (рис. 1.4.) – це офіційний веб-сайт компанії, яка займається продажом мотоциклів, снігоходів, квадроциклів та спорядження бренду Yamaha.

Перше, що ми бачимо при вході на сайт – це головна сторінка, яка оформлена згідно з «рекомендаціями» щодо її структури. На цій сторінці міститься реклама власної продукції, інформація про оновлення модельного ряду. Загалом, дизайн мінімалістичний. Хоча даний сайт має значно більше елементів ніж попередньо розглянутий сайт «Ducati», він не виглядає загромадженим.

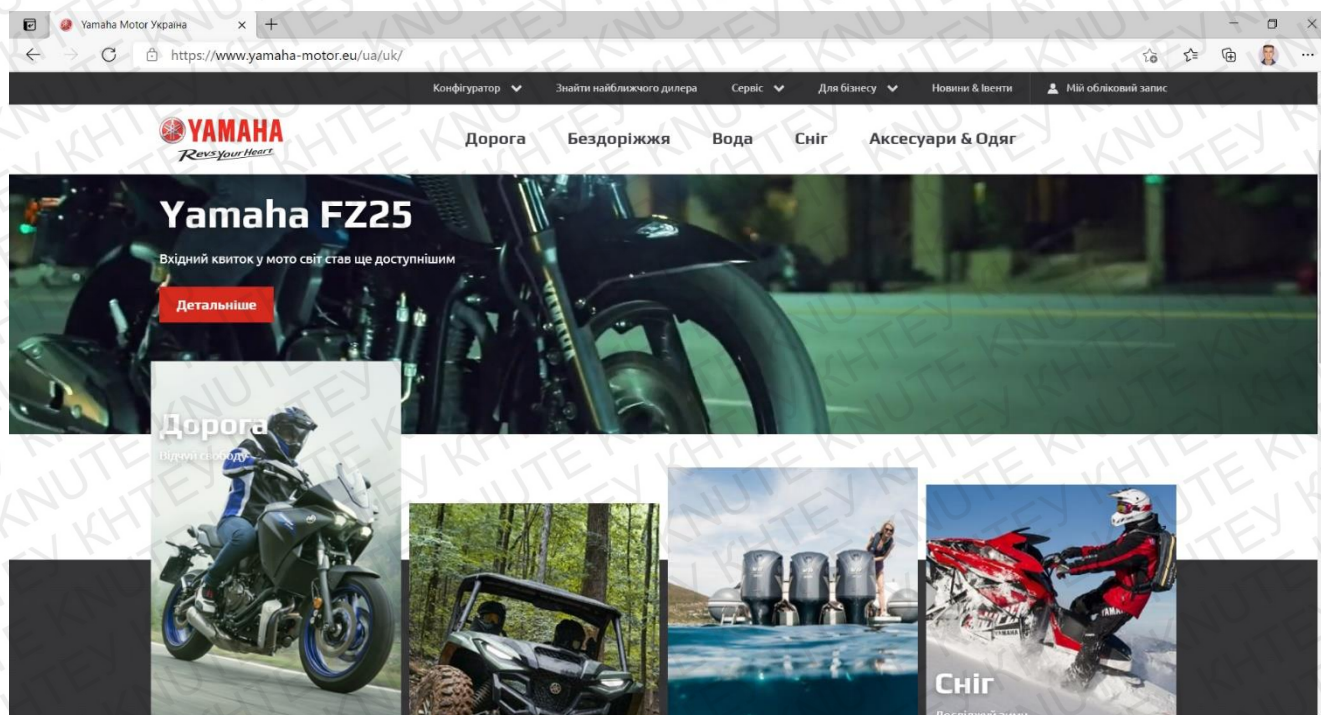


Рис 1.4. Головна сторінка веб-сайту «Yamaha».

Серед сильних сторін даного інтернет-магазину можна відзначити:

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		16

- яскравий, сучасний та все ще мінімалістичний інтерфейс;
- можливість авторизації на сайті;
- наявність особистого кабінету клієнта;
- чудова адаптивність на різних типах дисплеїв;
- наявність блоку з новинами сайту та компанії;
- великий асортимент;
- добре оформлений каталог товарів (рис. 1.5.);
- наявність добре оформленої інформаційної сторінки;
- можливість детальної конфігурації більшості товарів.

Розглянувши всі позитивні аспекти даного ресурсу, можна зробити висновок, що компанія багато уваги приділила як зовнішньому вигляду, так і оптимізації свого інтернет-магазину; всі анімації та елементи з відео працюють швидко та плавно як на персональних комп'ютерах, так і на мобільних пристроях.

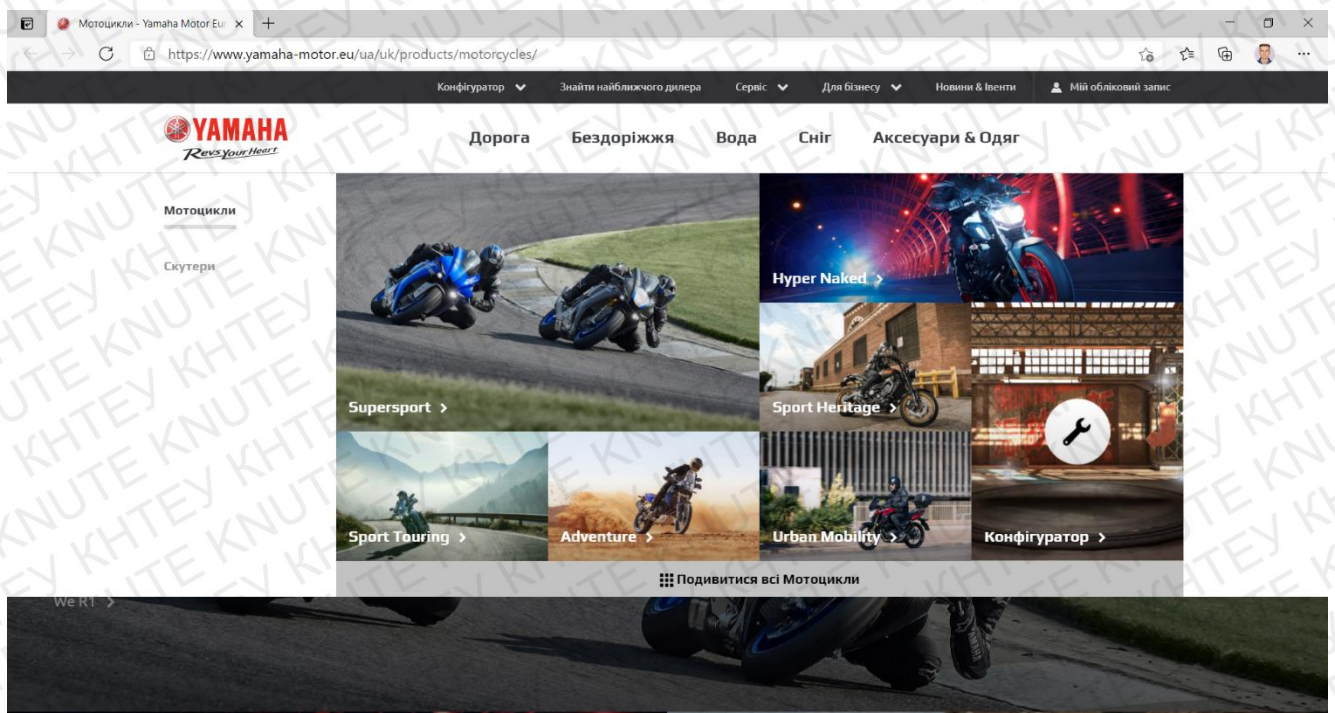


Рис. 1.5. Каталог товарів інтернет-магазину «Yamaha».

Хоча й робота з візуальною частиною даного сайту була проведена на «відмінно» і його можна вважати еталонним по зручності інтерфейсу та

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		17

зовнішньому вигляді, інтернет-магазин «Yamaha» має мінуси, перш-за-все, пов'язані з процесом покупки товарів та оформлення замовлення.

Процес покупки на даному сайті відбувається за такою послідовністю:

- вибір товару в каталозі, його комплектації та кольору;
- повідомлення оператора кол-центру даного ресурсу про артикул обраного вами товару;
- оператор повідомляє чи є даний товар в наявності, і якщо він є, то інформує про точку дистрибуції, де можна забрати товар.

Тобто ми бачимо, що не дивлячись на те, що сайт виглядає дуже сучасно, головна функція інтернет-магазину, а саме функція покупки товару, виконана дещо архаїчно. Всі наведені вище кроки можна з легкістю автоматизувати, що в свою чергу пришвидшить процес покупки, мінімізує людський фактор та покращить досвід від користування даним веб-сайтом.

Виходячи з цього, можна відокремити наступні недоліки структури інтернет-магазину «Yamaha»:

- відсутність спільного каталогу товарів, де можна було б продивитись усі товари, фільтруючи їх за потрібними критеріями вручну;
- відсутність пошуку товарів;
- відсутність кошика покупця;
- архаїчний, неавтоматизований процес оформлення замовлення;

Хоча даний сайт є позитивним прикладом того, як має виглядати сучасний інтернет-магазин, його все ж не можна назвати взірцевим, адже перелічені вище недоліки є, по суті, ключовими недоліками у функціоналі веб-ресурсу з роздрібною торгівлі, бо стосуються саме процесу покупки та оформлення замовлень клієнтів.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		18

1.4 Висновки до розділу 1

Ознайомившись із загальною концепцією використання веб-сайтів в галузі торгівлі, особливостями існуючих застосунків у сфері роздрібної торгівлі можна зробити висновок, що створення інтернет-магазину з продажу мотоциклів «2WHEELS» є досить трудомістким та відповідальним процесом. Адже реалізований сайт повинен відповідати всім критеріям, які ставляться до сучасних веб-ресурсів, та володіти визначеними властивостями та функціями.

На першому етапі було сформоване технічне завдання, з основними вимогами до інтерфейсу та функціоналу майбутнього веб-застосунку.

На другому етапі проведено аналіз та детальний розбір аналогічних веб-ресурсів з боку користувача, відзначено сильні сторони та недоліки розглянутих застосунків, зроблено висновки, які в подальшому вплинуть на процес розробки інтерфейсу та функціоналу майбутнього проекту.

					КНТЕУ 121.07-11.БР	Аркуш
						19
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ БАЗИ ДАНИХ

2.1. Характеристика предметної області інтернет-магазину з продажу мотоциклів «2WHEELS»

Інтернет-магазин з продажу мотоциклів «2WHEELS» – це веб-застосунок компанії «2WHEELS», що пропонує у продаж транспортні засоби, а саме мотоцикли найпопулярніших світових виробників.

Основною метою даного веб-застосунку є задоволення потреб споживача та надання своїх послуг поза межами міст, де представлені їхні торгові точки.

Предметна область бази даних для веб-застосунку з продажу мотоциклів «2WHEELS» являє собою магазин, що має асортимент товарів, поділений на категорії, виробників цих товарів та деяку інформацію про них.

До основних об'єктів даного інтернет-магазину можна віднести:

1) Товар – головний об'єкт магазину, що має наступні характеристики:

- id-номер товару;
- назву товару;
- виробника товару;
- ціну товару;
- категорія, до якої він відноситься (тип мотоцикла);
- потужність мотоциклу у кінських силах;
- об'єм двигуна у літрах;
- використання пального у літрах на 100 кілометрів;

					КНТЕУ 121 07-11.БР			
					Розробка web-сайту інтернет-магазину з продажу мотоциклів «2WHEELS» РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS»	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум	Підпис	Дата		P2	20	61
Зав. кафедри		Криворучко О.В.				Факультет інформаційних технологій, 4 курс, 7 група		
Керівник		Пашорін В.І.						
Гарант		Цензура М.О.						
Розроб.		Кравченко А.Ю.						

- опис товару;
- фото товару;

2) Виробник товару – це будь-який бренд з вироблення транспортних засобів, а саме мотоциклів, з яким співпрацює компанія «2WHEELS».

Виробник має наступні характеристики:

- назва виробника;
- країна походження;
- зображення логотипа виробника;
- інформація про виробника;

3) Споживач – це авторизований, або не авторизований користувач веб-застосунку, який може на сайті: змінювати свої облікові дані, переглядати та сортувати товари, додавати товари в корзину, оформлювати замовлення. Серед основних характеристик споживача, у випадку, якщо він виконав авторизацію можна відзначити всі його облікові дані:

- ім'я облікового запису (логін у вигляді псевдоніму або електронна пошта);
- пароль;
- персональні дані (дата народження, реальне ім'я, стать);

4) Продаж товару є основною характеристикою роботи інтернет-магазину.

1. Вимоги до БД по обліку товарів та покупок споживачів:

- можливість додавати нові товари, бренди, категорії;
- облік проданих товарів;
- облік замовлень;

2. Вимоги з продажу:

- інформація про проданий товар;
- інформація про споживача, необхідна для оформлення замовлення;
- інформація про групи товарів;

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		21

2.2. Інфологічна та даталогічна модель даних

У якості логічної моделі бази даних була обрана реляційна модель, оскільки саме реляційна модель використовується у більшості веб-застосунків з продажу товарів.

Проаналізувавши всю інформацію і вивчивши предметну область перейдемо до створення сутностей та заповнення їх атрибутами. Між сутностями встановимо зв'язки, між більшістю сутностей зв'язок необов'язковий неідентифікований – це означає, що значення атрибутів не може бути нуль.

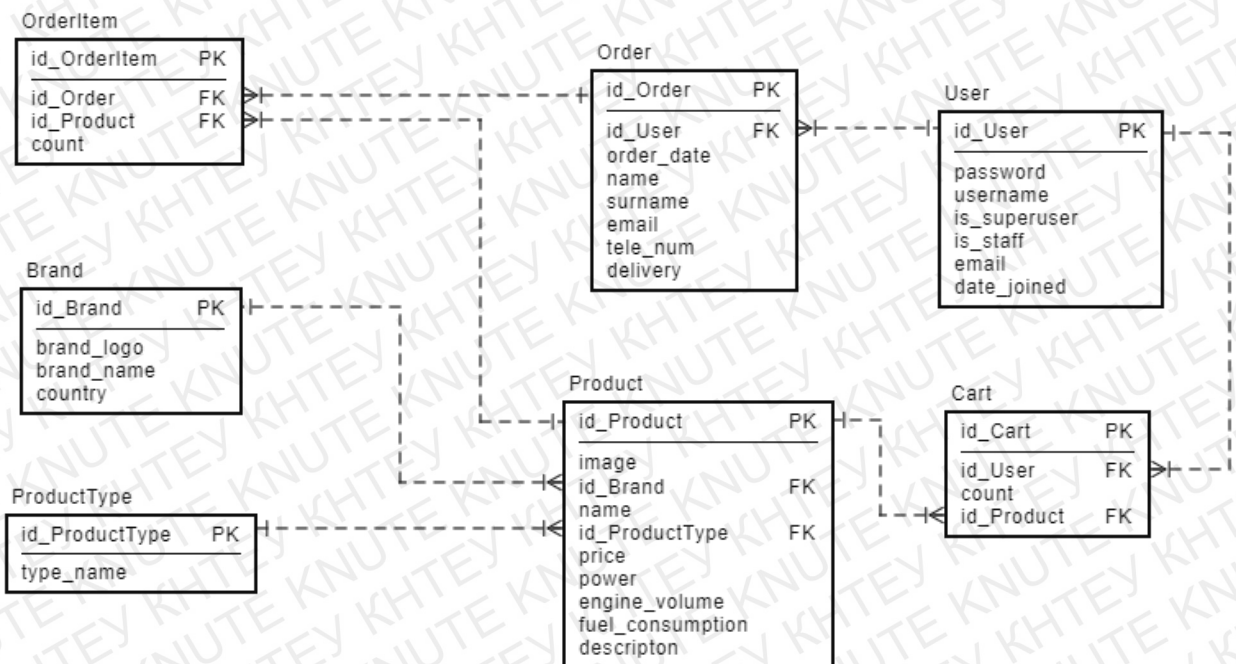


Рис. 2.1. Логічний рівень проектування ІЛМ

В таблицях 2.1 – 2.7 відображені основні атрибути створених сутностей та їхні характеристики.

Таблиця Виробник (Brand) (таблиця 2.1) знаходиться в 2НФ і 3НФ.

					КНТЕУ 121.07-11.БР	Аркуш
						22
Зм.	Аркуш	№ докум	Підпис	Дата		

Таблиця 2.1

Таблиця Виробник (Brand)

Ім'я поля	Тип даних	Розмір поля	Наявність пустого рядка	Наявність ключового поля
id_Brand	int	-	-	PK
brand_logo	varchar	100	NOT NULL	
brand_name	varchar	40	NOT NULL	
country	varchar	40	NOT NULL	

Таблиця Тип товару (ProductType) (таблиця 2.2) знаходиться в 2НФ і 3НФ.

Таблиця 2.2

Таблиця Тип товару (ProductType)

Ім'я поля	Тип даних	Розмір поля	Наявність пустого рядка	Наявність ключового поля
id_ProductType	integer	-	-	PK
type_name	varchar	40	NOT NULL	

Таблиця Товар (Product) (таблиця 2.3) знаходиться в 2НФ і 3НФ.

Таблиця 2.3

Таблиця Товар (Product)

Ім'я поля	Тип даних	Розмір поля	Наявність пустого рядка	Наявність ключового поля
id_Product	integer	-	-	PK
image	varchar	100	NOT NULL	
id_Brand	integer	-	NOT NULL	FK
name	varchar	50	NOT NULL	
id_ProductType	integer	-	NOT NULL	FK
price	integer	-	NOT NULL	
power	integer	-	NOT NULL	
engine_volume	integer	-	NOT NULL	
fuel_consumption	integer	-	NOT NULL	
description	text	1200	NOT NULL	

Таблиця Замовлення (Order) (таблиця 2.4) знаходиться в 2НФ і 3НФ.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		23

Таблиця Замовлення (Order)

Ім'я поля	Тип даних	Розмір поля	Наявність пустого рядка	Наявність ключового поля
id_Order	integer	-	-	PK
id_User	integer	-	NOT NULL	FK
order_date	datetime	-	NOT NULL	
name	varchar	30	NOT NULL	
surname	varchar	30	NOT NULL	
email	varchar	254	NULL	
tele_num	integer	-	NOT NULL	
delivery	varchar	100	NOT NULL	

Таблиця Замовлений товар (OrderItem) (таблиця 2.5) знаходиться у 2НФ і 3НФ.

Таблиця Замовлений товар (OrderItem)

Ім'я поля	Тип даних	Розмір поля	Наявність пустого рядка	Наявність ключового поля
id_OrderItem	integer	-	-	PK
id_Order	integer	-	NOT NULL	
id_Product	integer	-	NOT NULL	
count	integer	-	NOT NULL	

Таблиця Кошик (Cart) (таблиця 2.6) знаходиться у 2НФ і 3НФ .

Таблиця Кошик (Cart)

Ім'я поля	Тип даних	Розмір поля	Наявність пустого рядка	Наявність ключового поля
id_Cart	integer	-	NOT NULL	PK
id_User	integer	-	NOT NULL	FK
count	integer	-	NOT NULL	

					КНТЕУ 121.07-11.БР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

id_Product	integer	-	NOT NULL	FK
------------	---------	---	----------	----

Таблиця Користувач (User) (таблиця 2.7) знаходиться у 2НФ і 3НФ.

Таблиця 2.7

Таблиця Користувач (User)

Ім'я поля	Тип даних	Розмір поля	Наявність пустого рядка	Наявність ключового поля
id_User	integer	-	-	PK
password	varchar	128	NOT NULL	
username	varchar	150	NOT NULL	
is_superuser	boolean	-	NOT NULL	
is_staff	boolean	-	NOT NULL	
email	varchar	254	NULL	
date_joined	datetime	-	NOT NULL	

При виборі ключових полів було враховано, що вони повинні однозначно ідентифікувати записи визначеної сутності.

Ключовим моментом при визначенні структури бази даних є визначення типів відношень між атрибутами, які можуть бути 1:1 – один до одного, 1:Б – один до багатьох, Б:1 – багато до одного і Б:Б – багато до багатьох.

В таблиці 2.7 подано інформацію про відношення між об'єктами даної бази даних.

Таблиця 2.7

Відношення між об'єктами БД

Батьківська сутність	Дочірна сутність	Тип відношення	Примітка
Order	OrderItem	1:Б	В кожному замовленні може бути безліч товарів.
User	Order	1:Б	Кожен користувач може робити безліч замовлень.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

User	Cart	1:Б	Кожен користувач може додавати безліч товарів в кошик.
Brand	Product	1:Б	Кожен бренд може мати безліч товарів.
Product	Cart	1:Б	Кожен товар може бути в безлічі кошиків користувачів.
Product	OrderItem	1:Б	Кожен товар може бути замовлений безліч разів.
ProductType	Product	1:Б	Кожен тип товару може бути відповідним безлічі товарів.

Розроблені сутності та зв'язки між ними можна відобразити на фізичну модель спроектованої бази даних (рис. 2.2).

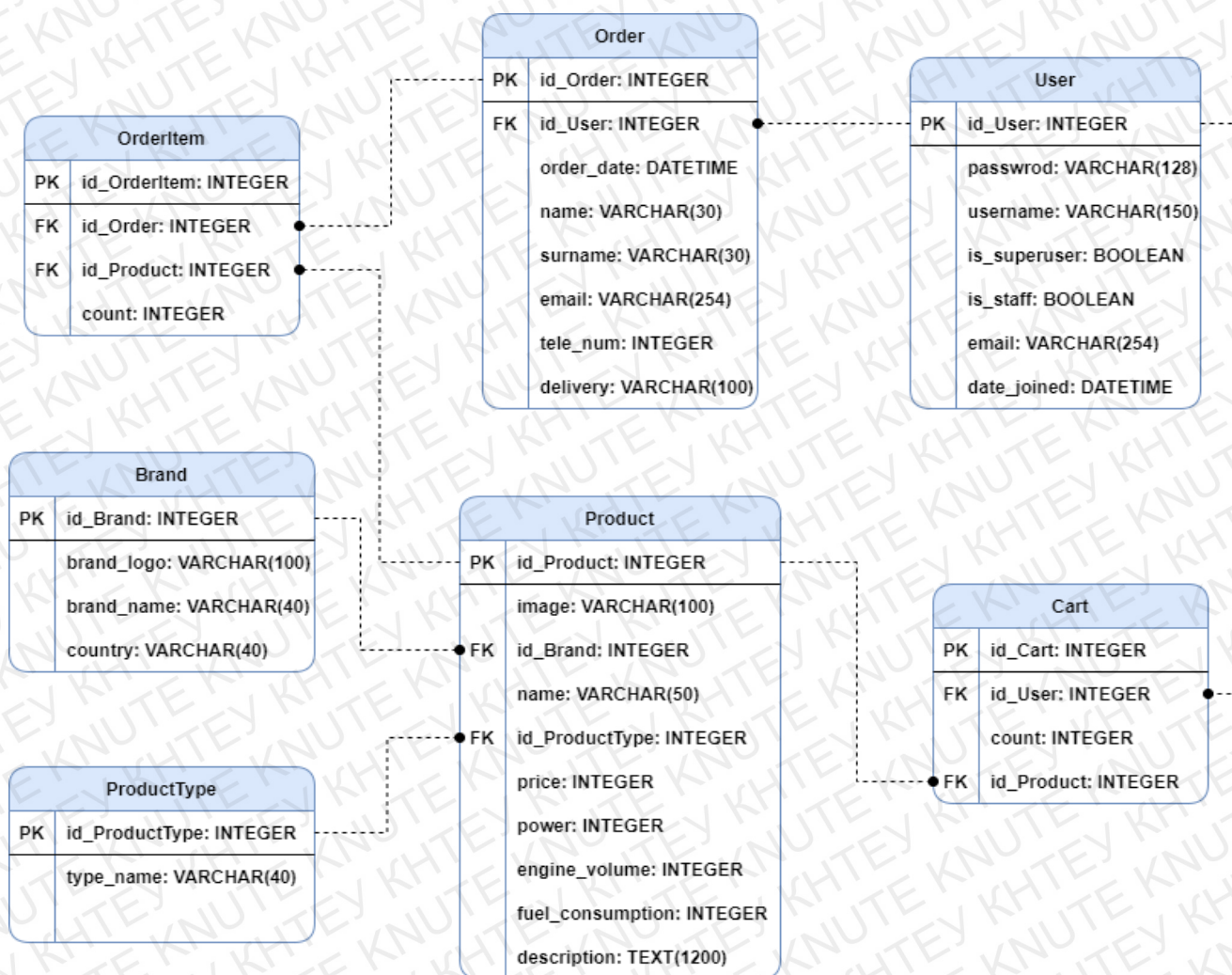


Рис. 2.2 Фізична модель бази даних

					КНТЕУ 121.07-11.БР	Аркуш
						26
Зм.	Аркуш	№ докум	Підпис	Дата		

Проаналізувавши всі зв'язки та ключі, для того щоб данні не були суперечними та сумісними, було створено каскадне оновлення даних. Суть каскадного оновлення даних полягає в тому, що при видаленні рядка із батьківської таблиці, усі пов'язані записи, що посилаються на первинний ключ видаляються і з дочірніх таблиць.

В таблиці 2.8 наведено правила оновлення посилальної цілісності.

Таблиця 2.8

Правила оновлення записів посилальної цілісності

№з/п	Батьківська сутність	Дочірна сутність	Обмеження
1	Order	OrderItem	Каскадне (cascade)
2	User	Order	Каскадне (cascade)
3	User	Cart	Каскадне (cascade)
4	Brand	Product	Каскадне (cascade)
5	Product	Cart	Каскадне (cascade)
6	Product	OrderItem	Каскадне (cascade)
7	ProductType	Product	Каскадне (cascade)

2.3. Вибір системи керування базою даних

Для створення бази даних в роботі обрано об'єктно-реляційну систему керування базами даних PostgreSQL 13.

Середовище PostgreSQL – це інтегроване середовище для доступу, налаштування, адміністрування, розробки всіх компонентів PostgreSQL Server і управління ними. У версії PostgreSQL 13 велике число графічних засобів поєднується з набором повнофункціональних редакторів скриптів для доступу розробників та адміністраторів з будь-яким рівнем знань до PostgreSQL Server.

Утиліта для безпосереднього керування компонентами БД в середовищі PostgreSQL має назву pgAdmin 4. Даний засіб являє собою графічний інтерфейс,

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		27

набір функцій для роботи з сервером, за допомогою якого ми можемо додавати, редагувати, видаляти об'єкти бази даних. Починаючи з PostgreSQL версії 13, pgAdmin 4 реалізований у вигляді веб-застосунку, який дозволяє керувати базою даних напряму з браузера (рис 2.3.).

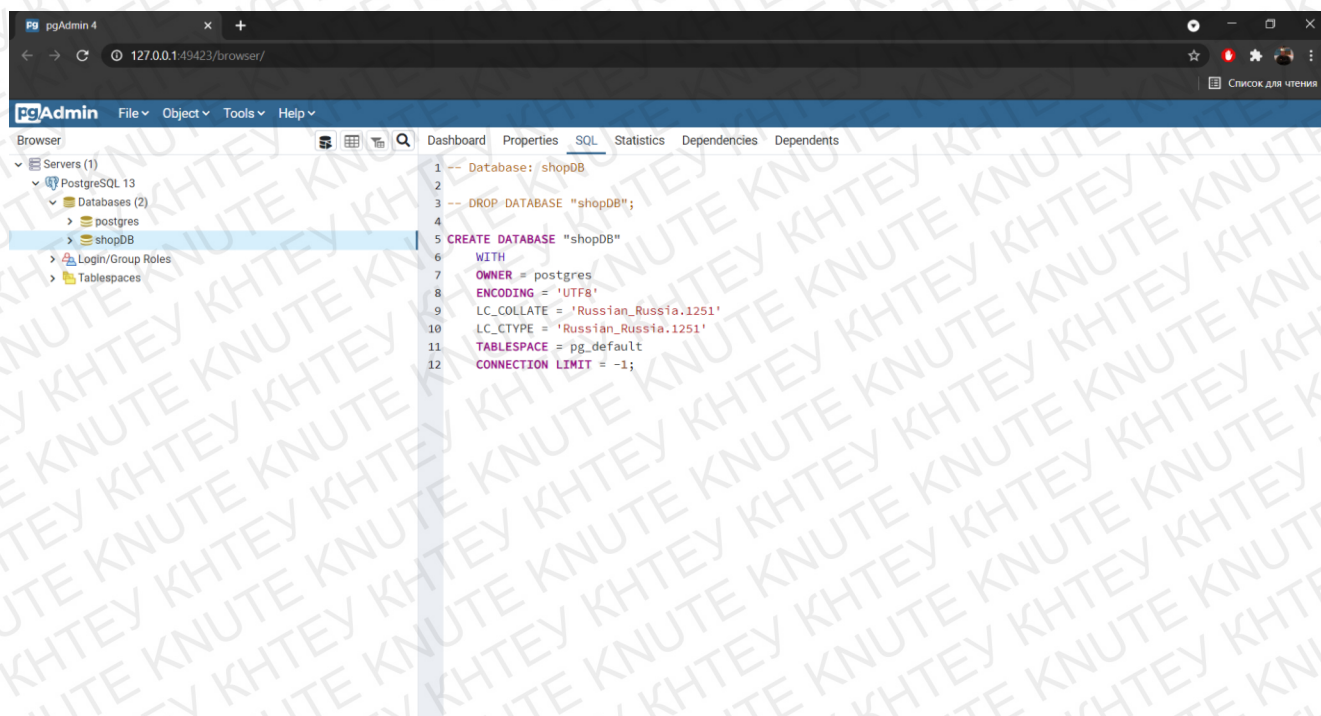


Рис. 2.3. Графічний інтерфейс утиліти pgAdmin 4

Середовище PostgreSQL 13 pgAdmin4 забезпечує наступні основні можливості:

- підтримує більшість адміністративних завдань для SQL Server;
- зручний та простий інтерфейс, що дозволяє максимально ефективно використовувати можливості PostgreSQL;
- швидке розвертання БД на видалених серверах, на платформі UNIX-систем;
- єдине інтегроване середовище для розробки та управління базою даних;
- має набір ефективних бібліотек для мови Python, яка була обрана для даного проекту;

Дана система керування базою даних працює із реляційною моделлю баз даних, тобто з моделлю, яка побудована у вигляді таблиць і зв'язків між ними. Дані, що

					КНТЕУ 121.07-11.БР	Аркуш
						28
Зм.	Аркуш	№ докум	Підпис	Дата		

зберігаються у цих таблицях є нормалізованими і описують певну предметну область. Всі запити створюються на мові Transact-SQL. Як вказано вище, PostgreSQL має ряд сумісних бібліотек для різних мов програмування, наприклад Python, що дозволяє створювати подання та запити до БД, керувати БД, на платформі PostgreSQL за допомогою синтаксису використовуваної мови програмування, в даному випадку – Python, а саме фреймворку Django.

Зважаючи на всі перелічені вище переваги і було обрано систему управління базами даних PostgreSQL 13.

2.4. Створення таблиць

Розробивши інфологічну та фізичну модель бази даних, ми отримали чітке уявлення про архітектуру, зв'язки між таблицями та їх атрибутами майбутньої БД.

Як було зазначено вище, завдяки бібліотекам Python для роботи з БД на платформі PostgreSQL, ми можемо створювати, видаляти, редагувати таблиці та їх поля за допомогою синтаксису обраної мови програмування. В даному випадку, була використана бібліотека psycopg2. Щоб розпочати роботу з БД, потрібно створити базу даних в засобі pgAdmin 4, та налаштувати її для роботи з Python проектом. Надалі переходимо до безпосереднього налаштування середовища розробки, для роботи з PostgreSQL та psycopg2, для створення бази даних shopDB, таблиць в ній, та заповнення їх даними. Для початку, в налаштуваннях проекту, файлі settings.py, потрібно вказати авторизаційні дані, з якими наш проект буде звертатись до БД, та бібліотеку для роботи саме з PostgreSQL (рис 2.4).

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29


```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'shopDB',
        'USER': 'postgres',
        'PASSWORD': 'password',
        'HOST': 'localhost',
        'PORT': '5432',
    }
}
```

Рис. 2.4. Авторизаційні дані для доступу до БД у файлі settings.py

Тепер, коли ми «з'єднали» БД з нашим проектом, можна перейти до створення таблиць, відповідно до раніше розроблених інфологічної та фізичної моделей, наповнення полів цих таблиць інформацією, згідно з встановленим типом даних, та утворення зв'язків між ними (рис. 2.5-2.7).

Заповнювати поля таблиць БД PostgreSQL, з якою ми працюємо в межах проекту Python + веб-фреймворк Django, ми можемо напряму, через панель адміністратора, яка встановлена в усіх Django-проектах за замовчуванням.

```
class Brand(models.Model):
    brand_logo = models.ImageField(null=True, blank=True, upload_to='images/')
    brand_name = models.CharField(max_length=40)
    country = models.CharField(max_length=40)

    class Meta:
        verbose_name = 'Виробник'
        verbose_name_plural = 'Виробникии'

    def __str__(self):
        return self.brand_name
```

Add Виробник

Brand logo: yamaha.png

Brand name:

Country:

Save and add another

Save and continue editing

SAVE

Рис. 2.5 Таблица Brand (Виробник)

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		30

Основною перевагою використання бібліотек для роботи з базами даних є відсутність потреби в досконалому володінні мовою Transact-SQL, адже для створення, редагування та видалення таблиць і подань використовується синтаксис мови програмування, на якій ведеться розробка проекту, а в комерційній розробці дозволяє скоротити витрати компанії на найманні спеціалістів з баз даних.

```
class ProductType(models.Model):
    type_name = models.CharField(max_length=40)

    class Meta:
        verbose_name = 'Тип'
        verbose_name_plural = 'Типи'

    def __str__(self):
        return self.type_name
```

Add Тип

Type name:

Кривізер

Save and add another

Save and continue editing

SAVE

Рис. 2.6. Таблица ProductType (Типы товаров)

```
class Product(models.Model):
    image = models.ImageField(null=True, blank=True, upload_to='images/')
    name = models.CharField(max_length=50)
    brand = models.ForeignKey(Brand, on_delete=models.CASCADE)
    product_type = models.ForeignKey(ProductType, on_delete=models.CASCADE)
    price = models.IntegerField(default=0)
    power = models.IntegerField(default=0)
    engine_volume = models.IntegerField(default=0)
    fuel_consumption = models.IntegerField(default=0)
    description = models.TextField(max_length=1000)
```

					Аркуш
					КНТЕУ 121.07-11.БР
Зм.	Аркуш	№ докум	Підпис	Дата	31

Image:

Name:

Brand:  

Product type:  

Price:

Power:

Engine volume:

Fuel consumption:

Description:

Круїзер, що виробляється компанією Yamaha з 1985 року, відомий завдяки своєму потужному двигуну V4 і оригінальному дизайну. У 2008 році, назву V-Max було замінено на VMAX.

Для розробки дизайну мотоцикла був запрошений англійський дизайнер Джон Рід. Взявши за основу Yamaha Venture Royale, Рід створив потужний круїзер.

Відразу після випуску першої моделі в 1985 році V-Max здобув визнання критиків і був названий "Мотоцикл року" за версією Cycle Guide. Незважаючи на продажі не тільки в Японії, але і за кордоном, V-Max випускався з незначними модифікаціями порівняно з першою випущеною моделлю. Швидкий розгін мотоцикла не можна було не відзначити, проте також не

Рис. 2.7 Таблиця Product (Товар)

Решта таблиць, такі як Cart (Кошик), Order (Замовлення), OrderItem (Замовлений товар) та User (Користувач), будуть заповнені користувачем при здійсненні певних операцій: додавання товару до кошика, оформлення замовлення та реєстрації відповідно. Створення перелічених таблиць та їхніх полів продемонстровано нижче (рис. 2.8 – 2.10).

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32


```
class Order(models.Model):
    ODESSA = 'Odessa'
    KIEV = 'Kiev'
    DNEPR = 'Dnepr'
    DELIVERY_TYPES = (
        (ODESSA, 'Одеса'),
        (KIEV, 'Київ'),
        (DNEPR, 'Дніпро'),
    )
    user = models.CharField(null=True, blank=True, max_length=30)
    order_date = models.DateTimeField(auto_now_add=True)
    name = models.CharField(max_length=30, verbose_name='Ім'я', null=True)
    surname = models.CharField(max_length=30, verbose_name='Фамілія', null=True)
    email = models.EmailField(blank=True, null=True)
    tele_num = models.IntegerField(default=380)
    delivery = models.CharField(max_length=100,
                                choices=DELIVERY_TYPES,
                                default=KIEV)

    def __str__(self):
        return f"Замовлення №{str(self.id)}"
```

Рис. 2.8. Таблиця Order (Замовлення)

```
class OrderItem(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE)
    item = models.ForeignKey(Product, on_delete=models.CASCADE)
    count = models.IntegerField(default=0)
```

Рис. 2.9. Таблиця OrderItem (Замовлений товар)

```
class Cart(models.Model):
    user = models.CharField(null=True, blank=True, max_length=30)
    item = models.ForeignKey(Product, on_delete=models.CASCADE)
    count = models.IntegerField(default=0)
```

Рис 2.10. Таблиця Cart (Кошик)

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		33

2.5. Висновки до розділу 2

Проаналізувавши предметну область інтернет-магазину з продажу мотоциклів «2WHEELS», було здійснено постановку завдання, де виділено вхідні дані та результати роботи бази даних.

На етапі проектування бази даних розроблено концептуальну, інфологічну та фізичну моделі, які стали основою під час її реалізації.

На етапі безпосередньої розробки бази даних, було обрано платформу, на основі якої була розроблена та заповнена база даних – PostgreSQL. Всі поля та зв'язки між таблицями було створено згідно з розробленими на етапі проектування моделями.

Таким чином, розроблена база даних складається із семи таблиць, які в подальшому будуть доповнені динамічними поданнями в процесі розробки архітектури проекту.

					КНТЕУ 121.07-11.БР	Аркуш
						34
Зм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS»

3.1. Вибір середовища програмування

PyCharm – інтегроване середовище професійної розробки, що входить до лінійки продуктів, що створені на основі середовища IntelliJ IDEA від компанії JetBrains. Можливості даного програмного засобу дозволяють розробляти як консольні додатки та програми інтерфейсом, так і великі проекти й веб-додатки на основі Python Django, Django REST Framework.

PyCharm включає в себе «розумний» редактор коду, який слідкує, щоб код на мові Python відповідав стандарту PEP-8, всі змінні в коді мали коректні імена, а структура проекту була цілісною. Крім того, в даному середовищі є вбудований відладчик, інструменти для зручної роботи з рір – системою керування пакетами на мові Python та синхронізація з Git – системою контролю версій. Крім .py (python) файлів, дане середовище дозволяє ефективно редагувати HTML документи, .css файли, файли на мові JavaScript та її популярних веб-фреймворків, зручно «підсвічує» синтаксис вище перелічених мов та слідкує за чистотою написаного коду.

В середовищі PyCharm наявна можливість створювати віртуальне середовище VENV, що дозволяє зберігати всі потрібні пакети, модулі та бібліотеки безпосередньо в директорії проекту, що є дуже корисною функцією при роботі з

					КНТЕУ 121 07-11.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка web-сайту інтернет-магазину з продажу мотоциклів «2WHEELS» РЕАЛІЗАЦІЯ ВЕБ-САЙТУ ІНТЕРНЕТ-МАГАЗИНУ З ПРОДАЖУ МОТОЦИКЛІВ «2WHEELS»	Стадія	Аркуш	Аркушів
Зав. кафедри		Криворучко О.В.				РЗ	35	61
Керівник		Пашорін В.І.				Факультет інформаційних технологій, 4 курс, 7 група		
Гарант		Цензура М.О.						
Розроб.		Кравченко А.Ю.						

декількома проектами, де встановлені одні і ті ж модулі, але різних версій, які можуть між собою конфліктувати.

Для розробки даного проекту було використано PyCharm Professional, яка є повнофункціональним, професійним розширюваним середовищем розробки для створення сучасних додатків на мові Python – компанія JetBrains надає безкоштовний доступ студентам вищих навчальних закладів.

3.2. Моделювання веб-сайту

Зазвичай веб-сайти являють собою складні динамічні структури. На їх розробку виділяють порівняно короткий проміжок часу, щоб сайт як можна швидше почав працювати та давати ефект. Іноді програмісти починають розробку сайту, відразу починаючи писати код, додавати таблиці та моделі до бази даних по мірі прогресу розробки, навіть не обдумавши в деталях як має працювати кінцевий продукт. У результаті архітектура проекту починає розвиватись у абсолютно непередбачуваному напрямку. Але за допомогою моделювання та чіткого підходу до розробки можна забезпечити більш високу продуктивність роботи програміста та гарантувати, що створену систему буде просто модифікувати та підтримувати у майбутньому.

UML (Unified Modeling Language) – це уніфікована мова моделювання, яка використовується для об'єктно-орієнтованого аналізу та опису систем, а отже добре підходить для опису архітектури нашого майбутнього проекту. Суть UML полягає в візуалізації сутностей системи за допомогою діаграм.

Для того, щоб забезпечити продуктивну роботу та цілісну архітектуру майбутнього проекту цілком вистачить діаграми діяльності (activity diagram), діаграми варіантів використання (use case diagram), діаграми станів (state chart diagram) і діаграми класів (class diagram).

					КНТЕУ 121.07-11.БР	Аркуш
						36
Зм.	Аркуш	№ докум	Підпис	Дата		

Діаграма класів (class diagram) (рис. 3.1) – слугує для зображення статичної структури моделей системи, їх атрибутів та відносин між ними. Даний тип діаграми використовується для:

- зазначення ролей і обов'язків сутності, яка забезпечує поведінку системи;
- фіксації структури класів та зв'язків між ними, що в ході проектування формує системну архітектуру проекту.

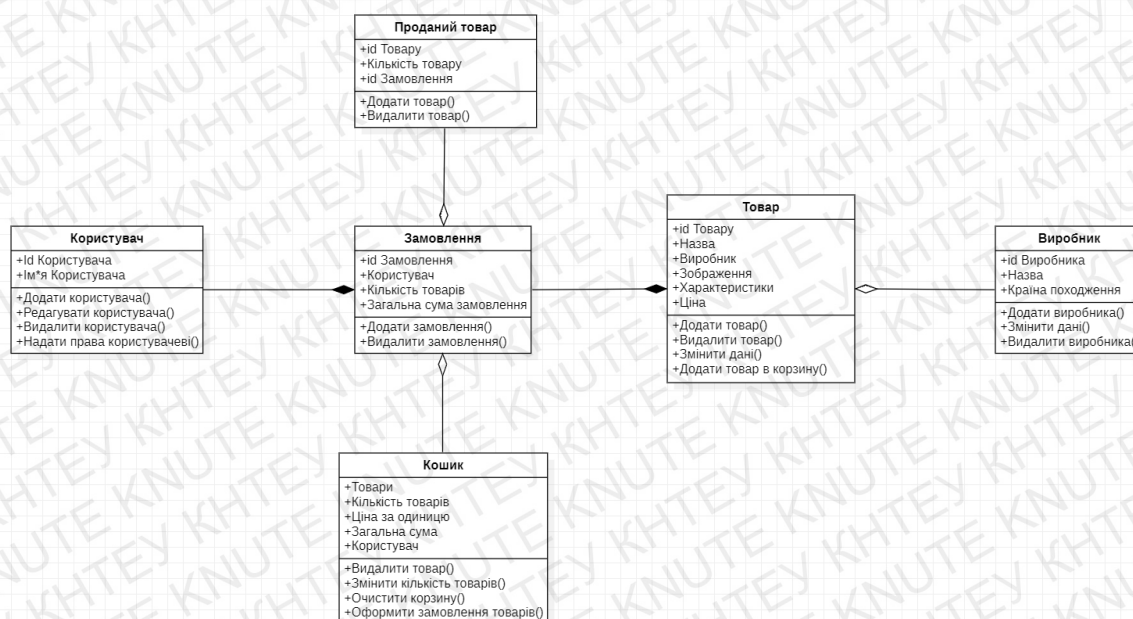


Рис. 3.1 Діаграма класів (class diagram) інтернет-магазину з продажу мотоциклів «2WHEELS»

Діаграма станів (use case diagram) (рис. 3.2) використовується для зображення поведінки системи з огляду на акторів – будь-яких сутностей, що взаємодіють із системою ззовні. Кожен варіант використання визначає набір дій, що виконує система при діалозі з актором.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		37

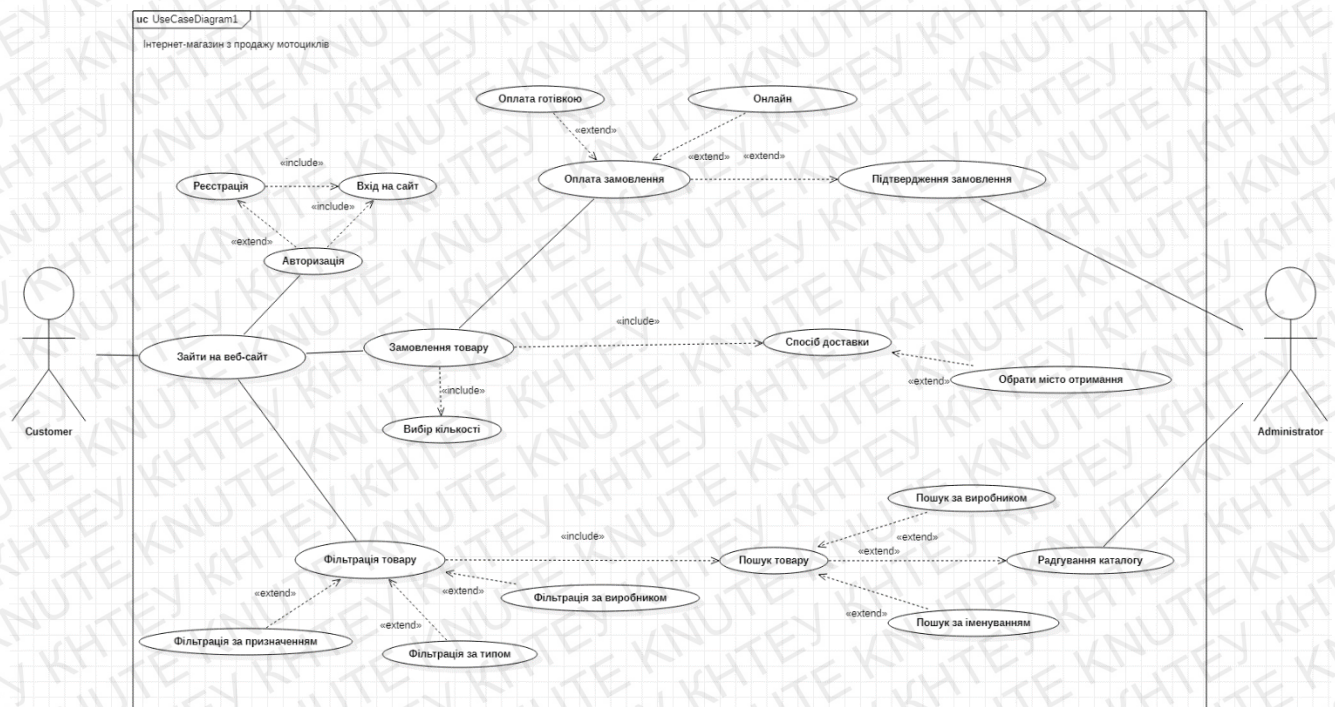


Рис. 3.2. Діаграма варіантів використання (use case diagram) інтернет-магазину з продажу мотоциклів «2WHEELS»

Діаграма діяльності (activity diagram) (рис. 3.3.) описує логіку процедур та потоки робіт. Зважаючи на предметну область сайту інтернет-магазину з продажу мотоциклів «2WHEELS», в даному випадкові, на діаграмі діяльності відображено переходи від одного діяльності до іншої в двох найголовніших аспектах роботи подібних ресурсів: доступ користувача до особистого кабінету та сам процес покупки товару.

Такі аспекти як перехід до сторінки з контактною інформацією, пошук товару, зміна авторизаційних даних, фільтрація та сортування товару розглянуті не будуть через їхню структурну елементарність, тож з точки зору моделювання не вимагають особливої уваги.

					КНТЕУ 121.07-11.БР	Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата		

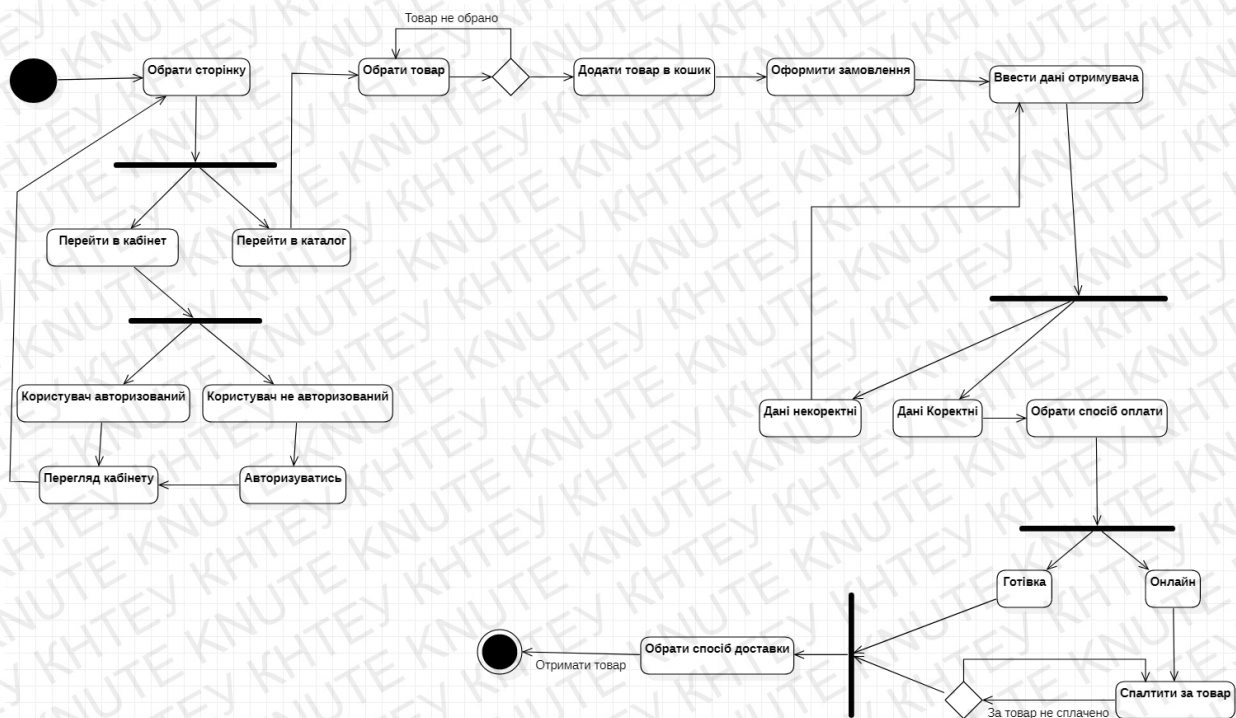


Рис. 3.3. Діаграма діяльності (activity diagram) інтернет-магазину з продажу мотоциклів «2WHEELS»

Діаграма станів (state chart diagram) (рис. 3.4.) використовується для опису поведінки системи протягом виконання тієї чи іншої операції. В даному випадкові буде розглянуто діаграму станів процесу покупки товарів в інтернет-магазині з продажу мотоциклів «2WHEELS», адже процес покупки є і найбільш архітектурно складним, і водночас, найголовнішим аспектом в даній предметній області.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		39

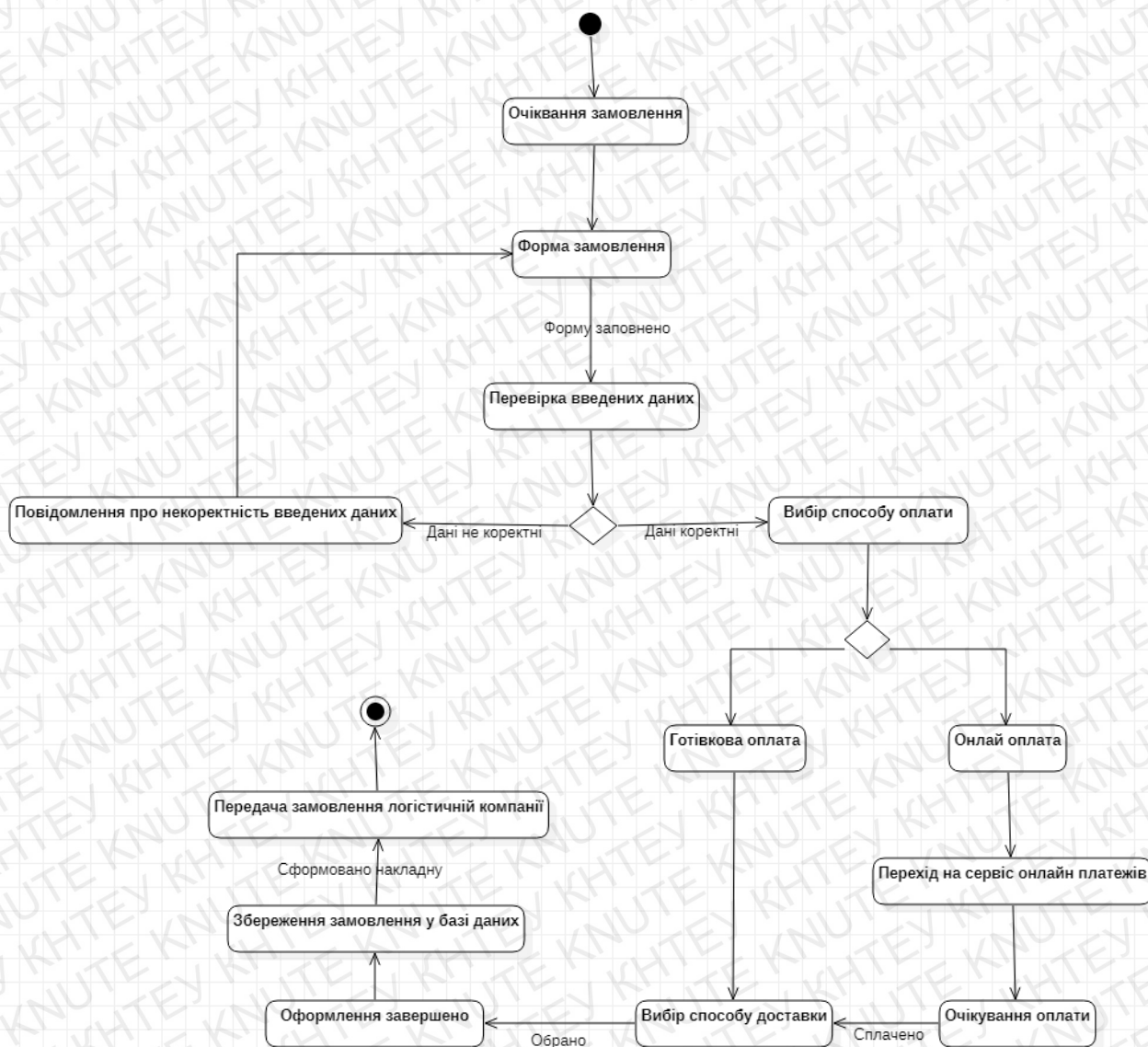


Рис. 3.4. Діаграма станів (state chart diagram) інтернет-магазину з продажу мотоциклів «2WHEELS»

Визначившись із потрібними діаграмами та спроектувавши їх ми, по суті, змодельовали, забезпечили архітектурну цілісність та більш високу ефективність програмування майбутнього проекту.

Тепер можна перейти безпосередньо до розробки веб-сайту, створення проекту та налаштування середовища розробки.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		40

3.3. Розробка серверної частини веб-сайту

Для розробки серверної частини сайту інтернет-магазину з продажу мотоциклів «2WHEELS» було обрано мову Python та написаний на ній фреймворк Django. Django – це найпопулярніший в світі (за даними Git) фреймворк для розробки веб-додатків, який можна використовувати як для роботи над швидкими проектами, так і для вирішення обширних та архітектурно складних задач. На Django працюють такі ресурси як Instagram, Pinterest і Google.

Головною архітектурною особливістю проекту написаного на Django є те, що він складається з деякої кількості модулів (додатків – apps), кожен з яких відповідає за той чи інший функціонал сайту. Архітектура Django проекту подібна на MVC (Модель-вигляд-контролер), але з деякими змінами. Наприклад те, що в архітектурному шаблоні MVC називається «controller», в Django називається «view», а те, що мало б називатись «view» в MVC в Django називається «template». А отже архітектурний шаблон MVC серед Django розробників зветься MTV (Модель-шаблон-вид).

Через архітектурні особливості Django, проекти написані на даному фреймворку мають наступну структуру (рис 3.5).

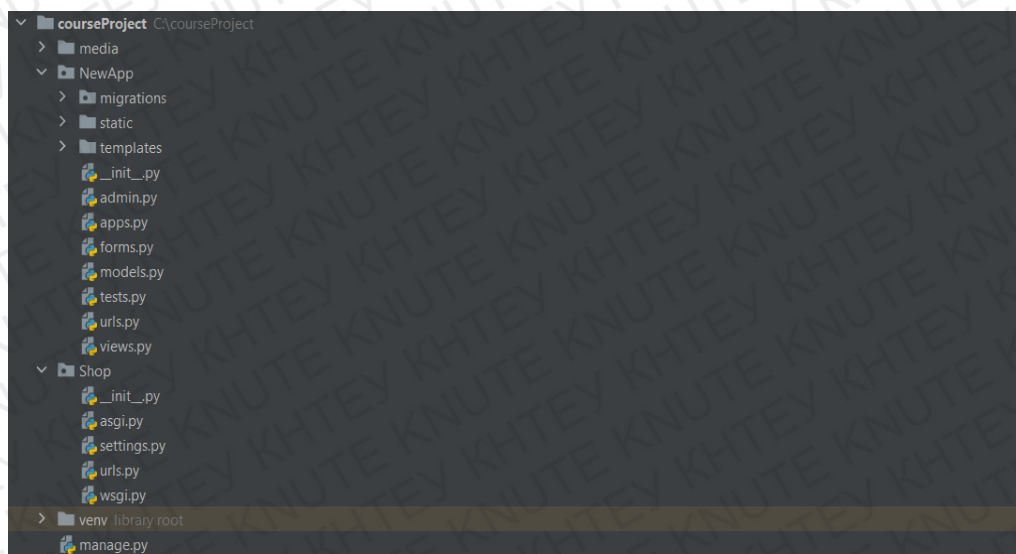


Рис. 3.5 Структура проекту інтернет-магазину з продажу мотоциклів «2WHEELS»

					КНТЕУ 121.07-11.БР	Аркуш
						41
Зм.	Аркуш	№ докум	Підпис	Дата		

Папка Shop це головна директорія проекту, в якій містяться файли:

- settings.py - налаштування проекту;
- urls.py – файл, в якому прописані url-адреси сайту і посилання на модулі (apps);
- wsgi.py – файл для налаштування WSGI-конфігурації;
- asgi.py – файл для налаштування ASGI-конфігурації.

Папка NewApp – це створений модуль, в якому містяться;

- migrations – директорія, в якій зберігаються файли з інформацією про міграції БД;
- static – директорія з статичними файлами (JavaScript та CSS файли);
- templates – директорія з HTML-шаблонами;
- admin.py – файл для налаштування панелі адміністратора;
- apps.py – файл з конфігурацією даного модуля;
- forms.py – файл з формами, які наповнюють ту чи іншу модель бази даних, наприклад при реєстрації користувача чи оформленні замовлення;
- models.py – файл, в якому містяться моделі бази даних, тобто саме тут створюються і редагуються всі таблиці БД;
- tests.py – файл для тестування коду проекту;
- urls.py – аналогічний файлу в директорії Shop, але містить вже url-адреси даного модуля;
- views.py – це файл, для «видів», тобто саме тут прописана логіка проекту, створені запити до БД, тощо;

Директорія media містить медіафайли (зображення товарів), а venv – налаштування віртуального середовища та встановлені бібліотеки, що використовуються в проєкті.

					КНТЕУ 121.07-11.БР	Аркуш
						42
Зм.	Аркуш	№ докум	Підпис	Дата		

```

from django.urls import path
from NewApp import views

urlpatterns = [
    path('', views.home, name='home'),
    path('shop/', views.shop, name='shop'),
    path('search/', views.search, name='search'),
    path('add_to_cart/<int:pk>', views.add_to_cart, name='add_to_cart'),
    path('cart/', views.cart, name='cart'),
    path('order/', views.order, name='order'),
    path('shop/filter/', views.shop_filter, name='filter'),
    path('delete_item/<int:pk>', views.delete_item, name='delete_item'),
    path('change_count/<int:pk>/<str:oper>', views.change_count, name='change_count'),
    path('success/', views.success_order, name='success'),
    path('cabinet/', views.cabinet, name='cabinet'),
    path('contacts/', views.contacts, name='contacts'),
    path('guarantee/', views.guarantee, name='guarantee'),
    path('delivery/', views.delivery, name='delivery'),
]

```

Рис. 3.6 Структура файлу NewApp/urls.py проекту інтернет-магазину з продажу мотоциклів «2WHEELS»

В середині файлу urls.py прописані шляхи (path), що будуть доступні користувачеві при взаємодії з сайтом. Функція path приймає три аргументи, де перший – це url-адреса сторінки, другий – це «вид» з файлу views.py, який оброблює запит і видає відповідь, а третій – це внутрішнє ім'я даного шляху.

Також, слід звернути увагу на файл views.py, де прописана логіка проекту та створені запити до БД. В основі цього файлу лежать види (views), реалізовані у вигляді функцій, написаних на синтаксисі мови Python. Кожен вид приймає запит (request) та довільну кількість аргументів, обробляє запит та повертає HttpResponse, який може містити HTML-документ та текстові дані у вигляді QuerySet чи будь-яку іншу інформацію.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		43

Для більш точного розуміння роботи views в Django проекті, розглянемо приклад виду (view) shop_filter (рис 3.7), що відповідає за фільтрацію товарів в каталозі.

```
def shop_filter(request):
    types = ProductType.objects.all()
    brands = Brand.objects.all()
    print(request.GET.get("price"))
    if len(request.GET) == 3:
        products = Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand')),
                                            Q(product_type__type_name__in=request.GET.getlist('type')) \
                                            .order_by(f'{request.GET.get("price")}')
    elif len(request.GET) == 2:
        if 'brand' and 'type' in request.GET:
            products = Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand')),
                                                Q(product_type__type_name__in=request.GET.getlist('type')) \
                                                .order_by(f'{request.GET.get("price")}')
        else:
            products = Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand')) |
                                                Q(product_type__type_name__in=request.GET.getlist('type')) \
                                                .order_by(f'{request.GET.get("price")}')
    elif len(request.GET) == 1:
        if 'price' in request.GET:
            products = Product.objects.order_by(f'{request.GET.get("price")}')
        else:
            products = Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand')) |
                                                Q(product_type__type_name__in=request.GET.getlist('type'))
    else:
        return redirect('/shop/')
    return render(request, 'shop.html', {'products': products, 'form': LoginForm,
                                          'types': types, 'brands': brands,
                                          'signup': SignupForm})
```

Рис. 3.7 Вид shop_filter файлу NewApp/views.py проекту інтернет-магазину з продажу мотоциклів «2WHEELS»

Логіка даного виду (view) полягає в тому, що він приймає запит від користувача, який надходить у вигляді словаря, визначає його довжину і повертає користувачеві QuerySet з відфільтрованими за допомогою запитів об'єктами бази даних. На сайті наявна можливість фільтру по виробникам (змінна brands) і типам (змінна types), а також сортування по ціні – для цього і потрібно визначити довжину запиту, щоб view зрозумів за якими саме критеріями потрібно фільтрувати товари.

У випадку, якщо користувач натиснув кнопку «фільтрувати», не обравши жодних фільтрів, то відбудеться перенаправлення його на сторінку каталогу – redirect('/shop/').

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		44

Якщо ж помилок з боку серверу чи користувача не відбулось, то користувач отримає відповідь у вигляді команди «render». Дана команда приймає три аргументи: request – запит, назву HTML- шаблону і context у вигляді словаря, в якому містяться запитувані користувачем дані, в нашому випадкові – QuerySet з відфільтрованими мотоциклами.

Файл forms.py містить в собі форми та їх конфігурацію, за допомогою яких користувач може додавати, видаляти чи редагувати дані в БД. Даний файл складається з форм у вигляді класів, що успадковуються від батьківського класу ModelForm з модулю django.forms. Розглянемо приклад форми оформлення замовлення на сайті.

```
from django.forms import ModelForm
from .models import Order

class OrderForm(ModelForm):
    class Meta:
        model = Order
        fields = '__all__'
        exclude = ['user']
```

Рис. 3.8. Структура файлу NewApp/forms.py проекту інтернет-магазину з продажу мотоциклів «2WHEELS»

Форма OrderForm відповідає за наповнення користувачем моделі Order (таблиці Order в базі даних). В налаштуванні даної форми вказана модель, яку вона заповнює, та значення змінної fields встановлено на ‘__all__’, що означає можливість редагування всіх полів таблиці Order. Змінна exclude визначає поля таблиці, що не підлягають редагуванню користувачем, в даному випадку це поле ‘user’ (користувач), адже дане поле заповнюється автоматично згідно з тим, який користувач заповнює форму.

					КНТЕУ 121.07-11.БР	Аркуш
						45
Зм.	Аркуш	№ докум	Підпис	Дата		

Однією з важливих особливостей та переваг фреймворку Django є простота налаштування панелі адміністратора проекту. Всі налаштування адмін-панелі відбуваються у файлі `admin.py`. Розглянемо роботу в даному файлі на прикладі додавання до панелі адміністратора розділу з товарами на сайті.

```
from django.contrib import admin
from .models import Brand, ProductType, Product, Cart, Order, OrderItem

class AdminProduct(admin.ModelAdmin):
    list_display = ['name', 'brand', 'product_type', 'price', 'power', 'engine_volume']

admin.site.register(Brand)
admin.site.register(Product, AdminProduct)
admin.site.register(ProductType)
admin.site.register(Cart)
admin.site.register(Order)
admin.site.register(OrderItem)
```

Рис. 3.9. Структура файлу `NewApp/admin.py` проекту інтернет-магазину з продажу мотоциклів «2WHEELS»

Файл `admin.py` складається з переліку моделей БД, що додаються розробником для того, щоб з'явилась можливість редагувати дані моделі з адмін-панелі. За допомогою команди `admin.site.register()` реєструє моделі в панелі адміністратора, шляхом передавання двох аргументів: перший – це сама модель з файлу `models.py`, другий – це специфічні налаштування (за потреби). У випадку з моделлю `Product` (товар), були внесені певні зміни в налаштування.

Всі конфігурації відбуваються за допомогою класів-надбудов, що успадковуються від батьківського класу `admin.ModelAdmin` з модулю `django.contrib`. Клас `AdminProduct` додає можливість переглядати інформацію про товар, не заходячи при цьому в саму модель. Це досягнуто завдяки зміні значення змінної `list_display` на список (`list`), в якому перераховані всі поля моделі, що мають

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		46

відображатись при перегляді списку товарів з панелі адміністратора. В результаті маємо такий вигляд даного розділу (рис. 3.10).

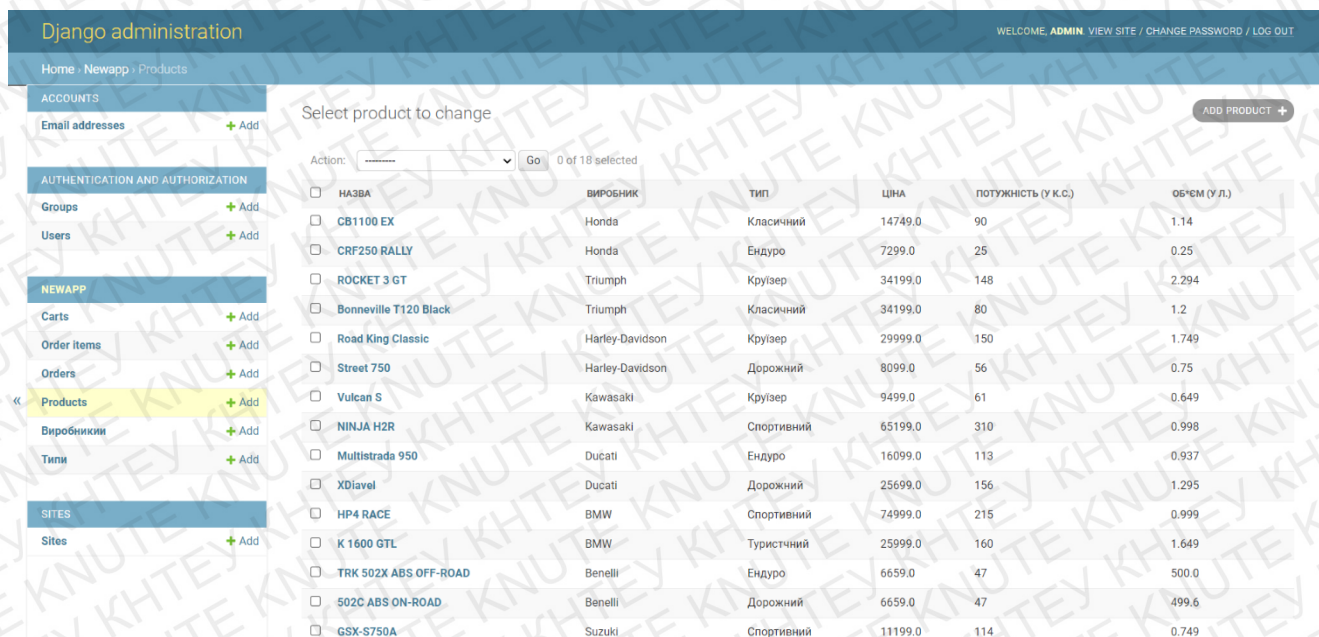


Рис. 3.10. Розділ товарів (Products) в панелі адміністратора сайту інтернет-магазину з продажу мотоциклів «2WHEELS»

Як було зазначено раніше, Django надає можливість «спілкуватись» з базою даних на синтаксисі мови Python. Це досягнуто завдяки Django ORM – системі, що надсилає запити до бази даних, інтерпретуючи мову Python в мову SQL. Це дозволяє значно скоротити час розробки серверної частини проекту, швидко створювати запити до БД і в результаті мати чистий та читабельний код. Запити прописуються безпосередньо в середині видів (views), а результатом виконання запиту є QuerySet – зрозумілий для Python формат даних. Розглянемо деякі використані в даному проекті запити (рис. 3.11 – 3.14).

```
products = Product.objects.filter(
    Q(brand__brand_name__icontains=search_req) | Q(name__icontains=search_req))
```

Рис. 3.11. Запит на пошук товару по назві чи виробникові

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		47


```
cart_items = Cart.objects.filter(user=user).order_by('item')
```

Рис. 3.12. Запит на відтворення товарів у корзині для конкретного користувача

```
item = Cart.objects.filter(pk=pk)
item.delete()
```

Рис. 3.13. Запит на видалення конкретного товару з корзини

```
orders = Order.objects.filter(user=request.user.username).order_by('-order_date')
order_items = OrderItem.objects.filter(Q(order_id__in=orders))
```

Рис. 3.14. Запит на історію покупок користувача в кабінеті користувача

Авторизація користувачів на сайті відбувається за допомогою бібліотеки з безпечною логікою реєстрації та авторизації django-allauth. Дана бібліотека являє собою перелік форм потрібних для роботи з моделлю користувача, наприклад: форма реєстрації, форма входу, форма зміни паролю, форма відновлення паролю тощо.

Налаштувавши проект, дослідивши архітектуру проектів Python Django та розробивши серверну частину сайту інтернет-магазину з продажу мотоциклів «2WHEELS» ми отримали оптимізований та детально пропрацьований продукт, наскільки це можливо на даному етапі розробки.

3.4. Розробка інтерфейсу

Користувацький інтерфейс це перше, що оцінює користувач при вході на сайт. Тому швидкий, зрозумілий та сучасний front-end є дуже важливим етапом розробки будь-якого веб-додатку.

Для розробки інтерфейсу було обрано, звичайно ж, HTML – мова гіпертекстової розмітки для верстки сторінок, для стилізації було обрано фреймворк CSS Bootstrap 5, що включає в себе і JavaScript рішення.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		48

Згідно з визначеним раніше технічним завданням, сайт має бути мінімалістичним, кольори, що використовуються при стилізації сторінок повинні бути не яскравими.

Перша сторінка сайту – «Домашня» або «Головна» має містити в собі інформацію про компанію, переваги, про нові товари чи акції.

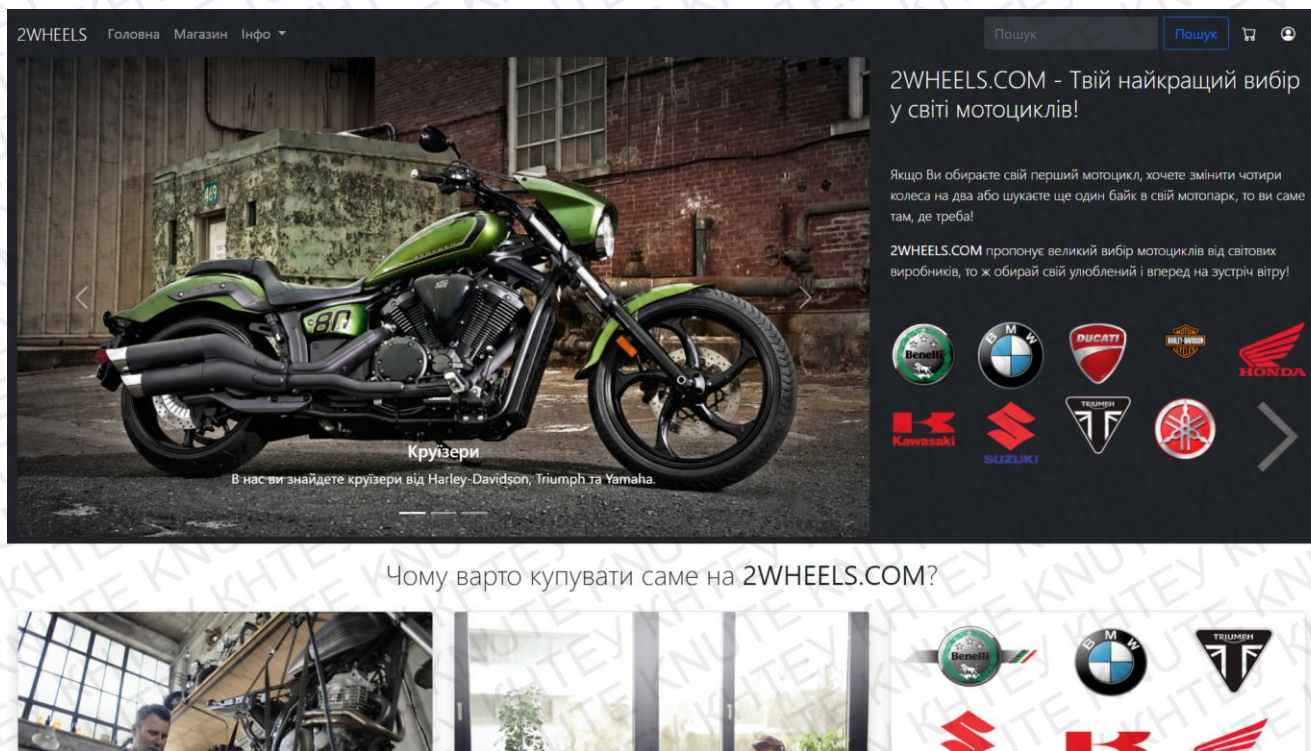


Рис. 3.15. Головна сторінка сайту інтернет-магазину з продажу мотоциклів «2WHEELS»

На домашній сторінці є слайдер з інформацією про нові товари, та колонка з привітанням користувача. Також дана колонка містить перелік брендів, що представлені на сайті у вигляді логотипів, при натисканні на які користувач потрапить в каталог де йому будуть показані товари згідно з тим брендом, на який він натиснув.

Якщо прогорнути головну сторінку вниз (рис. 3.16), то видно загальну інформацію про компанію та її переваги.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		49

Чому варто купувати саме на 2WHEELS.COM?



Сервіс

Завдяки мережі власних авторизованих сервісних центрів, компанія 2WHEELS пропонує якісне обслуговування мотоциклів всіх виробників представлених на сайті



Підтримка 24/7

Якщо у вас з'явилися питання щодо продукції або виникли проблеми в процесі експлуатації і ви хочете викликати евакуатор – консультанти нашого call-центру завжди допоможуть вам вирішити ці проблеми.



Світові бренди

Ми співпрацюємо лише з найкращими світовими виробниками мотоциклів та є їх офіційними представниками в Україні, що забезпечує Вам високий рівень сервісу та офіційну гарантію.

Про компанію

Зареєстрована торгова марка «2WHEELS» - одна з перших компаній, професійно зайнялася продажем мототехніки і сервісним обслуговуванням скутерів, мотоциклів, квадроциклів і мопедів в Україні.

Ми на ринку з 2005 року і маємо величезний досвід у своїй сфері. Працюємо тільки з дійсно якісними виробниками мототехніки, які за багато років добре зарекомендували себе, і на модельний ряд яких завжди є в наявності будь-які запчастини. Вся нова техніка ретельно збирається і тестується.

Рис. 3.16. Інформація про компанію та переваги на сайті інтернет-магазину «2WHEELS»

Також слід розглянути шапку і підвал сайту (рис. 3.17).

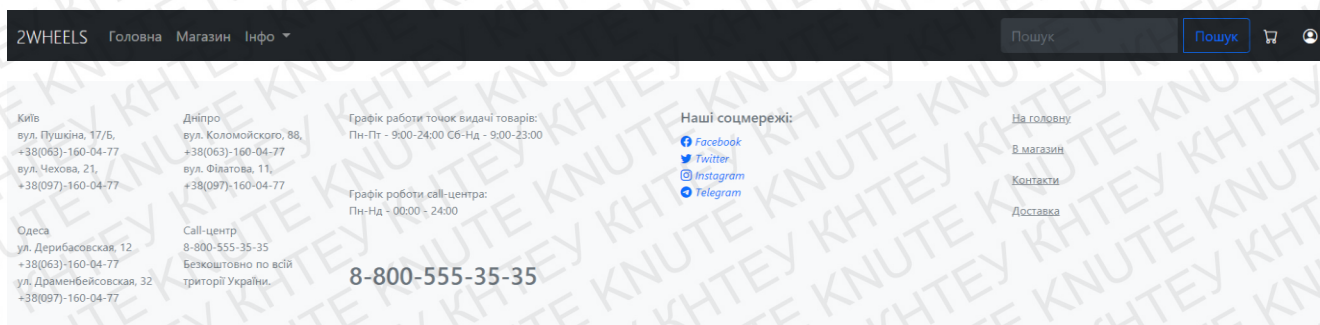


Рис. 3.17 Шапка та підвал сайту інтернет-магазину з продажу мотоциклів «2WHEELS»

Шапка містить: графу для пошуку товарів, навігацію по сторінкам, посилання на корзину і особистий кабінет користувача – якщо користувач авторизований, при натисканні, він потрапить в свій кабінет, якщо ні, то йому буде запропоновано

					КНТЕУ 121.07-11.БР	Аркуш 50
Зм.	Аркуш	№ докум	Підпис	Дата		

авторизуватись. Підвал містить контактну інформацію, посилання на соцмережі і мініатюрну навігацію по сайту.

Головна частина будь-якого сайту з продажу товарів в інтернет – це каталог товарів. Для сайту інтернет-магазину з продажу мотоциклів «2WHEELS» було обрано мінімалістичну концепцію, щоб вся увага користувача була спрямована саме на товари.

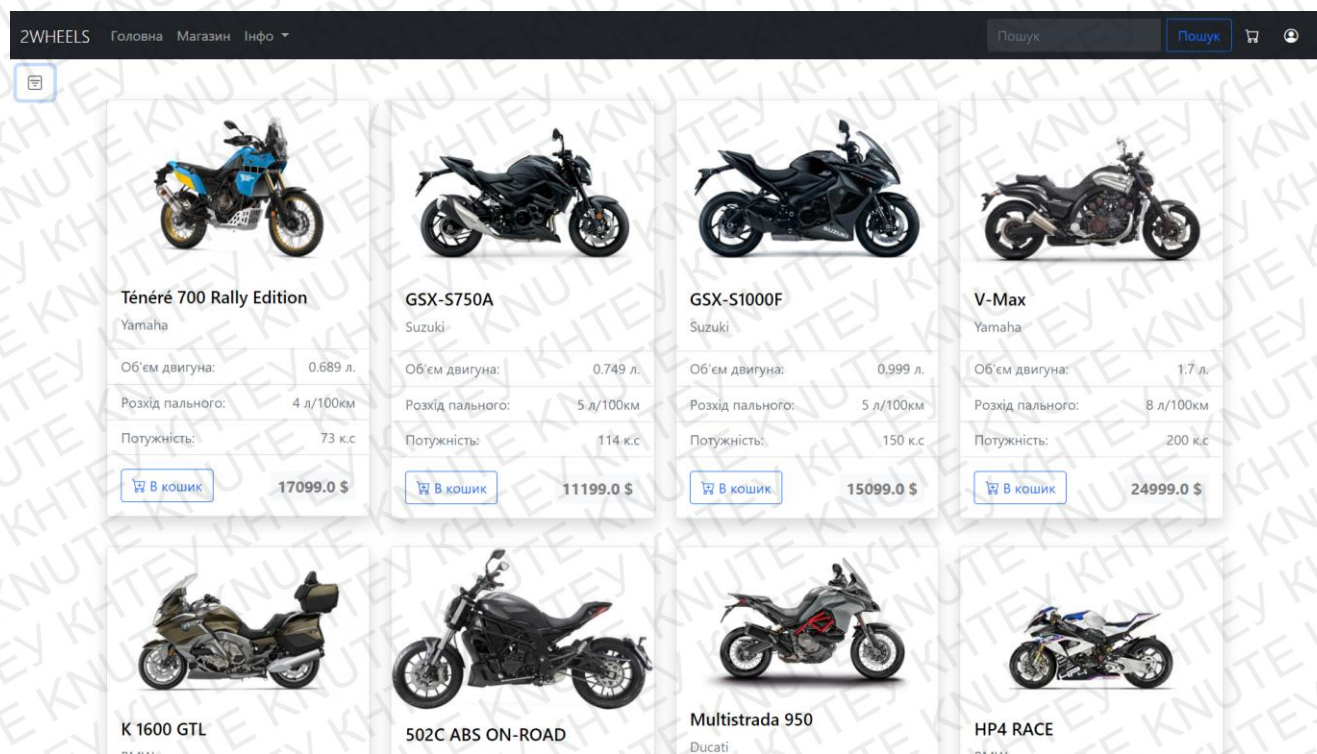


Рис. 3.18 Каталог товарів сайту інтернет-магазину з продажу мотоциклів «2WHEELS»

Каталог складається з карток товарів, де міститься зображення товару, назва, виробник, ціна та деякі основні технічні характеристики. На кожній картці наявна кнопка «В кошик», при натисканні на яку обраний товар автоматично потрапляє в кошик користувача.

В лівому верхньому вуглі знаходиться кнопка «Фільтри», при натисканні на яку виїжджає шторка (рис. 3.19) з переліком фільтрів та можливістю сортування товарів.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		51

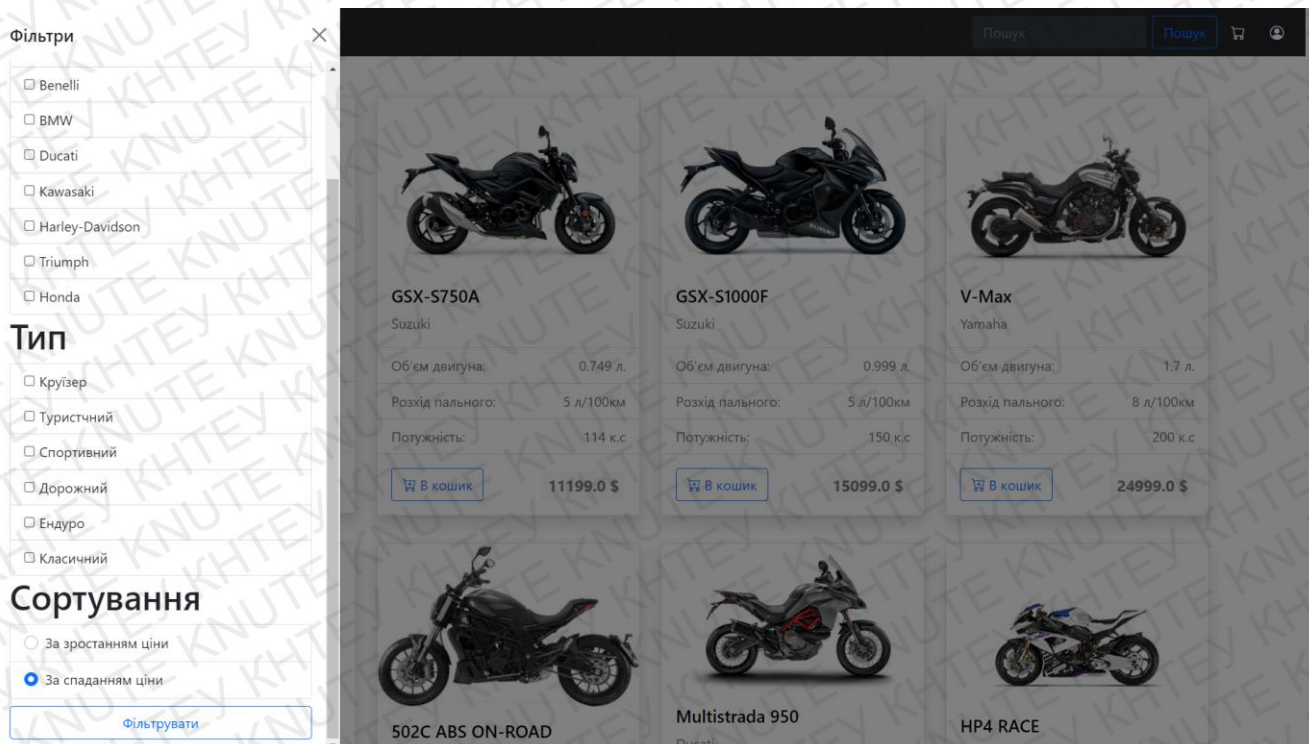


Рис. 3.19 Шторка з фільтрами в каталозі товарів інтернет-магазину з продажу мотоциклів «2WHEELS»

На той випадок, якщо користувач хоче переглянути більш детальну інформацію про товар, на сайті з продажу товарів має бути окрема для кожної позиції сторінка, що надає таку можливість. На сайті «2WHEELS» в неї можна потрапити, натиснувши на картку бажаного товару в каталозі. Реалізована дана сторінка наступним чином (рис. 3.20).

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		52

2WHEELS
Головна
Магазин
Інфо
Пошук
Пошук



GSX-S750A
Suzuki
11199.0\$
В кошик

Доставка	Безкоштовно
Термін доставки	1-2 дні
Можна перевірити	В Одесі, Дніпрі, Києві
Щоб отримати	Водійські права категорії А, А1
Оплата	Готівкою в магазині, онлайн

На холостых Suzuki GSX-S750 звучит просто отлично, подзадоривая и подзуживая. Он шикарно выстреливает из поворотов, удобен в городе, прекрасно справляется в любом потоке и легко уезжает от него. В реальных условиях больше просто не требуется, то есть для повседневного дорожного нейкеда 750-ки более чем достаточно. У него весьма неплохая тяга на средних оборотах, приятный плавный газ, отличная приёмистость, в городе он весёлый и послушный одновременно. Но когда отпускаешь вожжи и выжимаешь все его 115 лошадок — вот уж тогда он действительно увлекает! Рядная четвёрка раскручивается очень высоко, выдаёт пиковую мощность на 10500об/мин и крутится ещё дальше, позволяя меньше переключаться, если нет нужды выжать из него всё. Коробка Suzuki GSX-S750 чёткая и внятная, передачи у неё сближенные, и единственное, чего не хватает, так это квикшифтера. Его, и, пожалуй, тонинговый выхлоп — вот и всё, что я бы докупил себе, доведись мне обладать этим мотоциклом.

Назва моделі	GSX-S750A
Виробник	Suzuki
Країна походження	Японія
Тип мотоциклу	Спортивний
Потужність двигуна	114 к.с.
Об'єм двигуна	0.749 л.
Споживання пального	5 л/100км
Ціна	11199.0 \$

Рис. 3.20. Сторінка товару на сайті інтернет-магазину з продажу мотоциклів «2WHEELS»

На даній сторінці міститься зображення товару, назва, назва виробника, ціна, кнопка «В кошик» та скорочена інформація про умови доставки та оплати товарів. Крім цього ця сторінка вже містить повний перелік технічних характеристик товару і опис конкретної моделі.

Також дуже важливою для репутації сайту є сторінка з інформацією про доставку, гарантію, оплату та контакти сайту. На сайті інтернет-магазину з продажу мотоциклів «2WHEELS» посилання на ці сторінки знаходиться в випадальному меню «Інфо» в шапці сайту. Розглянемо приклад сторінки з контактною інформацією (рис. 3.21).

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		53

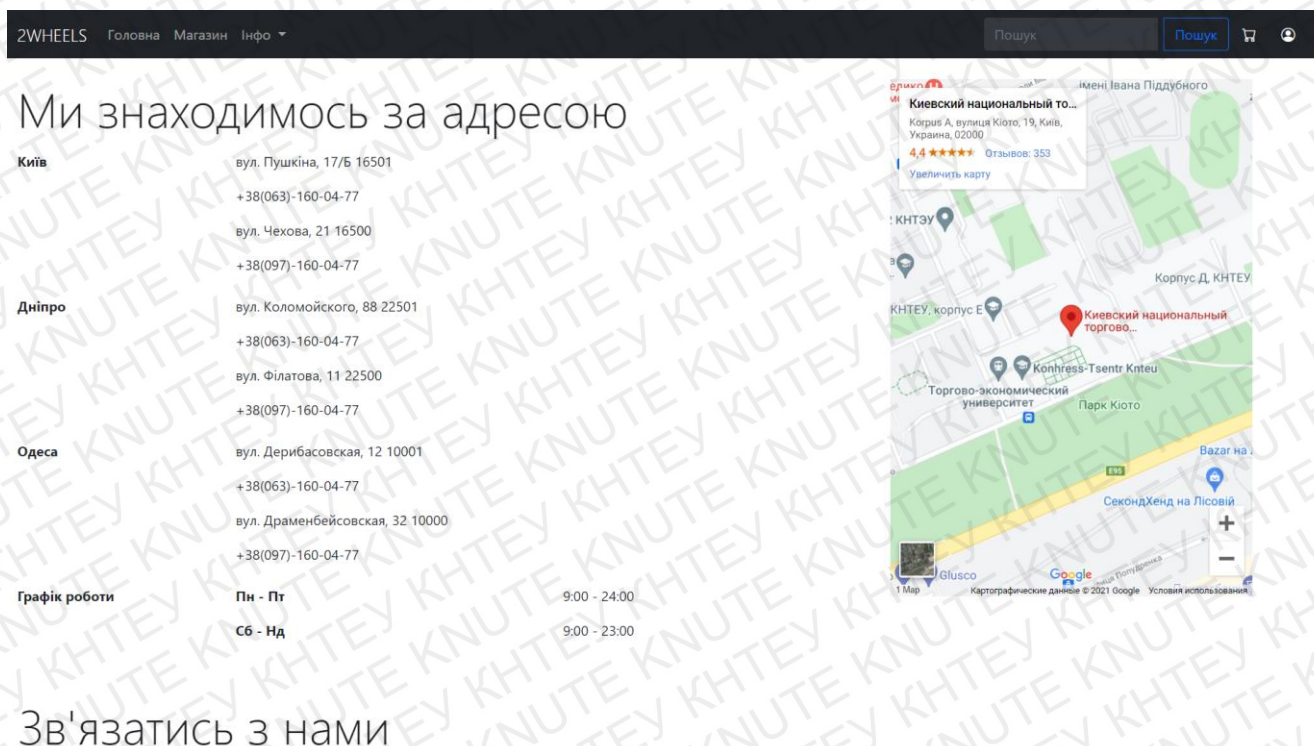


Рис. 3.21. Сторінка з контактною інформацією на сайті інтернет-магазину з продажу мотоциклів «2WHEELS»

На даній сторінці містяться адреси, графік роботи, номери телефонів точок видачі товарів (центрів дистрибуції) та контактна інформація call-центру (служби підтримки). Також в правій частині знаходиться інтерактивна карта з API Google Maps, що мала б вказувати на місцезнаходження найближчої точки дистрибуції до користувача.

Далі слід розглянути сторінку кошику товарів (рис. 3.22).

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		54

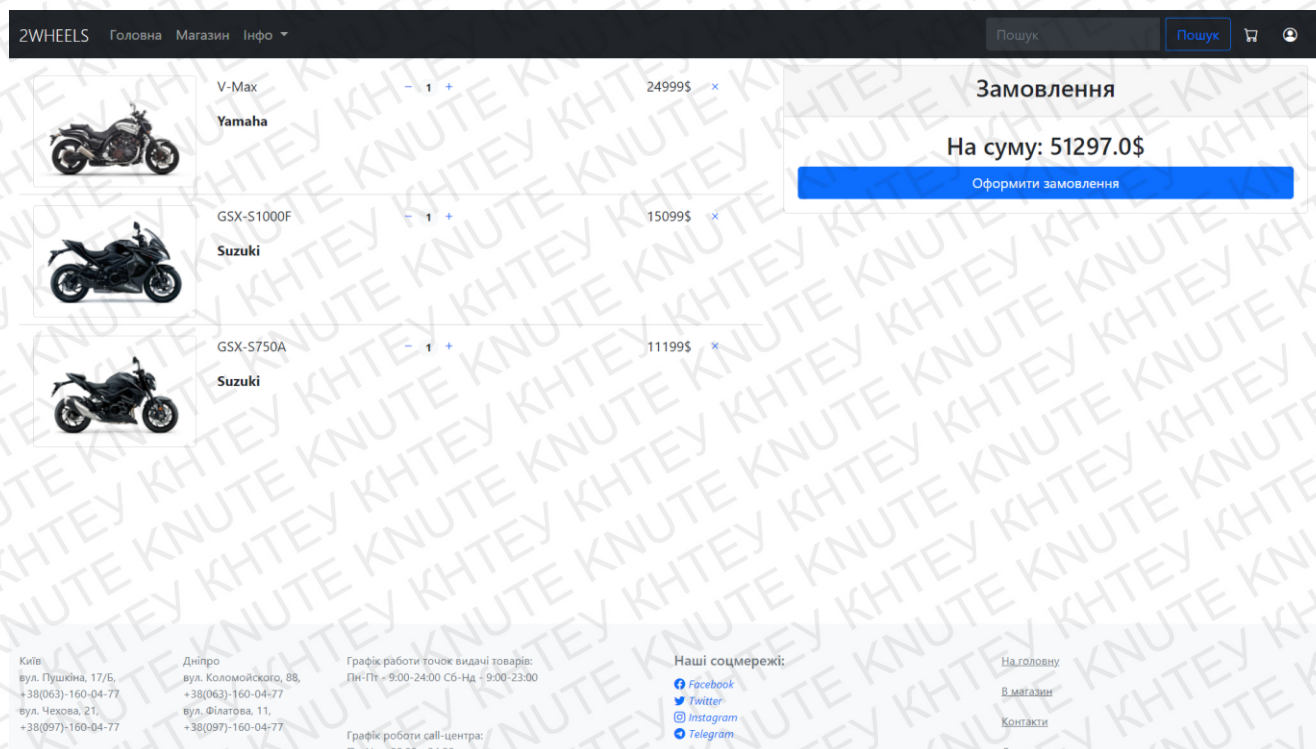


Рис. 3.22. Сторінка кошику покупця на сайті інтернет-магазину з продажу мотоциклів «2WHEELS»

Дана сторінка має перелік товарів доданих користувачем до кошика, кількість яких можна змінювати, або ж повністю видаляти. В правій частині міститься таблиця з сумою замовлення та кнопкою «Оформити замовлення», що при натисканні викликає модальне вікно з формою оформлення замовлення (рис. 3.23).

The screenshot shows the 'Оформлення замовлення' (Order form) modal window. It contains the following fields:

- Ім'я* (Name*)
- Фамілія* (Surname*)
- Пошта (Email)
- Номер телефону* (Phone number*)
- Доставка* (Delivery*)

A blue button 'Оформити' (Place order) is located at the bottom right of the form.

Рис. 3.23. Форма оформлення замовлення на сайті інтернет-магазину з продажу мотоциклів «2WHEELS»

					КНТЕУ 121.07-11.БР	Аркуш
						55
Зм.	Аркуш	№ докум	Підпис	Дата		

Після заповнення форми та натискання кнопки «Оформити» користувач отримує повідомлення про успішне замовлення (рис. 3.24) та ще раз бачить список замовлених товарів.

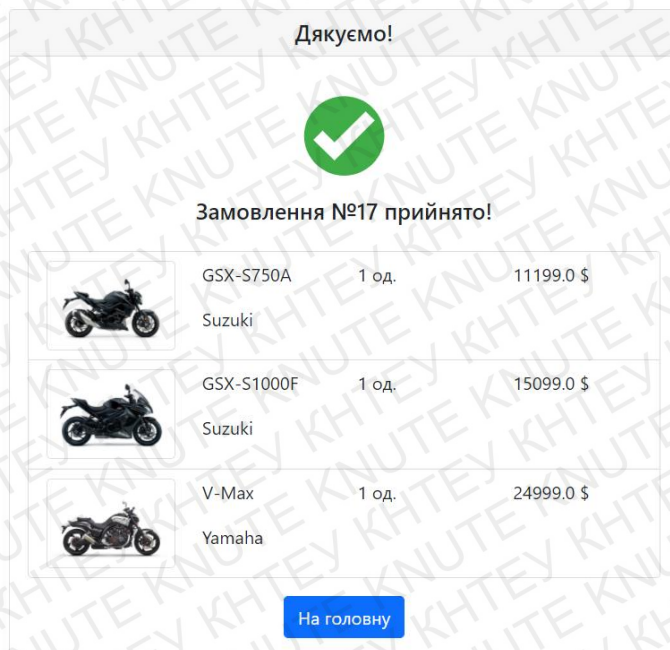


Рис. 3.24. Повідомлення про успішне замовлення на сайті інтернет-магазину з продажу мотоциклів «2WHEELS»

Далі, якщо користувач зареєстрований, він може перейти в особистий кабінет, де щойно створене замовлення з'явиться в історії його покупок (рис. 3.25).

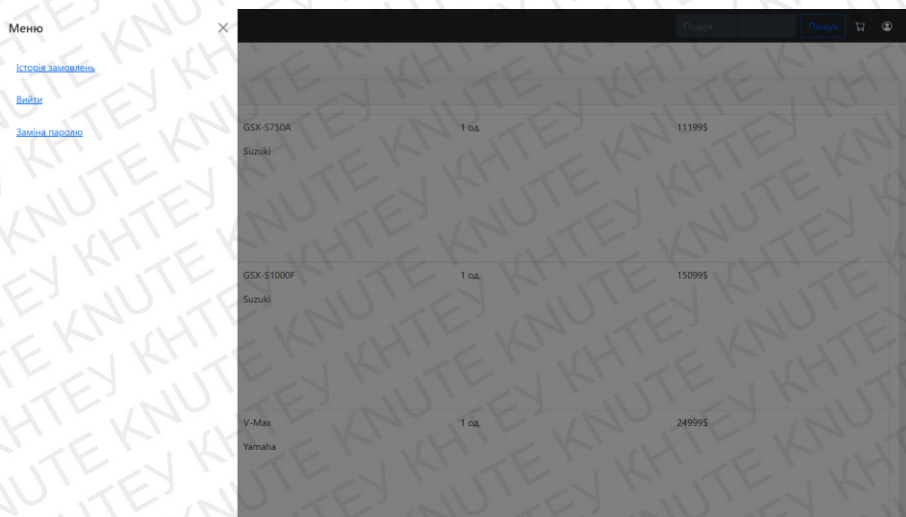


Рис 3.25. Кабінет користувача на сайті інтернет-магазину з продажу мотоциклів «2WHEELS»

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		56

Кабінет користувача являє собою сторінку, з впливаючим меню (аналогічно шторці фільтрів на сторінці «Магазин»), де користувач може переглянути історію власних покупок, змінити пароль та вийти з акаунту.

Отже, розробивши інтерфейс користувача, ми можемо перейти до розміщення сайту у вільному доступі на будь-якому з онлайн хостингів.

Для розміщення даного проекту було обрано хмарну платформу Heroku, адже вона має потрібні для роботи з PostgreSQL та Python інструменти. Готовий сайт розміщено за адресою: <https://moto-project-knute.herokuapp.com/>.

3.5. Висновки до розділу 3

Таким чином, було проаналізовано і обґрунтовано вибір програмних засобів та технологій обраних для розробки проекту сайту інтернет-магазину з продажу мотоциклів «2WHEELS».

Під час моделювання проекту було створено низку UML-діаграм, а саме: діаграма діяльності, діаграма варіантів використання, діаграма станів і діаграма класів, що гарантувало цілісність проекту та простоту його подальшої підтримки.

Під час першого етапу розробки було створено серверну частину сайту, додано представлення БД та розроблено «скелет» майбутнього проекту.

В заключному етапі, розробивши інтерфейс користувача, стилі та розмітку сторінок, було завершено основну частину розробки, а сайт розміщено на хмарному хостинзі.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		57

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В результаті виконання даної випускної кваліфікаційної роботи було розроблено веб-сайт інтернет-магазину з продажу мотоциклів «2WHEELS».

Розробка даного веб-додатку відбувалася в кілька етапів.

На першому етапі було здійснено опис загальної концепції використання веб-сайтів в галузі торгівлі, особливостей даної предметної області та вже існуючих сайтів у сфері роздрібної торгівлі. В результаті можна зробити висновок, що створення сайту інтернет-магазину з продажу мотоциклів «2WHEELS» є досить трудомістким та відповідальним процесом.

На другому етапі було здійснено постановку завдання, обрано програмні засоби для управління базою даних, а також розроблено інфологічну та даталогічну моделі, які стали основою під час її реалізації. Створення бази даних для інтернет-магазину з продажу мотоциклів «2WHEELS» відбувалося на платформі середовища PostgreSQL, яка являє собою інтегроване середовище для доступу, налаштування, адміністрування, розробки всіх компонентів SQL Server та управління ними. В результаті, розроблена база даних складається із семи реляційних таблиць.

Для реалізації основної частини сайту було обрано мову програмування Python, а зокрема фреймворк Django, який є сучасним інструментом для розробки веб-додатків за допомогою синтаксису мови Python. Вибір середовища розробки зупинився на PyCharm Professional.

					КНТЕУ 121 07-11.БР			
					Розробка web-сайту інтернет-магазину продажу мотоциклів «2WHEELS»	Стадія	Аркуш	Аркушів
						ВП	58	61
Зм.	Аркуш	№ докум	Підпис	Дата		Факультет інформаційних технологій, 4 курс, 7 група		
Зав. кафедри		Криворучко О.В.						
Керівник		Пашорін В.І.						
Гарант		Цензура М.О.						
Розроб.		Кравченко А.Ю.			ВИСНОВКИ ТА ПРОПОЗИЦІЇ			

Для розробки інтерфейсу користувача були застосовані такі сучасні технології як CSS Bootstrap 5, HTML-5 та JavaScript.

В результаті тестування даного застосунку було виявлено, що результати його роботи є коректними та задовільними.

Таким чином, розроблений сайт інтернет-магазину з продажу мотоциклів «2WHEELS» реалізує вимоги, заявлені в технічному завданні та є сучасним і вдалим рішенням у сфері роздрібної торгівлі.

					КНТЕУ 121.07-11.БР	Аркуш
						59
Зм.	Аркуш	№ докум	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Swaroop С.Н. A Byte of Python / С.Н. Swaroop, 2008. – 159 р.
2. Lutz M. Learning Python / М. Lutz, D. Ascher, 1999. – 1648 р.
3. Роббінс Д.Н. HTML5 Кишеньковий довідник: швидкий, всебічний, незамінний / Д.Н. Роббінс, 2013. – 192 с.
4. 2020 Digital Trends Report [Електронний ресурс]. – URL: <https://www.similarweb.com/corp/reports/2020-digital-trends-report/>
5. Obe R. PostgreSQL: Up and Running / R. Obe, L. Hsu, 2012. – 168 р.
6. Django: The web framework for perfectionists with deadlines [Електронний ресурс]. – URL: <https://docs.djangoproject.com/>
7. Форсьє Д. Django: Розробка веб-додатків на Python / Д. Форсьє, П. Біссекс, У. Чан, 2018. – 416 с.
8. Мейер Е.А. CSS: Повний довідник / Е.А. Мейер, Е. Уейл, 2019. – 1019 с.
9. Bootstrap 5 Framework: Documentation [Електронний ресурс]. – URL: <https://getbootstrap.com/docs/5.0>
10. Heroku: Heroku Python Documentation [Електронний ресурс]. – URL: <https://devcenter.heroku.com/categories/python-support>
11. PyCharm JetBrains: Documentation [Електронний ресурс]. – URL: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>
12. PEP 8: Style Guide for Python Code [Електронний ресурс]. – URL: <https://www.python.org/dev/peps/pep-0008/>
13. Python 3.9.5: Documentation [Електронний ресурс]. – URL: <https://docs.python.org/3/>

					КНТЕУ 121 07-11.БР			
Зм.	Аркуш	№ докум	Підпис	Дата	Розробка web-сайту інтернет-магазину продажу мотоциклів «2WHEELS»	Стадія	Аркуш	Аркушів
Зав. кафедри	Криворучко О.В.			СД		60	61	
Керівник	Пашорін В.І.			Факультет інформаційних технологій, 4 курс, 7 група				
Гарант	Цензура М.О.							
Розроб.	Кравченко А.Ю.							
					СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ			

14. Spurlock J. Bootstrap. Responsive Web Development / J. Spurlock, D. Winer, 2013. – 128р.
15. Холден С. Python. Довідник: Повний опис мови / С. Холден, А. Мартелі, А. Рейвенскрофт, 2018. – 896 с.
16. S. Franklin Planning Your Web-Site with UML [Електронний ресурс]. – URL: https://people.apache.org/~jim/NewArchitect/webrevu/2001/05_18/developers/index01.html
17. Django ORM для начинающих: оптимизируем запросы [Електронний ресурс]. – URL: <https://habr.com/ru/post/503526/>
18. Django-allauth: Documentation [Електронний ресурс]. – URL: <https://django-allauth.readthedocs.io/en/latest>
19. Git: Documentation [Електронний ресурс]. – URL: <https://git-scm.com/doc>
20. Chacon S. Pro Git / S. Chacon, B. Straub, 2012. – 554 p.

					КНТЕУ 121.07-11.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		61

ДОДАТКИ

Додаток А

1. Файл BASEDIR/urls.py

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('accounts/', include('allauth.urls')),
    path('admin/', admin.site.urls),
    path('', include('NewApp.urls'))
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

2. Файл NewApp/urls.py

```
from NewApp import views

urlpatterns = [
    path('', views.home, name='home'),
    path('shop/', views.shop, name='shop'),
    path('search/', views.search, name='search'),
    path('add_to_cart/<int:pk>', views.add_to_cart, name='add_to_cart'),
    path('cart/', views.cart, name='cart'),
    path('order/', views.order, name='order'),
    path('shop/filter/', views.shop_filter, name='filter'),
    path('delete_item/<int:pk>', views.delete_item, name='delete_item'),
    path('change_count/<int:pk>/<str:oper>', views.change_count,
name='change_count'),
    path('success/', views.success_order, name='success'),
    path('cabinet/', views.cabinet, name='cabinet'),
    path('contacts/', views.contacts, name='contacts'),
    path('guarantee/', views.guarantee, name='guarantee'),
    path('delivery/', views.delivery, name='delivery'),
    path('product/<int:pk>', views.product_detail, name='detail'),
]
```

3. Файл NewApp/models.py

```
from django.db import models

class Brand(models.Model):
    brand_logo = models.ImageField(null=True, blank=True, upload_to='images/')
    brand_name = models.CharField(max_length=40)
    country = models.CharField(max_length=40)

    class Meta:
        verbose_name = 'Виробник'
        verbose_name_plural = 'Виробникии'

    def __str__(self):
        return self.brand_name

class ProductType(models.Model):
    type_name = models.CharField(max_length=40)

    class Meta:
        verbose_name = 'Тип'
        verbose_name_plural = 'Типи'
```



```

def __str__(self):
    return self.type_name

class Product(models.Model):
    image = models.ImageField(null=True, blank=True, upload_to='images/',
verbose_name='Зображення')
    name = models.CharField(max_length=50, verbose_name='Назва')
    brand = models.ForeignKey(Brand, on_delete=models.CASCADE,
verbose_name='Виробник')
    product_type = models.ForeignKey(ProductType, on_delete=models.CASCADE,
verbose_name='Тип')
    price = models.FloatField(default=0, verbose_name='Ціна')
    power = models.IntegerField(default=0, verbose_name='Потужність (у к.с.)')
    engine_volume = models.FloatField(default=0, verbose_name='Об'єм (у л.)')
    fuel_consumption = models.IntegerField(default=0, verbose_name='Витрата
пального (л. на 100км)')
    description = models.TextField(max_length=1000, verbose_name='Опис')

class Order(models.Model):
    ODESSA = 'Odessa'
    KIEV = 'Kiev'
    DNEPR = 'Dnepr'
    DELIVERY_TYPES = (
        (ODESSA, 'Одеса'),
        (KIEV, 'Київ'),
        (DNEPR, 'Дніпро'),
    )
    user = models.CharField(null=True, blank=True, max_length=30,
verbose_name='Користувач')
    order_date = models.DateTimeField(auto_now_add=True, verbose_name='Дата
замовлення')
    name = models.CharField(max_length=30, verbose_name="Ім'я", null=True)
    surname = models.CharField(max_length=30, verbose_name='Фамілія', null=True)
    email = models.EmailField(blank=True, null=True, verbose_name='Пошта')
    tele_num = models.BigIntegerField(default=380, verbose_name='Номер телефону')
    delivery = models.CharField(max_length=100,
choices=DELIVERY_TYPES,
default=KIEV, verbose_name='Доставка')

    def __str__(self):
        return f"Замовлення №{str(self.id)}"

class OrderItem(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE)
    item = models.ForeignKey(Product, on_delete=models.CASCADE)
    count = models.IntegerField(default=0)

class Cart(models.Model):
    user = models.CharField(null=True, blank=True, max_length=30)
    item = models.ForeignKey(Product, on_delete=models.CASCADE)
    count = models.IntegerField(default=0)

```

4. Файл NewApp/views.py

```

5. from django.shortcuts import render, redirect
   from .models import Product, Cart, OrderItem, Order, ProductType, Brand
   from django.db.models import Q
   from allauth.account.forms import LoginForm, ChangePasswordForm, SignupForm
   from .forms import OrderForm

def home(request):
    return render(request, 'home.html', {'form': LoginForm, 'signup':
    SignupForm})

def shop(request):
    products = Product.objects.all()
    types = ProductType.objects.all()
    brands = Brand.objects.all()
    return render(request, 'shop.html', {'products': products, 'form':
    LoginForm,
                                         'types': types, 'brands': brands,
                                         'signup': SignupForm})

def product_detail(request, pk):
    product = Product.objects.filter(pk=pk)
    return render(request, 'product.html', {'product': product, 'form':
    LoginForm,
                                         'signup': SignupForm})

def shop_filter(request):
    types = ProductType.objects.all()
    brands = Brand.objects.all()
    print(request.GET.get("price"))
    if len(request.GET) == 3:
        products =
        Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand'))
        ,
        Q(product_type__type_name__in=request.GET.getlist('type')))\
        .order_by(f'{request.GET.get("price")}')
        elif len(request.GET) == 2:
            if 'brand' and 'type' in request.GET:
                products =
                Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand'))
                ,
                Q(product_type__type_name__in=request.GET.getlist('type')))\
                .order_by(f'{request.GET.get("price")}')
            else:
                products =
                Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand'))
                |
                Q(product_type__type_name__in=request.GET.getlist('type')))\
                .order_by(f'{request.GET.get("price")}')

```



```

elif len(request.GET) == 1:
    if 'price' in request.GET:
        products = Product.objects.order_by(f'{request.GET.get("price")}')
    else:
        products =
Product.objects.filter(Q(brand__brand_name__in=request.GET.getlist('brand')) |
Q(product_type__type_name__in=request.GET.getlist('type'))))
    else:
        return redirect('/shop/')
    return render(request, 'shop.html', {'products': products, 'form': LoginForm,
                                         'types': types, 'brands': brands,
                                         'signup': SignupForm})

def search(request):
    search_req = request.GET.get('search')
    print(search_req)
    products = Product.objects.filter(
        Q(brand__brand_name__icontains=search_req) |
Q(name__icontains=search_req))
    return render(request, 'shop.html', {'products': products, 'form': LoginForm,
                                         'signup': SignupForm})

def add_to_cart(request, pk):
    if request.user.is_authenticated:
        user = request.user.username
    else:
        x_forwarded_for = request.META.get('HTTP_X_FORWARDED_FOR')
        if x_forwarded_for:
            user = x_forwarded_for.split(',')[0]
        else:
            user = request.META.get('REMOTE_ADDR')
    cart_item, create = Cart.objects.get_or_create(item_id=pk, user=user)
    cart_item.count += 1
    cart_item.save()
    return redirect('/shop/')

def cart(request):
    if request.user.is_authenticated:
        user = request.user.username
    else:
        x_forwarded_for = request.META.get('HTTP_X_FORWARDED_FOR')
        if x_forwarded_for:
            user = x_forwarded_for.split(',')[0]
        else:
            user = request.META.get('REMOTE_ADDR')
    cart_items = Cart.objects.filter(user=user).order_by('item')
    total = 0
    for item in cart_items:
        count = item.count
        product = Product.objects.filter(pk=item.item_id)
        total += product.values('price')[0].get('price') * count
    return render(request, 'cart.html', {'items': cart_items, 'form': LoginForm,
                                         'order': OrderForm, 'total': total,
                                         'signup': SignupForm})

```

```

def delete_item(request, pk):
    item = Cart.objects.filter(pk=pk)
    item.delete()
    return redirect('/cart/')

def change_count(request, pk, oper):
    if request.user.is_authenticated:
        user = request.user.username
    else:
        x_forwarded_for = request.META.get('HTTP_X_FORWARDED_FOR')
        if x_forwarded_for:
            user = x_forwarded_for.split(',')[0]
        else:
            user = request.META.get('REMOTE_ADDR')
    item, create = Cart.objects.get_or_create(pk=pk, user=user)
    if oper == 'minus':
        item.count -= 1
    else:
        item.count += 1
    if item.count == 0:
        item.delete()
    else:
        item.save()
    return redirect('/cart/')

def order(request):
    if request.user.is_authenticated:
        user = request.user.username
    else:
        x_forwarded_for = request.META.get('HTTP_X_FORWARDED_FOR')
        if x_forwarded_for:
            user = x_forwarded_for.split(',')[0]
        else:
            user = request.META.get('REMOTE_ADDR')
    cart_items = Cart.objects.filter(user=user)
    form = OrderForm(request.POST)
    if form.is_valid():
        order = form.save(commit=False)
        order.user = user
        order.name = form.cleaned_data['name']
        order.surname = form.cleaned_data['surname']
        order.email = form.cleaned_data['email']
        order.tele_num = form.cleaned_data['tele_num']
        order.delivery = form.cleaned_data['delivery']
        order.save()
        for item in cart_items:
            order_item = OrderItem.objects.create(item=item.item, order=order,
count=item.count)
            order_item.save()
        cart_items.delete()
        return redirect('/success/')

def success_order(request):
    if request.user.is_authenticated:
        user = request.user.username
    else:
        x_forwarded_for = request.META.get('HTTP_X_FORWARDED_FOR')

```



```

if x_forwarded_for:
    user = x_forwarded_for.split(',')[0]
else:
    user = request.META.get('REMOTE_ADDR')
    order = Order.objects.filter(user=user).order_by('-order_date')[0]
    order_items = OrderItem.objects.filter(order=order.pk)
    return render(request, 'success.html', {'order': order, 'order_items':
order_items,
                                           'form': LoginForm, 'signup':
SignupForm})

def cabinet(request):
    orders = Order.objects.filter(user=request.user.username).order_by('-
order_date')
    order_items = OrderItem.objects.filter(Q(order_id__in=orders))
    return render(request, 'cabinet.html', {'orders': orders,
                                           'order_items': order_items,
                                           'form': LoginForm,
                                           'change': ChangePasswordForm,
                                           'signup': SignupForm})

def contacts(request):
    return render(request, 'contacts.html', {'form': LoginForm,
                                           'signup': SignupForm})

def guarantee(request):
    return render(request, 'guarantee.html', {'form': LoginForm,
                                           'signup': SignupForm})

def delivery(request):
    return render(request, 'delivery.html', {'form': LoginForm,
                                           'signup': SignupForm})

```