

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА TELEGRAM-БОТУ «АВТОФОНД» З ВИКОРИСТАННЯМ
TELEGRAM API, PYTHON ТА POSTGRE SQL**

**DEVELOPMENT OF THE TELEGRAM BOT «AUTOFOND» USING
TELEGRAM API, PYTHON AND POSTGRE SQL**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-41
Петрущенко Антон Олегович

Керівник:
Вознюк Анастасія Вадимівна

Кваліфікаційну роботу
допущено до захисту

«10» 06 2025р.

Гарант освітньої програми:

к.т.н., доцент

Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри


«20» 12 2024 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Пертущенко Антону Олеговичу

(прізвище, ім'я, по батькові)

Тема кваліфікаційної роботи «Розробка Telegram-боту «АвтоФонд» з використанням Python, telegram API, PostgreSQL»

Керівник роботи: к.т.н., доцент, Вознюк Анастасія Вадимівна

затверджені наказом закладу вищої освіти від «08» березня 2025 р. № 137/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра

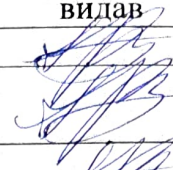
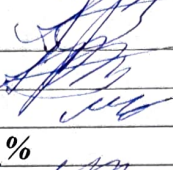
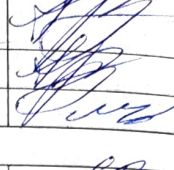
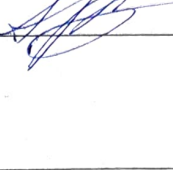
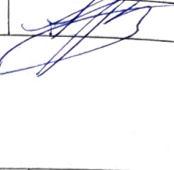
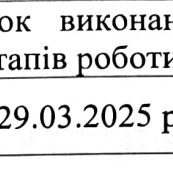
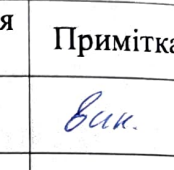
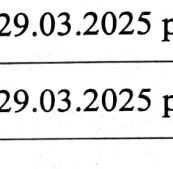
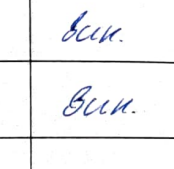
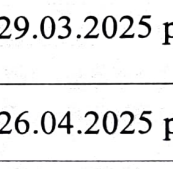
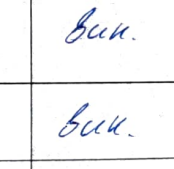
«10» червня 2025 р.

3. Вихідні дані до роботи: Python, бібліотека Aiogram, Excel, PostgreSQL, методичні вказівки до виконання кваліфікаційної роботи бакалавра

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): аналіз актуальності проблеми обліку імпортованих автомобілів та огляд існуючих рішень; визначення вимог до Telegram-бота та обґрунтування вибору технологій; проектування структури програмного забезпечення та розробка функціоналу бота з авторизацією, фільтрами та підтримкою зображень; тестування роботи бота та оцінка його зручності й надійності; створення інструкцій із встановлення, використання та опису основних компонентів; забезпечення збереження даних у таблиці з перспективою переходу до СУБД

5. Перелік графічного матеріалу: 14 рисунків, 3 таблиці

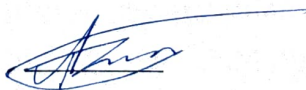
6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Вознюк А. В.		
Специфікація вимог до розробленої системи	Вознюк А. В.		
Розробка об'єкта проектування	Вознюк А. В.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	2,18 %		
Академічна доброчесність	Вознюк А. В.		

7. Дата видачі завдання «11» березня 2025 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 29.03.2025 р.	вик.
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 29.03.2025 р.	вик.
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 29.03.2025 р.	вик.
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	вик.
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	вик.
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	вик.
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	до 10.06.2025 р.	вик.

Здобувач вищої освіти



Антон ПЕТРУШЕНКО

Керівник кваліфікаційної роботи



Анастасія ВОЗНЮК

АНОТАЦІЯ

Петрущенко А. О. Розробка Telegram-боту «АвтоФонд» з використанням Telegram API, Python та PostgreSQL. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків та списку використаних джерел.

У першому розділі здійснено аналіз потреб волонтерських організацій, а також досліджено переваги використання чат-ботів у роботі з базами даних. У другому розділі визначено вимоги до системи, обґрунтовано вибір інструментів розробки, таких як Python, бібліотека Aiogram, Excel як база даних, Telegram Bot API. У третьому розділі реалізовано Telegram-бота «АвтоФонд» з функцією додавання, пошуку, редагування автомобілів, а також збереження фотографій. Здійснено тестування функціоналу, створено документацію та описано процес встановлення. У висновках узагальнено результати розробки, визначено перспективи розвитку проєкту.

Ключові слова: Telegram-бот, Python, Aiogram, Excel, волонтерський облік, автоматизація.

ABSTRACT

Petrushchenko A. Development of the Telegram Bot "AutoFond" Using Telegram API, Python and PostgreSQL. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's thesis consists of an introduction, three chapters, conclusions, and a list of references.

The first chapter analyzes the needs of volunteer organizations and the benefits of using chatbots for working with data. The second chapter defines system requirements and justifies the choice of development tools such as Python, the Aiogram library, Excel as the database, and the Telegram Bot API. The third chapter presents the implementation of the «AutoFond» Telegram bot, which includes functionality for adding, searching, and editing car data, as well as photo saving. Functional testing was conducted, documentation was created, and installation instructions were provided. The conclusions summarize the development results and outline prospects for future improvements.

Keywords: Telegram bot, Python, Aiogram, Excel, volunteer record-keeping, automation.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ ТЕЛЕГРАМ БОТІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми.....	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра.....	13
Висновки до розділу 1	14
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	15
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	15
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	26
Висновки до розділу 2	31
РОЗДІЛ 3 РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ОБЛІКУ ВИТРАТ ТА ДОХОДІВ	33
3.1 Практична реалізація об'єкта проектування	33
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	36
3.3 Розробка бази даних	38
3.4 Розробка документації для програмного продукту	40
3.5 Захист інформаційно-комп'ютерної системи	43
Висновки до розділу 3	45
ВИСНОВКИ.....	47
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49

ВСТУП

Актуальність теми. У сучасних умовах цифрової трансформації особливого значення набуває автоматизація рутинних процесів у різних сферах діяльності, зокрема у сфері обліку та ведення баз даних. Однією з таких сфер є облік ввезених автомобілів, що є актуальним як для бізнесу, так і для приватних осіб, які займаються купівлею, продажем чи оформленням транспортних засобів.

Зважаючи на стрімке зростання популярності месенджерів, таких як Telegram, та доступність API для автоматизації взаємодії з користувачами, доцільною є розробка спеціалізованих Telegram-ботів, які можуть спростити та прискорити процес ведення обліку. Telegram-бот є зручним інструментом, що дозволяє оперативно додавати нові записи, шукати необхідну інформацію, прикріплювати фото та формувати звіти без потреби у складних системах чи спеціалізованому програмному забезпеченні.

Розробка Telegram-бота для обліку ввезених автомобілів дозволить зменшити кількість ручної роботи, уникнути помилок під час ведення записів, забезпечити швидкий доступ до даних, а також організувати чітку рольову модель доступу для різних користувачів.

Метою кваліфікаційної роботи є розробка Telegram-бота «АвтоФонд», який забезпечує зручне введення, пошук, редагування та зберігання даних про ввезені автомобілі, з можливістю прикріплення фотографій, а також з реалізацією обмеження доступу на основі Telegram ID. Для реалізації рішення використовуються технології Python, Telegram API та Google Sheets (через бібліотеку gspread або pandas).

Об'єктом кваліфікаційної роботи є Telegram-бот для обліку ввезених автомобілів.

Предметом кваліфікаційної роботи є процес розробки Telegram-бота з використанням мови програмування Python, Telegram Bot API та інтеграції з електронними таблицями (Excel Shit) для зберігання інформації.

Завдання на кваліфікаційну роботу:

- визначити актуальність проблеми обліку імпортованих автомобілів та переваги автоматизації цього процесу та дослідити наявні Telegram-боти зі схожим функціоналом для виявлення сильних та слабких сторін;
- сформулювати функціональні та нефункціональні вимоги до Telegram-бота «АвтоФонд»;
- обґрунтувати вибір технологій та архітектурного підходу;
- реалізувати Telegram-бота з підтримкою авторизації за ролями (admin/manager), введення автомобіля в єдиному повідомленні, додавання фото, зберігання даних у таблиці та пошуку за фільтрами;
- протестувати роботу бота на функціональність, зручність використання та стабільність, проаналізувати результати;
- розробити супровідну документацію, яка міститиме інструкції щодо встановлення, запуску та використання Telegram-бота, а також опис його основних компонентів;
- реалізувати збереження даних у вигляді таблиці (Excel), що забезпечить простоту інтеграції з іншими системами та можливість подальшої міграції до повноцінної бази даних (наприклад, PostgreSQL).

Таким чином, дана робота спрямована на вирішення проблеми ефективного обліку ввезених автомобілів шляхом створення простого у використанні, доступного з будь-якого пристрою Telegram-бота, що поєднує в собі сучасні підходи до автоматизації та цифровізації процесів.

РОЗДІЛ 1

АНАЛІЗ РОЗРОБКИ ТЕЛЕГРАМ БОТІВ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сучасному цифровому світі Telegram-боти стали важливим інструментом автоматизації рутинних завдань, зокрема для обліку, управління інформацією та взаємодії з користувачами. Існує безліч прикладів Telegram-ботів, які дозволяють вести облік товарів, клієнтів, замовлень або послуг, адаптуючись до різноманітних сфер бізнесу. Однією з актуальних ніш є облік імпортованих автомобілів – сегмент, у якому Telegram-бот може значно полегшити процес збирання, структурування та доступу до інформації [1].

Більшість існуючих ботів забезпечують базові функції – додавання записів, пошук за ключовими параметрами, фільтрацію даних. Деякі реалізують синхронізацію з Google Таблицями, хмарними базами даних або локальними файлами, що спрощує інтеграцію з наявними обліковими системами.

Одним із ключових аспектів, на який звертають увагу користувачі, є простота взаємодії з ботом та логічна структура команд. Хороший бот повинен надавати швидкий доступ до основних функцій, запитуючи лише необхідну інформацію і не перевантажуючи інтерфейс зайвими елементами. Також важливо, щоб бот міг працювати навіть у мінімальному середовищі – без потреби встановлення додаткового ПЗ або доступу до повноцінної CRM-системи [2].

Серед важливих характеристик виділяється можливість адаптації бота до індивідуальних потреб команди або бізнесу: підтримка кастомізованих параметрів авто, додавання фото, можливість призначати ролі користувачам (наприклад, менеджер або адміністратор), ведення статистики та звітності.

Однак, незважаючи на потенціал, багато Telegram-ботів мають обмежений функціонал, реалізований без урахування потреб реальних користувачів. Часто бракує гнучкості, зручного імпорту/експорту даних, або ж

бот є прив'язаний до сторонніх сервісів із незрозумілим рівнем безпеки. Також проблеми можуть виникати з розмежуванням доступу, що ускладнює використання в командній роботі.

Дослідження та аналіз існуючих Telegram-ботів у сфері обліку дозволяє виявити ключові тренди й очікування користувачів. Це, своєю чергою, дає змогу сформулювати чіткі вимоги до розробки власного Telegram-бота «АвтоФонд», який забезпечить зручний, безпечний і гнучкий інструмент для обліку ввезених автомобілів, оптимізований під потреби автодилерів та менеджерів з продажу.

Прикладом подібного рішення є бот «InventoryBot» (рис. 1.1) [13].

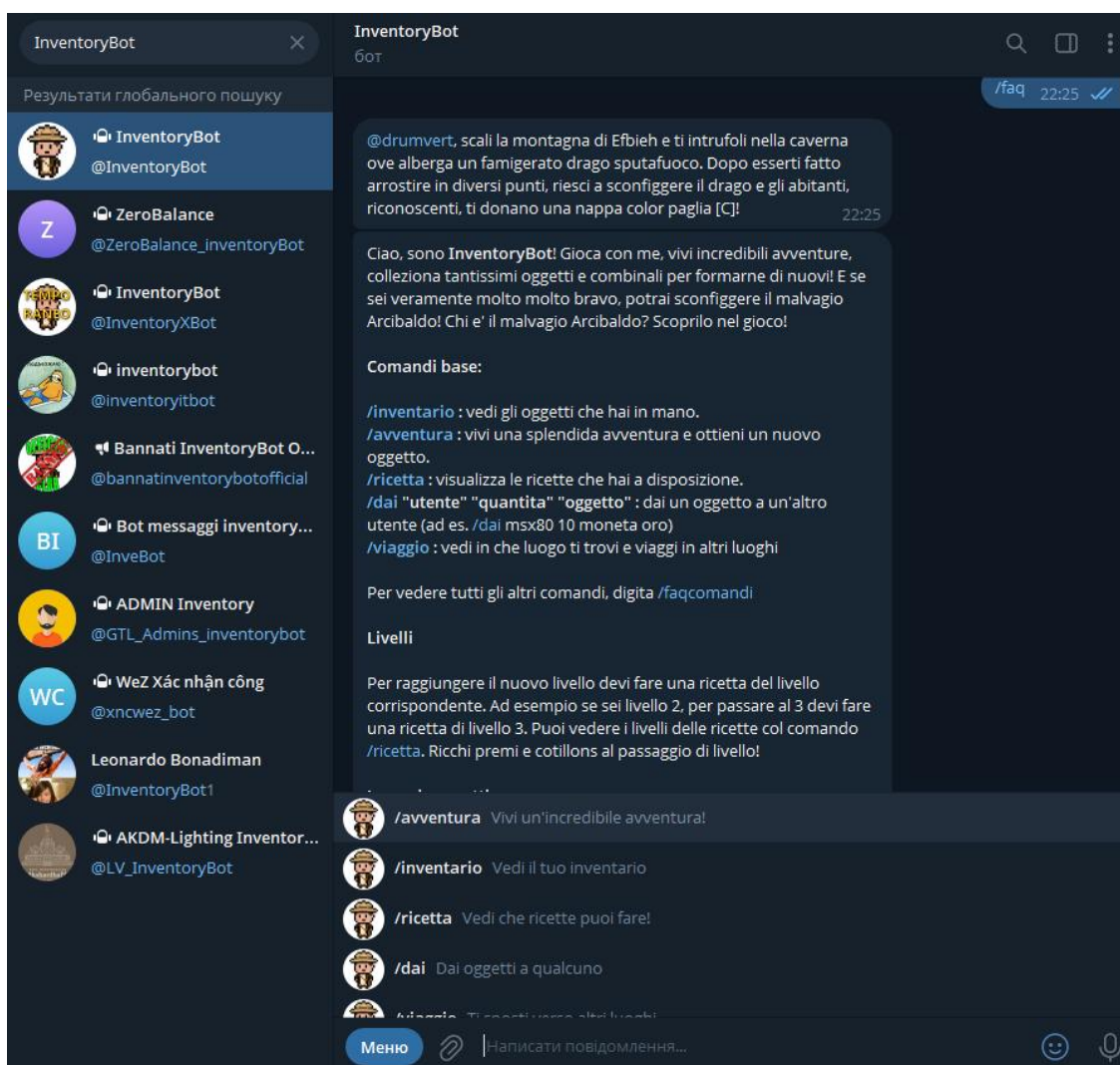


Рисунок 1.1 – Telegram-бот «InventoryBot»

«InventoryBot» – це Telegram-бот для ведення обліку товарів на складі або в магазині. Його функціонал дозволяє додавати нові одиниці товару, редагувати дані, фільтрувати записи за параметрами (наприклад, назвою або штрихкодом), а також переглядати залишки на складі. Дані зберігаються у Google Таблицях або у локальній базі на сервері [13].

Бот підтримує обмеження прав доступу, де адміністратор може змінювати дані, а менеджери – лише переглядати.

Хоча бот орієнтований на товарний облік, його структура чудово підходить і для обліку авто, оскільки має простий механізм додавання записів та можливість прикріплення зображень.

Ще одним прикладом є Telegram-бот «AutoSeller Bot» (рис. 1.2) [19].

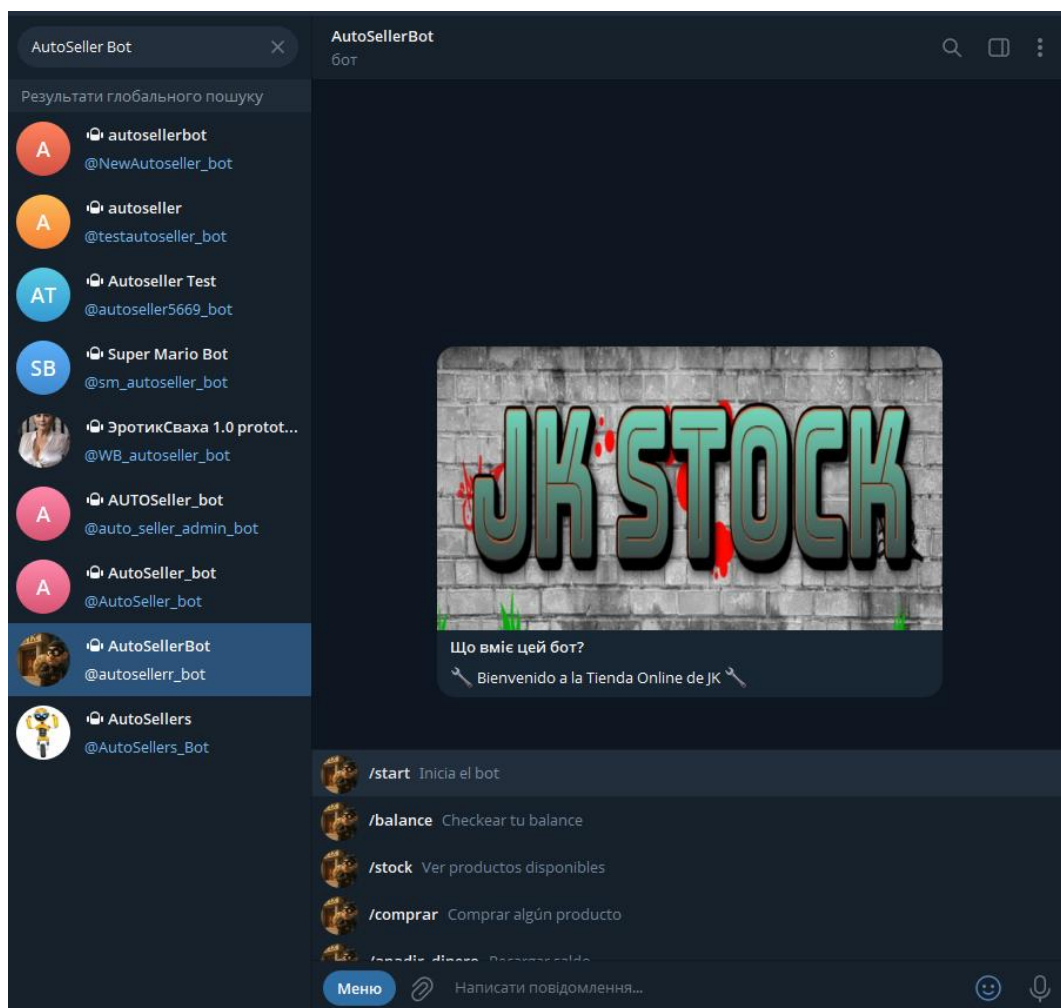


Рисунок 1.2 – Telegram-бот «AutoSeller Bot»

«AutoSeller Bot» – бот, створений для допомоги авто-дилерам у продажі вживаних автомобілів. Він дозволяє додавати авто за шаблоном (марка, модель, рік, пробіг, фото), які потім можна переглядати у форматі картки товару [19].

Користувач може запитати повний список доступних авто або здійснити пошук за окремими характеристиками.

Бот інтегрується з каналами Telegram, куди можна автоматично надсилати публікації про нові авто. Такий підхід дозволяє одночасно вести облік та просувати товар.

Також вартий уваги Telegram-бот «SheetManager» (рис. 1.3) [14].

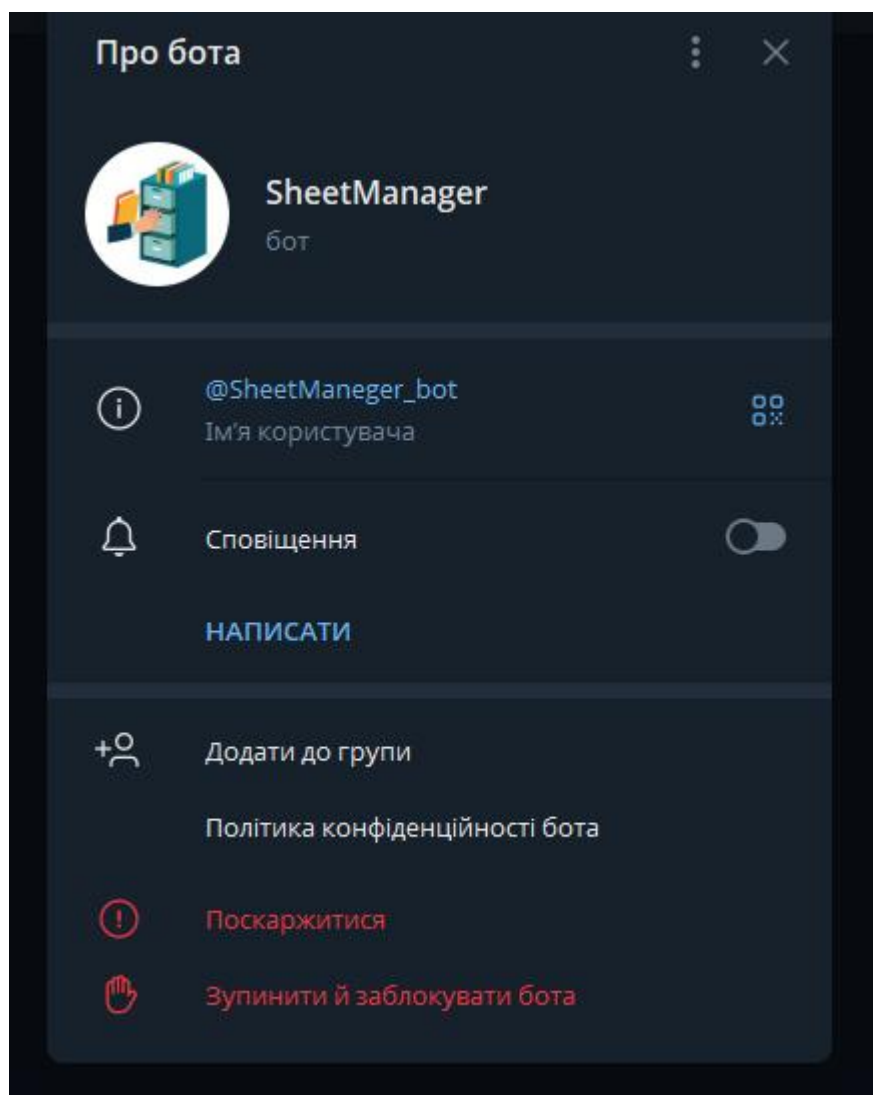


Рисунок 1.3 – Telegram-бот «SheetManager»

«SheetManager» – універсальний бот для роботи з Google Таблицями через Telegram. Його функціонал дозволяє створювати та заповнювати таблиці прямо в чаті, працюючи з будь-якими даними: від клієнтів до продуктів або послуг [14].

Користувачі можуть вводити дані у вигляді одного повідомлення з роздільниками (наприклад, комами), а бот автоматично розміщує їх у потрібних стовпцях таблиці.

Він підтримує обмеження за Telegram ID, зберігає історію дій, має можливість експорту в CSV, а також дозволяє додавати посилання на файли або фото.

Хоча бот не є спеціалізованим саме для автомобілів, його гнучкість дозволяє адаптувати його під будь-який тип обліку, включаючи імпортовані авто.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

Після аналізу існуючих рішень у сфері обліку даних через Telegram-ботів було прийнято рішення розробити Telegram-бот «АвтоФонд», який дозволить вести облік імпортованих автомобілів, забезпечити простоту взаємодії, швидке введення даних та гнучке управління доступом.

Telegram обрано як платформу через його популярність, зручний інтерфейс, наявність бот-API та можливість миттєвої взаємодії з користувачами.

Було поставлено наступні завдання, які необхідно реалізувати в межах кваліфікаційної роботи:

- визначити актуальність проблеми обліку імпортованих автомобілів та переваги автоматизації цього процесу та дослідити наявні Telegram-боти зі схожим функціоналом для виявлення сильних та слабких сторін;
- сформулювати функціональні та нефункціональні вимоги до Telegram-бота «АвтоФонд»;
- обґрунтувати вибір технологій та архітектурного підходу;

- реалізувати Telegram-бота з підтримкою авторизації за ролями (admin/manager), введення автомобіля в єдиному повідомленні, додавання фото, зберігання даних у таблиці та пошуку за фільтрами;
- протестувати роботу бота на функціональність, зручність використання та стабільність, проаналізувати результати;
- розробити супровідну документацію, яка міститиме інструкції щодо встановлення, запуску та використання Telegram-бота, а також опис його основних компонентів;
- реалізувати збереження даних у вигляді таблиці (Excel), що забезпечить простоту інтеграції з іншими системами та можливість подальшої міграції до повноцінної бази даних (наприклад, PostgreSQL).

Висновки до розділу 1

У першому розділі було проведено детальний аналіз сучасних рішень у сфері Telegram-ботів, орієнтованих на збереження і обробку даних. Розглянуто приклади таких ботів, як InventoryBot, AutoSeller Bot і SheetManager, що дозволило виявити поточні підходи до реалізації облікових систем у Telegram-середовищі.

Проаналізовані рішення стали основою для формування технічних вимог до Telegram-бота «АвтоФонд» – інструмента, який має забезпечити простий облік імпортованих авто з можливістю додавання фото, фільтрації, пошуку та розмежування доступу між користувачами.

Було визначено перелік завдань для реалізації кваліфікаційної роботи, серед яких: аналіз, розробка, тестування та впровадження повнофункціонального Telegram-бота для автоматизації обліку автомобілів.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Аналіз вимог до програмного забезпечення є ключовим етапом у процесі створення Telegram-бота для обліку ввезених автомобілів «АвтоФонд». Основною метою цього етапу є виявлення, систематизація та документування функціональних і нефункціональних вимог, що визначають, яким має бути функціонал системи, її структура та поведінка [13].

Telegram-бот «АвтоФонд» призначений для спрощення процесу обліку ввезених автомобілів, забезпечення централізованого зберігання інформації, швидкого доступу до даних та розмежування ролей користувачів (наприклад, адміністратор, менеджер). З огляду на цілі розробки, важливим є вибір відповідної моделі програмного забезпечення, яка найкраще відповідатиме архітектурі системи та характеру задач.

Існує кілька підходів до моделювання програмного забезпечення, серед яких найпоширенішими є [4,5]:

- об'єктно-орієнтована модель (ООП);
- функціонально-орієнтована модель;
- модель на основі компонентів;
- модель, орієнтована на дані.

Об'єктно-орієнтована модель побудована на концепції класів та об'єктів. Цей підхід дозволяє будувати програму як сукупність взаємодіючих об'єктів, що інкапсулюють стан і поведінку. Основні переваги:

- інкапсуляція – приховування внутрішніх даних об'єкта від зовнішнього середовища;
- поліморфізм – можливість використовувати один інтерфейс для реалізації різної поведінки;
- наслідування – повторне використання коду і спрощення підтримки;

- модульність – чіткий поділ системи на незалежні частини.

Функціонально-орієнтована модель зосереджена на реалізації послідовності функцій і алгоритмів. Зазвичай використовується у вигляді діаграм потоків даних. Цей підхід менш гнучкий при зміні вимог і складний у підтримці великої системи [5].

Модель на основі компонентів використовує незалежні функціональні блоки, які можуть бути перевикористані. Придатна для великих систем, але має певну складність в інтеграції та залежностях.

Модель, орієнтована на дані підходить для систем, де основною задачею є обробка та зберігання великих обсягів даних, як-от бази даних або сховища інформації.

З огляду на завдання Telegram-бота «АвтоФонд», об'єктно-орієнтована модель є найбільш доцільною, оскільки:

дозволяє логічно структурувати бот за допомогою класів (наприклад «Автомобіль», «Користувач», «Фото», «Запис»);

- спрощує підтримку та масштабування системи;
- забезпечує гнучкість у додаванні нових функцій, як-от експорт даних, додавання фото або ведення статистики;
- забезпечує безпеку та цілісність даних завдяки інкапсуляції;
- дає можливість зручно реалізувати перевірку прав доступу через Telegram ID.

Для ефективної реалізації Telegram-бота необхідно визначити чіткі вимоги до програмного забезпечення, які поділяються на функціональні та нефункціональні. Функціональні вимоги описують, що має виконувати система, зокрема:

- додавання записів про автомобіль із зазначенням параметрів через одне повідомлення;
- збереження фото та прив'язка до запису;
- перевірка доступу користувача за Telegram ID;
- пошук автомобіля за фільтрами;

- вивід статистики;
- редагування та видалення записів.

Нефункціональні вимоги визначають, як система повинна виконувати функції:

- продуктивність – бот має швидко обробляти запити;
- надійність – система повинна стабільно працювати без збоїв;
- безпека – обмеження доступу до функціоналу за роллю (admin, manager);
- масштабованість – можливість додавати нові функції без кардинальної зміни структури;
- зручність використання – інтуїтивно зрозумілий інтерфейс взаємодії в Telegram.

Таким чином, на цьому етапі було сформульовано ключові вимоги до Telegram-бота «АвтоФонд», що дозволяють перейти до розробки архітектури та реалізації функціоналу з урахуванням потреб кінцевих користувачів і цілей проєкту.

У таблиці 2.1 наведено список функціональних вимог до системи.

Таблиця 2.1 – Функціональні вимоги

Вимога	Опис	Деталі
Додавання автомобіля	Менеджер може додати нове авто в базу.	Дані вводяться в одному повідомленні через кому: Марка, модель, рік, VIN, номер кузова, стан, відповідальний. Після цього бот запитує про додавання фото.
Завантаження фото	Користувач може прикріпити фото до запису авто.	Бот пропонує завантажити 1 повідомлення з фото. Фото зберігається у папці images/, шлях додається в таблицю у колонку img.

Продовження таблиці 2.1

Редагування автомобіля	Можливість змінити дані про вже додане авто.	Адміністратор або менеджер вибирає запис за VIN або іншими параметрами та редагує конкретні поля (наприклад, стан, відповідального, фото).
Пошук авто	Швидкий пошук потрібного авто у базі.	Пошук за VIN, маркою, статусом. Результат виводиться з фото, якщо воно додане, та повною інформацією.
Статистика	Бот може згенерувати статистику по всіх авто.	Наприклад: кількість ввезених авто, скільки передано, середній час від ввезення до передачі, тощо.
Перевірка доступу	Ролі користувачів визначають їх можливості у боті.	Дані про ролі зберігаються в admins.xlsx. Бот перевіряє ID користувача і дозволяє: адміну – все, менеджеру – додавати/редагувати, іншим – лише перегляд.
Пошук транзакцій	Користувач може швидко шукати транзакції за назвою.	Пошук проводиться за назвою транзакції та повинен бути розташований біля списку транзакцій. Пошук повинен враховувати помилки при наборі.
Зберігання даних	Дані повинні зберігатись надійно та зручно для подальшої обробки.	Основний формат – Excel (можливість зберігання в Google Sheets або PostgreSQL). Дані вносяться построчно із збереженням структури.

У таблиці 2.2 наведено список нефункціональних вимог.

Таблиця 2.2 – Нефункціональні вимоги

Вимога	Опис
Продуктивність	Бот має миттєво реагувати на команди, обробка даних – не більше 1-2 секунд.
Масштабованість	Можливість розширення функціоналу без повної переробки архітектури.
Юзабіліті	Інтерфейс діалогу з ботом має бути інтуїтивним навіть для користувача без досвіду.
Безпека	Доступ до змін у базі мають лише авторизовані користувачі (за Telegram ID та роллю).
Надійність	У разі помилок введення бот не завершує роботу, а просить користувача повторити.

На основі вимог до функціональності та зручності взаємодії було реалізовано Telegram-бот, що виконує роль системи обліку імпортованих автомобілів. Бот реалізований на основі бібліотеки Aiogram, з використанням бази даних у вигляді Excel-файлу, що дозволяє зручно зберігати й обробляти інформацію. Інтерфейс побудовано з урахуванням мінімалізму та простоти, з орієнтацією на ефективне виконання конкретних задач: додавання авто, пошук і перегляд статистики [5].

Нижче наведено скріншоти головних екранів взаємодії з ботом, а також детальний опис кожної функціональної частини.

Головне меню бота. Після запуску бота користувач потрапляє на головний екран, на якому відображаються основні функції: «**+** Додати авто», «**🔍** Пошук авто» та «**📊** Статистика». Меню реалізовано у вигляді клавіатури Telegram, що дозволяє швидко викликати необхідні функції.

Інтерфейс не перевантажений зайвими елементами, що забезпечує інтуїтивно зрозумілу взаємодію. Кнопка «Додати авто» відкриває діалог додавання нового запису. «Пошук авто» дозволяє швидко знаходити інформацію за ключовими словами. «Статистика» забезпечує миттєвий аналіз бази даних авто.

Основні умови та шлях юзера телеграм боту «АвтоФонд» наведено на UML діаграмі (рис. 2.1).

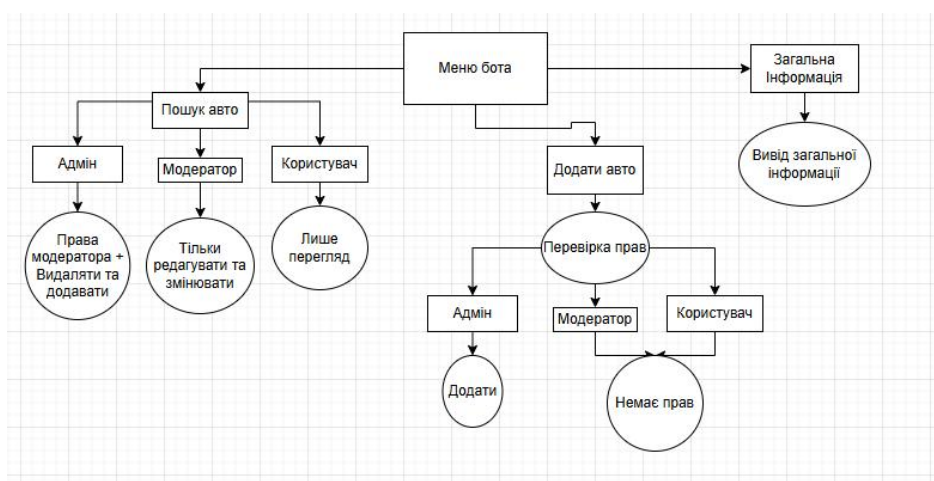


Рисунок 2.1 – Схема використання телеграм боту

Привітання від телеграм боту «АвтоФонд» та його запуск зображено на рисунку 2.2.

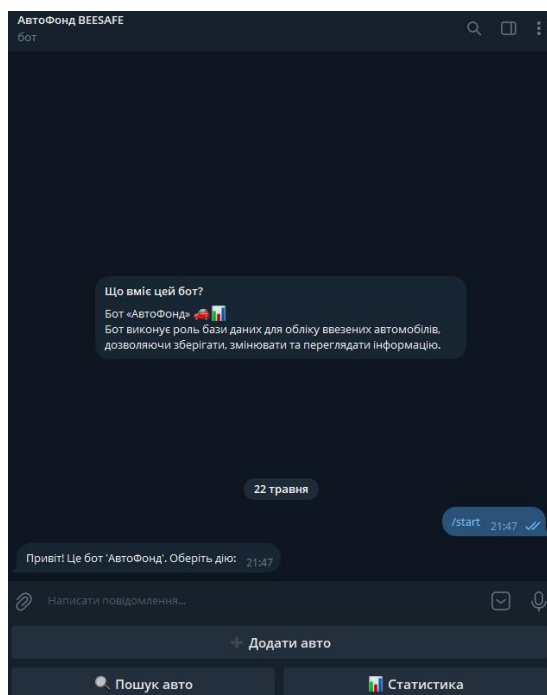


Рисунок 2.2 – Головне меню Telegram-бота «АвтоФонд»

Додавання авто. При натисканні на кнопку «**+** Додати авто» бот переходить у режим очікування введення даних (рис. 2.3). Користувачеві надсилається повідомлення з інструкцією, в якому вказано, що необхідно ввести всю інформацію в одному повідомленні через кому. Такий підхід дозволяє зменшити кількість запитів і забезпечити швидкий облік авто.

Користувач вводить дані у наступному порядку:

- марка;
- модель;
- рік;
- VIN;
- номер кузова;
- статус;
- дата додавання.

Як приклад як би це виглядало: BMW, X5, 2017, WBAVL12030TR00001, 345AB1234, У процесі розмитнення, 21.05.2025.

Бот розбиває цей рядок на частини, перевіряє кількість полів, і, якщо всі дані вірні – надсилає повідомлення з запитом, чи потрібно додати фотографію. Це реалізовано через інлайн-кнопки: «Так» / «Ні» (рис. 2.3).

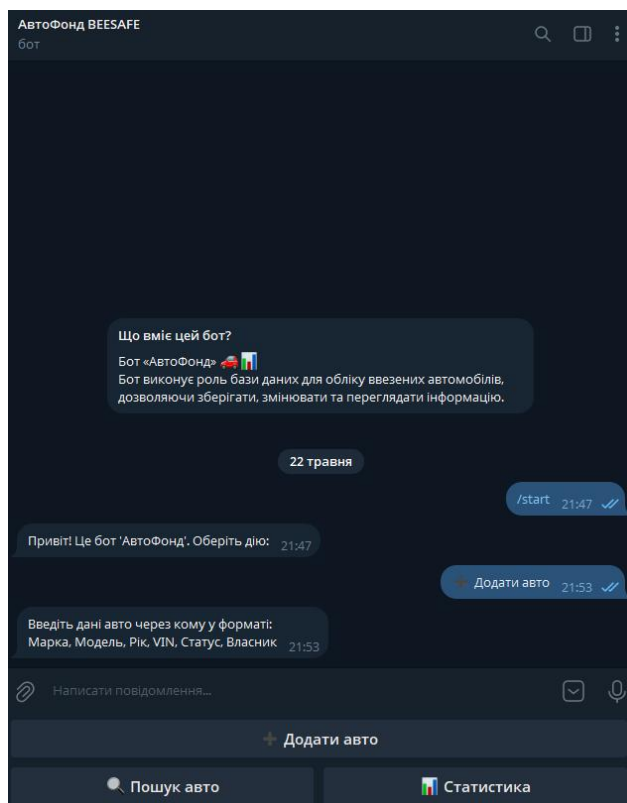


Рисунок 2.3 – Інструкція до введення даних при додаванні авто

Завантаження фотографії. Якщо користувач натискає кнопку «Так» після введення даних – бот переходить у режим очікування зображення (рис. 2.4). Користувач повинен надіслати одне повідомлення з фото, яке буде автоматично збережено в папці images. Шлях до зображення додається до відповідного рядка в Excel-файлі у вигляді словника.

Це дозволяє при подальшому пошуку авто отримати не лише текстову інформацію, але й переглянути прикріплене фото. Якщо користувач не хоче додавати фото - бот відразу записує дані в таблицю.

Функція обробки фото реалізована з урахуванням обмежень Telegram API щодо формату файлів. Ім'я фото зберігається у первісному вигляді, що спрощує подальше вивантаження при потребі [4].

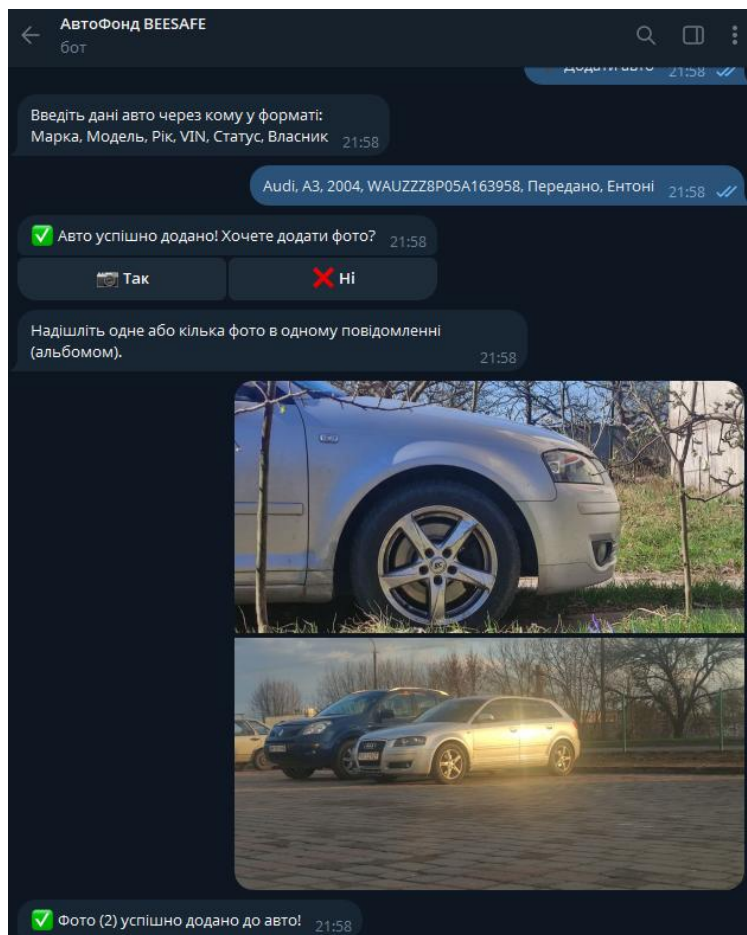


Рисунок 2.4 – Додавання авто з фото. Повідомлення із підтвердженням збереження

Пошук авто. Функція пошуку реалізована максимально просто та ефективно. При натисканні кнопки «🔍 Пошук авто», бот надсилає повідомлення з проханням ввести будь-яке ключове слово (частину моделі, статус, рік тощо). У відповідь користувач вводить запит.

Бот фільтрує рядки таблиці за вказаним словом, використовуючи case-insensitive пошук у всіх полях записів. Усі знайдені авто відправляються користувачеві у вигляді текстових повідомлень з повною інформацією про кожне авто. Якщо авто має фото – воно також прикріплюється до відповіді.

Це дозволяє миттєво знаходити записи серед великої кількості імпортованих авто (рис. 2.5).

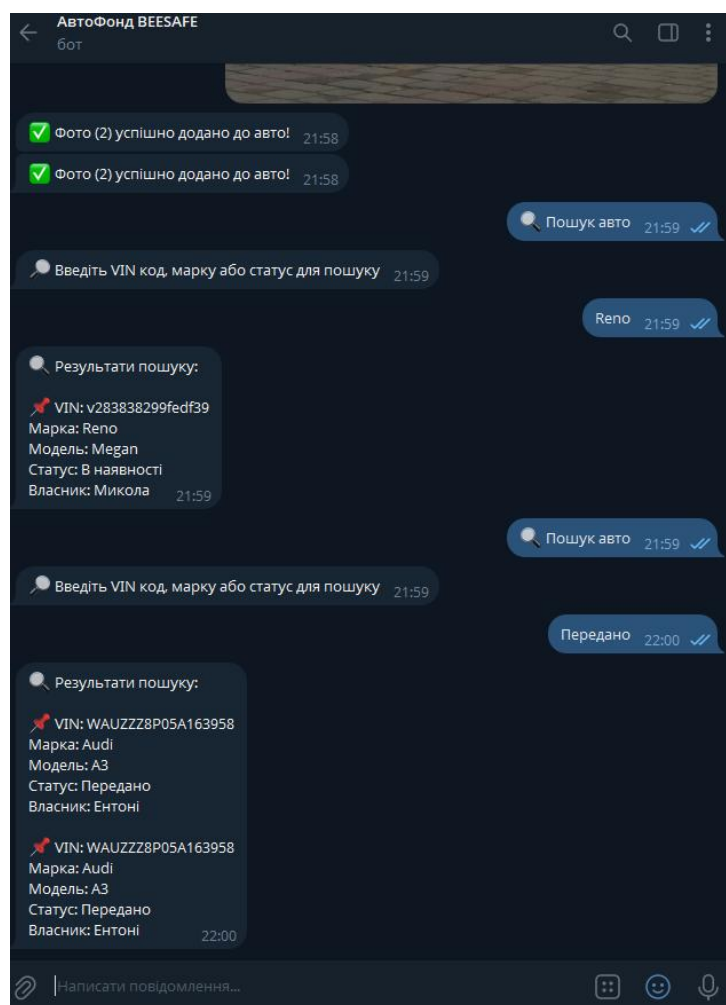


Рисунок 2.5 – Введення ключового слова для пошуку авто

Результати пошуку. Після обробки запиту бот надсилає один або кілька результатів у текстовому форматі. Формат повідомлення виглядає приблизно так:

- марка: BMW;
- модель: X5;
- рік: 2020;
- VIN: WBAVL12030TR00001;
- номер кузова: 345AB1234;
- статус: У розмитненні;

– дата: 21.05.2025.

Якщо до запису прикріплено фото - бот надсилає зображення разом із текстовою інформацією. Це особливо корисно, коли потрібно швидко ідентифікувати авто візуально (рис. 2.6).

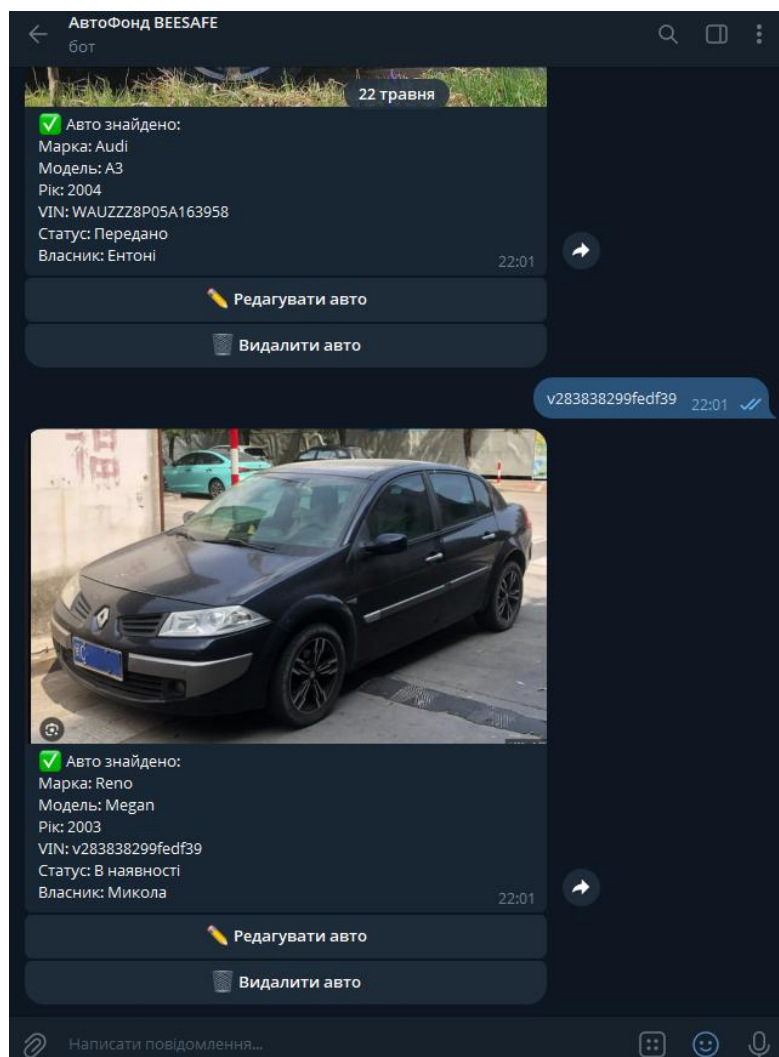


Рисунок 2.6 – Результат пошуку авто з прикріпленим фото

Перегляд статистики. Функція « Статистика» дозволяє адміністратору отримати агреговану інформацію про всі авто, що зберігаються в базі. Бот обробляє Excel-файл і виводить наступну інформацію:

Загальна кількість авто. Кількість авто у кожному статусі (наприклад: «У розмитненні», «Продано», «Очікує огляд» тощо).

Найстаріший та найновіший рік випуску авто. Статистика формується в реальному часі, що дозволяє бачити актуальний стан автопарку в будь-який момент.

Цей функціонал є важливим для керівництва, яке може оперативно аналізувати обсяг поставок, відстежувати динаміку поповнення автопарку та виявляти проблемні зони (рис. 2.7).

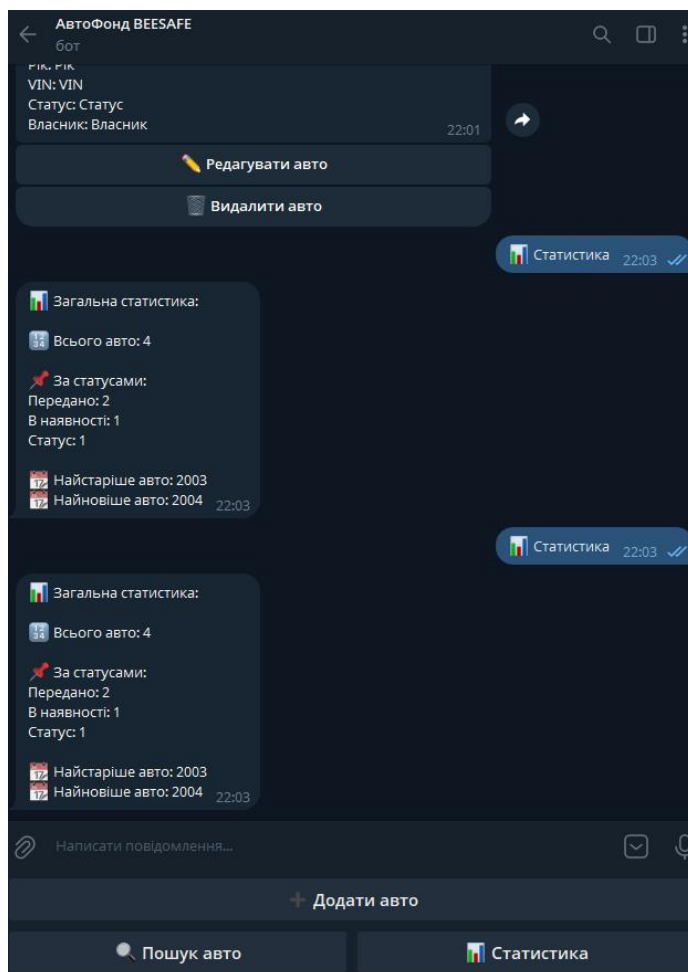


Рисунок 2.7 – Статистика автопарку, згенерована ботом

Telegram-бот «АвтоФонд» реалізує всі ключові функції обліку імпортованих авто: додавання, зберігання фото, пошук, аналітика. Простота інтерфейсу робить його зручним для щоденного використання, а використання таблиці Excel як бази даних – легкою в інтеграції для невідготовлених користувачів. В майбутньому передбачено перенесення даних на повноцінну

реляційну СУБД (наприклад, PostgreSQL) для розширення можливостей та забезпечення масштабування [9].

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

При розробці Telegram-бота важливо правильно обрати мову програмування, інструменти, бібліотеки та структуру, яка дозволить ефективно реалізувати необхідну функціональність, враховуючи зручність масштабування, простоту підтримки та швидкодію. У нашому випадку йдеться про створення Telegram-бота, який виконує роль обліку автомобілів, має розмежування доступу, працює з Excel-файлом, отримує зображення та працює з локальними ресурсами.

Вибір платформи: Telegram API – Telegram надає відкритий API, який дозволяє створювати ботів, що можуть приймати повідомлення, відповідати на команди, обробляти медіа, надсилати клавіатури, виконувати inline-запити тощо. API підтримується як у форматі HTTP-запитів, так і через Webhooks або довготривале опитування (long polling). Для більшості невеликих та середніх проєктів з низькою затримкою, найчастіше використовується саме long polling, який не вимагає окремого серверного сертифікату [4].

Вибір мови програмування: Python – є однією з найпопулярніших мов для розробки Telegram-ботів завдяки простоті синтаксису, великій кількості бібліотек та швидкій розробці. Вона ідеально підходить для автоматизації, роботи з файлами, API, а також взаємодії з Excel і Telegram API [6].

Серед ключових переваг Python:

- велика спільнота та наявність готових рішень (бот-фреймворки, плагіни, інтеграції);
- лаконічний синтаксис;
- можливість легкої інтеграції з бібліотеками для роботи з Excel (openpyxl, pandas), зображеннями (Pillow), файловою системою та мережею.

Вибір бібліотеки для роботи з Telegram API: Aiogram. Для створення бота використано Aiogram – асинхронний фреймворк для роботи з Telegram Bot API. Aiogram заснований на `asyncio` та дозволяє реалізувати продуктивніші рішення порівняно з синхронними бібліотеками, такими як `python-telegram-bot` [5,11].

Переваги Aiogram:

- повністю асинхронна архітектура;
- висока швидкодія;
- зручна структура розділення логіки на хендлери;
- підтримка FSM (машини станів), що дозволяє організувати послідовний сценарій введення даних [5];
- легко масштабуються за допомогою `middlewares`, `filters` та окремих файлів логіки.

Недоліки Aiogram:

- для початківців може бути складніше через використання `async/await`;
- документація менш детальна, ніж у деяких інших бібліотек, однак активно оновлюється.

Зберігання та обробка даних: Excel (`openpyxl`, `pandas`). Для зберігання даних про автомобілі використовується Excel-файл, який є зручним способом для невеликої локальної бази даних. З ним працює Python-бібліотека `openpyxl`, яка дозволяє [8,12]:

- створювати, читати та змінювати Excel-файли (`.xlsx`);
- працювати зі стилями, форматами, формулами;
- отримувати дані по комірках і додавати нові рядки.

Також за потреби можна застосовувати `pandas` для більш складного аналізу або агрегації даних, наприклад, під час створення звітів або пошуку [7].

Недоліки використання Excel:

- не підходить для високонавантажених проєктів;
- при одночасному доступі кількох користувачів можуть виникати конфлікти;
- немає транзакційності, як у СУБД.

Для масштабування в майбутньому можливо перейти на PostgreSQL або Firebase [11].

Бот підтримує завантаження одного фото на кожен автомобіль. Фото зберігаються у локальній директорії images/, а в Excel-файлі фіксується шлях до фото (наприклад: «img»: «images/IMG_20240515_143322.jpg»). Це дозволяє легко завантажувати і виводити фото у відповідь на запити користувачів.

Переваги локального зберігання:

- немає необхідності в інтеграції з хмарними сервісами;
- повний контроль над структурою та доступом;
- зручність у розробці та тестуванні.

Недоліки:

- прив'язаність до одного сервера;
- складність із масштабуванням.

У разі розгортання на хостингу або VPS важливо враховувати місце на диску та можливість резервного копіювання.

Алгоритм розмежування доступу. Доступ до дій бота обмежується за допомогою перевірки Telegram ID користувача у таблиці admins.xlsx, де вказано:

- Telegram ID (унікальний ідентифікатор);
- роль користувача (admin, manager тощо).

Після кожної команди бот перевіряє роль і надає доступ до певних функцій відповідно до неї. Це забезпечує просту і гнучку систему контролю доступу без використання повноцінної бази даних.

FSM (машина станів) для послідовного введення даних. Для зменшення кількості повідомлень і забезпечення послідовності введення даних про авто, бот використовує FSM. Користувач вводить всі параметри через кому (наприклад: «Audi A6, 2015, дизель, 150000 км, чорний»), після чого бот виконує .split(«»), перевіряє кількість елементів, і зберігає їх у відповідні стовпці Excel [5].

В загальному це впливає саме так:

- прискорює процес вводу;
- зменшує спам у чаті;
- спрощує обробку даних.

Керування фото через Inline-кнопки. Після введення даних бот запитує: «Чи потрібно додати фото?». Якщо так – приймає одне повідомлення з фото, зберігає його локально та записує шлях у таблицю. Якщо ні – просто додає запис без фото.

Це реалізовано через inline-кнопки, що дозволяє уникати зайвого спаму командами і робить інтерфейс дружнім для користувача.

Архітектура проєкту. Проєкт структуровано наступним чином:

- `main.py` – точка входу, запуск бота;
- `handlers/` – окремі модулі з обробниками команд (`add_car.py`, `search.py`, `access_control.py`);
- `utils/` – функції для роботи з Excel, фото, перевірками;
- `admins.xlsx` – таблиця з правами доступу;
- `cars.xlsx` – база даних автомобілів;
- `images/` – папка з усіма фото.

Така архітектура дозволяє легко масштабувати логіку, додавати нові модулі (наприклад, статистику, фільтри пошуку, завантаження PDF) без порушення існуючої структури.

Обробка повідомлень та модульність логіки. Aiogram дозволяє чітко розділяти логіку обробки команд, повідомлень, `callback`-ів та медіа у вигляді хендлерів, які можуть бути організовані по модулях [5]. Це дає змогу:

- не перевантажувати головний файл (`main.py`);
- швидко знаходити потрібні частини коду;
- підключати нові функції без втручання в базову логіку.

Наприклад, команду `/add_car` обробляє окремий модуль, який керує всією FSM-послідовністю, перевіркою доступу, записом до Excel та, за потреби, обробкою фото.

Кожен хендлер має обмеження через filters – наприклад, фільтр ролі, типу повідомлення, стану FSM. Це зменшує ймовірність помилок і забезпечує чітку маршрутизацію [13].

Асинхронність і обробка навантаження. Завдяки `asyncio` в `Aiograt` можна обробляти десятки або навіть сотні запитів одночасно, не блокуючи основний потік. У нашому випадку:

- бот приймає повідомлення про додавання авто;
- одночасно може приймати фото;
- паралельно обробляє запити на пошук, перегляд чи видалення;
- надсилає відповіді без затримок навіть при високій активності.

Це особливо важливо при використанні FSM, де кожен користувач перебуває в окремому «сценарії». Асинхронність дозволяє не чекати завершення дій одного користувача, перш ніж обробити іншого.

Логування та обробка помилок. У бота реалізована система логування, яка дозволяє:

- відслідковувати помилки при читанні або записі до Excel;
- логувати дії користувачів (додавання, пошук, помилки доступу);
- зберігати інформацію про помилкові фото (неправильний формат, відсутність медіа тощо).

Це реалізовано за допомогою стандартного модуля `logging` Python, з можливістю виведення в консоль або в лог-файл на сервері. У майбутньому логування можна доповнити повідомленнями адміну у Telegram про критичні помилки.

Валідація даних користувача. Оскільки бот приймає всі параметри автомобіля через одне повідомлення (через кому), важливо ретельно перевіряти:

- кількість введених полів (наприклад, 5 параметрів – марка, рік, паливо, пробіг, колір);
- формат (перевірка, що рік – це число, пробіг – теж число, колір – текст);
- допустимі значення (наприклад, рік у діапазоні 1990-2025).

Це мінімізує помилки при записі до Excel і дозволяє уникнути зламу структури таблиці. У разі помилки бот відповідає користувачу з проханням ввести дані повторно, вказуючи причину помилки.

Зворотний зв'язок та інформування користувача. Бот побудований з урахуванням UX-принципів, аби користувач розумів, що саме відбувається:

- після успішного додавання авто з'являється повідомлення-підтвердження;
- якщо завантажено фото – бот виводить шлях і зразок збереженого імені;
- при пошуку або перегляді бот надсилає не лише текст, а й відповідне фото (якщо воно є).

Хоча зараз дані зберігаються локально в Excel, структура побудована так, щоб у будь-який момент:

- замінити Excel на базу даних (наприклад, PostgreSQL) [6];
- перенести фото у хмарне сховище (наприклад, Amazon S3 або Google Drive API);
- використовувати кешування (Redis) для швидкого доступу до частих запитів;
- організувати адмін-панель у браузері (наприклад, через Flask або FastAPI) [10].

Це дозволить масштабувати бота на більшу кількість користувачів, підключити менеджерів із різних регіонів, додати фільтрацію, аналітику, CRM-функціонал тощо.

Незважаючи на просту структуру, бот забезпечує базову безпеку:

- перевірка Telegram ID запобігає несанкціонованому доступу;
- доступ до таблиць і фото обмежений у межах сервера;
- можна налаштувати резервне копіювання cars.xlsx і папки images/.

Висновки до розділу 2

У цьому розділі було проведено ґрунтовний аналіз засобів, методів і алгоритмів, необхідних для реалізації Telegram-бота з функціоналом обліку імпортованих автомобілів. Визначено ключові вимоги до системи, серед яких: підтримка ролей користувачів (адміністратор/менеджер), збереження даних у структурованому вигляді, зручний процес додавання авто, обробка зображень та забезпечення доступу лише авторизованим особам.

На основі поставлених вимог обрано мову програмування Python та асинхронний фреймворк Aiogram, який дозволяє ефективно реалізовувати логіку взаємодії з користувачами за допомогою хендлерів, станів (FSM) та callback-кнопок. Реалізовано асинхронну обробку повідомлень, що забезпечує стабільну роботу бота навіть при великій кількості одночасних запитів.

Для збереження даних було обрано формат Excel-таблиці, що забезпечує швидкий старт, простоту інтеграції та можливість ручного редагування. Також впроваджено механізм обробки та збереження зображень автомобілів, із збереженням їх на сервері та внесенням відповідних посилань у таблицю.

Окрему увагу приділено перевірці введених даних, структурі команд, побудові гнучкої FSM-послідовності та безпеці: система перевіряє права доступу на основі Telegram ID та не допускає несанкціонованих дій.

Обрані підходи до реалізації, зокрема модульна структура, логування дій, та гнучке керування даними, дозволяють масштабувати проєкт у майбутньому: мігрувати з Excel на повноцінну базу даних, інтегрувати з хмарними сховищами для фото або реалізувати веб-інтерфейс для менеджменту.

У результаті, обрані інструменти та розроблені алгоритми дозволяють створити функціональний, зручний у використанні та розширюваний Telegram-бот, що повністю відповідає вимогам до сучасної системи обліку імпортованих авто.

РОЗДІЛ 3

РОЗРОБКА ТЕЛЕГРАМ БОТУ ДЛЯ ОБЛІКУ АВТОМОБІЛІВ

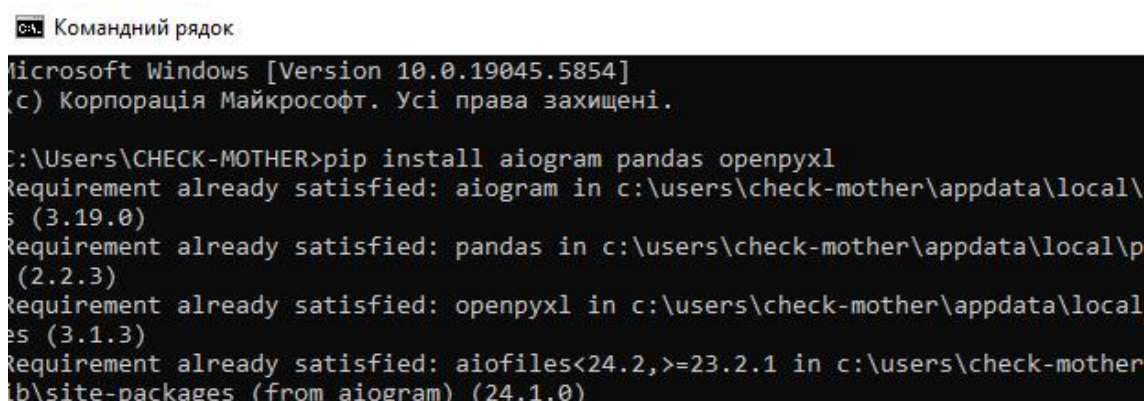
3.1 Практична реалізація об'єкта проєктування

Розроблюваний об'єкт – Telegram-бот під назвою «АвтоФонд», створений для ведення обліку імпортованих автомобілів, з можливістю додавання, пошуку, редагування, зберігання фото та статистики. Бот має перевірку доступу за ролями (admin/manager) та працює з Excel-файлом як основною базою даних.

Бот реалізований за допомогою таких засобів:

- мова програмування: Python 3.11 [6];
- бібліотека для Telegram: Aiogram (асинхронний фреймворк для ботів) [5,11];
- обробка таблиць: pandas + openpyxl [7,8];
- збереження фото: локально в папці images [15];
- захист доступу: на основі Telegram ID через файл admins.xlsx [5,7].

Для початку необхідно встановити залежності. Це робиться через команду, яка прописана на рисунку 3.1.



```

Командний рядок
Microsoft Windows [Version 10.0.19045.5854]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\CHECK-MOTHER>pip install aiogram pandas openpyxl
Requirement already satisfied: aiogram in c:\users\check-mother\appdata\local\
s (3.19.0)
Requirement already satisfied: pandas in c:\users\check-mother\appdata\local\p
(2.2.3)
Requirement already satisfied: openpyxl in c:\users\check-mother\appdata\local
es (3.1.3)
Requirement already satisfied: aiofiles<24.2,>=23.2.1 in c:\users\check-mother
b\site-packages (from aiogram) (24.1.0)
  
```

Рисунок 3.1 – Встановлення необхідних бібліотек

Архітектура бота. Бот побудований за принципами чистої архітектури з розділенням логіки на модулі [6,12]:

- handlers/ – обробники команд та повідомлень;

- database/ – робота з Excel-файлом (запис/читання);
- utils/ – допоміжні функції: форматування, генерація відповідей;
- config.py – базові налаштування та шляхи;
- main.py – запуск бота.

Це дозволяє масштабувати проєкт без втрати керованості коду, що можна побачити у лістингу 3.1.

Лістинг 3.1 – Компонент файлу main.py

```

from aiogram import Bot, Dispatcher
from aiogram.types import Message
from aiogram.filters import CommandStart
from handlers import register_handlers
from config import BOT_TOKEN
import asyncio

bot = Bot(token=BOT_TOKEN)
dp = Dispatcher()

@dp.message(CommandStart())
async def cmd_start(message: Message):
    await message.answer("Вітаю! Це бот АвтоФонд.")

register_handlers(dp)

if __name__ == "__main__":
    asyncio.run(dp.start_polling(bot))

```

кінець лістингу 3.1

Цей код ініціалізує бота, обробляє команду /start і підключає інші обробники з модуля handlers. Запуск відбувається через асинхронний цикл asyncio. Керування ролями можна побачити у файлі access.py (ліст. 3.2).

Лістинг 3.2 – Перевірка доступу за роллю (файл utils/access.py)

```

import pandas as pd

def get_user_role(user_id: int) -> str:
    df = pd.read_excel("admins.xlsx")
    row = df[df["id"] == user_id]
    if row.empty:

```

```

    return "unauthorized"
    return row.iloc[0]["role"]

```

кінець лістингу 3.2

Функція перевіряє роль користувача за Telegram ID. Права беруться з Excel-файлу admins.xlsx, де вказано ID та роль (admin, manager) (ліст. 3.3).

Лістинг 3.3 – Додавання автомобіля в таблицю (файл database/add_car.py)

```

import pandas as pd
from openpyxl import load_workbook

def add_car_to_excel(data: list[str]):
    file = "data/autofond.xlsx"
    df = pd.read_excel(file)
    new_row = pd.DataFrame([data], columns=df.columns)
    df = pd.concat([df, new_row], ignore_index=True)
    df.to_excel(file, index=False)

```

кінець лістингу 3.3

Передбачено введення даних користувачем через кому (наприклад: «Audi A6, 2017, дизель, чорний, \$13000»). Потім ці дані обробляються та додаються в таблицю autofond.xlsx (ліст. 3.4).

Лістинг 3.4 – Завантаження фото автомобіля (файл handlers/upload_photo.py)

```

import os
from aiogram import Router, F
from aiogram.types import Message
from utils.access import get_user_role

router = Router()

@router.message(F.photo)
async def handle_photo(message: Message):
    user_role = get_user_role(message.from_user.id)
    if user_role == "unauthorized":
        return await message.answer("У вас немає доступу.")

    photo = message.photo[-1]
    file = await message.bot.get_file(photo.file_id)
    file_path = file.file_path
    file_name = f"images/{photo.file_unique_id}.jpg"

```

```
await message.bot.download_file(file_path, file_name)
```

```
await message.answer("Фото збережено.")
```

кінець лістингу 3.4

Фото зберігається у папці images під унікальним іменем. Цей шлях потім можна додати до Excel у колонку img.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

У процесі розробки Telegram-бота «АвтоФонд» було проведено поетапне тестування з метою забезпечення його стабільної та надійної роботи. Основна мета тестування – виявлення помилок на ранніх етапах, перевірка логіки обробки даних, взаємодії з Excel-файлом, фото та авторизації користувачів [4-5].

Методика тестування. Було застосовано такі методи:

- модульне тестування – перевірка окремих функцій, таких як запис даних в Excel, обробка фото, перевірка прав доступу [6,7];
- функціональне тестування – перевірка роботи кожної команди бота згідно з технічним завданням (додавання авто, пошук, фото, статистика) [4];
- тестування граничних випадків – наприклад, введення неповних даних, порожніх полів, помилкових ID, файлів без фото [11];
- інтеграційне тестування – взаємодія всіх частин системи: команда → обробник → перевірка прав → запис в Excel → відповідь користувачу [14].

Інструменти:

- Telegram для взаємодії з ботом;
- Python-інтерпретатор із виводом логів (print, logging);
- Excel-файли для перевірки правильності збережених даних;
- Aiogram у режимі debug для виводу помилок.

Виявлені недоліки та їх усунення. Під час розробки було виявлено кілька недоліків:

- некоректна обробка прав доступу: Якщо користувача не було в таблиці `admins.xlsx`, бот аварійно завершував роботу при доступі до `row.iloc[0]`. Виправлення: додано перевірку `if row.empty`: перед отриманням даних;
- неправильне збереження фото при одночасному завантаженні кількох: Через однакові назви файлів деякі фото перезаписувалися. Виправлення: ім'я фото генерується за унікальним ID кожного зображення (`file_unique_id`);
- проблеми з кодуванням Excel-файлу: Деякі символи не відображались правильно при відкритті Excel після запису (ліст. 3.5). Виправлення: встановлено кодування UTF-8 і використано `openpyxl` як engine для запису [8].

Лістинг 3.5 – Додавання автомобіля в таблицю (файл `database/add_car.py`)

```
import pandas as pd
def get_user_role(user_id: int) -> str:
    try:
        df = pd.read_excel("admins.xlsx")
        row = df[df["id"] == user_id]
        if row.empty:
            return "unauthorized"
        return row.iloc[0]["role"]
    except Exception as e:
        print(f"Помилка доступу до admins.xlsx: {e}")
        return "unauthorized"
```

кінець лістингу 3.5

Ця версія функції перевірки ролі обгорнута в `try-except`, що дозволяє уникнути краху програми при будь-яких помилках читання файлу або формату. У разі помилки повертається роль `unauthorized` [13].

У результаті проведеного тестування було усунено основні недоліки, що забезпечило стабільну роботу Telegram-бота в реальних умовах використання.

3.3 Розробка бази даних

Для зберігання інформації про автомобілі в Telegram-боті «АвтоФонд» було обрано формат бази даних у вигляді таблиці Excel. Такий вибір є обґрунтованим з огляду на простоту реалізації, зручність використання та низький поріг входження для користувачів, які не мають досвіду роботи з повноцінними системами керування базами даних (СКБД).

Для роботи боту, було створено 3 бази даних :

- нові користувачі – для запису всіх юзерів, які взаємодіяли з ботом (можна використати для статистики в майбутньому);
- адміністратори та модератори – база яка зберігає ролі юзерів;
- база автомобілів – всі автомобілі та їх характеристики (дані).

Детальний приклад був приведений на схемі нижче (рис. 3.2).

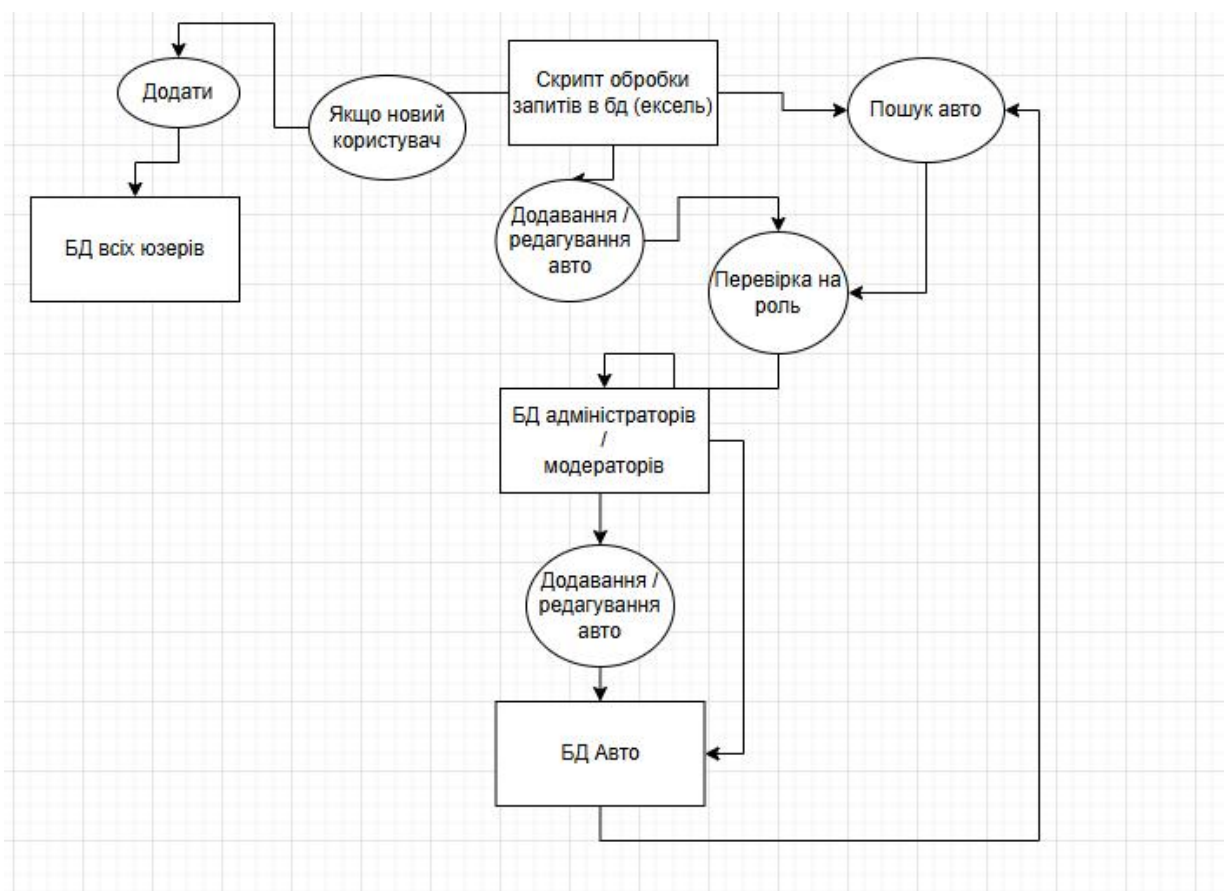


Рисунок 3.2 – Схема роботи баз даних

Причини вибору Excel:

- легкий доступ і редагування: таблицю можна легко переглянути і змінити вручну через Microsoft Excel або Google Sheets [7];
- відсутність потреби у сервері: немає потреби в налаштуванні СКБД, хостингу чи додаткових ресурсах;
- швидка реалізація: використання бібліотеки pandas дозволяє просто читати та записувати дані [15];
- прозорість зберігання: усі дані зберігаються у відкритому вигляді, що зручно при ручній перевірці (ліст. 3.6).

Лістинг 3.6 – Додавання автомобіля в таблицю (файл database/add_car.py)

```
import pandas as pd
from openpyxl import load_workbook

def save_car_data(data: dict, file_path: str = "cars.xlsx"):
    try:
        # Створення DataFrame з одного запису
        new_row = pd.DataFrame([data])

        # Якщо файл вже існує, додати новий рядок
        try:
            book = load_workbook(file_path)
            writer = pd.ExcelWriter(file_path, engine='openpyxl', mode='a',
if_sheet_exists='overlay')
            writer.book = book
            writer.sheets = {ws.title: ws for ws in book.worksheets}
            start_row = writer.sheets['Sheet1'].max_row
            new_row.to_excel(writer, index=False, header=False,
startrow=start_row)
        except FileNotFoundError:
            # Якщо файл не існує - створити новий
            new_row.to_excel(file_path, index=False)

        writer.close()
    except Exception as e:
        print(f"Помилка збереження у Excel: {e}")
```

кінець лістингу 3.6

Цей фрагмент коду дозволяє зберігати нові записи до файлу cars.xlsx. Якщо файл уже існує, новий запис додається в кінець таблиці без перезапису попередніх даних. У випадку відсутності файлу – створюється новий.

Перспективи переходу на СКБД (MySQL). Попри всі переваги Excel, такий підхід має обмеження у масштабованості та паралельному доступі до даних. У майбутньому можливою альтернативою стане використання MySQL або іншої СКБД, що дозволить:

- забезпечити багатокористувацький доступ до даних без конфліктів;
- підвищити надійність і безпеку зберігання інформації;
- виконувати складні запити та фільтрацію;
- реалізувати веб-інтерфейс або мобільний додаток з безпосереднім доступом до бази.

Впровадження MySQL потребуватиме додаткового налаштування сервера та зміни логіки взаємодії з базою даних у Telegram-боті, однак відкриває значно ширші можливості для масштабування проєкту [18].

3.4 Розробка документації для програмного продукту

Документація є невід'ємною частиною розробки будь-якого програмного продукту, оскільки вона дозволяє зрозуміти принципи його роботи, вимоги до середовища, порядок встановлення та використання. На цьому етапі було створено технічну документацію, яка включає опис середовища запуску, користувацьку інструкцію та технічні вимоги до системи.

Насамперед були визначені нефункціональні вимоги до програмного продукту, які описують умови для його стабільного та ефективного функціонування. У таблиці 3.1 наведено мінімальні та рекомендовані параметри для запуску Telegram-бота на цільовій системі.

Таблиця 3.1 – Нефункціональні вимоги

Компонент	Мінімальні вимоги	Рекомендовані параметри
ОС	Ubuntu 20.04 / Windows 10	Ubuntu 22.04 LTS / Windows 11
Процесор	2 ядра, 1.5 ГГц	4 ядра, 2.5 ГГц
Оперативна пам'ять	2 ГБ	4+ ГБ
Місце на диску	500 МБ	1+ ГБ
Python	Версія 3.9 або вище	Python 3.11
Інтернет	Постійне з'єднання	Швидке стабільне з'єднання

Інструкція з встановлення і запуску Telegram-бота [12].

1) клонування проєкту або копіювання файлів. Скопіювати вихідні файли Telegram-бота до локальної або серверної директорії. Структура проєкту має бути приблизно такою як на рисунку 3.3;

```
autofond_bot/
├─ main.py
├─ handlers.py
├─ excel_utils.py
├─ admins.xlsx
├─ cars.xlsx
└─ images/
```

Рисунок 3.3 – Структура проєкту

2) налаштування токена бота. У файлі main.py потрібно вказати токен Telegram-бота, отриманий через BotFather (рис. 3.4);

```
12
13 # Налаштовуємо логування
14 logging.basicConfig(level=logging.INFO)
15
16 # Ініціалізуємо бота та диспетчер
17 bot = Bot(token=TOKEN)
18 dp = Dispatcher()
19
```

Рисунок 3.4 – Підключення токена

3) запуск бота. Бот автоматично запуститься й почне обробляти вхідні повідомлення (рис. 3.5).

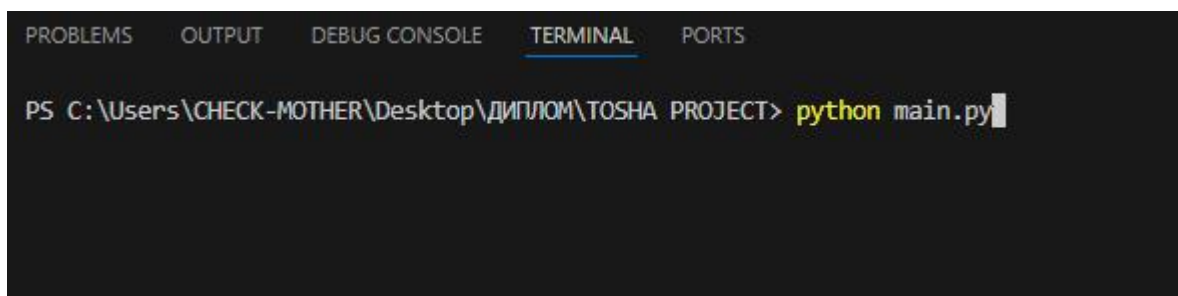


Рисунок 3.5 – Запуск бота на сервері

Довідкові матеріали для користувачів.

1. Інструкції для менеджерів та адміністраторів:

- для додавання автомобіля: ввести всі дані через кому у форматі Марка, Модель, Рік, VIN, Колір, Ціна, Нотатки;
- після цього бот запитає, чи потрібно додати фото;
- у разі позитивної відповіді потрібно надіслати лише одне повідомлення з усіма фото.

2. Формат фото:

- підтримуються зображення JPEG, PNG;
- фото зберігаються у вигляді словника зі шляхами в колонці img файлу cars.xlsx.

3. Файл адміністраторів (admins.xlsx):

- Telegram ID зберігаються у першій колонці;
- у другій колонці – роль (admin або manager).

Можливість інтеграції довідки в бота. У майбутньому планується реалізація внутрішньої довідки, яку користувач зможе викликати командою /help. Довідка буде містити:

- список доступних команд;
- інструкції щодо додавання авто;
- часті помилки й шляхи їх вирішення.

Для цього буде створено окремий обробник, наприклад як на рисунку 3.6.

```

30
31 @dp.message(Command("help"))
32 async def show_help(message: types.Message):
33     await message.answer(
34         "Довідка:\n"
35         "/start [ ] запустити бота\n"
36         "/add [ ] додати авто\n"
37         "Після введення даних, бот запитає про фото.\n"
38         "Детальна інструкція [ ] [ ] документації."
39     )
40

```

Рисунок 3.6 – Інтеграція довідки в бота

Telegram-бот «АвтоФонд» не потребує складної інсталяції та легко переноситься між машинами. Для невеликих проєктів та команд він є зручним інструментом, що не потребує розгортання на повноцінному сервері. Однак, при подальшому масштабуванні можливою є міграція на більш серйозні хостингові рішення та використання СКБД (наприклад, MySQL) для підвищення надійності зберігання даних.

1.5 Захист інформаційно-комп'ютерної системи

Захист інформації – один із найважливіших аспектів при розробці будь-яких інформаційних систем, особливо тих, що працюють із персональними або комерційними даними. Telegram, як платформа для створення чат-ботів, надає розробникам надійний і захищений канал взаємодії з користувачами, забезпечуючи високий рівень конфіденційності, шифрування та захисту даних.

Безпека платформи Telegram. Telegram був створений з урахуванням безпеки як пріоритету. Однією з головних особливостей є використання власного протоколу шифрування MTProto, який забезпечує захищену передачу даних між клієнтом (користувачем) і сервером. Для хмарних чатів Telegram застосовує схему клієнт-сервер/сервер-клієнт з шифруванням на основі 256-

бітного симетричного AES, 2048-бітного RSA та Diffie–Hellman обміну ключами [13].

Хоча боти Telegram не підтримують наскрізне шифрування (end-to-end), вони все одно працюють у межах цієї шифрованої інфраструктури, а доступ до API та webhook здійснюється через HTTPS, що запобігає перехопленню трафіку. Крім того, Telegram зберігає дані на розподілених серверах, що мінімізує ризики втрати чи несанкціонованого доступу до даних [14].

Авторизація та ідентифікація користувачів. У боті «АвтоФонд» реалізована проста, але ефективна система перевірки доступу. Кожен користувач, який взаємодіє з ботом, має унікальний Telegram ID, що не змінюється й автоматично надсилається з кожним повідомленням. Це дозволяє з високою точністю ідентифікувати користувача та порівнювати його ідентифікатор з базою прав доступу, що зберігається у файлі admins.xlsx.

Розмежування доступу за ролями – admin або manager – дозволяє контролювати, хто має право додавати нові дані, редагувати інформацію чи просто переглядати її. Таким чином, навіть якщо ботом скористається несанкціонована особа, вона не зможе виконати дії без відповідних прав.

Безпечне зберігання та обробка даних. Усі дані про автомобілі зберігаються в локальній таблиці Excel. Це забезпечує контроль над фізичним доступом до даних і виключає ризики, пов'язані з передачею інформації через мережу. Додатково, фотографії авто зберігаються локально в окремій директорії images/, і в таблиці Excel фіксується лише шлях до них. Це знижує ризик витоку зображень, оскільки немає потреби завантажувати їх на сторонні сервіси [15].

Щоб уникнути випадкового або навмисного видалення даних, таблиці можуть періодично архівуватись або синхронізуватись із хмарним сховищем із налаштованими правами доступу (наприклад, Google Drive, OneDrive з обмеженням за IP чи двофакторною автентифікацією).

Надійність Telegram-бота як рішення. Telegram надає стабільний інтерфейс API для ботів із високим рівнем доступності серверів і мінімальною

затримкою. Бот «АвтоФонд» не потребує встановлення додаткового ПЗ, оновлень чи складної інсталяції на пристрої користувача – достатньо мати Telegram. Це зменшує площину атаки, оскільки не залучає сторонніх бібліотек чи веб-доступу з браузера.

Крім того, Telegram активно бореться зі спамом і фішингом, регулярно оновлює свої протоколи захисту та проводить аудит систем безпеки. Додатково, можна обмежити взаємодію з ботом до певного кола користувачів, використовуючи white list на рівні коду або шляхом приховування бота з пошуку [18].

Потенційні загрози та шляхи їх мінімізації. Попри високий рівень захисту, Telegram-боти можуть бути вразливими до таких ризиків, як:

- 1) соціальна інженерія, коли зловмисник намагається отримати доступ через маніпуляції;
- 2) помилки у коді – наприклад, неправильна перевірка прав доступу;
- 3) несанкціонований доступ до файлової системи, де зберігаються Excel-файли або фото.

Для зменшення ризиків:

- 1) код бота має бути ретельно протестований, особливо логіка авторизації;
- 2) файли даних зберігати в директоріях з обмеженим доступом;
- 3) застосовувати шифрування резервних копій;
- 4) упровадити журналювання дій користувачів для аудиту.

У підсумку, Telegram-бот «АвтоФонд» є безпечним, зручним та функціональним рішенням для ведення обліку ввезених автомобілів. Його структура, інтеграція з Telegram, локальне зберігання даних і чітка система авторизації користувачів забезпечують високий рівень захисту, достатній для використання в умовах малого бізнесу чи внутрішніх потреб організації.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи було розроблено Telegram-бот для обліку ввезених автомобілів.

Для зберігання даних була використана база даних у форматі Excel, що забезпечує простоту у реалізації, редагуванні та аналізі інформації без необхідності налаштування складних серверів баз даних.

Бот був протестований у реальних умовах, функціонал відповідає поставленим вимогам: реалізовано авторизацію за Telegram ID, додавання даних про автомобілі, збереження фотографій, а також гнучку систему ролей для адміністраторів і менеджерів.

Описано процес встановлення, налаштування бота та запуску на локальному середовищі або сервері. Надано мінімальні системні вимоги для роботи програмного продукту.

Крім того, розроблена базова користувацька документація, яка допоможе новим користувачам швидко ознайомитися з принципами роботи бота.

У майбутньому можливе масштабування системи та перехід до використання серверної бази даних (наприклад, MySQL), що дозволить зберігати більші обсяги інформації та підвищити надійність зберігання даних.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було розроблено Telegram-бот «АвтоФонд» – сучасний інструмент для обліку ввезених автомобілів, що використовується у межах діяльності волонтерського фонду. Бот поєднує зручність, функціональність і актуальність, дозволяючи ефективно вести облік транспорту в реальному часі.

У процесі виконання кваліфікаційної роботи були досягнуті всі поставлені завдання:

- визначено актуальність проблеми обліку імпортованих автомобілів в умовах зростаючого попиту на ефективні цифрові інструменти, проаналізовано переваги автоматизації цього процесу, а також проведено дослідження існуючих Telegram-ботів таких як, «SheetManager», «AutoSeller Bot», «InventoryBot» зі схожим функціоналом з метою виявлення їхніх сильних і слабких сторін;

- сформульовано функціональні та нефункціональні вимоги до Telegram-бота «АвтоФонд», зокрема щодо швидкості роботи, зручності інтерфейсу, обмежень доступу, можливості обробки зображень та збереження даних;

- обґрунтовано вибір технологій, таких як Python, бібліотека Aiogram, Excel як проміжна база даних, а також архітектурного підходу, орієнтованого на простоту впровадження та подальшу масштабованість;

- реалізовано Telegram-бота «АвтоФонд», який підтримує авторизацію за ролями (admin/manager), введення даних про автомобіль у форматі одного повідомлення, додавання фотографій, зберігання інформації в таблиці Excel та пошук за вказаними параметрами;

- проведено тестування Telegram-бота на предмет функціональності, зручності використання, стабільності роботи та відповідності заявленим вимогам. Отримані результати підтвердили працездатність системи та її готовність до використання;

- розроблено супровідну документацію, що містить покрокові інструкції з встановлення, запуску та використання Telegram-бота, опис архітектури та основних компонентів, а також приклади типових сценаріїв роботи;

- реалізовано збереження даних у форматі таблиці Excel, що забезпечує просту інтеграцію з іншими системами та створює основу для майбутньої міграції до повноцінної реляційної бази даних, такої як PostgreSQL.

У майбутньому можливе розширення функціоналу бота, інтеграція з повноцінною базою даних (наприклад, MySQL), створення веб-інтерфейсу для зручного аналізу статистики, а також автоматичне формування звітів. Отже, Telegram-бот «АвтоФонд» є не лише готовим рішенням, але й перспективною платформою для подальшого розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ESPN. URL: <https://www.espn.com/> (дата звернення: 03.05.2025).
2. TheScore. URL: <https://www.thescore.com/> (дата звернення: 03.05.2025).
3. Google Fit. URL: <http://surl.li/ukfmz> (дата звернення: 05.05.2025).
4. Telegram Bot API. URL: <https://core.telegram.org/bots/api> (дата звернення: 09.05.2025).
5. Aiogram Documentation. URL: <https://docs.aiogram.dev/en/latest/> (дата звернення: 11.05.2025).
6. Python Official Documentation. URL: <https://docs.python.org/3/> (дата звернення: 11.05.2025).
7. Pandas Documentation. URL: <https://pandas.pydata.org/docs/> (дата звернення: 11.05.2025).
8. OpenPyXL Documentation. URL: <https://openpyxl.readthedocs.io/en/stable/> (дата звернення: 13.05.2025).
9. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/> (дата звернення: 14.05.2025).
10. Google Sheets API. URL: <https://developers.google.com/sheets/api> (дата звернення: 16.05.2025).
11. GitHub – Aiogram GitHub Repository. URL: <https://github.com/aiogram/aiogram> (дата звернення: 17.05.2025).
12. Real Python – Creating Telegram Bots with Python. URL: <https://realpython.com/python-telegram-bot/> (дата звернення: 17.05.2025).
13. Medium – How to Build a Telegram Bot with Python and Aiogram. URL: <https://medium.com/@codingfreak/building-a-telegram-bot-with-aiogram-6b2bdca5f04f> (дата звернення: 21.05.2025).
14. Stack Overflow. URL: <https://stackoverflow.com/> (дата звернення: 23.05.2025).
15. Microsoft Excel Help & Learning. URL: <https://support.microsoft.com/en-us/excel> (дата звернення: 25.05.2025).

16. InventoryBot – Telegram Example Bot. URL: <https://t.me/inventorybot> (дата звернення: 25.05.2025).

17. SheetManager Bot (Google Sheets integration). URL: https://t.me/SheetManager_bot (дата звернення: 28.05.2025).

18. GitHub – Python Telegram Bot Examples. URL: <https://github.com/python-telegram-bot/python-telegram-bot> (дата звернення: 30.05.2025)

19. AutoSeller Bot – Telegram-бот для продажу автомобілів. URL: <https://t.me/autosellerbot> (дата звернення: 02.06.2025).