

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПОШУКУ

АКСЕСУАРІВ ДЛЯ АВТОМОБІЛІВ

SOFTWARE FOR FINDING CAR ACCESSORIES

Виконав(ла): студент(ка) 4 курсу, групи КНТ-151

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ГУНЬКО І.А.

(ПРИЗВИЩЕ та ініціали)

Керівник ЛЬОВКІН В.М.

(ПРИЗВИЩЕ та ініціали)

Рецензент ЗЕЛІК О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет КНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
 (КОД І НАЙМЕНУВАННЯ)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2025 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ГУНЬКА Ігоря Андрійовича
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення пошуку аксесуарів для автомобілів. Software for Finding Car Accessories

керівник проєкту (роботи) к.т.н., доцент ЛЬОВКІН Валерій Миколайович,
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 року № 209

2. Строк подання студентом проєкту (роботи) 02 червня 2025 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програми. 3. Розробка програми. 4. Експлуатація та тестування програми.

5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЛЬОВКІН В.М., доцент		
Нормоконтроль	БСЛОВА А.В., асистент		

7. Дата видачі завдання « 14 » квітня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Ігор ГУНЬКО
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Валерій ЛЬОВКІН
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 93 с., 3 табл., 27 рис., 3 дод., 9 джерел.

ВЕБЗАСТОСУНОК, АВТОМОБІЛЬ, АКСЕСУАР, ПОШУК, БАЗА ДАНИХ.

Об'єкт роботи – процес пошуку аксесуарів для автомобілів. Предмет роботи – програмне забезпечення пошуку аксесуарів для автомобілів. Мета роботи – розробити програму пошуку аксесуарів для автомобілів для підтримки продажу автомобільних аксесуарів з наданням широкої інформації власниками.

Матеріали, методи та технічні засоби: мова Python для програмування, система MySQL для роботи з базою даних, фреймворк Django для веброзробки.

Результати. У результаті сукупності результатів, отриманих на виконання етапів роботи, розроблено програмне забезпечення пошуку аксесуарів для автомобілів, яке може використовуватися особами, які мають необхідність придбати аксесуари для автомобілів, особами, які мають необхідність продати аксесуари для автомобілів.

Висновки. Програмне забезпечення пошуку аксесуарів для автомобілів може використовуватися як для пошуку інформації про існуючі на ринку аксесуари для автомобілів, практику використання таких аксесуарів у своїх автомобілях власниками, так і для підтримки продажу і придбання аксесуарів для автомобілів через програму. Підвищений обмін інформацією щодо аксесуарів автомобілями має сприяти привертанню уваги до програми.

Галузь використання. Створення майданчика для придбання і продажу аксесуарів для автомобілів, як нових, так і використовуваних зокрема від власників.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 93 pages, 3 tables, 27 figures, 3 appendixes, 9 sources.

WEB APPLICATION, CAR, ACCESSORY, SEARCH, DATABASE.

The object of the work is the process of searching for car accessories. The subject of the work is car accessory search software. The purpose of the work is to develop a car accessory search program to support the sale of car accessories with the provision of extensive information by owners.

Materials, methods and tools. Python programming language, MySQL database system, Django framework for web development.

Results. As a result of the set of results obtained during the stages of the work, car accessory search software was developed, which can be used by persons who need to purchase car accessories, persons who need to sell car accessories.

Conclusions. Car accessory search software can be used both to search for information about existing car accessories on the market, the practice of using such accessories in their cars by owners, and to support selling and purchasing car accessories through the program. Increased information exchange regarding car accessories should help draw attention to the program.

Field of use. Creating a platform for buying and selling car accessories, both new and used, in particular from owners.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Огляд програм пошуку аксесуарів для автомобілів	10
1.2 Порівняння програм пошуку аксесуарів для автомобілів	13
1.3 Висновки за розділом 1	15
2 Проєктування програми.....	16
2.1 Вибір засобів проєктування і розробки програми пошуку аксесуарів для автомобілів	16
2.2 Моделювання роботи користувачів з програмним забезпеченням пошуку аксесуарів для автомобілів	17
2.3 Визначення структури бази даних програми пошуку аксесуарів для автомобілів	22
2.4 Висновки за розділом 2	31
3 Розробка програми	33
3.1 Визначення послідовності роботи з програмою пошуку аксесуарів для автомобілів	33
3.2 Особливості реалізації програми пошуку аксесуарів для автомобілів ..	34
3.3 Висновки за розділом 3	48
4 Експлуатація та тестування програми	49
4.1 Призначення програми пошуку аксесуарів для автомобілів.....	49
4.2 Умови виконання програми пошуку аксесуарів для автомобілів.....	49
4.3 Експлуатація програми пошуку аксесуарів для автомобілів	49
4.4 Тестування програми пошуку аксесуарів для автомобілів.....	54
4.5 Висновки за розділом 4	57
Висновки	58
Перелік джерел посилення	59
Додаток А Технічне завдання	60

Додаток Б Текст програми	65
Додаток В Слайди презентації.....	87

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

БД – база даних;

ФВ – фреймворк.

ВСТУП

Автомобілі потребують постійного обслуговування, а разом з тим багато автовласників бажають підвищити якість користування власними транспортними засобами, а тому використовують додаткові аксесуари, підвищуючи комфорт. Це створює загалом величезний ринок, для зручного орієнтування в якому потрібне програмне забезпечення, ефективно орієнтоване на користувачів.

У той же час продаж аксесуарів для автомобілів може здійснюватися не тільки в новому стані, але і тих, що були у використанні. У такому випадку досвід використання таких аксесуарів для автомобілів може бути корисним для потенційних покупців, а тому необхідний інструмент, який дозволить таким досвідом ділитися у самому програмному забезпеченні. На реалізацію таких можливостей і має бути направлене виконання роботи.

Оцінюючи перспективи, загалом варто враховувати, що розмір ринку аксесуарів для автомобілів у 2022 році оцінювався загалом у число в 196 мільярдів американських доларів, а з того моменту передбачалось, що середній річний темп зростання у період наступних 10 років складе 5,5 % [1].

Мета роботи пов'язана з тим, щоб розробити програму пошуку аксесуарів для автомобілів для підтримки продажу автомобільних аксесуарів з наданням широкої інформації власниками, а для цього у подальших розділах роботи має бути досягнуто завдання:

- аналіз проблеми розробки програми пошуку аксесуарів для автомобілів;
- проєктування програми пошуку аксесуарів для автомобілів;
- розробка програми пошуку аксесуарів для автомобілів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд програм пошуку аксесуарів для автомобілів

В Україні широковідомим сервісом продажу різних товарів дрібними торговцями є OLX [2] – це платформа, на якій зокрема серед всіх товарів є аксесуари і для автомобілів (рис. 1.1).

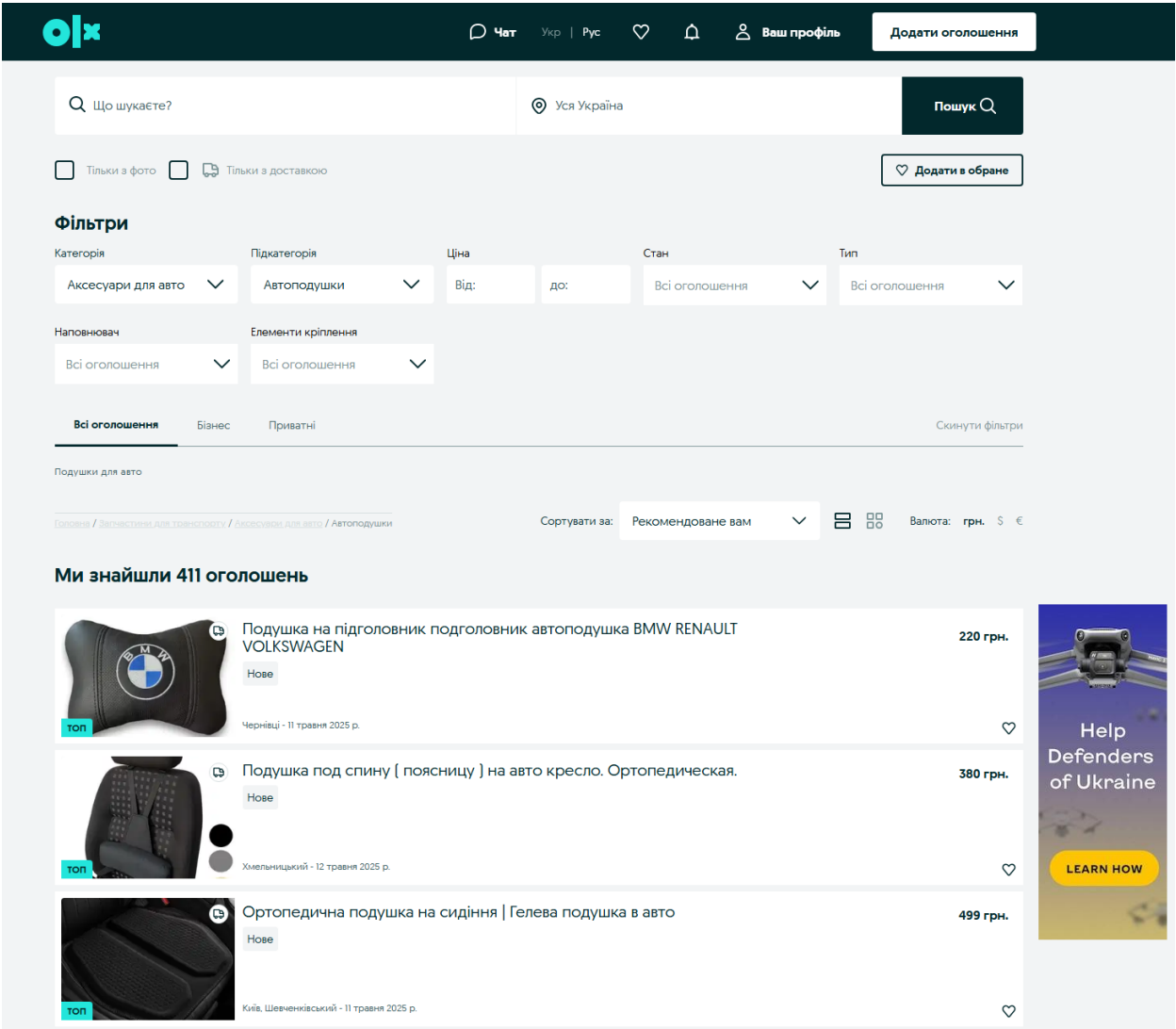


Рисунок 1.1 – Пошук аксесуарів для автомобіля на сервісі OLX

Вебзастосунок OLX містить пошук, в якому серед категорій можна як одну з можливих обрати аксесуари для автомобілів. Підкатегоріями аксесуарів для автомобілів можуть бути авто інструменти і набори ключів, автокилимки, автомобільний органайзери, автонасоси, компресори, манометри,

автоподушки, автосигналізації, автохолодильники, автошторки, ароматизатори, озонатори, багажники, брызговики, вентилятори, вогнегасники, аптечки, аварійки, декоративна підсвітка, засоби для тюнінгу, дефлектори, домкрати, елементи кріплення, каністри, ланцюги, браслети протиковзання, лебідки, наклейки, емблеми, значки, насадки на глушник, підігрів сидінь та автомасажери, підставки та тримачі, повітрозабірники, пороги та накладки, пускові та зарядні пристрої, рамки номерного знаку, салонний годинник, тонування та сонцезахист, фаркопи, чохли, тенти, накидки, шумоізоляція, щітки та скребки [2]. Під час пошуку можна задати діапазон вартості, стан оголошення (вживані чи нові). Знайдені аксесуари для автомобілів можна зберігати собі в профіль. Також можна задавати значення деяких параметрів, які залежать від обраної підкатегорії. Надається можливість взаємодіяти з продавцем.

На відміну від OLX, Lavita [3] є спеціалізованим вебсайтом саме стосовно аксесуарів для автомобілів (рис. 1.2).

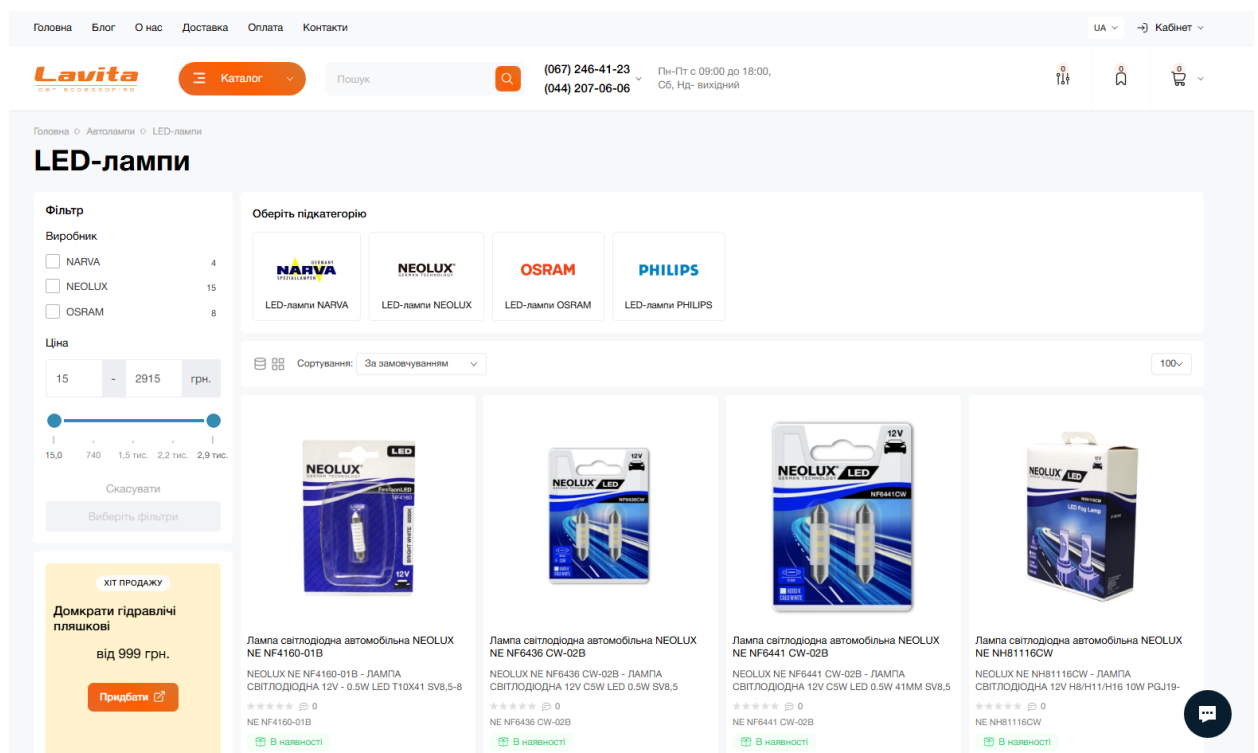


Рисунок 1.2 – Пошук аксесуарів для автомобілів у вебзастосунку Lavita

Усі аксесуари для автомобілів у сервісі Lavita представлені тільки одним продавцем, тобто на відміну від OLX продаж не може здійснюватись індивідуальними власниками. За аксесуарами можна створювати закладки, можна оформлювати замовлення на придбання аксесуарів для автомобілів.

Lavita дозволяє залишати відгуки за аксесуарами для автомобілів, при чому відгук може залишити будь-який користувач, тобто він не обов'язково є покупцем. Це не сприяє достовірності отриманих оцінок. Надається можливість поставити запитання за товаром, але такі повідомлення не є приватними.

Autobaza [4] (рис. 1.3) є вебсайтом, де надаються можливості придбання будь-яких товарів для автомобілів, не обмежуючись тільки аксесуарами.

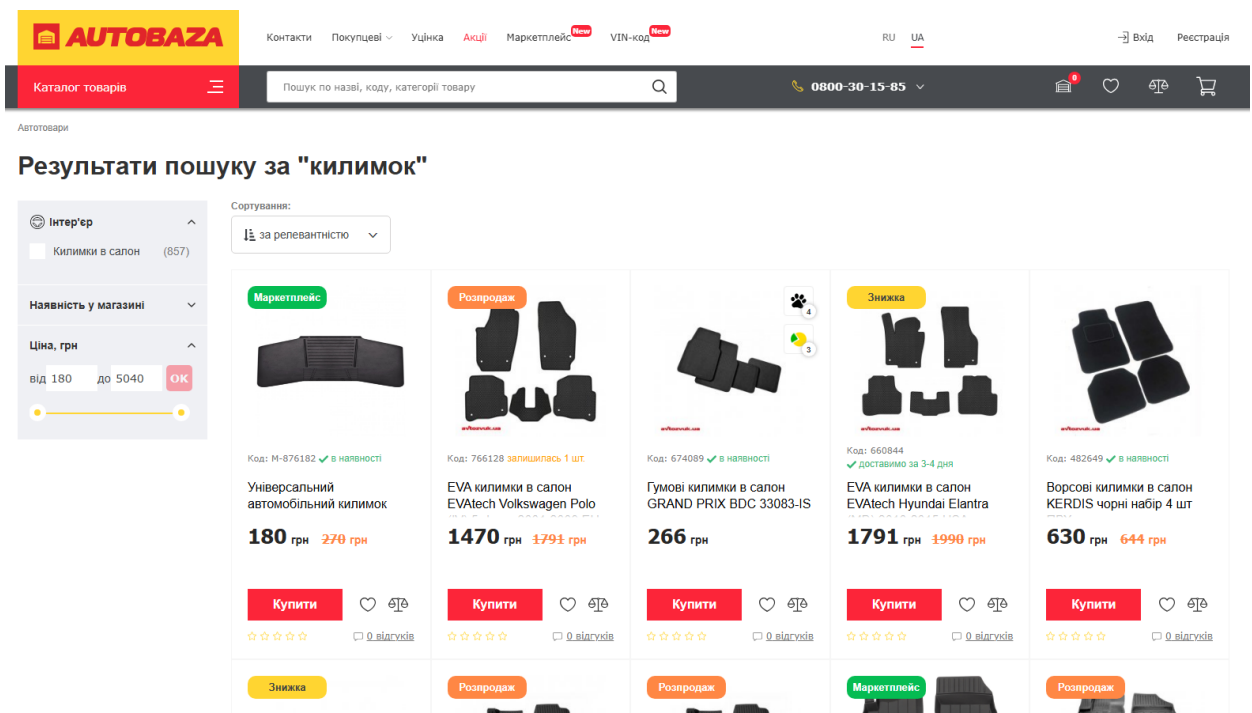


Рисунок 1.3 – Пошук аксесуарів для автомобілів у вебзастосунку Autobaza

Пошук є загальним для товарів для автомобілів, при цьому якщо обрати потрібну категорію товарів, то можна використати фільтр в тому числі за автомобілем, для якого підбирається товар, задаючи марку, модель, рік виробництва тощо (рис. 1.4). Також можна обрати потрібну категорію

продавця (сервіс або інші продавці), задати необхідність наявності у магазині, інтервал вартості, бренд, вид, застосовуваність, матеріали, тип кузова та інші атрибути, які залежать вже від обраної категорії товарів.

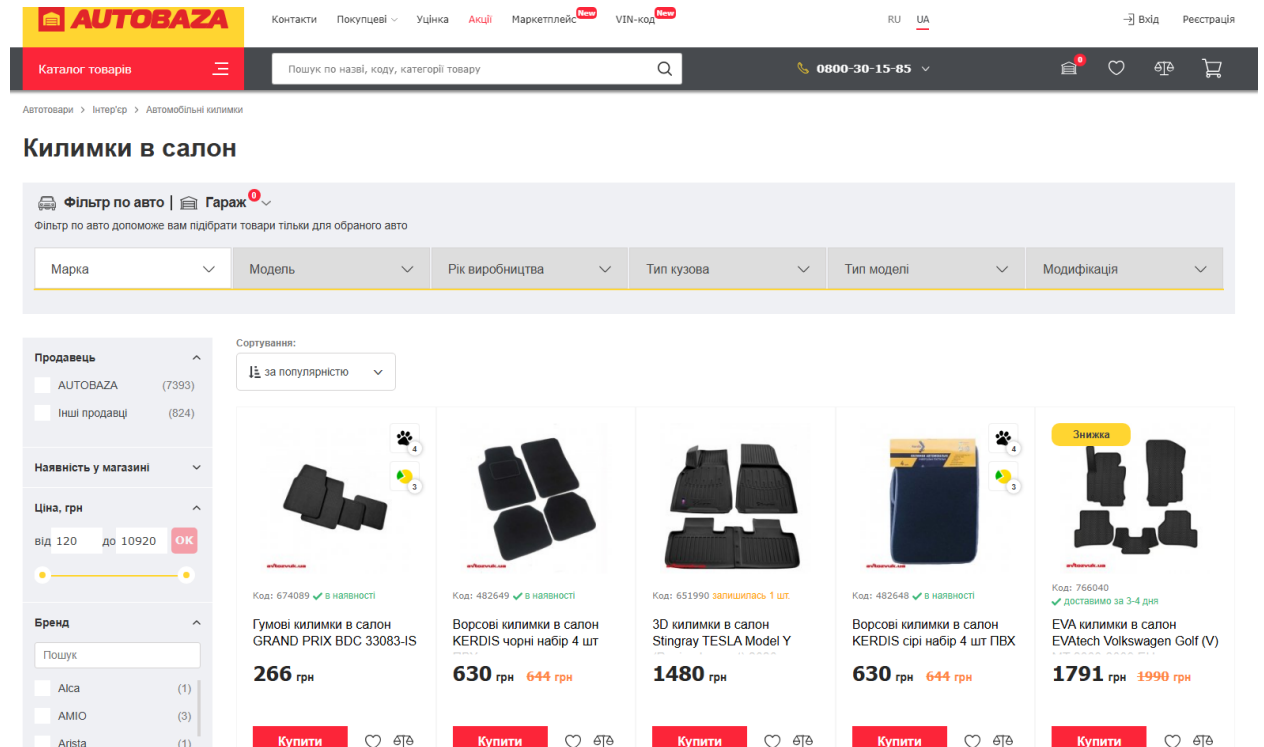


Рисунок 1.4 – Пошук за обраною категорією товарів для автомобілів у вебзастосунку Autobaza

1.2 Порівняння програм пошуку аксесуарів для автомобілів

Порівняння програм пошуку аксесуарів для автомобілів зведено до універсального майданчика продажу товарів OLX, спеціалізованого магазину продажу автоаксесуарів Lavita та спеціалізованого магазину продажу автомобільних товарів Autobaza разом з Cassgun як розробкою, що має бути виконана в цій роботі (табл. 1.1).

Cassgun в якості основної особливості має те, що ця програма має надавати можливості перегляду, створення історій використання аксесуарів від власників. Таким чином, коли власник продає вживаний аксесуар для автомобілів, яким сам раніше користувався, він може розповісти про власний

досвід, таким чином сприяючи залученості покупців, розумінню їхнього власного досвіду після придбання такого аксесуару для власного автомобіля. Така особливість може сприяти додатковому залученню користувачів до роботи з вебзастосунком.

Таблиця 1.1 – Порівняння програм пошуку аксесуарів для автомобілів

Характеристика	OLX	Lavita	Autobaza	Cassgun
Джерело продажу аксесуарів для автомобілів	Власники	Сервіс	Сервіс та інші продавці	Власники
Наявні товари	Не тільки для автомобілів	Аксесуари для автомобілів	Усі товари для автомобілів	Аксесуари для автомобілів
Історії використання аксесуарів від власників	—	—	—	+
Пошук за моделями автомобілів	—	—	+	+
Система відгуків	Після покупок	Будь-який користувач	Зареєстрований користувач	Після покупок

Cassgun має дозволити залишати відгуки тільки тим користувачам, які придбали аксесуар спершу, тобто після визначення продавцем, що аксесуар було продано. Тільки OLX використовує таку модель. При цьому OLX як універсальний майданчик не враховує підбір аксесуарів під конкретний автомобіль. Така можливість присутня тільки у Autobaza разом з виконуваною розробкою.

Концепція розробки програми пошуку аксесуарів для автомобілів передбачає, що самі власники або продавці аксесуарів для автомобілів

зможуть розміщувати власні аксесуари і залучати для себе покупців. Тобто участь масштабного продажу самим сервісом не передбачена. Це дозволяє спрямовувати діяльність на підтримку маленьких бізнесів.

1.3 Висновки за розділом 1

Проведений огляд програм пошуку аксесуарів для автомобілів було в результаті порівняння зведено до універсального майданчика продажу товарів OLX, спеціалізованого магазину продажу автоаксесуарів Lavita та спеціалізованого магазину продажу автомобільних товарів Autobaza. Розробка цієї роботи має відмінності порівняно з ними, узагальнюючи окремі переваги цих рішень, але окрім того має і ту відмінність, яка взагалі відсутня у аналогів. Такою відмінністю є можливість перегляду, створення історій використання аксесуарів від власників.

2 ПРОЄКТУВАННЯ ПРОГРАМИ

2.1 Вибір засобів проєктування і розробки програми пошуку аксесуарів для автомобілів

Програма пошуку аксесуарів для автомобілів виконувалась як веброзробка, а тому важливість фреймворків (ФВ) для веброзробки є високою. Вибір остаточного рішення зведений до Django [5]-[6] і Flask [7]-[8] (табл. 2.1) з врахуванням того, що рішення на Python дозволяють створити складні застосунки з меншими витратами, окрім того дозволяють працювати з універсальним засобом розробки, яким є сама по собі мова Python, що має багато напрямів застосування.

Таблиця 2.1 – Порівняння засобів розробки для програми пошуку аксесуарів для автомобілів

Критерій	Flask	Django
Тип ФВ	Серверний	Повний
Простота реалізації функціональних можливостей	Нижча	Вища, оскільки багато засобів інтегровано до складу, існує велика кількість засобів розширення
Захищеність	Велика кількість перевірок або засобів має бути додана для використання на практиці	Велика кількість інтегрованих засобів захисту

Рішення щодо розробки програми пошуку аксесуарів для автомобілів на Django прийнято, враховуючи, що Django надає підтримку всіх складових веброзробки, має багато інтегрованих засобів, в тому числі щодо захисту, що спрощує розробку як результат.

2.2 Моделювання роботи користувачів з програмним забезпеченням пошуку аксесуарів для автомобілів

У програмі пошуку аксесуарів для автомобілів до сценаріїв неавторизованого користувача включені (рис. 2.1):

- авторизація;
- пошук аксесуарів для автомобіля;
- перегляд інформації про аксесуар для автомобіля, виставлений на продаж;
- перегляд аксесуарів для автомобілів, виставлених на продаж заданим користувачем;
- перегляд історій застосування заданого аксесуару для автомобіля;
- перегляд сторінки продавця аксесуарів для автомобілів;
- перегляд сторінки покупця аксесуарів для автомобілів;
- перегляд відгуків про покупця аксесуарів для автомобілів;
- перегляд відгуків про продавця аксесуарів для автомобілів.

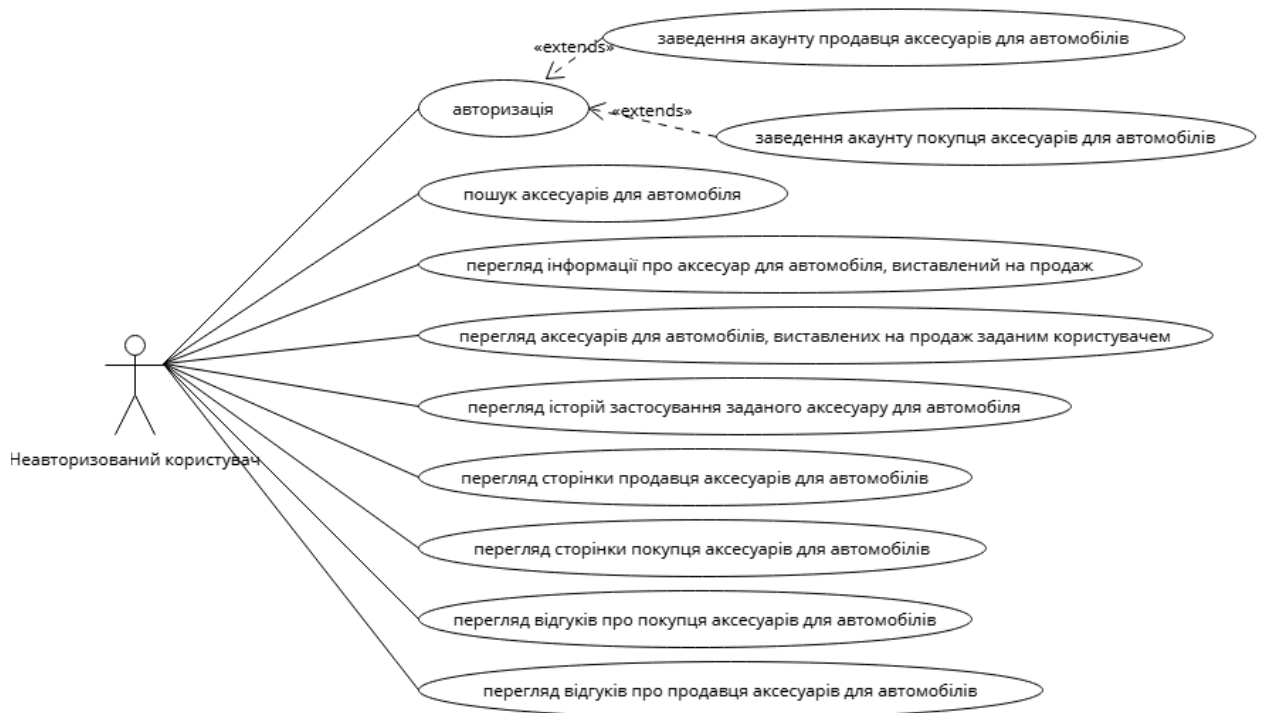


Рисунок 2.1 – Діаграма прецедентів для програми пошуку аксесуарів для автомобілів для неавторизованого користувача

Альтернативними сценаріями неавторизованого користувача програми пошуку аксесуарів для автомобілів є:

- заведення акаунту продавця аксесуарів для автомобілів стосовно сценарію авторизації, коли користувач бажає продовжити працювати з програмою як продавець аксесуарів для автомобілів, однак ще не має такого акаунту;

- заведення акаунту покупця аксесуарів для автомобілів стосовно сценарію авторизації, коли користувач бажає продовжити працювати з програмою як покупець аксесуарів для автомобілів, однак ще не має такого акаунту.

У програмі пошуку аксесуарів для автомобілів до сценаріїв продавця аксесуарів для автомобілів включені (рис. 2.2) разом зі сценаріями неавторизованого користувача:

- виставлення на продаж аксесуару для автомобіля;
- перегляд всіх нових приватних повідомлень щодо аксесуарів для автомобілів;
- відправлення приватного повідомлення щодо аксесуару для автомобіля;
- перегляд приватних повідомлень за аксесуаром для автомобіля між користувачами;
- управління замовленням придбанням аксесуару для автомобіля продавця;
- звернення до керуючого щодо замовлення придбання аксесуару для автомобіля;
- перегляд замовлених придбань аксесуару для автомобіля;
- створення відгуку про покупця аксесуару для автомобіля;
- перегляд усіх замовлених придбань аксесуарів для автомобіля за продавцем;
- публікація історії застосування власного аксесуару для автомобіля;
- відправлення приватного повідомлення щодо звернення до

керуючого;

– перегляд власних звернень до керуючого.

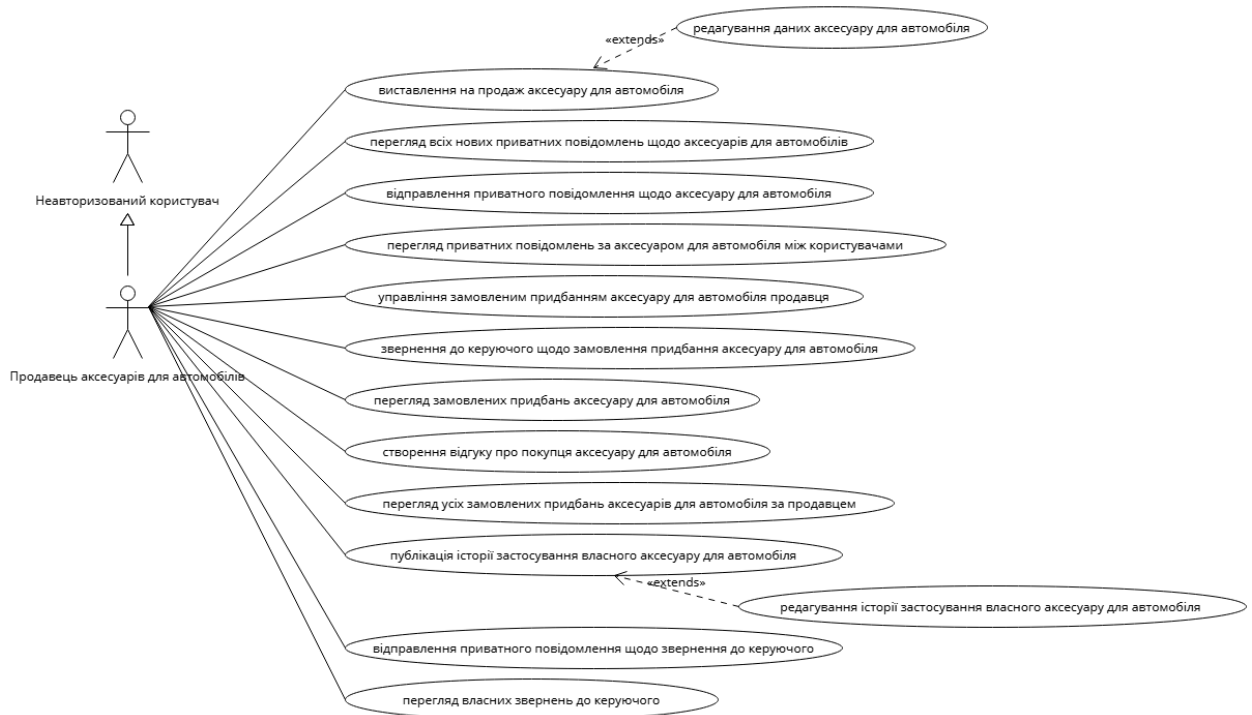


Рисунок 2.2 – Діаграма прецедентів для програми пошуку аксесуарів для автомобілів для продавця аксесуарів для автомобілів

Альтернативними сценаріями продавця аксесуарів для автомобілів є:

– редагування даних аксесуару для автомобіля, виставленого на продаж, стосовно сценарію виставлення на продаж аксесуару для автомобіля, коли продавець аксесуару для автомобілів вже виставив його на продаж, проте повинен внести зміни щодо цього;

– редагування історії застосування власного аксесуару для автомобіля стосовно сценарію публікації історії застосування власного аксесуару для автомобіля, коли історія застосування власного аксесуару для автомобіля опублікована, але в ній є неточності.

У програмі пошуку аксесуарів для автомобілів до сценаріїв покупця аксесуарів для автомобілів включені (рис. 2.3) разом зі сценаріями неавторизованого користувача:

- збереження аксесуару для автомобіля;
- перегляд збережених аксесуарів для автомобілів;
- відправлення приватного повідомлення щодо аксесуару для автомобіля;
- перегляд приватних повідомлень за аксесуаром для автомобіля між користувачами;
- управління замовленим придбанням аксесуару для автомобіля покупцем;
- створення відгуку про аксесуар для автомобіля;
- створення відгуку про продавця аксесуару для автомобіля;
- перегляд усіх замовлених придбань аксесуарів для автомобіля за покупцем.

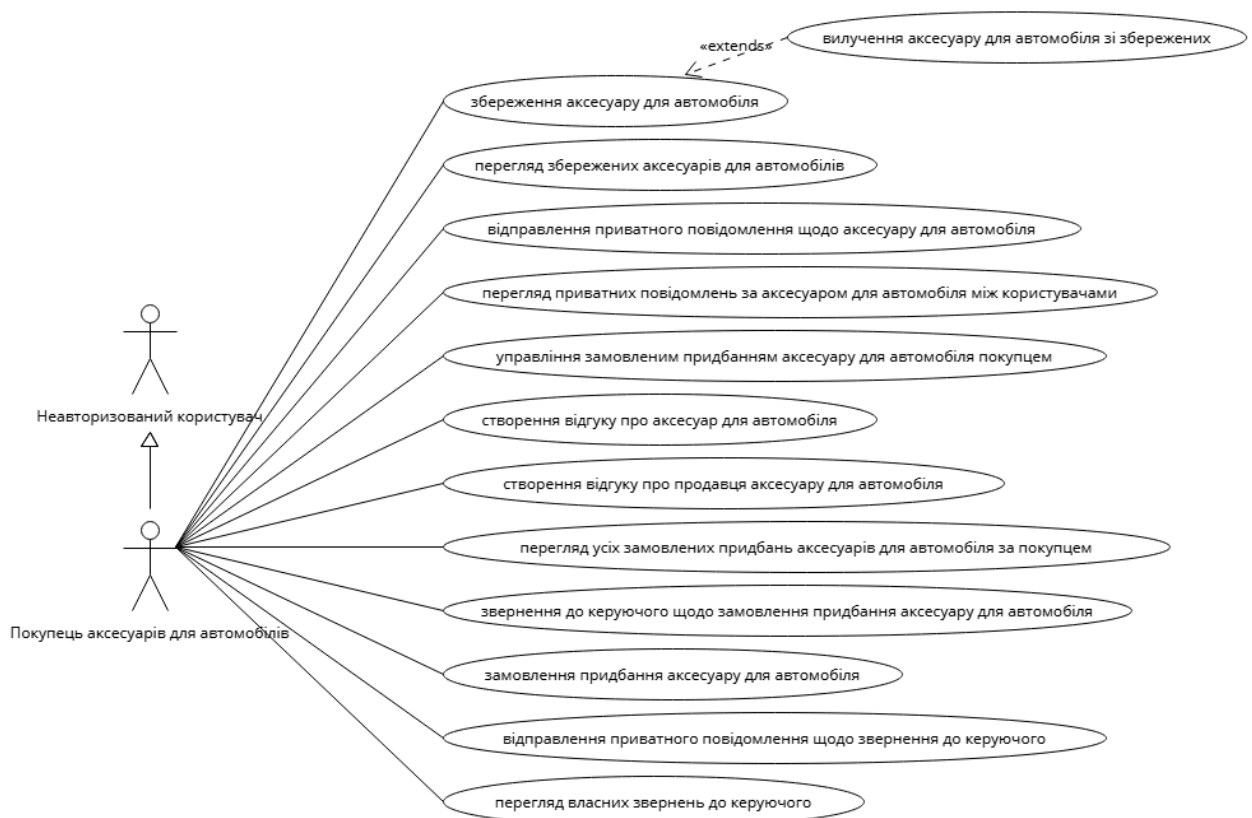


Рисунок 2.3 – Діаграма прецедентів для програми пошуку аксесуарів для автомобілів для покупця аксесуарів для автомобілів

Також до сценаріїв покупця аксесуарів для автомобілів включені:

- звернення до керуючого щодо замовлення придбання аксесуару для автомобіля;
- замовлення придбання аксесуару для автомобіля;
- відправлення приватного повідомлення щодо звернення до керуючого;
- перегляд власних звернень до керуючого.

Альтернативним сценарієм покупця аксесуарів для автомобілів є вилучення аксесуару для автомобіля зі збережених стосовно сценарію збереження аксесуару для автомобіля, коли цей аксесуар для автомобілів цим користувачем собі було збережено, проте в ньому більше немає потреби.

У програмі пошуку аксесуарів для автомобілів до сценаріїв керуючого включені (рис. 2.4) разом зі сценаріями неавторизованого користувача:

- управління замовленим придбанням аксесуару для автомобіля;
- перегляд звернень;
- перегляд замовлених придбань аксесуару для автомобіля;
- перегляд усіх замовлених придбань аксесуарів для автомобіля за продавцем та перегляд усіх замовлених придбань аксесуарів для автомобіля за покупцем;
- управління зверненням.



Рисунок 2.4 – Діаграма прецедентів для програми пошуку аксесуарів для автомобілів для керуючого

2.3 Визначення структури бази даних програми пошуку аксесуарів для автомобілів

База даних (БД) є одним з ключових рішень проектування програми пошуку аксесуарів для автомобілів, адже основна робота відбувається через взаємодію з даними, що лягають в основу роботи Django. Системою управління БД використано MySQL, що використовується в багатьох високопродуктивних практичних програмних проєктах, не вимагаючи додаткової оплати.

БД має таблицю `Caraccessorycategory`, в якій зберігаються категорії аксесуарів для автомобілів, у які об'єднані всі аксесуари для автомобілів, що мають продаватися у програмі, включаючи для цього стовпці:

- `id` – ідентифікатор категорії аксесуарів для автомобілів;
- `caraccessorycategorytitle` – назва категорії аксесуарів для автомобілів.

БД має таблицю `Caraccessorycategoryattribute`, в якій зберігаються допустимі для опису аксесуарів для автомобілів певної категорії атрибути, включаючи для цього стовпці:

- `id` – ідентифікатор допустимого для опису аксесуарів для автомобілів певної категорії атрибута;
- `caraccessorycategory` – категорія аксесуарів для автомобілів, для якої допустимо використати цей атрибут, що як зовнішній ключ пов'язана з таблицею `Caraccessorycategory`;
- `captionforaccessoryattribute` – назва допустимого для опису аксесуарів для автомобілів певної категорії атрибута.

БД має таблицю `Caraccessorycategoryattributeset`, в якій зберігаються допустимі значення для допустимих для опису аксесуарів для автомобілів певної категорії атрибутів, включаючи для цього стовпці:

- `id` – ідентифікатор допустимого значення для допустимого для опису аксесуарів для автомобілів певної категорії атрибута;

– `caraccessorycategoryattribute` – допустимий для опису аксесуарів для автомобілів певної категорії атрибут, для якого може бути визначено це допустиме значення, що як зовнішній ключ пов'язаний з таблицею `Caraccessorycategoryattribute`;

– `captionforaccessoryvalue` – саме допустиме значення для допустимого для опису аксесуарів для автомобілів певної категорії атрибута.

БД має таблицю `Carmakerbrand`, в якій зберігаються марки виробників автомобілів, включаючи для цього стовпці:

- `id` – ідентифікатор марки виробника автомобіля;
- `carmakerbrandtitle` – назва марки виробника автомобіля.

БД має таблицю `Carmakerbrandmodel`, в якій зберігаються моделі за марками автомобілів, включаючи для цього стовпці:

- `id` – ідентифікатор моделі за маркою автомобіля;
- `carmakerbrand` – марка автомобіля, за якою виробляється ця модель автомобіля, що як зовнішній ключ пов'язана з таблицею `Carmakerbrand`;
- `carmakerbrandmodeltitle` – назва моделі за маркою автомобіля.

БД має таблицю `Caraccessoryformodel`, в якій зберігаються допустимі співвідношення щодо використання аксесуарів для автомобілів для конкретних моделей за марками автомобілів, включаючи для цього стовпці:

– `id` – ідентифікатор використання аксесуара для автомобілів для конкретної моделі за маркою автомобілів;

– `caraccessory` – аксесуар для автомобілів, що використовується для певної моделі за маркою автомобіля за цим співвідношенням, який як зовнішній ключ пов'язаний з таблицею `Caraccessory`;

– `carmakerbrandmodel` – модель за маркою автомобіля, для якої може бути використаний цей аксесуар для автомобілів, яка як зовнішній ключ пов'язаний з таблицею `Carmakerbrandmodel`;

– `carmodelvariantyearstart` – рік виготовлення цієї моделі за цією маркою автомобіля, з якого може бути використаний для неї цей аксесуар для автомобілів;

– `carmodelvariantyearend` – рік виготовлення цієї моделі за цією маркою автомобіля, до якого може бути використаний для неї цей аксесуар для автомобілів.

БД має таблицю `Caraccessory`, в якій зберігаються аксесуари для автомобілів у продажі продавців, включаючи для цього стовпці:

- `id` – ідентифікатор аксесуара для автомобілів у продажі продавця;
- `caraccessorycategory` – категорія аксесуарів для автомобілів, до якої належить цей аксесуар для автомобілів, яка як зовнішній ключ пов'язана з таблицею `Caraccessorycategory`;
- `carmakerbrand` – виробник цього аксесуару для автомобілів, який як зовнішній ключ пов'язаний з таблицею `Carmakerbrand`;
- `caraccessorymodel` – модель цього аксесуара для автомобілів;
- `caraccessorydetail` – детальний опис цього аксесуара для автомобілів;
- `caraccessoryseller` – продавець аксесуарів для автомобілів, що продає даний, який як зовнішній ключ пов'язаний з таблицею `Caraccessoryseller`;
- `caraccessoryfeedbackaverage` – середній бал цього аксесуару для автомобілів, поставлений у відгуках;
- `caraccessorycondition` – стан аксесуару для автомобілів, що продається: новий або вживаний;
- `caraccessoryselling` – доступність аксесуару для автомобілів для продажу;
- `startquantity` – початкова кількість у партії в продажі цього аксесуару для автомобілів;
- `leftquantity` – поточна кількість у партії в продажі цього аксесуару для автомобілів;
- `pricesetforaunit` – вартість одиниці цього аксесуару для автомобілів.

БД має таблицю `Caraccessoryset`, в якій зберігаються значення атрибутів аксесуарів для автомобілів, включаючи для цього стовпці:

- `id` – ідентифікатор значення атрибуту аксесуару для автомобілів;

– caraccessory – аксесуар для автомобілів, що має це встановлене значення, як зовнішній ключ пов’язаний з таблицею Caraccessory;

– caraccessorycategoryattributeset – допустиме значення для допустимих для опису аксесуарів для автомобілів певної категорії атрибутів, що як конкретне значення встановлене для цього аксесуару для автомобілів, як зовнішній ключ пов’язане з таблицею Caraccessorycategoryattributeset.

БД має таблицю Caraccessoryphoto, в якій зберігаються фотографії аксесуарів для автомобілів, включаючи для цього стовпці:

- id – ідентифікатор фотографії аксесуару для автомобілів;
- caraccessory – аксесуар для автомобілів, що показаний на фотографії, як зовнішній ключ пов’язаний з таблицею Caraccessory;
- photopres – фотографія, на якій цей аксесуар для автомобілів показаний детально.

БД має таблицю Caraccessoryseller, в якій зберігаються дані продавців аксесуарів для автомобілів, включаючи для цього стовпці:

- id – ідентифікатор продавця аксесуарів для автомобілів;
- accountfromdjango – обліковий запис продавця аксесуарів для автомобілів у Django, як зовнішній ключ пов’язаний з таблицею User з Django;
- phone – номер телефону продавця аксесуарів для автомобілів;
- caraccessorysellerfeedbackaverage – середня оцінка продавця аксесуарів для автомобілів, надана у відгуках;
- caraccessorysellerdetail – детальний опис продавця аксесуарів для автомобілів про себе;
- personalphoto – фотографія продавця аксесуарів для автомобілів або презентаційний логотип, що він використовує.

БД має таблицю Caraccessorybuyer, в якій зберігаються дані покупців аксесуарів для автомобілів, включаючи для цього стовпці:

- id – ідентифікатор покупця аксесуарів для автомобілів;
- accountfromdjango – обліковий запис покупця аксесуарів для автомобілів у Django, як зовнішній ключ пов’язаний з таблицею User з Django;

- phone – номер телефону покупця аксесуарів для автомобілів;
- caraccessorybuyerfeedbackaverage – середня оцінка покупця аксесуарів для автомобілів, надана у відгуках;
- personalphoto – фотографія покупця аксесуарів для автомобілів або презентаційний логотип, що він використовує.

БД має таблицю Caraccessoryapplication, в якій зберігаються замовлення придбання аксесуарів для автомобілів, включаючи для цього стовпці:

- id – ідентифікатор замовлення придбання аксесуарів для автомобілів;
- caraccessoryseller – продавець аксесуарів для автомобілів, у якого замовлення придбання оформлено, як зовнішній ключ пов'язаний з таблицею Caraccessoryseller;
- caraccessorybuyer – покупець аксесуарів для автомобілів, який оформив замовлення придбання аксесуарів для автомобілів у цього продавця, як зовнішній ключ пов'язаний з таблицею Caraccessorybuyer;
- accessoriesprice – вартість всіх аксесуарів для автомобілів, що входять до цього замовлення придбання, в сумі;
- waitingaddress – адреса, за якою покупець аксесуарів для автомобілів передбачив необхідність доставки.

БД має таблицю Caraccessoryapplicationstatus, в якій зберігаються стани замовлень придбання аксесуарів для автомобілів, включаючи для цього стовпці:

- id – ідентифікатор стану замовлення придбання аксесуарів для автомобілів;
- caraccessoryapplication – замовлення придбання аксесуарів для автомобілів, яке характеризує цей стан, як зовнішній ключ пов'язане з таблицею Caraccessoryapplication;
- statusside – сторона (продавець або покупець аксесуарів для автомобілів) замовлення придбання аксесуарів для автомобілів, якою цей стан встановлено;

– caraccessoryapplicationstatus – конкретний стан, який встановлено для замовлення придбання аксесуарів для автомобілів;

– caraccessoryapplicationtime – час, коли замовлення придбання аксесуарів для автомобілів отримало цей стан.

БД має таблицю Caraccessoryapplicationappeal, в якій зберігаються звернення до керуючого за замовленнями придбання аксесуарів для автомобілів, включаючи для цього стовпці:

– id – ідентифікатор звернення до керуючого за замовленням придбання аксесуарів для автомобілів;

– caraccessoryapplicationstatus – стан замовлення придбання аксесуарів для автомобілів, під час отримання якого було оформлено звернення, як зовнішній ключ пов'язаний з таблицею Caraccessoryapplicationstatus;

– appealside – сторона, що створила звернення до керуючого за замовленням придбання аксесуарів для автомобілів (продавець або покупець аксесуарів для автомобілів);

– appealtime – час, коли звернення до керуючого за замовленням придбання аксесуарів для автомобілів було оформлено;

– appealprivate – приватний текст звернення до керуючого за замовленням придбання аксесуарів для автомобілів;

– appealprocessingstatus – стан опрацювання звернення до керуючого за замовленням придбання аксесуарів для автомобілів.

БД має таблицю Caraccessoryapplicationelement, в якій зберігається вміст замовлення придбання аксесуарів для автомобілів, включаючи для цього стовпці:

– id – ідентифікатор позиції в замовленні придбання аксесуарів для автомобілів;

– caraccessoryapplication – замовлення придбання аксесуарів для автомобілів, позиція в якому зазначена, представлене як зовнішній ключ, що пов'язаний з таблицею Caraccessoryapplication;

– `caraccessory` – аксесуар для автомобілів, який відноситься до цієї позиції в замовленні придбання аксесуарів для автомобілів, як зовнішній ключ пов’язаний з таблицею `Caraccessory`;

– `positionquantity` – кількість одиниць цього аксесуару для автомобілів, що відноситься до цієї позиції в замовленні придбання аксесуарів для автомобілів;

– `pricesetforaunit` – вартість одиниці цього аксесуару для автомобілів, що відноситься до цієї позиції в замовленні придбання аксесуарів для автомобілів.

БД має таблицю `Caraccessoryprivate`, в якій зберігаються приватні повідомлення щодо аксесуарів для автомобілів, включаючи для цього стовпці:

– `id` – ідентифікатор приватного повідомлення щодо аксесуарів для автомобілів;

– `caraccessory` – аксесуар для автомобілів, якого стосується приватне повідомлення, як зовнішній ключ пов’язаний з таблицею `Caraccessory`;

– `caraccessorybuyer` – покупець аксесуара для автомобіля, що бере участь в цих приватних повідомленнях, як зовнішній ключ пов’язаний з таблицею `Caraccessorybuyer`;

– `privateside` – сторона взаємодії у приватних повідомленнях між продавцем та покупцем аксесуару для автомобілів, яка створила це повідомлення;

– `privatemesstext` – текст приватного повідомлення, створеного у взаємодії між покупцем та продавцем аксесуару для автомобілів стосовно конкретного аксесуару для автомобілів;

– `privatemesssent` – час відправки приватного повідомлення, створеного у взаємодії між покупцем та продавцем аксесуару для автомобілів стосовно конкретного аксесуару для автомобілів;

– `seenbyreceiver` – позначка щодо того, чи побачила інша сторона взаємодії (не автор приватного повідомлення, а протилежна сторона) вже це приватне повідомлення щодо аксесуарів для автомобілів.

БД має таблицю Caraccessorylist, в якій знаходяться збереження аксесуарів для автомобілів користувачами, включаючи для цього стовпці:

- id – ідентифікатор збереження аксесуару для автомобілів користувачем;
- caraccessorybuyer – покупець аксесуарів для автомобілів, який це збереження зробив для себе, як зовнішній ключ пов'язаний з таблицею Caraccessorybuyer;
- caraccessory – аксесуар для автомобілів, який збережено цим покупцем аксесуарів для автомобілів для себе, як зовнішній ключ пов'язаний з таблицею Caraccessory.

БД має таблицю Caraccessorystory, в якій зберігаються історії застосування аксесуару для автомобіля, включаючи для цього стовпці:

- id – ідентифікатор історії застосування аксесуару для автомобіля;
- accessorystorycaption – назва історії застосування аксесуару для автомобіля;
- accessorystorytext – текст історії застосування аксесуару для автомобіля;
- accessorystorycreationtime – час публікації історії застосування аксесуару для автомобіля;
- storytype – тип історії застосування цього аксесуару для автомобілів: подорож, щоденне застосування, свято тощо.

БД має таблицю Caraccessorystoryelement, в якій зберігаються аксесуари для автомобілів, яких стосується ця історія застосування, включаючи для цього стовпці:

- id – ідентифікатор включення аксесуару для автомобілів до цієї історії застосування;
- caraccessorystory – історія застосування, якої стосується цей аксесуар для автомобілів, як зовнішній ключ пов'язана з таблицею Caraccessorystory;
- caraccessory – аксесуар для автомобілів, який входить до цієї історії застосування, як зовнішній ключ пов'язаний з таблицею Caraccessory.

БД має таблицю Caraccessorystoryphotoconnection, в якій зберігаються включені до історії застосування аксесуару для автомобілів фотографії, включаючи для цього стовпці:

- id – ідентифікатор включеної до історії застосування аксесуару для автомобілів фотографії;
- caraccessorystory – історія застосування аксесуару для автомобілів, в якій має бути використано цю фотографію, як зовнішній ключ пов'язана з таблицею Caraccessorystory;
- caraccessorystoryphoto – фотографія, яка має бути використана в цій історії застосування аксесуару для автомобілів, як зовнішній ключ пов'язана з таблицею Caraccessorystoryphoto.

БД має таблицю Caraccessorystoryphoto, в якій зберігаються фотографії, що можуть бути використані в історіях застосування аксесуарів для автомобілів, включаючи для цього стовпці:

- id – ідентифікатор фотографії, що може бути використана в історіях застосування аксесуарів для автомобілів;
- photopres – фотографія, яка може бути використана в історіях застосування аксесуарів для автомобілів.

БД має таблицю Caraccessoryfeedback, в якій зберігаються відгуки за замовленнями придбання аксесуарів для автомобілів, включаючи для цього стовпці:

- id – ідентифікатор відгуку за замовленням придбання аксесуарів для автомобілів;
- caraccessoryapplicationelement – вміст замовлення придбання аксесуарів для автомобілів, тобто фактично сам придбаний аксесуар для автомобілів, якого стосується цей відгук однієї зі сторін, тому що за ним відбувалась взаємодія, як зовнішній ключ пов'язаний з таблицею Caraccessoryapplicationelement;
- feedbacksetpoint – оцінка відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів;

- feedbacksettext – текст відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів;
- feedbacksettime – час визначення відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів;
- feedbacksidetype – тип визначення відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів: відгук про продавця аксесуарів для автомобілів, відгук про аксесуар для автомобілів, відгук про покупця аксесуарів для автомобілів (рис. 2.5).

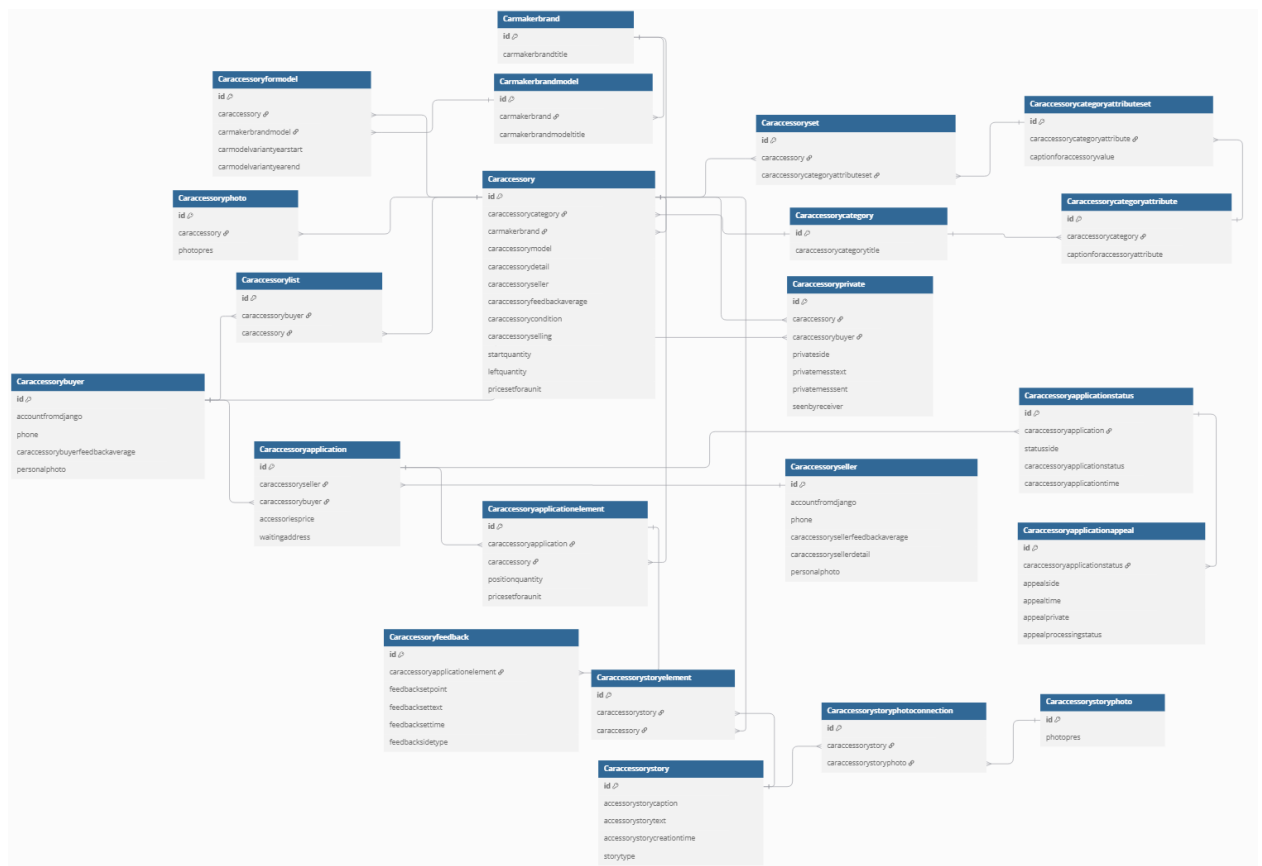


Рисунок 2.5 – Схема БД програми пошуку аксесуарів для автомобілів

2.4 Висновки за розділом 2

Розділ 2 накопичує отримані результати на етапі проєктування програми пошуку аксесуарів для автомобілів стосовно виконання відповідного завдання, що в підсумку охопило моделювання сценаріїв

покупця аксесуарів для автомобілів, продавця, керуючого та неавторизованого користувача для роботи з програмою пошуку аксесуарів для автомобілів, вибір для розробки системи MySQL, мови Python, ФВ Django, повного визначення структури БД.

3 РОЗРОБКА ПРОГРАМИ

3.1 Визначення послідовності роботи з програмою пошуку аксесуарів для автомобілів

Пошук аксесуарів для автомобілів є ключовою можливістю роботи з програмою, тому він доступний всім користувачам у всіх режимах, у тому числі до авторизації (рис. 3.1).

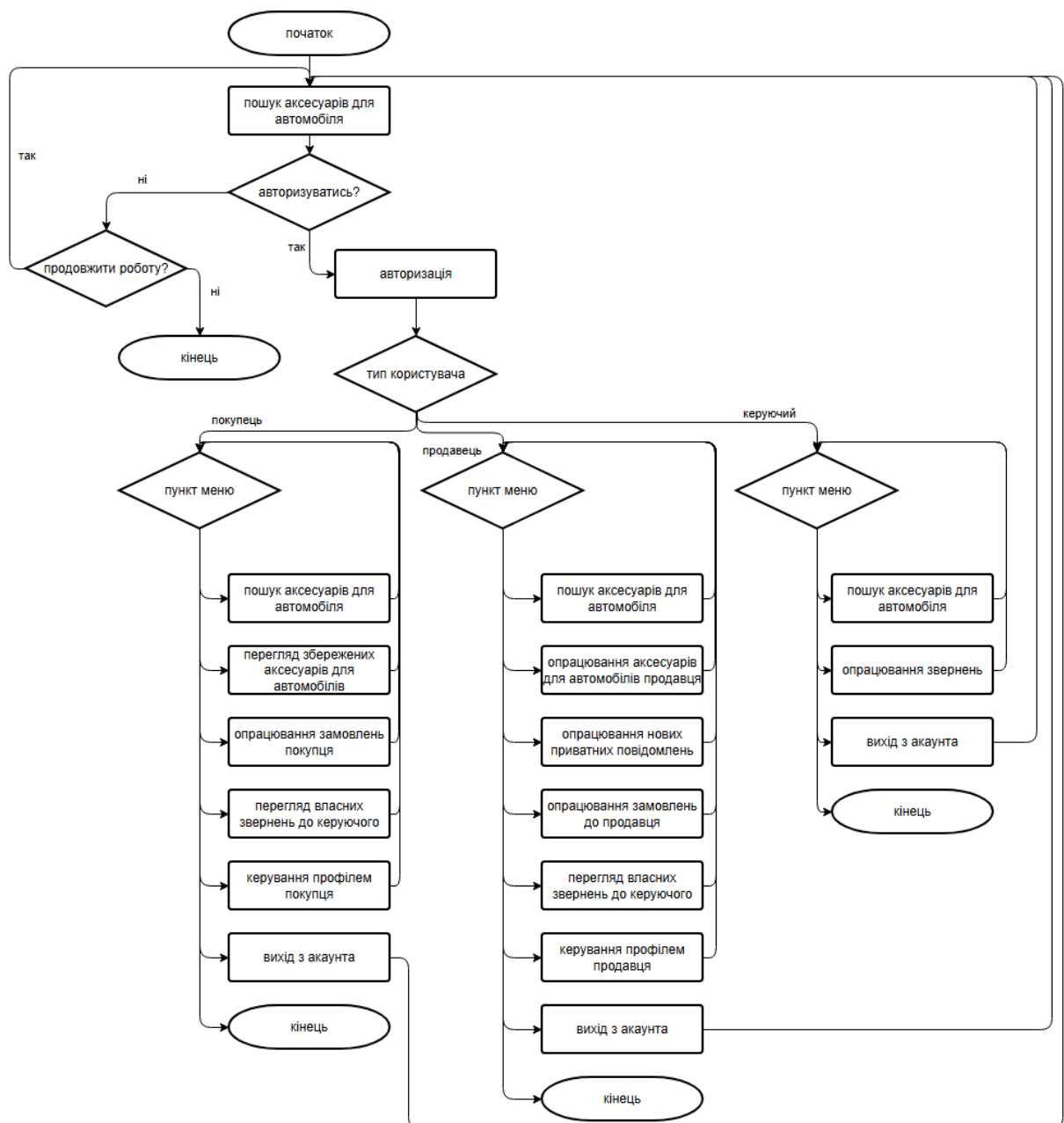


Рисунок 3.1 – Функціональна схема програми пошуку аксесуарів для автомобілів

3.2 Особливості реалізації програми пошуку аксесуарів для автомобілів

Представлення застосунку пошуку аксесуарів для автомобілів як для Django-застосунку програмно реалізовані функціями.

Функція `caraccessory_info_view` є реалізацією представлення перегляду інформації про аксесуар для автомобіля, виставлений на продаж, номер якого задається параметром `caraccessory_id`. Передбачено обробку відсутності даних такого аксесуару для автомобіля, що призведе до створення повідомлення про помилку доступу з програмно згенерованим текстом. Шаблон сторінки `caraccessory.html` використовується для заповнення даними аксесуару для автомобілів, включаючи:

- дані самого аксесуару для автомобіля за ключем `caraccessory_base_info`;
- останнє приватне повідомлення в комунікації за цим аксесуаром для автомобілів з цим покупцем (або з покупцями для продавця) за ключем `privates`;
- кількість відгуків за цим аксесуаром для автомобілів, створених покупцем, за ключем `personalfeedbacks` (тільки за доступу покупця);
- кількість замовлень придбання цього аксесуару для автомобілів, створених покупцем, за ключем `applicationnumber` (тільки за доступу покупця);
- загальна кількість відгуків за цим аксесуаром для автомобілів за ключем `feedbacknumber`;
- відгуки за цим аксесуаром для автомобілів за ключем `feedbacks`;
- фотографії цього аксесуару для автомобілів за ключем `photos`;
- останні 3 історії застосування цього аксесуару для автомобілів за ключем `stories`;
- загальна кількість історій застосування цього аксесуару для автомобілів його продавцем за ключем `storiesnumber`;

- атрибути, що описують цей аксесуар для автомобілів, за ключем `attributes`;
- моделі автомобілів, для яких може використовуватись цей аксесуар для автомобілів, за ключем `carmodels`;
- кількість приватних повідомлень, відправлена цим покупцем за цим аксесуаром для автомобілів, (або нових приватних повідомлень для продавця) за ключем `privatesnumber`.

Дизайн шаблонів сторінок створено за допомогою вільно розповсюдженого шаблону [9].

Функція `caraccessory_stories_view` є реалізацією представлення перегляду історій застосування заданого аксесуару для автомобіля з номером `caraccessory_id`, що є параметром функції. Окрім загальних даних про аксесуар для автомобілів надається повний перелік історій застосування аксесуару для автомобілів його продавцем, розташовуючи по 10 історій застосування цього аксесуару для автомобілів на сторінці. Використовується створений для цього шаблон сторінки `caraccessorystories.html`.

Функція `caraccessory_privates_view` є реалізацією представлення перегляду приватних повідомлень за аксесуаром для автомобілів між користувачами. Номер самого аксесуару для автомобілів задається параметром `caraccessory_id`. Параметр `caraccessorybuyer` є опційним та використовується тоді, коли доступ до сторінки отримує продавець цього аксесуару для автомобілів. У такому випадку, якщо значення `caraccessorybuyer` не задано, то повертається перелік тих покупців аксесуарів для автомобілів, які звертались з приватними повідомленнями за цим аксесуаром для автомобілів, а якщо значення задано, то повертаються приватні повідомлення саме між цим продавцем та заданим покупцем. Використовується створений для цього шаблон сторінки `caraccessoryprivates.html`.

Функція `caraccessory_applications_view` є реалізацією представлення перегляду замовлених придбань аксесуару для автомобіля з номером `caraccessory_id`, що є параметром, при цьому необов'язковий параметр

status_to_display визначає потрібний стан таких замовлень придбань аксесуару для автомобіля. Якщо стан визначений як -1, то стан не враховується. Використовується створений для цього шаблон сторінки caraccessoryapplications.html.

Функція caraccessories_list_view є реалізацією представлення перегляду збережених аксесуарів для автомобілів. Користувач обов'язково повинен бути авторизованим як покупець аксесуарів для автомобілів. Використовується створений для цього шаблон сторінки caraccessorieslist.html.

Функція seller_caraccessories_list_view є реалізацією представлення перегляду аксесуарів для автомобілів, виставлених на продаж заданим користувачем з логіном caraccessories_seller. Якщо доступ виконується самим цим продавцем аксесуарів для автомобілів, то вибираються всі його аксесуари для автомобілів, а якщо будь-яким іншим користувачем, то відповідно тільки ті, які доступні зараз для продажу. Використовується створений для цього шаблон сторінки sellercaraccessories.html.

Функція caraccessoryseller_view є реалізацією представлення перегляду сторінки продавця аксесуарів для автомобілів з логіном caraccessoryseller_login. Якщо значення не вказано, то користувач має авторизуватись як продавець аксесуарів для автомобілів. За продавцем відбираються 10 його доступних аксесуарів для автомобілів, 10 відгуків про нього. Використовується створений для цього шаблон сторінки caraccessoryseller.html.

Функція caraccessoryseller_feedbacks_view є реалізацією представлення перегляду повного набору відгуків про продавця аксесуарів для автомобілів з розділенням на сторінки по 10 відгуків на одній. Використовується створений для цього шаблон сторінки caraccessorysellerbacks.html.

Функція caraccessorybuyer_view є реалізацією представлення перегляду сторінки покупця аксесуарів для автомобілів з логіном caraccessorybuyer_login. Якщо значення не вказано, то користувач має

авторизуватись як покупець аксесуарів для автомобілів. За покупцем аксесуарів для автомобілів окрім інформації про нього відбираються 10 відгуків про нього. Використовується створений для цього шаблон сторінки `caraccessorybuyer.html`.

Функція `caraccessorybuyer_feedbacks_view` є реалізацією представлення перегляду повного набору відгуків про покупця аксесуарів для автомобілів з розділенням на сторінки по 10 відгуків на одній. Використовується створений для цього шаблон сторінки `caraccessorybuyerbacks.html`.

Функція `caraccessory_create_view` є реалізацією представлення виставлення на продаж аксесуару для автомобіля, а підтримку роботи з самим аксесуаром для автомобіля забезпечує клас форми `CaraccessoryForm`, користувач повинен бути авторизований як продавець аксесуарів для автомобілів. Використовується створений для цього шаблон сторінки `caraccessorydet.html`. За цим класом `CaraccessoryForm` влаштовано і функцію `caraccessory_edit_view` представлення редагування даних аксесуару для автомобіля, виставленого на продаж.

Функція `caraccessorystory_create_view` є реалізацією представлення публікації історії застосування власного аксесуару для автомобіля, а підтримку роботи з самою історією застосування власного аксесуару для автомобіля забезпечує клас форми `CaraccessorystoryForm`, користувач повинен бути авторизований як продавець аксесуарів для автомобілів. Використовується створений для цього шаблон сторінки `caraccessorystorydet.html`. За цим класом `CaraccessorystoryForm` влаштовано і функцію `caraccessorystory_edit_view` представлення редагування історії застосування власного аксесуару для автомобіля.

Функція `caraccessoryfeedback_buyer_create_view` є реалізацією представлення створення відгуку про покупця аксесуару для автомобіля, при цьому відгук створюється на основі існуючого замовлення придбання аксесуарів для автомобілів з позицією в ньому з номером

caraccessoryapplicationelement_id. Параметри feedbacksetpoint і feedbacksettext визначають відповідно число оцінки відгуку та текст.

Функція caraccessoryfeedback_accessory_create_view є реалізацією представлення створення відгуку про аксесуар для автомобіля, при цьому відгук створюється на основі існуючого замовлення придбання аксесуарів для автомобілів з позицією в ньому з номером caraccessoryapplicationelement_id. Параметри feedbacksetpoint і feedbacksettext визначають відповідно число оцінки відгуку та текст.

Функція caraccessoryfeedback_seller_create_view є реалізацією представлення створення відгуку про продавця аксесуару для автомобіля, при цьому відгук створюється на основі існуючого замовлення придбання аксесуарів для автомобілів з позицією в ньому з номером caraccessoryapplicationelement_id. Параметри feedbacksetpoint і feedbacksettext визначають відповідно число оцінки відгуку та текст.

Функція buyer_caraccessories_applications_view є реалізацією представлення перегляду всіх замовлених придбань аксесуарів для автомобіля за покупцем з логіном caraccessorybuyer або з логіном, взятим з авторизації покупця аксесуарів для автомобілів як користувача.

Функція caraccessories_search_view є реалізацією представлення пошуку аксесуарів для автомобіля. Ключі запиту містять параметри пошуку, включаючи зокрема: caraccscateg – номер категорії аксесуарів для автомобілів (-1 для цього параметру та інших для того, щоб не обмежувати значення за параметром), caraccsbrand – номер бренду виробника аксесуарів для автомобілів, carbrand та carmodel – виробник та модель автомобіля, для якого покупець збирається використовувати цей аксесуар для автомобілів, caraccsmodel – назва моделі аксесуара для автомобілів, caraccscond – стан аксесуарів для автомобілів (нові, вживані), mincaraccsprice, maxcataccsprice – мінімальна та максимальна вартість аксесуара для автомобілів, caraccsproperty та caraccspropertyval – номер властивості та значення, яке повинен мати аксесуар для автомобілів, caraccsorder – напрямок сортування аксесуарів для

автомобілів. Використовується створений для цього шаблон сторінки `caraccessoryfind.html`.

Також відповідно створено зокрема функції представлень перегляду всіх нових приватних повідомлень щодо аксесуарів для автомобілів `seller_new_caraccessoryprivates_view`, відправлення приватного повідомлення щодо аксесуару для автомобіля `caraccessoryprivate_create_view`, управління замовленим придбанням аксесуару для автомобіля продавця, управління замовленим придбанням аксесуару для автомобіля та управління замовленим придбанням аксесуару для автомобіля покупцем `caraccessoryapplication_manage_view` (дії виконуються або менеджером, або перевіряється, що замовлення належить продавцю аксесуарів для автомобілів, що звернувся, або покупцю), звернення до керуючого щодо замовлення придбання аксесуару для автомобіля `caraccessoryapplication_appeal_create_view`, перегляду всіх замовлених придбань аксесуарів для автомобіля за продавцем `seller_all_caraccessoryapplications_view`, відправлення приватного повідомлення щодо звернення до керуючого `caraccessoryprivate_create_view`, перегляду власних звернень до керуючого `personal_caraccessoryapplicationappeals_view`, замовлення придбання аксесуару для автомобіля `caraccessoryapplication_create_view`, перегляду звернень до керуючого `caraccessoryapplication_appeals_view`, управління зверненням до керуючого `caraccessoryapplication_appeal_manage_view`.

Клас `Caraccessorycategory` реалізує модель категорії аксесуарів для автомобілів, у які об'єднані всі аксесуари для автомобілів, що мають продаватися у програмі. Оскільки ідентифікатор категорії аксесуарів для автомобілів присутній у класі моделі Django одразу через наслідування класу `django.db.models.Model`, то має тільки поле `caraccessorycategorytitle` типу `CharField` з назвою категорії аксесуарів для автомобілів.

Клас `Caraccessorycategoryattribute` реалізує модель допустимих для опису аксесуарів для автомобілів певної категорії атрибутів.

У класі `Caraccessorycategoryattribute` поле `caraccessorycategory` задає категорію аксесуарів для автомобілів, для якої допустимо використати цей атрибут, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessorycategory`.

У класі `Caraccessorycategoryattribute` поле `captionforaccessoryattribute` типу `CharField` задає назву допустимого для опису аксесуарів для автомобілів певної категорії атрибута.

Клас `Caraccessorycategoryattributeset` реалізує модель допустимих значень для допустимих для опису аксесуарів для автомобілів певної категорії атрибутів.

У класі `Caraccessorycategoryattributeset` поле `caraccessorycategoryattribute` задає допустимий для опису аксесуарів для автомобілів певної категорії атрибут, для якого може бути визначено це допустиме значення, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessorycategoryattribute`.

У класі `Caraccessorycategoryattributeset` поле `captionforaccessoryvalue` типу `CharField` задає саме допустиме значення для допустимого для опису аксесуарів для автомобілів певної категорії атрибута.

Клас `Carmakerbrand` реалізує модель марок виробників автомобілів.

У класі `Carmakerbrand` поле `carmakerbrandtitle` типу `CharField` задає назву марки виробника автомобіля.

Клас `Carmakerbrandmodel` реалізує модель моделей за марками автомобілів.

У класі `Carmakerbrandmodel` поле `carmakerbrand` задає марку автомобіля, за якою виробляється ця модель автомобіля, як зовнішній ключ має тип `ForeignKey` на основі класу `Carmakerbrand`.

У класі `Carmakerbrandmodel` поле `carmakerbrandmodeltitle` типу `CharField` задає назву моделі за маркою автомобіля.

Клас `Caraccessoryformodel` реалізує модель допустимих співвідношень щодо використання аксесуарів для автомобілів для конкретних моделей за марками автомобілів.

У класі `Caraccessoryformodel` поле `caraccessory` задає аксесуар для автомобілів, що використовується для певної моделі за маркою автомобіля за цим співвідношенням, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessory`.

У класі `Caraccessoryformodel` поле `carmakerbrandmodel` задає модель за маркою автомобіля, для якої може бути використаний цей аксесуар для автомобілів, як зовнішній ключ має тип `ForeignKey` на основі класу `Carmakerbrandmodel`.

У класі `Caraccessoryformodel` поле `carmodelvariantyearstart` типу `PositiveSmallIntegerField` задає рік виготовлення цієї моделі за цією маркою автомобіля, з якого може бути використаний для неї цей аксесуар для автомобілів;

У класі `Caraccessoryformodel` поле `carmodelvariantyearend` типу `PositiveSmallIntegerField` задає рік виготовлення цієї моделі за цією маркою автомобіля, до якого може бути використаний для неї цей аксесуар для автомобілів.

Клас `Caraccessory` реалізує модель аксесуарів для автомобілів у продажі продавців.

У класі `Caraccessory` поле `caraccessorycategory` задає категорію аксесуарів для автомобілів, до якої належить цей аксесуар для автомобілів, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessorycategory`.

У класі `Caraccessory` поле `carmakerbrand` задає виробника цього аксесуару для автомобілів, як зовнішній ключ має тип `ForeignKey` на основі класу `Carmakerbrand`.

У класі `Caraccessory` поле `caraccessorymodel` типу `CharField` задає модель цього аксесуара для автомобілів.

У класі `Caraccessory` поле `caraccessorydetail` типу `TextField` задає детальний опис цього аксесуара для автомобілів.

У класі `Caraccessory` поле `caraccessoryseller` задає продавця аксесуарів для автомобілів, що продає даний, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessoryseller`.

У класі `Caraccessory` поле `caraccessoryfeedbackaverage` типу `DecimalField` задає середній бал цього аксесуару для автомобілів, поставлений у відгуках.

У класі `Caraccessory` поле `caraccessorycondition` типу `BooleanField` задає стан аксесуару для автомобілів, що продається: новий або вживаний.

У класі `Caraccessory` поле `caraccessoryselling` типу `BooleanField` задає доступність аксесуару для автомобілів для продажу.

У класі `Caraccessory` поле `startquantity` типу `PositiveIntegerField` задає початкову кількість у партії в продажі цього аксесуару для автомобілів.

У класі `Caraccessory` поле `leftquantity` типу `PositiveIntegerField` задає поточну кількість у партії в продажі цього аксесуару для автомобілів.

У класі `Caraccessory` поле `pricesetforaunit` типу `DecimalField` задає вартість одиниці цього аксесуару для автомобілів.

Клас `Caraccessoryset` реалізує модель значень атрибутів аксесуарів для автомобілів.

У класі `Caraccessoryset` поле `caraccessory` задає аксесуар для автомобілів, що має це встановлене значення, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessory`.

У класі `Caraccessoryset` поле `caraccessorycategoryattributeset` задає допустиме значення для допустимих для опису аксесуарів для автомобілів певної категорії атрибутів, що як конкретне значення встановлене для цього аксесуару для автомобілів, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessorycategoryattributeset`.

Клас `Caraccessoryphoto` реалізує модель фотографій аксесуарів для автомобілів.

У класі Caraccessoryphoto поле caraccessory задає аксесуар для автомобілів, що показаний на фотографії, як зовнішній ключ має тип ForeignKey на основі класу Caraccessory.

У класі Caraccessoryphoto поле photopres типу ImageField задає фотографію, на якій цей аксесуар для автомобілів показаний детально.

Клас Caraccessoryseller реалізує модель дані продавців аксесуарів для автомобілів.

У класі Caraccessoryseller поле accountfromdjango задає обліковий запис продавця аксесуарів для автомобілів у Django, як зовнішній ключ має тип OneToOneField на основі класу User.

У класі Caraccessoryseller поле phone типу CharField задає номер телефону продавця аксесуарів для автомобілів;

У класі Caraccessoryseller поле caraccessorysellerfeedbackaverage типу DecimalField задає середню оцінку продавця аксесуарів для автомобілів, надана у відгуках;

У класі Caraccessoryseller поле caraccessorysellerdetail типу TextField задає детальний опис продавця аксесуарів для автомобілів про себе;

У класі Caraccessoryseller поле personalphoto типу ImageField задає фотографію продавця аксесуарів для автомобілів або презентаційний логотип, що він використовує.

Клас Caraccessorybuyer реалізує модель даних покупців аксесуарів для автомобілів.

У класі Caraccessorybuyer поле accountfromdjango задає обліковий запис покупця аксесуарів для автомобілів у Django, як зовнішній ключ має тип OneToOneField на основі класу User.

У класі Caraccessorybuyer поле phone типу CharField задає номер телефону покупця аксесуарів для автомобілів.

У класі Caraccessorybuyer поле caraccessorybuyerfeedbackaverage типу DecimalField задає середню оцінку покупця аксесуарів для автомобілів, надана у відгуках.

У класі Caraccessorybuyer поле personalphoto типу ImageField задає фотографію покупця аксесуарів для автомобілів або презентаційний логотип, що він використовує.

У класі Caraccessorybuyer поле caraccessorylist задає перелік збережених аксесуарів для автомобілів користувачем, як зовнішній ключ має тип ManyToManyField на основі класу Caraccessory.

Клас Caraccessoryapplication реалізує модель замовлень придбання аксесуарів для автомобілів.

У класі Caraccessoryapplication поле caraccessoryseller задає продавця аксесуарів для автомобілів, у якого замовлення придбання оформлено, як зовнішній ключ має тип ForeignKey на основі класу Caraccessoryseller.

У класі Caraccessoryapplication поле caraccessorybuyer задає покупця аксесуарів для автомобілів, який оформив замовлення придбання аксесуарів для автомобілів у цього продавця, як зовнішній ключ має тип ForeignKey на основі класу Caraccessorybuyer.

У класі Caraccessoryapplication поле accessoriesprice типу DecimalField задає вартість всіх аксесуарів для автомобілів, що входять до цього замовлення придбання, в сумі.

У класі Caraccessoryapplication поле waitingaddress типу CharField задає адресу, за якою покупець аксесуарів для автомобілів передбачив необхідність доставки.

Клас Caraccessoryapplicationstatus реалізує модель станів замовлень придбання аксесуарів для автомобілів.

У класі Caraccessoryapplicationstatus поле caraccessoryapplication задає замовлення придбання аксесуарів для автомобілів, яке характеризує цей стан, як зовнішній ключ має тип ForeignKey на основі класу Caraccessoryapplication.

У класі Caraccessoryapplicationstatus поле statusside типу BooleanField задає сторону (продавець або покупець аксесуарів для автомобілів) замовлення придбання аксесуарів для автомобілів, якою цей стан встановлено.

У класі `Caraccessoryapplicationstatus` поле `caraccessoryapplicationstatus` типу `PositiveSmallIntegerField` задає конкретний стан, який встановлено для замовлення придбання аксесуарів для автомобілів.

У класі `Caraccessoryapplicationstatus` поле `caraccessoryapplicationtime` типу `DateTimeField` задає час, коли замовлення придбання аксесуарів для автомобілів отримало цей стан.

Клас `Caraccessoryapplicationappeal` реалізує модель звернень до керуючого за замовленнями придбання аксесуарів для автомобілів.

У класі `Caraccessoryapplicationappeal` поле `caraccessoryapplicationstatus` задає стан замовлення придбання аксесуарів для автомобілів, під час отримання якого було оформлено звернення, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessoryapplicationstatus`.

У класі `Caraccessoryapplicationappeal` поле `appealside` типу `BooleanField` задає сторону, що створила звернення до керуючого за замовленням придбання аксесуарів для автомобілів (продавець або покупець аксесуарів для автомобілів).

У класі `Caraccessoryapplicationappeal` поле `appealtime` типу `DateTimeField` задає час, коли звернення до керуючого за замовленням придбання аксесуарів для автомобілів було оформлено.

У класі `Caraccessoryapplicationappeal` поле `appealprivate` типу `CharField` задає приватний текст звернення до керуючого за замовленням придбання аксесуарів для автомобілів.

У класі `Caraccessoryapplicationappeal` поле `appealprocessingstatus` типу `PositiveSmallIntegerField` задає стан опрацювання звернення до керуючого за замовленням придбання аксесуарів для автомобілів.

Клас `Caraccessoryapplicationelement` реалізує модель вмісту замовлення придбання аксесуарів для автомобілів.

У класі `Caraccessoryapplicationelement` поле `caraccessoryapplication` задає замовлення придбання аксесуарів для автомобілів, позиція в якому

зазначена, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessoryapplication`.

У класі `Caraccessoryapplicationelement` поле `caraccessory` задає аксесуар для автомобілів, який відноситься до цієї позиції в замовленні придбання аксесуарів для автомобілів, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessory`.

У класі `Caraccessoryapplicationelement` поле `positionquantity` типу `PositiveSmallIntegerField` задає кількість одиниць цього аксесуару для автомобілів, що відноситься до цієї позиції в замовленні придбання аксесуарів для автомобілів;

У класі `Caraccessoryapplicationelement` поле `pricesetforaunit` типу `DecimalField` задає вартість одиниці цього аксесуару для автомобілів, що відноситься до цієї позиції в замовленні придбання аксесуарів для автомобілів.

Клас `Caraccessoryprivate` реалізує модель приватних повідомлень щодо аксесуарів для автомобілів.

У класі `Caraccessoryprivate` поле `caraccessory` задає аксесуар для автомобілів, якого стосується приватне повідомлення, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessory`.

У класі `Caraccessoryprivate` поле `caraccessorybuyer` задає покупця аксесуара для автомобіля, що бере участь в цих приватних повідомленнях, як зовнішній ключ має тип `ForeignKey` на основі класу `Caraccessorybuyer`.

У класі `Caraccessoryprivate` поле `privateside` типу `BooleanField` задає сторону взаємодії у приватних повідомленнях між продавцем та покупцем аксесуару для автомобілів, яка створила це повідомлення.

У класі `Caraccessoryprivate` поле `privatesesstext` типу `CharField` задає текст приватного повідомлення, створеного у взаємодії між покупцем та продавцем аксесуару для автомобілів стосовно конкретного аксесуару для автомобілів.

У класі `Caraccessoryprivate` поле `privatemesssent` типу `DateTimeField` задає час відправки приватного повідомлення, створеного у взаємодії між

покупцем та продавцем аксесуару для автомобілів стосовно конкретного аксесуару для автомобілів.

У класі Caraccessoryprivate поле seenbyreceiver типу BooleanField задає позначку щодо того, чи побачила інша сторона взаємодії (не автор приватного повідомлення, а протилежна сторона) вже це приватне повідомлення щодо аксесуарів для автомобілів.

Клас Caraccessorystory реалізує модель історії застосування аксесуару для автомобіля.

У класі Caraccessorystory поле accessorystorycaption типу CharField задає назву історії застосування аксесуару для автомобіля.

У класі Caraccessorystory поле accessorystorytext типу TextField задає текст історії застосування аксесуару для автомобіля.

У класі Caraccessorystory поле accessorystorycreationtime типу DateTimeField задає час публікації історії застосування аксесуару для автомобіля.

У класі Caraccessorystory поле storytype типу BooleanField задає тип історії застосування цього аксесуару для автомобілів: подорож, щоденне застосування, свято тощо.

У класі Caraccessorystory поле caraccessorystoryelements задає перелік аксесуарів для автомобілів, яких стосується ця історія застосування, як зовнішній ключ має тип ManyToManyField на основі класу Caraccessory.

У класі Caraccessorystory поле caraccessorystoryphotos задає перелік фотографій, які мають бути використані в цій історії застосування аксесуару для автомобілів, як зовнішній ключ має тип ManyToManyField на основі класу Caraccessorystoryphoto.

Клас Caraccessorystoryphoto реалізує модель фотографій, що можуть бути використані в історіях застосування аксесуарів для автомобілів.

У класі Caraccessorystoryphoto поле photopres типу ImageField задає фотографію, яку може бути використано в історіях застосування аксесуарів для автомобілів.

Клас Caraccessoryfeedback реалізує модель відгуків за замовленнями придбання аксесуарів для автомобілів.

У класі Caraccessoryfeedback поле caraccessoryapplicationelement задає вміст замовлення придбання аксесуарів для автомобілів, тобто фактично сам придбаний аксесуар для автомобілів, якого стосується цей відгук однієї зі сторін, тому що за ним відбувалась взаємодія, як зовнішній ключ має тип ForeignKey на основі класу Caraccessoryapplicationelement.

У класі Caraccessoryfeedback поле feedbacksetpoint типу PositiveSmallIntegerField задає оцінку відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів.

У класі Caraccessoryfeedback поле feedbacksettext типу CharField задає текст відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів.

У класі Caraccessoryfeedback поле feedbacksettime типу DateTimeField задає час визначення відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів.

У класі Caraccessoryfeedback поле feedbacksidetype типу PositiveSmallIntegerField задає тип визначення відгуку однією зі сторін за результатами замовлення придбання аксесуарів для автомобілів: відгук про продавця аксесуарів для автомобілів, відгук про аксесуар для автомобілів, відгук про покупця аксесуарів для автомобілів.

3.3 Висновки за розділом 3

Процес взаємодії користувачів різних ролей з програмою пошуку аксесуарів для автомобілів представлено у вигляді функціональної схеми, описано основні представлення з відповідними шаблонами представлень, що сукупно забезпечують підтримку функціональних можливостей пошуку аксесуарів для автомобілів разом з класами моделей.

4 ЕКСПЛУАТАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМИ

4.1 Призначення програми пошуку аксесуарів для автомобілів

Програмне забезпечення пошуку аксесуарів для автомобілів призначене для виставлення власних аксесуарів для автомобілів продавцями на продаж, для пошуку покупцями цих аксесуарів, отримання інформації про аксесуари на основі зокрема історії їх використання власниками.

4.2 Умови виконання програми пошуку аксесуарів для автомобілів

Розроблене програмне забезпечення пошуку аксесуарів для автомобілів передбачає виконання експлуатаційних вимог до вебзастосунку для роботи з ним, включаючи наявність спеціаліста-адміністратора, який має забезпечити хостинг, розгортання вебзастосунку на сервері з застосуванням обраного для реалізації ФВ Django, підтримку роботи БД на основі обраної в роботі системи керування MySQL, інсталювання засобів мови програмування Python. При цьому на технічному рівні стосовно організації сервера для програми пошуку аксесуарів для автомобілів необхідно забезпечити мінімально 2 Гб оперативної пам'яті, два ядра процесора, 100 Гб для організації сховища (передбачено для виконання вимоги 25000 запитів на місяць).

4.3 Експлуатація програми пошуку аксесуарів для автомобілів

Покупець аксесуарів для автомобілів може використовувати всі доступні йому функції після авторизації, що у випадку відсутності акаунта передбачає також заведення відповідного акаунта, а тоді авторизацію. Через меню він може звернутись до безпосередньо пошуку аксесуарів для автомобілів, опрацювання власних замовлень придбання аксесуарів для автомобілів, які він вже створив, збережених аксесуарів для автомобілів,

оформлених звернень до керуючого за замовлення придбання аксесуарів для автомобілів, вийти з акаунту (рис. 4.1). Під час виходу з акаунту ставиться запитання про підтвердження з можливістю переходу до власної сторінки покупця, де можна відредагувати дану акаунту.

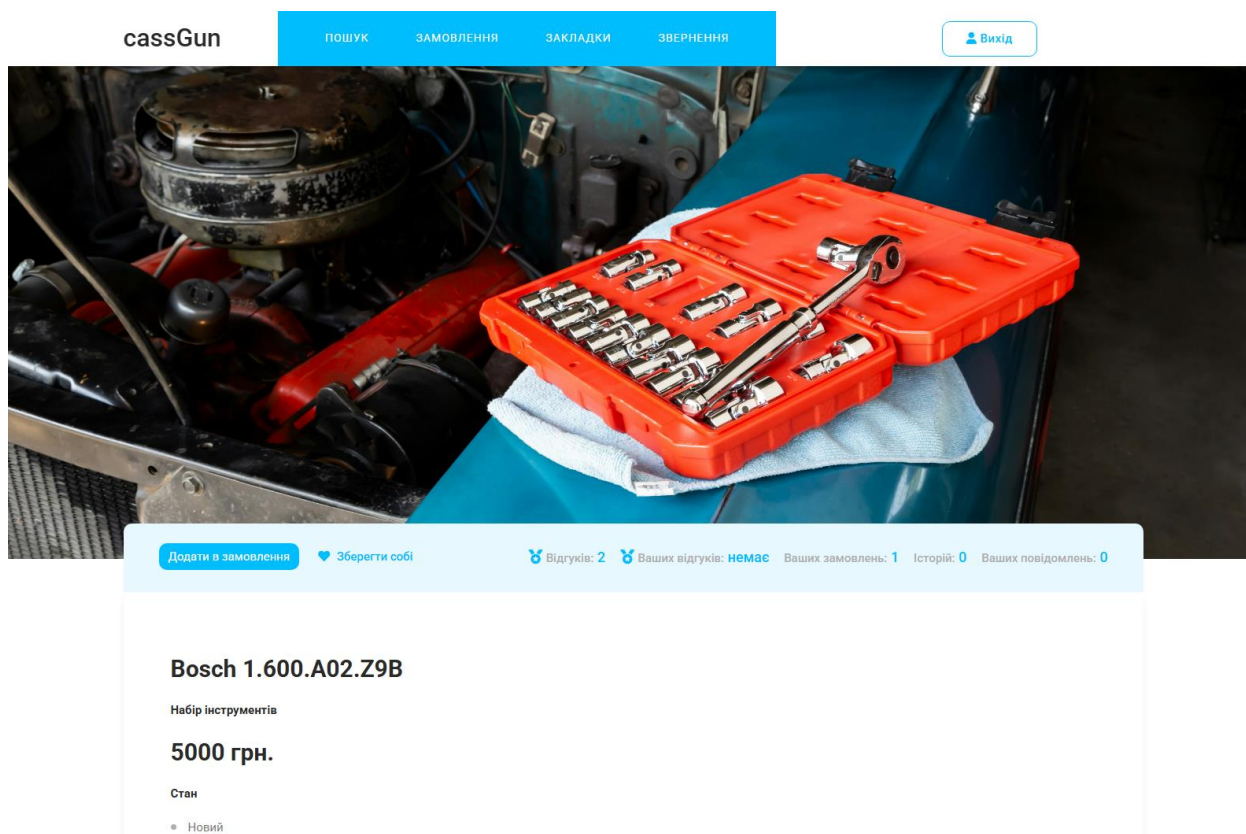


Рисунок 4.1 – Аксесуар для автомобілів у програмі пошуку аксесуарів для автомобілів зі сторони покупця

При роботі з аксесуарами для автомобілів покупець може зберігати аксесуари собі до переліку, додавати до поточного замовлення або створювати вже підготовлене замовлення на основі доданих аксесуарів. На самій сторінці аксесуарів зокрема представлена частина відгуків з можливістю відкриття всіх відгуків, 3 останні історії про застосування аксесуарів для автомобілів, фотографії аксесуара (згори сторінки з виведенням стрілок для переходу між фотографіями при наведенні на фотографію).

При роботі з власними аксесуарами для автомобілів продавець (рис. 4.2) може з такої сторінки переходити за допомогою кнопок згори

сторінки до редагування аксесуару, роботи з замовленнями за аксесуаром, додавання нової історії про застосування аксесуару для автомобіля, відтворення внизу сторінки приватних повідомлень.

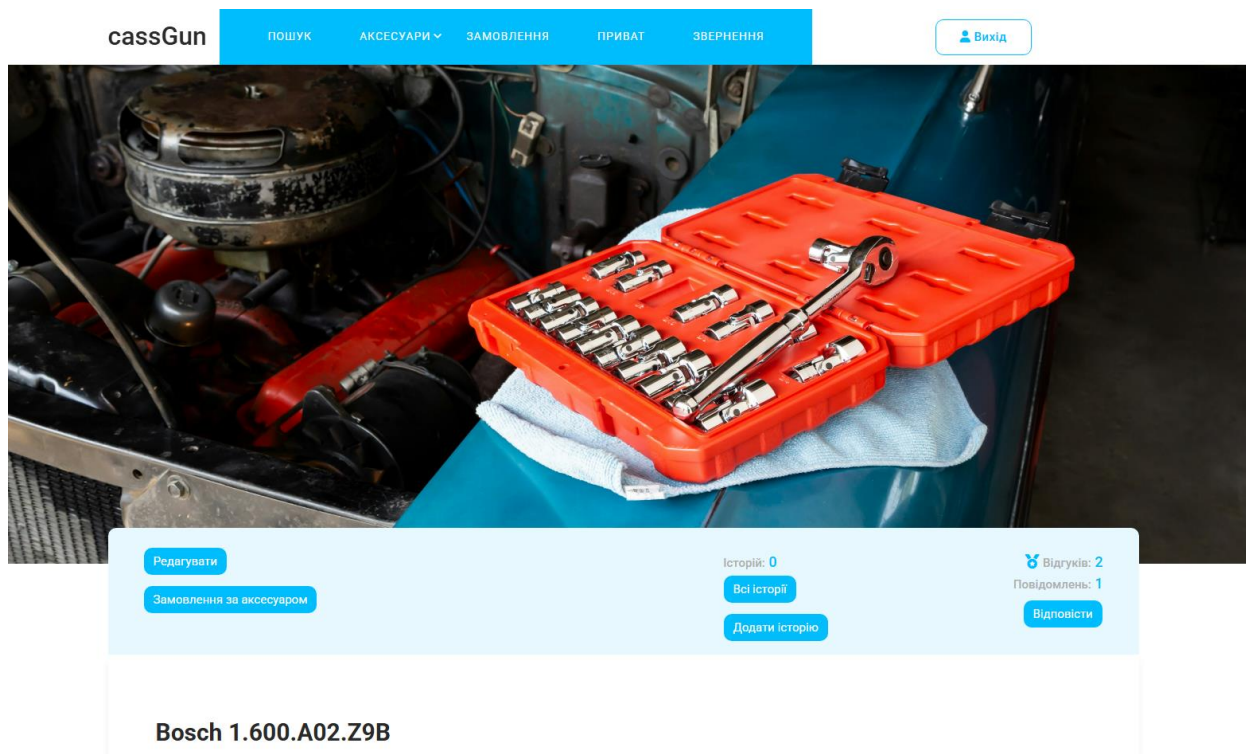


Рисунок 4.2 – Аксесуар для автомобілів у програмі пошуку аксесуарів для автомобілів зі сторони продавця

Ту ж саму сторінку може розглядати і неавторизований користувач (рис. 4.3), але маючи обмежені можливості, а в меню маючи тільки пункт для переходу до пошуку аксесуарів для автомобілів.

У випадку продавця аксесуарів для автомобілів з меню доступні засоби пошуку аксесуарів для автомобілів, розділ з випадаячими пунктами для розміщення нового аксесуару для автомобіля від продавця, власними аксесуарами для автомобілів у продажі, перегляду та управління замовленнями придбання аксесуарів для автомобілів, приватних повідомлень, які були отримані продавцем за аксесуарами для автомобілів, згруповані за аксесуарами та покупцями.

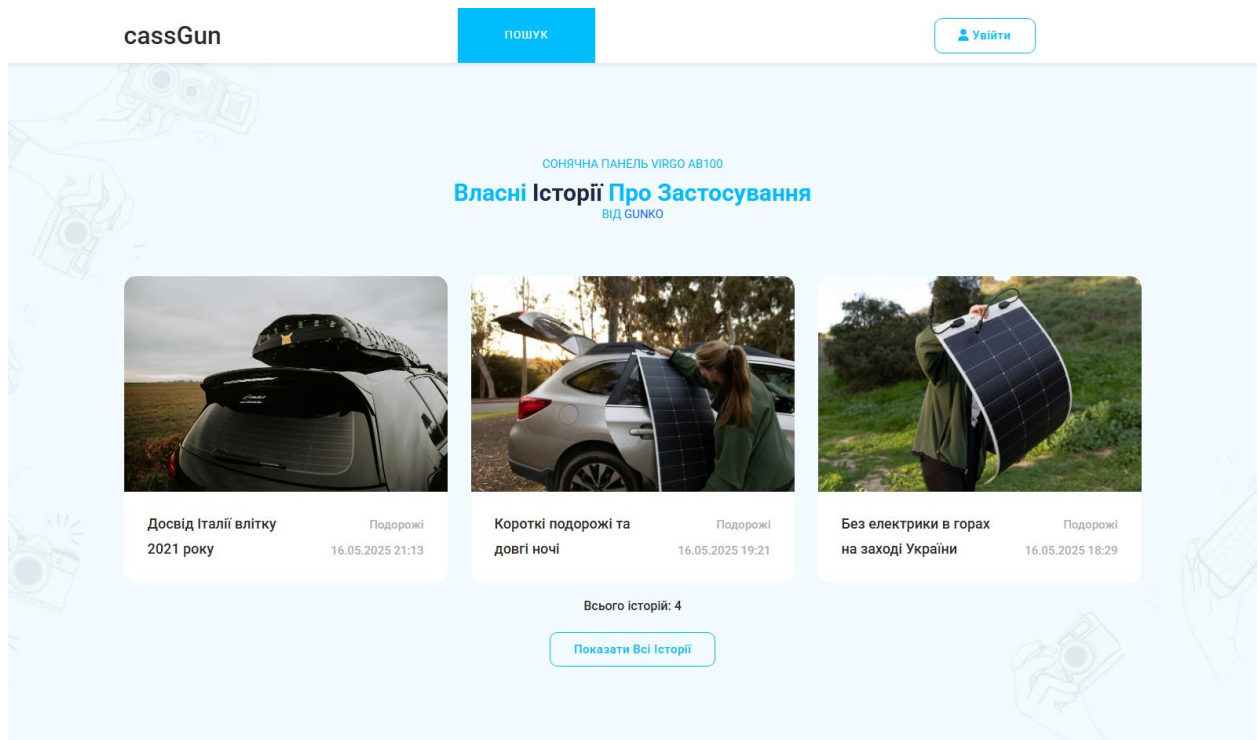


Рисунок 4.3 – Історії застосування на сторінці аксесуару для автомобілів у програмі пошуку аксесуарів для автомобілів зі сторони неавторизованого користувача

Продавець аксесуарів для автомобілів зі сторінки замовлень придбань аксесуара для автомобілів, тобто за конкретним аксесуаром, може переглядати всі замовлення, в яких міститься в переліку заданий аксесуар, встановлюючи за потреби необхідний стан цих замовлень (рис. 4.4). За замовленнями можна натиснути на кнопку детального вмісту, щоб відобразити склад замовлення (без відкриття окремої сторінки), кнопку відображення історії зміни станів, а також на кнопки, які доступні для зміни стану замовлення, виходячи з того, в якому стані зараз замовлення. У будь-якому стані за замовленням одразу продавець може звернутись до керуючого, позначаючи проблему взаємодії з покупцем. Якщо замовлення завершено, тобто отримання було підтверджено всіма сторонами, то може також залишити відгук. Продавець залишає відгук про покупця аксесуарів для автомобілів, а покупець – про продавця або про аксесуар для автомобілів.

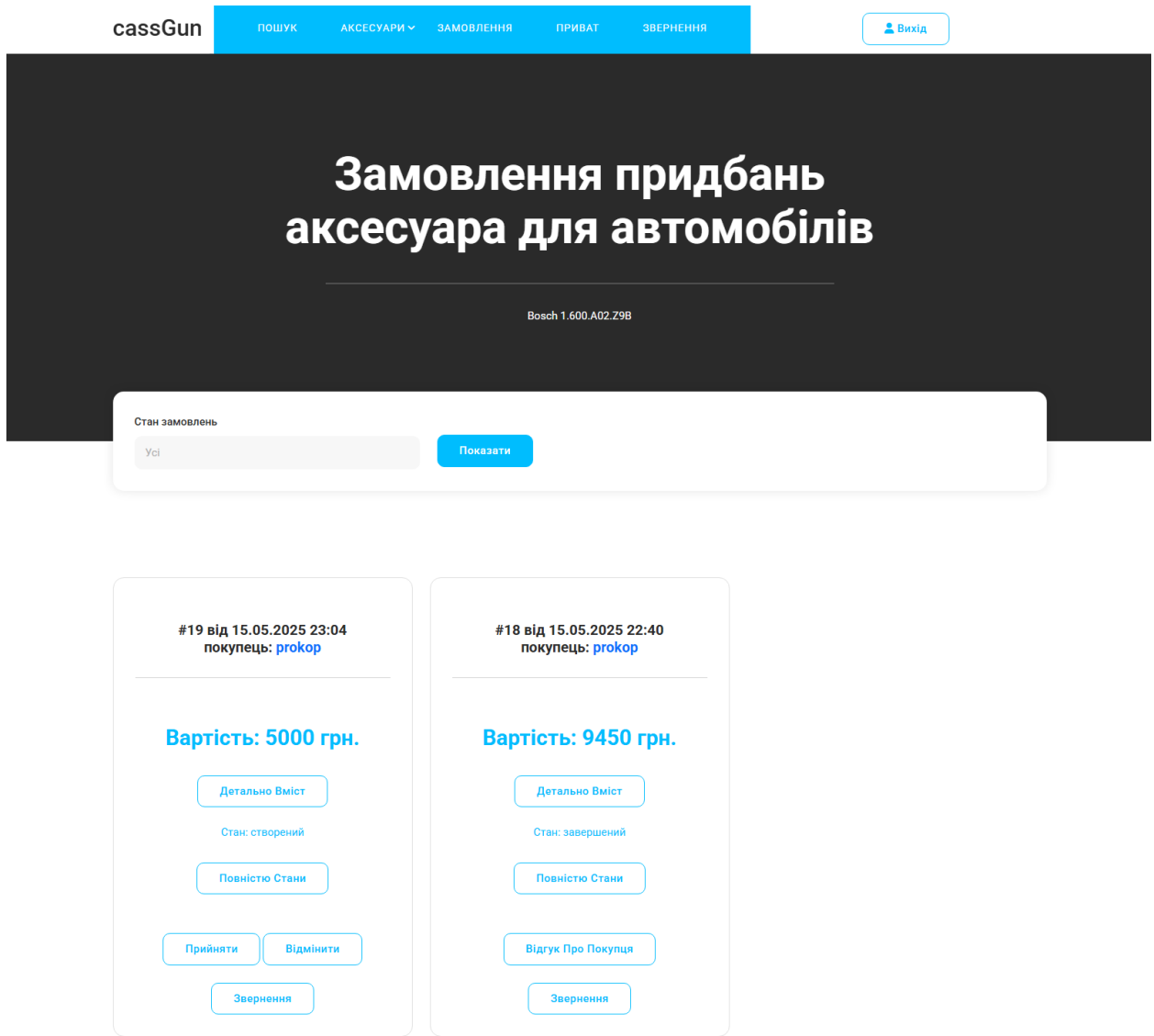


Рисунок 4.4 – Замовлення придбання аксесуарів для автомобілів у програмі пошуку аксесуарів для автомобілів зі сторони продавця

Керуючий також може працювати з замовленнями, при чому він може змінювати стан замовлення аксесуарів для автомобілів на будь-який інший (рис. 4.5), також він може опрацьовувати отримані до нього звернення щодо проблем з замовленнями придбання аксесуарів для автомобілів, взаємодіючи через повідомлення, змінюючи стан замовлення. Перехід до звернень доступний через однойменний розділ меню. Перехід до звернень за конкретним замовленням доступний через нижню кнопку в картці замовлення.

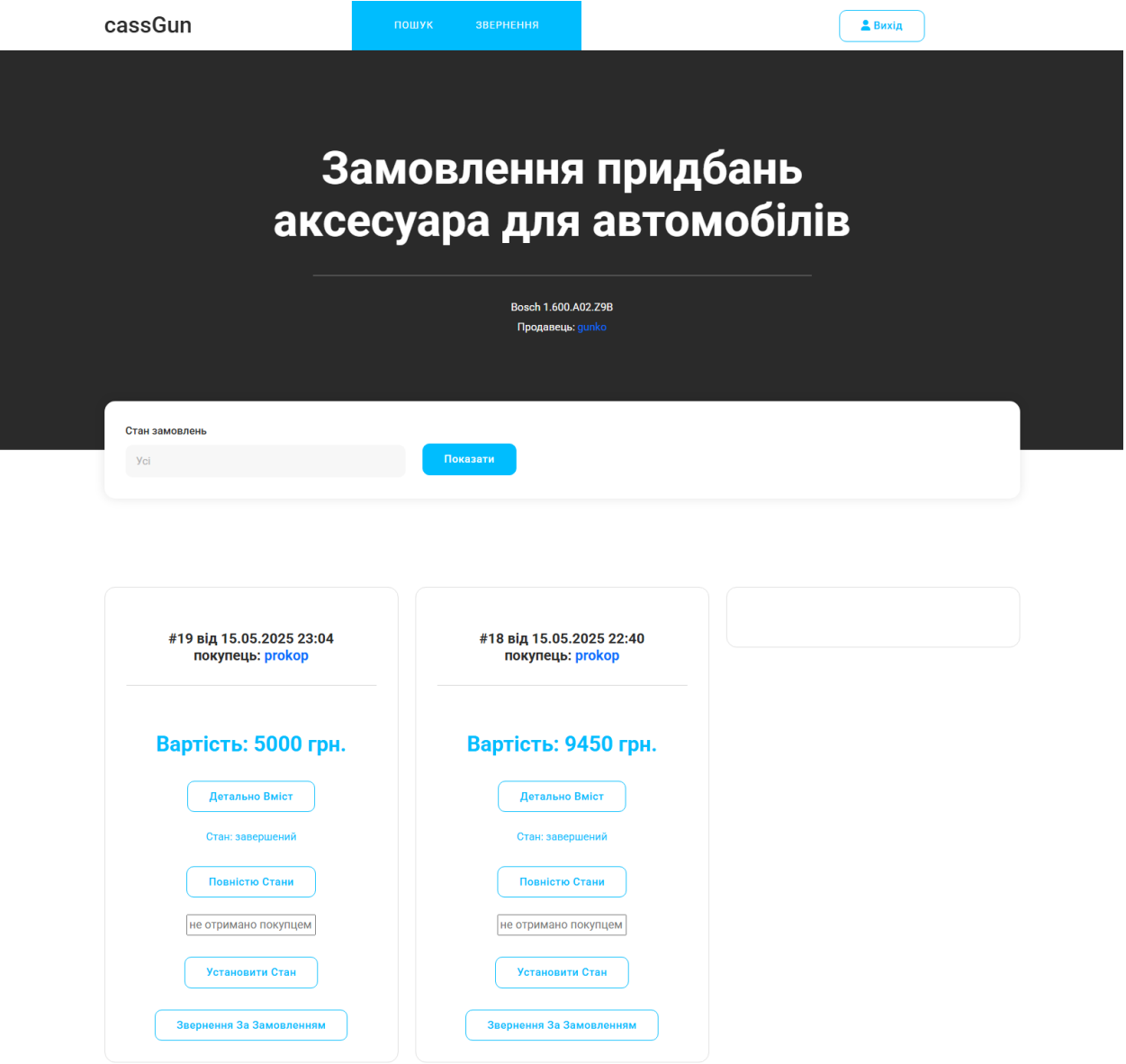


Рисунок 4.5 – Замовлення придбань аксесуара для автомобілів у програмі пошуку аксесуарів для автомобілів зі сторони керуючого

4.4 Тестування програми пошуку аксесуарів для автомобілів

Для перевірки того, чи виконані вимоги щодо функціональних можливостей програми пошуку аксесуарів для автомобілів, було забезпечено тестування, встановлюючи за кожною функцією позначку про успішне або неуспішне виконання її при доступі віддалено до вебзастосунку (табл. 4.1). Усі функції мали успішне завершення.

Таблиця 4.1 – Результати тестування програми пошуку аксесуарів для автомобілів

Функція	Результат
1	2
Пошук аксесуарів для автомобіля	+
Авторизація	+
Перегляд інформації про аксесуар для автомобіля, виставлений на продаж	+
Перегляд аксесуарів для автомобілів, виставлених на продаж заданим користувачем	+
Перегляд історій застосування заданого аксесуару для автомобіля	+
Перегляд сторінки продавця аксесуарів для автомобілів	+
Перегляд сторінки покупця аксесуарів для автомобілів	+
Перегляд відгуків про покупця аксесуарів для автомобілів	+
Перегляд відгуків про продавця аксесуарів для автомобілів	+
Заведення акаунту продавця аксесуарів для автомобілів	+
Заведення акаунту покупця аксесуарів для автомобілів	+
Виставлення на продаж аксесуару для автомобіля	+
Перегляд всіх нових приватних повідомлень щодо аксесуарів для автомобілів	+
Відправлення приватного повідомлення щодо аксесуару для автомобіля	+
Перегляд приватних повідомлень за аксесуаром для автомобіля між користувачами	+
Управління замовленим придбанням аксесуару для автомобіля продавця	+
Звернення до керуючого щодо замовлення придбання аксесуару для автомобіля	+

Продовження таблиці 4.1

1	2
Перегляд замовлених придбань аксесуару для автомобіля	+
Створення відгуку про покупця аксесуару для автомобіля	+
Перегляд усіх замовлених придбань аксесуарів для автомобіля за продавцем	+
Публікація історії застосування власного аксесуару для автомобіля	+
Відправлення приватного повідомлення щодо звернення до керуючого	+
Перегляд власних звернень до керуючого	+
Редагування даних аксесуару для автомобіля, виставленого на продаж	+
Редагування історії застосування власного аксесуару для автомобіля	+
Збереження аксесуару для автомобіля	+
Перегляд збережених аксесуарів для автомобілів	+
Управління замовленим придбанням аксесуару для автомобіля покупцем	+
Створення відгуку про аксесуар для автомобіля	+
Створення відгуку про продавця аксесуару для автомобіля	+
Перегляд усіх замовлених придбань аксесуарів для автомобіля за покупцем	+
Замовлення придбання аксесуару для автомобіля	+
Вилучення аксесуару для автомобіля зі збережених	+
Управління замовленим придбанням аксесуару для автомобіля	+
Перегляд звернень до керуючого	+
Управління зверненням до керуючого	+

4.5 Висновки за розділом 4

Принципи організації роботи створеної програми пошуку аксесуарів для автомобілів для продавців, покупців, керуючих та неавторизованих користувачів зазначені у розділі, тестування виконання всіх функціональних можливостей забезпечено і проведено з отриманням успішних результатів.

ВИСНОВКИ

В основі визначення концепції дипломної кваліфікаційної роботи знаходиться проведений огляд програм пошуку аксесуарів для автомобілів, який було в результаті порівняння зведено до універсального майданчика продажу товарів OLX, спеціалізованого магазину продажу автоаксесуарів Lavita та спеціалізованого магазину продажу автомобільних товарів Autobaza. Розробка цієї роботи має відмінності порівняно з ними, узагальнюючи окремі переваги цих рішень, але окрім того має і ту відмінність, яка взагалі відсутня у аналогів. Такою відмінністю є можливість перегляду, створення історій використання аксесуарів від власників.

Під час проєктування програми пошуку аксесуарів для автомобілів у підсумку охоплено моделювання сценаріїв покупця аксесуарів для автомобілів, продавця, керуючого та неавторизованого користувача, вибір для розробки системи MySQL, мови Python, ФВ Django, повне визначення структури БД.

Процес взаємодії користувачів різних ролей з програмою пошуку аксесуарів для автомобілів представлено у вигляді функціональної схеми, описано основні представлення створеної програми пошуку аксесуарів для автомобілів з відповідними шаблонами представлень, що сукупно забезпечують підтримку функціональних можливостей пошуку аксесуарів для автомобілів разом з класами моделей.

Принципи організації роботи створеної програми пошуку аксесуарів для автомобілів для продавців, покупців, керуючих та неавторизованих користувачів визначені та описані, тестування виконання всіх функціональних можливостей забезпечено і проведено з отриманням успішних результатів.

Програма може використовуватися як для пошуку інформації про існуючі на ринку аксесуари для автомобілів, практику використання таких аксесуарів у своїх автомобілях власниками, так і для підтримки продажу і придбання аксесуарів для автомобілів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Car Accessories Market Size, Growth Opportunities 2023-2032 [Electronic resource]. – Access mode : <https://www.gminsights.com/industry-analysis/car-accessories-market>.
2. Автоаксесуари для авто купити в Україні на OLX [Електрон. ресурс]. – Режим доступу : <https://www.olx.ua/uk/zapchasti-dlya-transporta/aksessuary-dlya-avto/?currency=UAH>.
3. Купити автоаксесуари за низькими цінами – Доставка по Україні. Lavita.ua [Електрон. ресурс]. – Режим доступу : <https://www.lavita.ua/>.
4. АвтоЗвук – Автомагазин, Автобаза, Автоаксесуари, Автоелектроніка, Автотовари, Київ, Одеса, Україна [Електрон. ресурс]. – Режим доступу : <https://avtozvuk.ua/ua>.
5. Django introduction – Learn web development. MDN [Electronic resource]. – Access mode : https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Server-side/Django/Introduction.
6. Mantosh, M. What Is the Django Web Framework? [Electronic resource] / M. Mantosh. – Access mode : <https://blog.jetbrains.com/pycharm/2023/11/what-is-the-django-web-framework/>.
7. Flask Tutorial. GeeksforGeeks [Electronic resource]. – Access mode : <https://www.geeksforgeeks.org/flask-tutorial/>.
8. Deery, M. What Is Flask and How Do Developers Use It? A Quick Guide [Electronic resource] / M. Deery. – Access mode : <https://careerfoundry.com/en/blog/web-development/what-is-flask/>.
9. Free Template 576 Snapx Photography [Electronic resource]. – Access mode : <https://templatemo.com/tm-576-snapx-photography>.

ДОДАТОК А
Технічне завдання

Вступ

Автомобілі потребують постійного обслуговування, а разом з тим багато автовласників бажають підвищити якість користування власними транспортними засобами, а тому використовують додаткові аксесуари, підвищуючи комфорт. Це створює загалом величезний ринок, для зручного орієнтування в якому потрібне програмне забезпечення, ефективно орієнтоване на користувачів. У той же час продаж аксесуарів для автомобілів може здійснюватися не тільки в новому стані, але і тих, що були у використанні. У такому випадку досвід використання таких аксесуарів для автомобілів може бути корисним для потенційних покупців, а тому необхідний інструмент, який дозволить таким досвідом ділитися у самому програмному забезпеченні. На реалізацію таких можливостей і має бути направлене виконання роботи.

A.1 Підстави для розробки

Основна підстава для виконання роботи сформульована у відповідності з Наказом № 209 від 28 квітня 2025 року, виданим у Національному університеті «Запорізька політехніка», де для виконання даної дипломної кваліфікаційної роботи бакалавра визначено «Програмне забезпечення пошуку аксесуарів для автомобілів» як тему роботи.

A.2 Призначення розробки

Програмне забезпечення пошуку аксесуарів для автомобілів призначене для виставлення власних аксесуарів для автомобілів продавцями на продаж, для пошуку покупцями цих аксесуарів, отримання інформації про аксесуари на основі зокрема історії їх використання власниками.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Вимоги до програми пошуку аксесуарів для автомобілів за набором функцій включають:

- заведення акаунту продавця аксесуарів для автомобілів;
- заведення акаунту покупця аксесуарів для автомобілів;
- авторизація продавця або покупця аксесуарів для автомобілів, керуючого;
- пошук аксесуарів для автомобіля;
- виставлення на продаж аксесуару для автомобіля;
- редагування даних аксесуару для автомобіля, виставленого на продаж;
- перегляд інформації про аксесуар для автомобіля, виставлений на продаж;
- перегляд аксесуарів для автомобілів, виставлених на продаж заданим користувачем;
- збереження аксесуару для автомобіля;
- перегляд збережених аксесуарів для автомобілів;
- публікація історії застосування власного аксесуару для автомобіля;
- редагування історії застосування власного аксесуару для автомобіля;
- перегляд історій застосування заданого аксесуару для автомобіля;
- перегляд сторінки продавця аксесуарів для автомобілів;
- перегляд сторінки покупця аксесуарів для автомобілів;
- перегляд відгуків про покупця аксесуарів для автомобілів;
- перегляд відгуків про продавця аксесуарів для автомобілів;
- замовлення придбання аксесуару для автомобіля;
- відправлення приватного повідомлення щодо аксесуару для автомобіля;
- перегляд приватних повідомлень за аксесуаром для автомобіля між

користувачами;

- перегляд всіх нових приватних повідомлень щодо аксесуарів для автомобілів;

- звернення до керуючого щодо замовлення придбання аксесуару для автомобіля;

- створення відгуку про аксесуар для автомобіля, його продавця та покупця;

- перегляд звернень керуючим;

- перегляд замовлених придбань аксесуару для автомобіля;

- перегляд усіх замовлених придбань аксесуарів для автомобіля за продавцем;

- перегляд усіх замовлених придбань аксесуарів для автомобіля за покупцем;

- управління замовленим придбанням аксесуару для автомобіля;

- відправлення приватного повідомлення щодо звернення до керуючого;

- перегляд власних звернень до керуючого.

A.3.2 Умови експлуатації

Експлуатаційні вимоги до вебзастосунку пошуку аксесуарів для автомобілів включають вимоги до спеціаліста-адміністратора, який має забезпечити хостинг, розгортання вебзастосунку на сервері з застосуванням обраного для реалізації вебфреймворку, підтримку роботи бази даних на основі обраної в роботі системи керування. Відповідно експлуатація такого вебзастосунку пошуку аксесуарів для автомобілів має бути забезпечена можливістю доступу до технологій, на основі яких реалізовано вебзастосунок та які відповідно були обрані в роботі.

А.3.3 Вимоги до складу та параметрів технічних засобів

Сервер на технічному рівні повинен щонайменше мати 2 Гб оперативної пам'яті, два ядра процесора, 100 Гб для організації сховища, що має надати можливість виконання 25000 запитів на місяць до вебзастосунку, але в подальшому кількість запитів та інші варіанти навантажень мають бути оцінені додатково, що може призвести до розширення мінімальних вимог, окрім того відсоток вільного простору для організації сховища та темпи його змінювання мають оцінюватися та призводити до прийняття рішень щодо розширення доступного простору за необхідності.

А.4 Порядок контролю та приймання

Сумарно 8 тижнів виділено на виконання дипломної кваліфікаційної роботи бакалавра, розподіл між якими має виконуватися за календарним планом, визначеним у завданні на роботі, з забезпеченням результатів, які мають бути досягнуті та які зазначені в календарному плані. У підсумку робота оцінюється керівником зі створенням відгуку, рецензентом зі створенням рецензії та екзаменаційною комісією за наперед затвердженим графіком.

ДОДАТОК Б
Текст програми

```

from .models import Caraccessorycategory,
Caraccessorycategoryattribute, Caraccessorycategoryattributeset,
Carmakerbrand, Carmakerbrandmodel, Caraccessoryformodel, Caraccessory,
Caraccessoryset, Caraccessoryphoto, Caraccessoryseller,
Caraccessorybuyer, Caraccessoryapplication,
Caraccessoryapplicationstatus, Caraccessoryapplicationappeal,
Caraccessoryapplicationelement, Caraccessoryprivate,
Caraccessorystory, Caraccessorystoryphoto, Caraccessoryfeedback
from django.shortcuts import render
from django.core.paginator import Paginator

def caraccessory_info_view(request, caraccessory_id):
    caraccessory_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(id = caraccessory_id,
caraccessoryselling = True)
    author_seller = False
    seller_data = None
    if request.user.is_authenticated:
        seller_data =
Caraccessoryseller.objects.select_related("accountfromdjango").filter(
accountfromdjango = request.user)
    if caraccessory_base_info:
        caraccessory_base_info = caraccessory_base_info[0]
    else:
        return render(
            request,
            "cassgun/problem.html",
            {"situation": "Доступ до сторінки аксесуару для
автомобілів за визначеним номером",
            "problem": "Такий аксесуар для автомобілів
мені, на жаль, невідомий. Можливо він більше не доступний.
Скористайтесь пошуком, ми маємо багато пропозицій інших аксесуарів для
автомобілів!"})
    if seller_data and seller_data ==
caraccessory_base_info.caraccessoryseller:
        author_seller = True
    elif not caraccessory_base_info.caraccessoryselling:
        caraccessory_base_info = None
    return render(
        request,
        "cassgun/problem.html",

```

```

        {"situation": "Доступ до сторінки аксесуару для
автомобілів за визначеним номером",
        "problem": "Такий аксесуар для автомобілів
мені, на жаль, невідомий. Можливо він більше не доступний.
Скористайтесь пошуком, ми маємо багато пропозицій інших аксесуарів для
автомобілів!"})

        carmodels_for_this_caraccessory =
Caraccessoryformodel.objects.select_related("carmakerbrandmodel",
"carmakerbrandmodel__carmakerbrand").filter(caraccessory =
caraccessory_base_info)
        carmodels_for_this_caraccessory =
carmodels_for_this_caraccessory.order_by("carmakerbrandmodel",
"carmakerbrandmodel__carmakerbrand", "-carmodelvariantyearstart")
        attributes_for_this_caraccessory =
Caraccessoryset.objects.select_related("caraccessorycategoryattributes
et",
"caraccessorycategoryattributeset__caraccessorycategoryattribute").fil
ter(caraccessory = caraccessory_base_info)
        stories_for_this_caraccessory =
Caraccessorystory.objects.prefetch_related("caraccessorystoryelements"
, "caraccessorystoryphotos").filter(caraccessorystoryelements__in =
[caraccessory_base_info])
        stories_for_this_caraccessory =
stories_for_this_caraccessory.order_by("-accessorystorycreationtime")
        stories_for_this_caraccessory_number =
stories_for_this_caraccessory.count()
        stories_for_this_caraccessory =
stories_for_this_caraccessory[:3]
        photos_for_this_caraccessory =
Caraccessoryphoto.objects.filter(caraccessory =
caraccessory_base_info)
        feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info)
        feedbacks_for_this_caraccessory =
feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
        feedbacks_for_this_caraccessory_number =
feedbacks_for_this_caraccessory.count()
        author_buyer = False
        caraccessory_base_info.listed_by_buyer = False

```

```

        if request.user.is_authenticated:
            buyer_data =
Caraccessorybuyer.objects.select_related("accountfromdjango").prefetch
_related("caraccessorylist").filter(accountfromdjango = request.user)
            if buyer_data:
                author_buyer = True
            if author_buyer:
                if caraccessory_base_info in
buyer_data.caraccessorylist:
                    caraccessory_base_info.listed_by_buyer = True
                personal_feedbacks_for_this_caraccessory = None
                personal_feedbacks_for_this_caraccessory_number = 0
                personal_applications_number_for_this_caraccessory = 0
                personal_privates_for_this_caraccessory = None
                personal_privates_for_this_caraccessory_number = 0
                if author_buyer:
                    personal_feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info,
caraccessoryapplicationelement__caraccessoryapplication__caraccessoryb
uyer = buyer_data)
                    personal_feedbacks_for_this_caraccessory =
personal_feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
                    personal_feedbacks_for_this_caraccessory_number =
personal_feedbacks_for_this_caraccessory.count()
                    personal_applications_number_for_this_caraccessory =
Caraccessoryapplicationelement.objects.filter(caraccessory =
caraccessory_base_info, caraccessoryapplication__caraccessorybuyer =
buyer_data)
                    personal_applications_number_for_this_caraccessory =
personal_applications_number_for_this_caraccessory.count()
                    personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.filter(caraccessorybuyer = buyer_data,
caraccessory = caraccessory_base_info)
                    personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.order_by("-privatemesssent")
                    personal_privates_for_this_caraccessory_number =
personal_privates_for_this_caraccessory.count()
                    personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory[0]

```

```

        return render(
            request,
            "cassgun/caraccessory.html",
            {"caraccessory": caraccessory_base_info,
            "privates":
personal_privates_for_this_caraccessory,
            "personalfeedbacks":
personal_feedbacks_for_this_caraccessory_number,
            "applicationnumber":
personal_applications_number_for_this_caraccessory,
            "feedbacknumber":
feedbacks_for_this_caraccessory_number,
            "feedbacks": feedbacks_for_this_caraccessory,
            "photos": photos_for_this_caraccessory,
            "stories": stories_for_this_caraccessory,
            "storiesnumber":
stories_for_this_caraccessory_number,
            "attributes": attributes_for_this_caraccessory,
            "carmodels": carmodels_for_this_caraccessory,
            "privatesnumber":
personal_privates_for_this_caraccessory_number })
        if author_seller:
            personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.select_related("caraccessorybuyer",
"caraccessorybuyer__accountfromdjango ").filter(caraccessory =
caraccessory_base_info)
            personal_privates_for_this_caraccessory_new =
personal_privates_for_this_caraccessory.filter(seenbyreceiver = False)
            personal_privates_for_this_caraccessory_new =
personal_feedbacks_for_this_caraccessory_new.count()
            personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.values("caraccessorybuyer").di
stinct()

        return render(
            request,
            "cassgun/caraccessory.html",
            {"caraccessory": caraccessory_base_info,
            "privates":
personal_privates_for_this_caraccessory,
            "feedbacknumber":
feedbacks_for_this_caraccessory_number,
            "feedbacks": feedbacks_for_this_caraccessory,
            "photos": photos_for_this_caraccessory,

```

```

        "stories": stories_for_this_caraccessory,
        "storiesnumber":
stories_for_this_caraccessory_number,
        "attributes": attributes_for_this_caraccessory,
        "carmodels": carmodels_for_this_caraccessory,
        "privatesnumber":
personal_privates_for_this_caraccessory_new })

```

```

def caraccessory_stories_view(request, caraccessory_id):
    caraccessory_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(id = caraccessory_id,
caraccessoryselling = True)
    author_seller = False
    seller_data = None
    if request.user.is_authenticated:
        seller_data =
Caraccessoryseller.objects.select_related("accountfromdjango").filter(
accountfromdjango = request.user)
        caraccessory_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(id = caraccessory_id)
        if caraccessory_base_info:
            caraccessory_base_info = caraccessory_base_info[0]
        else:
            return render(
                request,
                "cassgun/problem.html",
                {"situation": "Доступ до сторінки історій
застосування аксесуару для автомобілів за визначеним номером",
                "problem": "Такий аксесуар для автомобілів
мені, на жаль, невідомий. Можливо він більше не доступний.
Скористайтесь пошуком, ми маємо багато пропозицій інших аксесуарів для
автомобілів!"})
    if seller_data and seller_data ==
caraccessory_base_info.caraccessoryseller:
        author_seller = True
    elif not caraccessory_base_info.caraccessoryselling:
        caraccessory_base_info = None
    return render(
        request,

```

```

        "cassgun/problem.html",
        {"situation": "Доступ до сторінки аксесуару для
автомобілів за визначеним номером",
        "problem": "Такий аксесуар для автомобілів
мені, на жаль, невідомий. Можливо він більше не доступний.
Скористайтесь пошуком, ми маємо багато пропозицій інших аксесуарів для
автомобілів!"})
        stories_for_this_caraccessory =
Caraccessorystory.objects.prefetch_related("caraccessorystoryelements"
, "caraccessorystoryphotos").filter(caraccessorystoryelements__in =
[caraccessory_base_info])
        stories_for_this_caraccessory =
stories_for_this_caraccessory.order_by("-accessorystorycreationtime")
        stories_for_this_caraccessory_number =
stories_for_this_caraccessory.count()
        photos_for_this_caraccessory =
Caraccessoryphoto.objects.filter(caraccessory =
caraccessory_base_info)
        photos_for_this_caraccessory =
photos_for_this_caraccessory[0]
        feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info)
        feedbacks_for_this_caraccessory =
feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
        feedbacks_for_this_caraccessory_number =
feedbacks_for_this_caraccessory.count()
        author_buyer = False
        caraccessory_base_info.listed_by_buyer = False
        if request.user.is_authenticated:
            buyer_data =
Caraccessorybuyer.objects.select_related("accountfromdjango").prefetch
_related("caraccessorylist").filter(accountfromdjango = request.user)
            if buyer_data:
                author_buyer = True
            if author_buyer:
                if caraccessory_base_info in
buyer_data.caraccessorylist:
                    caraccessory_base_info.listed_by_buyer = True
                    personal_feedbacks_for_this_caraccessory_number = 0

```

```

        personal_applications_number_for_this_caraccessory = 0
        personal_privates_for_this_caraccessory_number = 0
        stories_for_this_caraccessory_paginator =
Paginator(stories_for_this_caraccessory, 10)
        stories_for_this_caraccessory_portion =
request.GET.get('portion')
        stories_for_this_caraccessory =
stories_for_this_caraccessory_paginator.get_page(stories_for_this_caraccessory_portion)
        if author_buyer:
            personal_feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationelement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessorybuyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info,
caraccessoryapplicationelement__caraccessoryapplication__caraccessorybuyer = buyer_data)
            personal_feedbacks_for_this_caraccessory =
personal_feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
            personal_feedbacks_for_this_caraccessory_number =
personal_feedbacks_for_this_caraccessory.count()
            personal_applications_number_for_this_caraccessory =
Caraccessoryapplicationelement.objects.filter(caraccessory =
caraccessory_base_info, caraccessoryapplication__caraccessorybuyer =
buyer_data)
            personal_applications_number_for_this_caraccessory =
personal_applications_number_for_this_caraccessory.count()
            personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.filter(caraccessorybuyer = buyer_data,
caraccessory = caraccessory_base_info)
            personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.order_by("-privatemesssent")
            personal_privates_for_this_caraccessory_number =
personal_privates_for_this_caraccessory.count()
            return render(
                request,
                "cassgun/caraccessorystories.html",
                {"caraccessory": caraccessory_base_info,
                "personalfeedbacks":
personal_feedbacks_for_this_caraccessory_number,
                "applicationnumber":
personal_applications_number_for_this_caraccessory,

```

```

        "feedbacknumber":
feedbacks_for_this_caraccessory_number,
        "photos": photos_for_this_caraccessory,
        "stories": stories_for_this_caraccessory,
        "storiesnumber":
stories_for_this_caraccessory_number,
        "privatesnumber":
personal_privates_for_this_caraccessory_number })
    if author_seller:
        personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.select_related("caraccessorybuyer",
"caraccessorybuyer__accountfromdjango ").filter(caraccessory =
caraccessory_base_info)
        personal_privates_for_this_caraccessory_new =
personal_privates_for_this_caraccessory.filter(seenbyreceiver = False)
        personal_privates_for_this_caraccessory_new =
personal_feedbacks_for_this_caraccessory_new.count()
        personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.values("caraccessorybuyer").di
stinct()

    return render(
        request,
        "cassgun/caraccessorystories.html",
        {"caraccessory": caraccessory_base_info,
        "privates":
personal_privates_for_this_caraccessory,
        "feedbacknumber":
feedbacks_for_this_caraccessory_number,
        "photos": photos_for_this_caraccessory,
        "stories": stories_for_this_caraccessory,
        "storiesnumber":
stories_for_this_caraccessory_number,
        "privatesnumber":
personal_privates_for_this_caraccessory_new })

def caraccessory_privates_view(request, caraccessory_id,
caraccessorybuyer = None):
    caraccessory_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(id = caraccessory_id,
caraccessoryselling = True)
    author_seller = False

```

```

        seller_data = None
        if request.user.is_authenticated:
            seller_data =
Caraccessoryseller.objects.select_related("accountfromdjango").filter(
accountfromdjango = request.user)
            caraccessory_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(id = caraccessory_id)
            if caraccessory_base_info:
                caraccessory_base_info = caraccessory_base_info[0]
            else:
                return render(
                    request,
                    "cassgun/problem.html",
                    {"situation": "Доступ до сторінки приватних
повідомлень за аксесуаром для автомобілів за визначеним номером",
                    "problem": "Такий аксесуар для автомобілів
мені, на жаль, невідомий. Можливо він більше не доступний.
Скористайтесь пошуком, ми маємо багато пропозицій інших аксесуарів для
автомобілів!"})
                if seller_data and seller_data ==
caraccessory_base_info.caraccessoryseller:
                    author_seller = True
                elif not caraccessory_base_info.caraccessoryselling:
                    caraccessory_base_info = None
                return render(
                    request,
                    "cassgun/problem.html",
                    {"situation": "Доступ до сторінки аксесуару для
автомобілів за визначеним номером",
                    "problem": "Такий аксесуар для автомобілів
мені, на жаль, невідомий. Можливо він більше не доступний.
Скористайтесь пошуком, ми маємо багато пропозицій інших аксесуарів для
автомобілів!"})
                stories_for_this_caraccessory =
Caraccessorystory.objects.prefetch_related("caraccessorystoryelements"
, "caraccessorystoryphotos").filter(caraccessorystoryelements__in =
[caraccessory_base_info])
                stories_for_this_caraccessory =
stories_for_this_caraccessory.order_by("-accessorystorycreationtime")
                stories_for_this_caraccessory_number =
stories_for_this_caraccessory.count()

```

```

        photos_for_this_caraccessory =
Caraccessoryphoto.objects.filter(caraccessory =
caraccessory_base_info)
        photos_for_this_caraccessory =
photos_for_this_caraccessory[0]
        feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info)
        feedbacks_for_this_caraccessory =
feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
        feedbacks_for_this_caraccessory_number =
feedbacks_for_this_caraccessory.count()
        author_buyer = False
        caraccessory_base_info.listed_by_buyer = False
        if request.user.is_authenticated:
            buyer_data =
Caraccessorybuyer.objects.select_related("accountfromdjango").prefetch
_related("caraccessorylist").filter(accountfromdjango = request.user)
            if buyer_data:
                author_buyer = True
            if author_buyer:
                if caraccessory_base_info in
buyer_data.caraccessorylist:
                    caraccessory_base_info.listed_by_buyer = True
                personal_feedbacks_for_this_caraccessory_number = 0
                personal_applications_number_for_this_caraccessory = 0
                personal_privates_for_this_caraccessory_number = 0
                if author_buyer:
                    personal_feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info,
caraccessoryapplicationelement__caraccessoryapplication__caraccessoryb
uyer = buyer_data)
                    personal_feedbacks_for_this_caraccessory =
personal_feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
                    personal_feedbacks_for_this_caraccessory_number =
personal_feedbacks_for_this_caraccessory.count()

```

```

        personal_applications_number_for_this_caraccessory =
Caraccessoryapplicationelement.objects.filter(caraccessory =
caraccessory_base_info, caraccessoryapplication__caraccessorybuyer =
buyer_data)

        personal_applications_number_for_this_caraccessory =
personal_applications_number_for_this_caraccessory.count()
        personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.filter(caraccessorybuyer = buyer_data,
caraccessory = caraccessory_base_info)
        personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.order_by("-privatemesssent")
        personal_privates_for_this_caraccessory_number =
personal_privates_for_this_caraccessory.count()
        personal_privates_for_this_caraccessory_paginator =
Paginator(personal_privates_for_this_caraccessory, 100)
        personal_privates_for_this_caraccessory_portion =
request.GET.get('portion')
        personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory_paginator.get_page(personal_pr
ivates_for_this_caraccessory_portion)
        if auth_buyer:
            return render(
                request,
                "cassgun/caraccessoryprivates.html",
                {"caraccessory": caraccessory_base_info,
                "privates":
personal_privates_for_this_caraccessory,
                "personalfeedbacks":
personal_feedbacks_for_this_caraccessory_number,
                "applicationnumber":
personal_applications_number_for_this_caraccessory,
                "feedbacknumber":
feedbacks_for_this_caraccessory_number,
                "photos": photos_for_this_caraccessory,
                "storiesnumber":
stories_for_this_caraccessory_number,
                "privatesnumber":
personal_privates_for_this_caraccessory_number })
        if author_seller and not caraccessorybuyer:
            personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.select_related("caraccessorybuyer",
"caraccessorybuyer__accountfromdjango ").filter(caraccessory =
caraccessory_base_info)

```

```

        personal_privates_for_this_caraccessory_new =
personal_privates_for_this_caraccessory.filter(seenbyreceiver = False)
        personal_privates_for_this_caraccessory_new =
personal_feedbacks_for_this_caraccessory_new.count()
        personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.values("caraccessorybuyer").di
stinct()

        return render(
            request,
            "cassgun/caraccessoryprivates.html",
            {"caraccessory": caraccessory_base_info,
            "privates":
personal_privates_for_this_caraccessory,
            "feedbacknumber":
feedbacks_for_this_caraccessory_number,
            "photos": photos_for_this_caraccessory,
            "stories": stories_for_this_caraccessory,
            "storiesnumber":
stories_for_this_caraccessory_number,
            "privatesnumber":
personal_privates_for_this_caraccessory_new })
        elif author_seller:
            buyer_data =
Caraccessorybuyer.objects.select_related("accountfromdjango").prefetch
_related("caraccessorylist").filter(accountfromdjango__username =
caraccessorybuyer)
            if buyer_data:
                buyer_data = buyer_data[0]
                personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.filter(caraccessory =
caraccessory_base_info, caraccessorybuyer = buyer_data)
                personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.order_by("-privatemesssent")
                personal_privates_for_this_caraccessory_new =
personal_privates_for_this_caraccessory.filter(seenbyreceiver = False)
                personal_privates_for_this_caraccessory_new =
personal_feedbacks_for_this_caraccessory_new.count()
            return render(
                request,
                "cassgun/caraccessoryprivates.html",
                {"caraccessory": caraccessory_base_info,
                "privates":
personal_privates_for_this_caraccessory,

```

```

        "feedbacknumber":
feedbacks_for_this_caraccessory_number,
        "photos": photos_for_this_caraccessory,
        "stories": stories_for_this_caraccessory,
        "storiesnumber":
stories_for_this_caraccessory_number,
        "privatesnumber":
personal_privates_for_this_caraccessory_new })

def caraccessory_applications_view(request, caraccessory_id,
status_to_display = -1):
    caraccessory_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(id = caraccessory_id,
caraccessoryselling = True)
    author_seller = False
    seller_data = None
    if request.user.is_authenticated:
        seller_data =
Caraccessoryseller.objects.select_related("accountfromdjango").filter(
accountfromdjango = request.user)
    if caraccessory_base_info:
        caraccessory_base_info = caraccessory_base_info[0]
    else:
        return render(
            request,
            "cassgun/problem.html",
            {"situation": "Доступ до сторінки замовлень за
аксесуаром для автомобілів за визначеним номером",
            "problem": "Такий аксесуар для автомобілів
мені, на жаль, невідомий. Можливо він більше не доступний.
Скористайтесь пошуком, ми маємо багато пропозицій інших аксесуарів для
автомобілів!"})
    if seller_data and seller_data ==
caraccessory_base_info.caraccessoryseller:
        author_seller = True
    else:
        return render(
            request,
            "cassgun/problem.html",
            {"situation": "Доступ до сторінки замовлень
аксесуару для автомобілів за визначеним номером",

```

```

        "problem": "Дані за таким аксесуаром для
автомобілів недоступні для вас. Дякуємо за розуміння!")})
        stories_for_this_caraccessory =
Caraccessorystory.objects.prefetch_related("caraccessorystoryelements"
, "caraccessorystoryphotos").filter(caraccessorystoryelements__in =
[caraccessory_base_info])
        stories_for_this_caraccessory =
stories_for_this_caraccessory.order_by("-accessorystorycreationtime")
        stories_for_this_caraccessory_number =
stories_for_this_caraccessory.count()
        stories_for_this_caraccessory =
stories_for_this_caraccessory[:3]
        photos_for_this_caraccessory =
Caraccessoryphoto.objects.filter(caraccessory =
caraccessory_base_info)
        photos_for_this_caraccessory =
photos_for_this_caraccessory[0]
        feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info)
        feedbacks_for_this_caraccessory =
feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
        feedbacks_for_this_caraccessory_number =
feedbacks_for_this_caraccessory.count()
        personal_applications_number_for_this_caraccessory = 0
        personal_privates_for_this_caraccessory = None
        personal_privates_for_this_caraccessory_number = 0
        personal_applications_for_this_caraccessory =
Caraccessoryapplicationelement.objects.select_related("caraccessoryapp
lication", "caraccessoryapplication__caraccessorybuyer",
"caraccessoryapplication__caraccessorybuyer__accountfromdjango").filte
r(caraccessory = caraccessory_base_info)
        personal_applications_number_for_this_caraccessory =
personal_applications_for_this_caraccessory.count()
        personal_applications_for_this_caraccessory =
personal_applications_for_this_caraccessory.order_by("-
caraccessoryapplication__id")
        personal_applications_for_this_caraccessory_paginator =
Paginator(personal_applications_for_this_caraccessory, 100)

```

```

        personal_applications_for_this_caraccessory_portion =
request.GET.get('portion')
        personal_applications_for_this_caraccessory =
personal_applications_for_this_caraccessory_paginator.get_page(persona
l_applications_for_this_caraccessory_portion)
        personal_applications_for_this_caraccessory_on_status = []
        for personal_application_for_this_caraccessory in
personal_applications_for_this_caraccessory:
            personal_applications_for_this_caraccessory_statuses =
Caraccessoryapplicationstatus.objects.filter(caraccessoryapplication =
personal_application_for_this_caraccessory)
            personal_applications_for_this_caraccessory_statuses =
personal_applications_for_this_caraccessory_statuses.order_by("-
caraccessoryapplicationtime")
            personal_application_for_this_caraccessory_statuses =
personal_applications_for_this_caraccessory_statuses
            if status_to_display >= 0 and
personal_applications_for_this_caraccessory_statuses[0].caraccessoryap
plicationstatus == status_to_display:

personal_applications_for_this_caraccessory_on_statusa.append(personal
_application_for_this_caraccessory)
            elif status_to_display == -1:

personal_applications_for_this_caraccessory_on_statusa.append(personal
_application_for_this_caraccessory)
            personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.select_related("caraccessorybuyer",
"caraccessorybuyer__accountfromdjango ").filter(caraccessory =
caraccessory_base_info)
            personal_privates_for_this_caraccessory_new =
personal_privates_for_this_caraccessory.filter(seenbyreceiver = False)
            personal_privates_for_this_caraccessory_new =
personal_feedbacks_for_this_caraccessory_new.count()
            personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.values("caraccessorybuyer").di
stinct()

        return render(
            request,
            "cassgun/caraccessoryapplications.html",
            {"caraccessory": caraccessory_base_info,
            "privates":
personal_privates_for_this_caraccessory,

```

```

        "feedbacknumber":
feedbacks_for_this_caraccessory_number,
        "photos": photos_for_this_caraccessory,
        "storiesnumber":
stories_for_this_caraccessory_number,
        "privatesnumber":
personal_privates_for_this_caraccessory_new,
        "applications":
personal_applications_for_this_caraccessory_on_status,
        "applicationsnumber":
personal_applications_number_for_this_caraccessory })

def caraccessories_list_view(request):
    if request.user.is_authenticated:
        buyer_data =
Caraccessorybuyer.objects.select_related("accountfromdjango").prefetch
_related("caraccessorylist", "caraccessorylist__caraccessorycategory",
"caraccessorylist__carmakerbrand",
"caraccessorylist__caraccessoryseller",
"caraccessorylist__caraccessoryseller__accountfromdjango").filter(acco
untfromdjango = request.user)
        if buyer_data:
            buyer_data = buyer_data[0]
        else:
            return render(
                request,
                "cassgun/problem.html",
                {"situation": "Доступ до сторінки збережених
аксесуару для автомобілів",
                "problem": "Авторизація виконана некоректно."})
    else:
        return render(
            request,
            "cassgun/problem.html",
            {"situation": "Доступ до сторінки збережених
аксесуару для автомобілів",
            "problem": "Спочатку потрібно авторизуватись, а
тоді звертатись до цієї сторінки."})
        caraccessorylist_paginator =
Paginator(buyer_data.caraccessorylist, 10)
        caraccessorylist_portion = request.GET.get('portion')
        caraccessorylist =
caraccessorylist_paginator.get_page(caraccessorylist_portion)

```

```

        for caraccessory_base_info in caraccessorylist:
            stories_for_this_caraccessory =
Caraccessorystory.objects.prefetch_related("caraccessorystoryelements"
, "caraccessorystoryphotos").filter(caraccessorystoryelements__in =
[caraccessory_base_info])
            stories_for_this_caraccessory =
stories_for_this_caraccessory.order_by("-accessorystorycreationtime")
            stories_for_this_caraccessory_number =
stories_for_this_caraccessory.count()
            caraccessory_base_info.stories_number =
stories_for_this_caraccessory_number
            photos_for_this_caraccessory =
Caraccessoryphoto.objects.filter(caraccessory =
caraccessory_base_info)
            photos_for_this_caraccessory =
photos_for_this_caraccessory[0]
            caraccessory_base_info.photo =
photos_for_this_caraccessory
            feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info)
            feedbacks_for_this_caraccessory =
feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
            feedbacks_for_this_caraccessory_number =
feedbacks_for_this_caraccessory.count()
            caraccessory_base_info.feedbacks_number =
feedbacks_for_this_caraccessory_number
            personal_feedbacks_for_this_caraccessory_number = 0
            personal_applications_number_for_this_caraccessory = 0
            personal_privates_for_this_caraccessory_number = 0
            personal_feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info,
caraccessoryapplicationelement__caraccessoryapplication__caraccessoryb
uyer = buyer_data)
            personal_feedbacks_for_this_caraccessory =
personal_feedbacks_for_this_caraccessory.order_by("-feedbacksettime")

```

```

        personal_feedbacks_for_this_caraccessory_number =
personal_feedbacks_for_this_caraccessory.count()
        personal_applications_number_for_this_caraccessory =
Caraccessoryapplicationelement.objects.filter(caraccessory =
caraccessory_base_info, caraccessoryapplication__caraccessorybuyer =
buyer_data)
        personal_applications_number_for_this_caraccessory =
personal_applications_number_for_this_caraccessory.count()
        personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.filter(caraccessorybuyer = buyer_data,
caraccessory = caraccessory_base_info)
        personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.order_by("-privatemesssent")
        personal_privates_for_this_caraccessory_number =
personal_privates_for_this_caraccessory.count()
        caraccessory_base_info.personal_feedbacks =
personal_feedbacks_for_this_caraccessory_number
        caraccessory_base_info.personal_applications =
personal_applications_number_for_this_caraccessory
        caraccessory_base_info.peronal_privates =
personal_privates_for_this_caraccessory_number
        return render(
            request,
            "cassgun/caraccessorieslist.html",
            {"caraccessorylist": caraccessorylist })

```

```

def seller_caraccessories_list_view(request,
caraccessories_seller):
    seller_data =
Caraccessoryseller.objects.select_related("accountfromdjango").filter(
accountfromdjango__username = caraccessories_seller)
    buyer_data = None
    seller_data_auth = None
    if request.user.is_authenticated:
        seller_data_auth =
Caraccessoryseller.objects.select_related("accountfromdjango").filter(
accountfromdjango = request.user)
        buyer_data =
Caraccessorybuyer.objects.select_related("accountfromdjango").prefetch
_related("caraccessorylist", "caraccessorylist__caraccessorycategory",
"caraccessorylist__carmakerbrand",
"caraccessorylist__caraccessoryseller",

```

```

"caraccessorylist__caraccessoryseller__accountfromdjango").filter(accountfromdjango = request.user)
    if buyer_data:
        buyer_data = buyer_data[0]
    else:
        return render(
            request,
            "cassgun/problem.html",
            {"situation": "Доступ до сторінки аксесуарсд для автомобілів продавця",
            "problem": "Авторизація виконана некоректно."})
    if seller_data_auth and seller_data_auth == seller_data:
        seller_data_auth = seller_data_auth[0]
        caraccessories_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(caraccessoryseller =
seller_data_auth)
    else:
        caraccessories_base_info =
Caraccessory.objects.select_related("caraccessorycategory",
"carmakerbrand", "caraccessoryseller",
"caraccessoryseller__accountfromdjango").filter(caraccessoryseller =
seller_data, caraccessoryselling = True)
        caraccessorylist_pagination =
Paginator(caraccessories_base_info, 10)
        caraccessorylist_portion = request.GET.get('portion')
        caraccessorylist =
caraccessorylist_pagination.get_page(caraccessorylist_portion)
        for caraccessory_base_info in caraccessorylist:
            stories_for_this_caraccessory =
Caraccessorystory.objects.prefetch_related("caraccessorystoryelements"
, "caraccessorystoryphotos").filter(caraccessorystoryelements__in =
[caraccessory_base_info])
            stories_for_this_caraccessory =
stories_for_this_caraccessory.order_by("-accessorystorycreationtime")
            stories_for_this_caraccessory_number =
stories_for_this_caraccessory.count()
            caraccessory_base_info.stories_number =
stories_for_this_caraccessory_number
            photos_for_this_caraccessory =
Caraccessoryphoto.objects.filter(caraccessory =
caraccessory_base_info)

```

```

        photos_for_this_caraccessory =
photos_for_this_caraccessory[0]
        caraccessory_base_info.photo =
photos_for_this_caraccessory
        feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info)
        feedbacks_for_this_caraccessory =
feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
        feedbacks_for_this_caraccessory_number =
feedbacks_for_this_caraccessory.count()
        caraccessory_base_info.feedbacks_number =
feedbacks_for_this_caraccessory_number
        personal_feedbacks_for_this_caraccessory_number = 0
        personal_applications_number_for_this_caraccessory = 0
        personal_privates_for_this_caraccessory_number = 0
        if buyer_data:
            personal_feedbacks_for_this_caraccessory =
Caraccessoryfeedback.objects.select_related("caraccessoryapplicationel
ement", "caraccessoryapplicationelement__caraccessoryapplication",
"caraccessoryapplicationelement__caraccessoryapplication__caraccessory
buyer").filter(feedbacksidetype = 0,
caraccessoryapplicationelement__caraccessory = caraccessory_base_info,
caraccessoryapplicationelement__caraccessoryapplication__caraccessoryb
uyer = buyer_data)
            personal_feedbacks_for_this_caraccessory =
personal_feedbacks_for_this_caraccessory.order_by("-feedbacksettime")
            personal_feedbacks_for_this_caraccessory_number =
personal_feedbacks_for_this_caraccessory.count()
            personal_applications_number_for_this_caraccessory =
Caraccessoryapplicationelement.objects.filter(caraccessory =
caraccessory_base_info, caraccessoryapplication__caraccessorybuyer =
buyer_data)
            personal_applications_number_for_this_caraccessory =
personal_applications_number_for_this_caraccessory.count()
            personal_privates_for_this_caraccessory =
Caraccessoryprivate.objects.filter(caraccessorybuyer = buyer_data,
caraccessory = caraccessory_base_info)
            personal_privates_for_this_caraccessory =
personal_privates_for_this_caraccessory.order_by("-privatemesssent")

```

```

        personal_privates_for_this_caraccessory_number =
personal_privates_for_this_caraccessory.count()
        caraccessory_base_info.personal_feedbacks =
personal_feedbacks_for_this_caraccessory_number
        caraccessory_base_info.personal_applications =
personal_applications_number_for_this_caraccessory
        caraccessory_base_info.peronal_privates =
personal_privates_for_this_caraccessory_number
    else:
        caraccessory_base_info.personal_feedbacks = 0
        caraccessory_base_info.personal_applications = 0
        caraccessory_base_info.peronal_privates = 0
    return render(
        request,
        "cassgun/sellercaraccessories.html",
        {"caraccessorylist": caraccessorylist })

```

ДОДАТОК В
Слайди презентації



Рисунок В.1 – Початковий слайд

ОБ'ЄКТ, ПРЕДМЕТ, МЕТА, ЗАВДАННЯ РОБОТИ

Об'єкт роботи – процес пошуку аксесуарів для автомобілів.

Предмет роботи – програмне забезпечення пошуку аксесуарів для автомобілів.

Мета роботи – розробити програму пошуку аксесуарів для автомобілів для підтримки продажу автомобільних аксесуарів з наданням широкої інформації власниками.

Завдання роботи:

- аналіз проблеми розробки програми пошуку аксесуарів для автомобілів;
- проєктування програми пошуку аксесуарів для автомобілів;
- розробка програми пошуку аксесуарів для автомобілів.



Рисунок В.2 – Об'єкт, предмет, мета, завдання роботи

ПОРІВНЯННЯ ПРОГРАМ ПОШУКУ АКСЕСУАРІВ ДЛЯ АВТОМОБІЛІВ

Характеристика	OLX	Lavita	Autobaza	Cassgun
Джерело продажу аксесуарів для автомобілів	Власники	Сервіс	Сервіс та інші продавці	Власники
Наявні товари	Не тільки для автомобілів	Аксесуари для автомобілів	Усі товари для автомобілів	Аксесуари для автомобілів
Історії використання аксесуарів від власників	-	-	-	+
Пошук за моделями автомобілів	-	-	+	+
Система відгуків	Після покупок	Будь-який користувач	Зареєстрований користувач	Після покупок

3

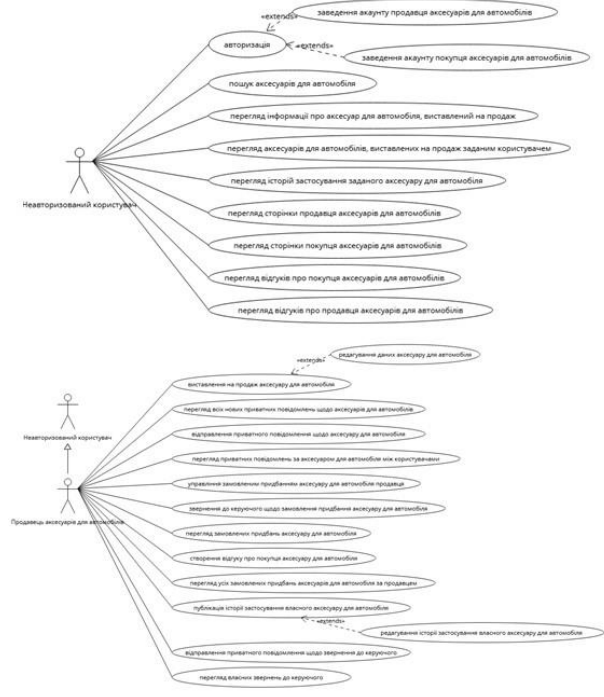
Рисунок В.3 – Порівняння програм пошуку аксесуарів для автомобілів

ПОРІВНЯННЯ ЗАСОБІВ РОЗРОБКИ ДЛЯ ПРОГРАМИ ПОШУКУ АКСЕСУАРІВ ДЛЯ АВТОМОБІЛІВ

Критерій	Flask	Django
Тип фреймворку	Серверний	Повний
Простота реалізації функціональних можливостей	Нижча	Вища, оскільки багато засобів інтегровано до складу, існує велика кількість засобів розширення
Захищеність	Велика кількість перевірок або засобів має бути додана для використання на практиці	Велика кількість інтегрованих засобів захисту

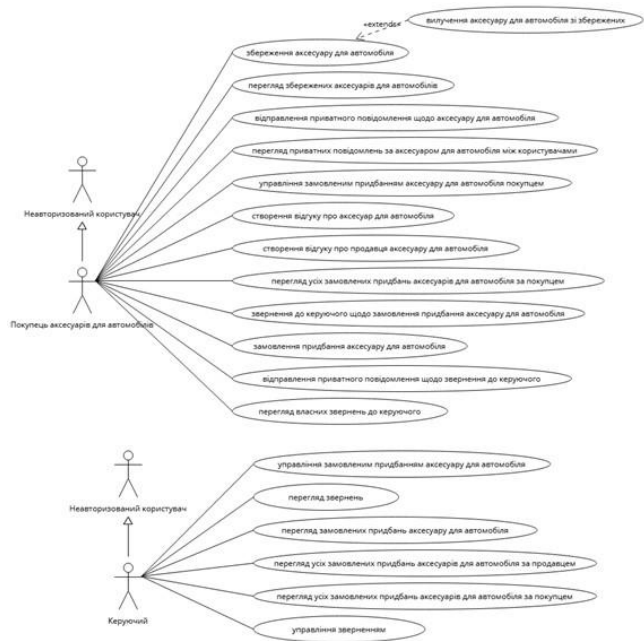
4

Рисунок В.4 – Порівняння засобів розробки для програми пошуку аксесуарів для автомобілів



**ДІАГРАМИ
ПРЕЦЕДЕНТІВ
ДЛЯ ПРОГРАМИ
ПОШУКУ
АКСЕСУАРІВ ДЛЯ
АВТОМОБІЛІВ
ДЛЯ НЕАВТОРИ-
ЗОВАНОГО
КОРИСТУВАЧА ТА
ПРОДАВЦЯ**

Рисунок В.5 – Діаграми прецедентів для програми пошуку аксесуарів для автомобілів для неавторизованого користувача та продавця



**ДІАГРАМИ
ПРЕЦЕДЕНТІВ
ДЛЯ ПРОГРАМИ
ПОШУКУ
АКСЕСУАРІВ ДЛЯ
АВТОМОБІЛІВ
ДЛЯ ПОКУПЦЯ ТА
КЕРУЮЧОГО**

Рисунок В.6 – Діаграми прецедентів для програми пошуку аксесуарів для автомобілів для покупця та керуючого

Рисунок В.7 – Схема бази даних програми пошуку аксесуарів для автомобілів

ФУНКЦІОНАЛЬНА СХЕМА ПРОГРАМИ ПОШУКУ АКСЕСУАРІВ ДЛЯ АВТОМОБІЛІВ

Рисунок В.8 – Функціональна схема програми пошуку аксесуарів для автомобілів

ІНТЕРФЕЙСИ ПРОГРАМИ ПОШУКУ АКСЕСУАРІВ ДЛЯ АВТОМОБІЛІВ

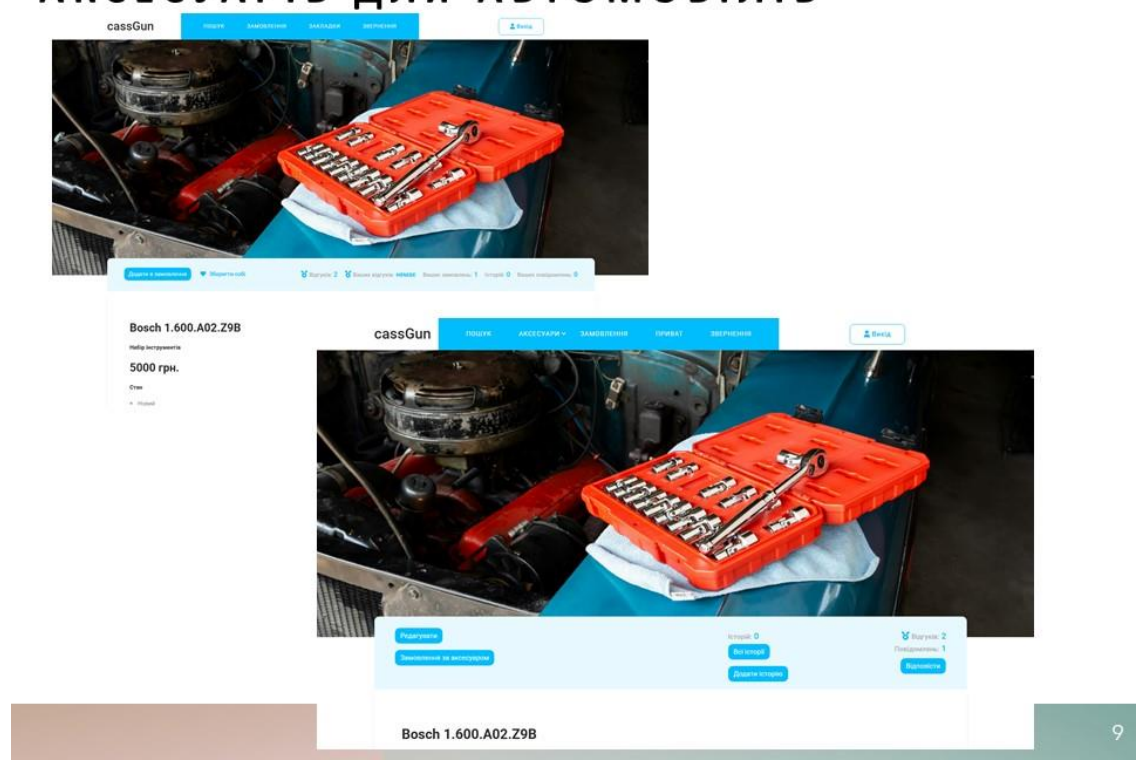


Рисунок В.9 – Інтерфейси програми пошуку аксесуарів для автомобілів

ІНТЕРФЕЙСИ ПРОГРАМИ ПОШУКУ АКСЕСУАРІВ ДЛЯ АВТОМОБІЛІВ

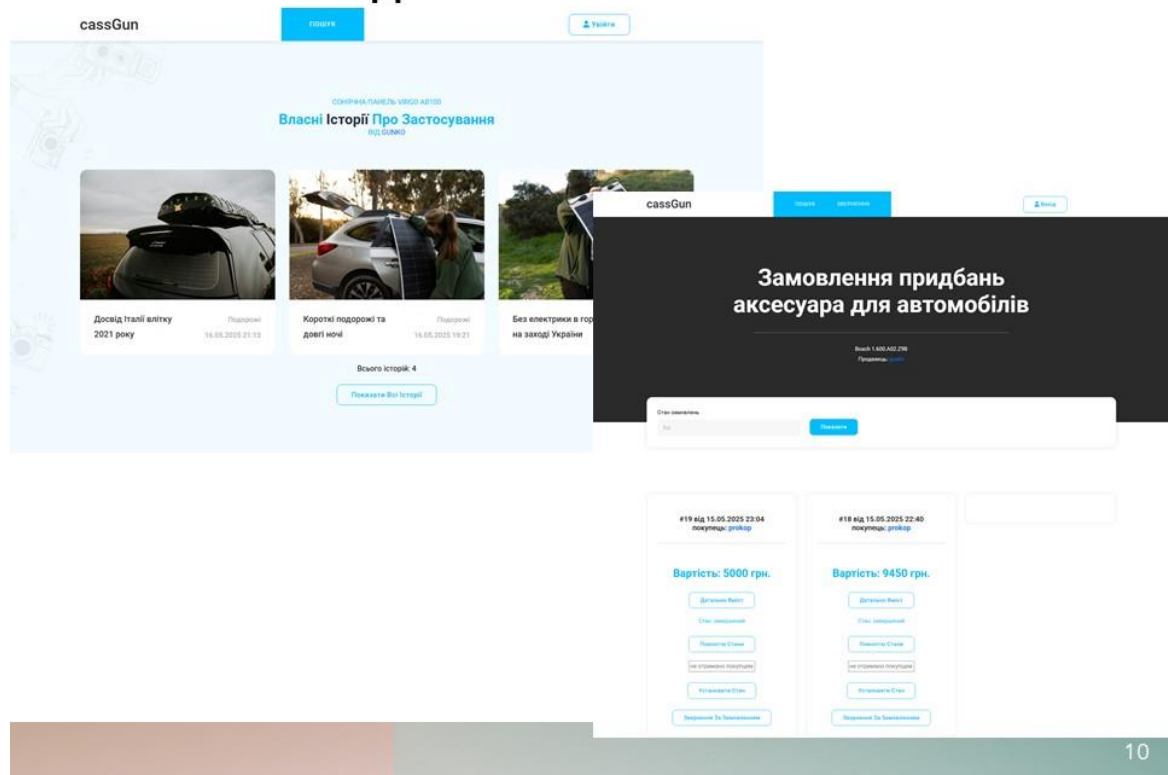


Рисунок В.10 – Інтерфейси програми пошуку аксесуарів для автомобілів

РЕЗУЛЬТАТИ ТЕСТУВАННЯ ПРОГРАМИ ПОШУКУ АКСЕСУАРІВ ДЛЯ АВТОМОБІЛІВ

Функція	Результат	Функція	Результат
Пошук аксесуарів для автомобіля	+	Перегляд замовлених придбань аксесуару для автомобіля	+
Авторизація	+	Створення відгуку про покупця аксесуару для автомобіля	+
Перегляд інформації про аксесуар для автомобіля, виставлений на продаж	+	Перегляд усіх замовлених придбань аксесуарів для автомобіля за продавцем	+
Перегляд аксесуарів для автомобілів, виставлених на продаж заданим користувачем	+	Публікація історії застосування власного аксесуару для автомобіля	+
Перегляд історій застосування заданого аксесуару для автомобіля	+	Відправлення приватного повідомлення щодо звернення до керуючого	+
Перегляд сторінки продавця аксесуарів для автомобілів	+	Перегляд власних звернень до керуючого	+
Перегляд сторінки покупця аксесуарів для автомобілів	+	Редагування даних аксесуару для автомобіля, виставленого на продаж	+
Перегляд відгуків про покупця аксесуарів для автомобілів	+	Редагування історії застосування власного аксесуару для автомобіля	+
Перегляд відгуків про продавця аксесуарів для автомобілів	+	Збереження аксесуару для автомобіля	+
Заведення акаунту продавця аксесуарів для автомобілів	+	Перегляд збережених аксесуарів для автомобілів	+
Заведення акаунту покупця аксесуарів для автомобілів	+	Управління замовленим придбанням аксесуару для автомобіля покупцем	+
Виставлення на продаж аксесуару для автомобіля	+	Створення відгуку про аксесуар для автомобіля	+
Перегляд всіх нових приватних повідомлень щодо аксесуарів для автомобілів	+	Створення відгуку про продавця аксесуару для автомобіля	+
Відправлення приватного повідомлення щодо аксесуару для автомобіля	+	Перегляд усіх замовлених придбань аксесуарів для автомобіля за покупцем	+
Перегляд приватних повідомлень за аксесуаром для автомобіля між користувачами	+	Замовлення придбання аксесуару для автомобіля	+
Управління замовленим придбанням аксесуару для автомобіля продавця	+	Вилучення аксесуару для автомобіля зі збережених	+
Звернення до керуючого щодо замовлення придбання аксесуару для автомобіля	+	Управління замовленим придбанням аксесуару для автомобіля	+
		Перегляд звернень до керуючого	+
		Управління зверненням до керуючого	+

11

Рисунок В.11 – Результати тестування програми пошуку аксесуарів для автомобілів

ВИСНОВКИ

- Під час проектування програми пошуку аксесуарів для автомобілів у підсумку охоплено моделювання сценаріїв покупця аксесуарів для автомобілів, продавця, керуючого та неавторизованого користувача, вибір для розробки системи MySQL, мови Python, фреймворку Django, повне визначення структури бази даних.
- Процес взаємодії користувачів різних ролей з програмою пошуку аксесуарів для автомобілів представлено у вигляді функціональної схеми, описано основні представлення створеної програми пошуку аксесуарів для автомобілів з відповідними шаблонами представлень, що сукупно забезпечують підтримку функціональних можливостей пошуку аксесуарів для автомобілів разом з класами моделей.
- Принципи організації роботи створеної програми пошуку аксесуарів для автомобілів для продавців, покупців, керуючих та неавторизованих користувачів визначені та описані, тестування виконання всіх функціональних можливостей забезпечено і проведено з отриманням успішних результатів.

12

Рисунок В.12 – Висновки