

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Автоматизація взаємодії інформаційної системи інтернет-магазину з
потенційними та існуючими користувачами»

ХІМЧЕНКО ОЛЕКСІЙ СЕРГІЙОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

„___” _____ 2024 року

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Автоматизація взаємодії інформаційної системи інтернет-магазину з потенційними та існуючими користувачами»

Виконав: студент 4-го курсу, групи КН-20-1 _____

Спеціальності: 122 «Комп'ютерні науки _____

Спеціалізація: «Інформаційні управляючі системи і технології» _____

(шифр і назва напрямку підготовки, спеціальності)

Хімченко О. С.

(прізвище та ініціали)

Керівник к.т.н., доц. Гончаренко Т. А.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ, 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: «бакалавр» за ОП

Спеціальність: 122 «Комп'ютерні науки»

Спеціалізація: Інформаційні управляючі системи і технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

„___” _____ 2024 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Хімченко Олексій Сергійович

Тема роботи: Автоматизація взаємодії інформаційної системи інтернет-магазину з потенційними та існуючими користувачами

затверджена наказом ректора КНУБА № _____ від « » лютого 2024 р.

2. Керівник роботи: Гончаренко Тетяна Андріївна, к.т.н, доцент
кафедри інформаційних технологій

3. Строк подання студентом роботи до захисту: _____

4. Зміст пояснювальної записки за розділами:

P.1. Аналіз предметної області та постановка задачі

P.2. Проектування клієнт-серверного додатку

P.3. Реалізація проекту. Опис програмного продукту

5. Інформаційні слайди:

S.1. _____

S.2. _____

S.3. _____

S.4. _____

S.5. _____

S.6. _____

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
Р. 1. Аналіз предметної області та постановка задачі	Лютий 2024 р.
Р. 2. Проектування клієнт-серверного додатку	Березень 2024 р.
Р. 3. Реалізація проекту. Опис програмного продукту	Квітень 2024 р.
Остаточне оформлення роботи	Травень 2024 р.
Направлення роботи на рецензування	Червень 2024 р.
Попередній захист роботи на кафедрі	Червень 2024 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Прийом програмного продукту			

8. Дата видачі завдання: грудня 2023 р.

Керівник

Гончаренко Т .А.

 (підпис) (прізвище та ініціали)

Бакалавр

Хімченко О. С.

 (підпис) (прізвище та ініціали)

АНОТАЦІЯ

Хімченко О. С. Автоматизація взаємодії інформаційної системи інтернет-магазину з потенційними та існуючими користувачами.

Атестаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки», спеціалізація: «Інформаційні управляючі системи і технології». – Київський національний університет будівництва та архітектури. – Київ, 2024.

Робота присвячена можливості автоматизації взаємодії інтернет-магазину з користувачами, пропонуючи товари та оформлюючи замовлення. Інструментом реалізації розробленої системи є бібліотеки та фреймворки мови програмування Python.

Ключові слова: інтернет-магазин, автоматизація, API, aiogram, Python.

SUMMARY

Khimchenko O. S. Automation of interaction of the information system of an online store with potential and existing users.

Bachelor's thesis in the specialty: 122 "Computer Science", specialization: "Information managing systems and technologies". - Kyiv National University of Construction and Architecture - Kyiv, 2024.

This thesis is devoted to the possibility of automating the interaction of an online store with users, offering goods and placing orders. The tool for implementing the system under development is the libraries and frameworks of the Python programming language.

Keywords: online store, automation, API, aiogram, Python.

ЗМІСТ

Вступ	7
Перелік умовних позначень	8
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	9
1.1 Опис предметної області	9
1.2 Аналіз об'єкта дослідження	12
1.3 Опис предмету дослідження	16
1.3.1 Складові та безпека доступу Telegram	16
1.3.2 Вибір БД і їх безпека	18
1.4 Аналіз актуальності	22
1.5 Стан вже існуючих рішень	24
1.6 Визначення цілей дослідження та постановка задачі	25
2. ПРОЕКТУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ	26
2.1 Telegram як платформа для розміщення бота	26
2.2 Мова програмування Python	28
2.3 БД Sqlite	32
2.4 Бібліотеки та фреймворки Python	34
2.5 Діаграма послідовності	36
3. РЕАЛІЗАЦІЯ ПРОЄКТУ. ОПИС ПРОГРАМНОГО ПРОДУКТУ	37
3.1 Початок взаємодії з інформаційною системою	37
3.2 Реалізація функціоналу бота для адміністратора. FSM	40
3.2.1 Додавання товару до БД	42
3.2.2 Видалення товару з БД	50
3.3 Взаємодія користувача з ботом	52
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	54

Вступ

З плином часу та розвитком технологій, традиційна торгівля стала можливою і в Інтернеті. Автоматизація взаємодії інформаційної системи інтернет-магазину з користувачами стала важливим фактором для досягнення конкурентних переваг та забезпечення ефективного обслуговування клієнтів.

У сучасному світі важко уявити життя без покупок товарів, замовлення продуктів або консультації з будь-яких питань онлайн. Швидкий розвиток технологій інформаційних систем та зростаюча конкуренція на ринку змушують компанії активно використовувати інновації для залучення та утримання клієнтів, тому виникає автоматизована взаємодія магазину з відвідувачем задля полегшення та покращення досвіду користування клієнтом певним сервісом.

Розроблювана в даній роботі система покликана автоматизувати пошук товару клієнтом, скоротити час відповіді та спростити роботу працівників.

Для реалізації головної мети буде впроваджено інтернет-магазин у месенджер Telegram за допомогою API-ключа, шляхом створення чат-боту, який взаємодіятиме з користувачами.

Розробка системи буде відбуватись за допомогою мови програмування Python у середовищі PyCharm.

Перелік умовних позначень

API – application programming interface

FSM – finite state machine

ІС – інформаційна система

БД – база даних

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

Опис предметної області

Дана дипломна робота зосереджена на розробці автоматизованої взаємодії інформаційної системи інтернет-магазину з потенційними та існуючими користувачами. Метою цього проєкту є підвищення маркетингу, продажів та покращення обслуговування клієнтів в інтернет-магазині за рахунок автоматизації їх взаємодії.

Предметною областю цієї дипломної роботи є електронна комерція (e-commerce), завдяки якій спрощується процес придбання товару для покупця та процес продажу для продавця або магазину.

Інтернет-магазини стали необхідною складовою сучасної торгівлі, оскільки дають можливість отримати консультацію, обрати та замовити товар з будь-якого місця, де є доступ до інтернету. Автоматизована взаємодія забезпечує ефективну роботу, якісне та швидке обслуговування.

Основними видами електронної комерції є:

- C2C (Consumer-to-Consumer). Схеми «споживач-до-споживача». Прикладом є торгові майданчики OLX, eBay, і подібні до них, в яких людина, навіть не будучи підприємцем має можливість виставляти на продаж якусь річ.
- B2B (Business-to-Business). Схеми «бізнес-до-бізнесу». Характеризується вона наданням послуг або продажем товару однією компанією іншій, а не роздрібному споживачеві. Прикладом є співпраця компаній Samsung та Apple, де Samsung виробляє дисплеї для продукції компанії Apple.
- B2C (Business-to-Consumer). Схеми «бізнес-споживач». Її основою є електронна роздрібна торгівля, де компанія надає свої товари, послуги та сервіси напряму кінцевому споживачеві, тобто будь-яка роздрібна угода між юридичними та фізичними особами. Прикладами можуть бути сайти IKEA і Nike, які пропонують продукцію безпосередньо від виробника.
- C2B (Consumer-to-Business). Схеми «споживач-бізнес». Дана схема є повною протилежністю B2C, тому що в цьому випадку споживачі

продають товари або послуги бізнесу, прикладом чого може слугувати надання фахівцями послуг компаніям на фріланс-платформах, або виготовлення продукції та її продаж інтернет-магазину.

- B2A (Business-to-Administration). Схеми «бізнес-адміністрація». У цьому випадку охоплюються онлайн-транзакції, що проводять компанії та державні структури. Останнім часом є прагнення урядів розвинених країн і керівництва міжнародних організацій активно використовувати Інтернет для проведення закупівель товарів і послуг. Наприклад, електронні тендери чи системи закупівель для державних установ.
- C2A (Consumer-to-Administration). Схеми «споживач-адміністрація». На даний момент є не дуже популярним типом електронної комерції, який включає транзакції між споживачами та органами державного управління. Прикладом є запис на прийом до лікаря, оплата медичних послуг, формування податкових декларацій, дистанційне навчання тощо.

Дана дипломна робота розробляється за принципом C2C, де споживач продає товари безпосередньо іншому за допомогою-інтернет магазину, хоча сам їх не виробляє.

Розглянемо переваги та недоліки електронної комерції. До переваг можна віднести:

- можливість обирати товар та замовляти його в будь-який час доби для покупця і можливість продавати товар для продавця;
- бізнес-процеси стають швидшими, ефективнішими та простішими;
- можливість порівняння товарів в різних магазинах;
- можна залишити свій коментар або побачити відгуки про товар інших покупців перш, ніж прийняти рішення про покупку;
- завжди присутня докладна інформація та опис про товар або послугу;

- покращення якості обслуговування клієнтів.

Недоліками електронної комерції є:

- мало довіри до нових інтернет-магазинів з боку користувачів;
- відсутність знань про роботу в мережі;
- недосконала законодавча база;
- неможливість замовником прийти і самостійно забрати посилку;
- відповідальних віддалених співробітників важко знайти і навчити;
- велика конкуренція в популярних напрямках онлайн-торгівлі, що супроводжується витратами на маркетинг, SMM просування та ін.

З цього випливає, що тут присутні як плюси, так і мінуси, але з подальшим розвитком цього виду комерції, буде більша обізнаність людей у цьому, кращі умови для створення такого бізнесу та підвищення рівня попиту.

Аналіз об'єкта дослідження

Інтернет-магазин – одна з основних ніш в E-commerce, де виробник або ж продавець, що не є виробником, може реалізувати товар кінцевому споживачеві.

Перші інтернет-магазини з'явилися у 90-ті роки ХХ століття, коли Інтернет тільки почав ставати загальнодоступним.

З 1994 року стали з'являтися перші інтернет-магазини, в яких замовлення оформлялися по телефону і підтверджувалися електронною поштою. Офіційно перший інтернет-магазин був запущений в 1994 році компанією NetMarket.

Існує кілька основних видів створення інтернет-магазинів, які відрізняються за технічними особливостями, складністю реалізації та вартістю. Ось деякі з найпоширеніших видів:

- 1) ручна розробка з нуля (цей підхід вимагає розробки інтернет-магазину вручну з використанням програмування і веб-розробки). Цей метод може бути дуже часо- та ресурсомістким, але дає повний контроль над функціональністю та дизайном;

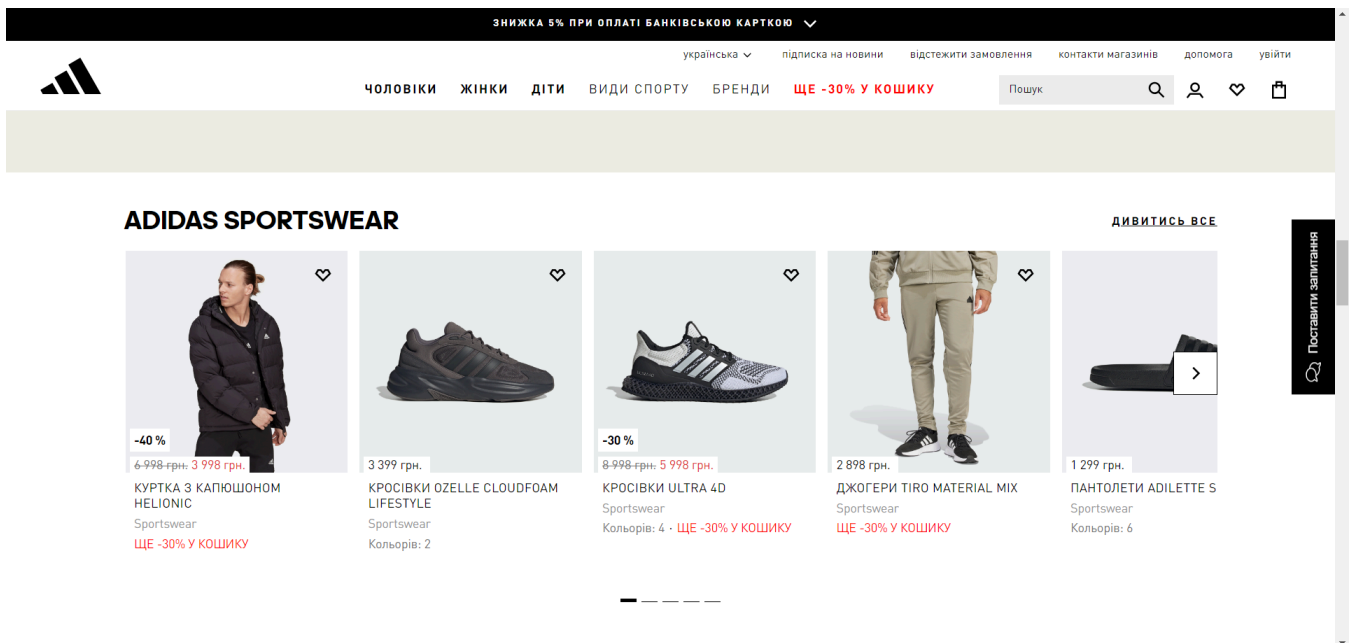


Рисунок 1.1 Сайт Adidas

- 2) використання готових платформ e-commerce (використання готових рішень, таких як Shopify, Magento, OpenCart тощо). Ці платформи зазвичай

мають шаблони для дизайну, готові функції оплати та доставки, а також можливості розширення за допомогою плагінів або додаткових модулів;

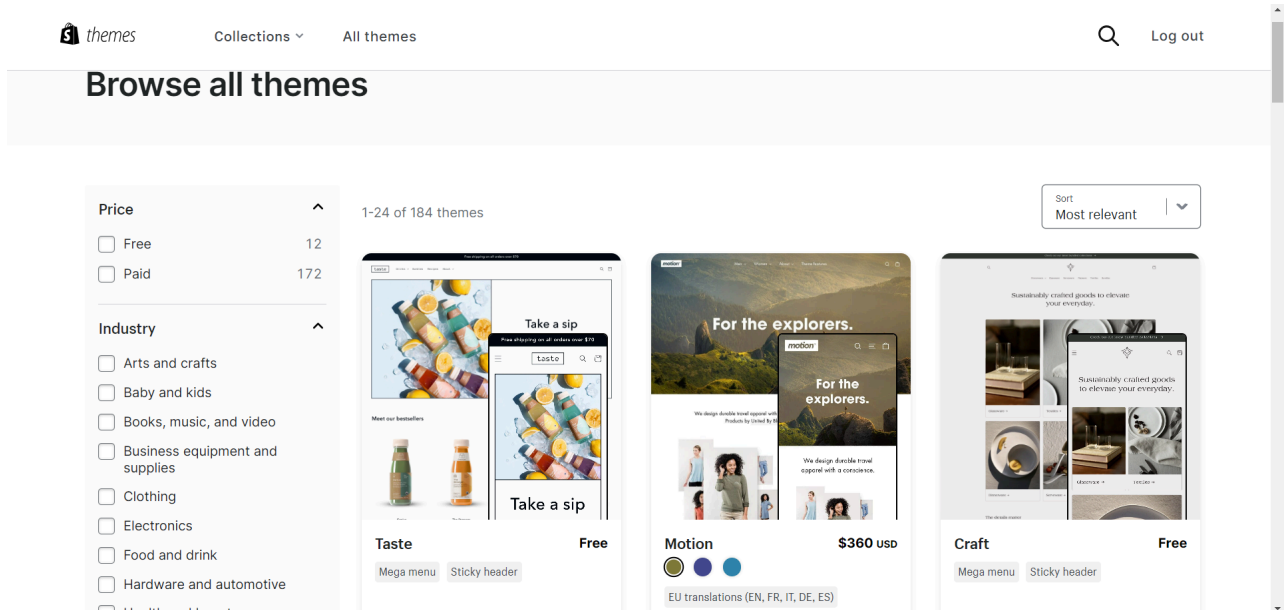


Рисунок 1.2 Сайт Shopify

3) Dropshipping (використання моделі бізнесу dropshipping, де інтернет-магазин продає товари, але не має їх на складі). Замовлення передаються постачальникам, які відправляють товари безпосередньо клієнтам.

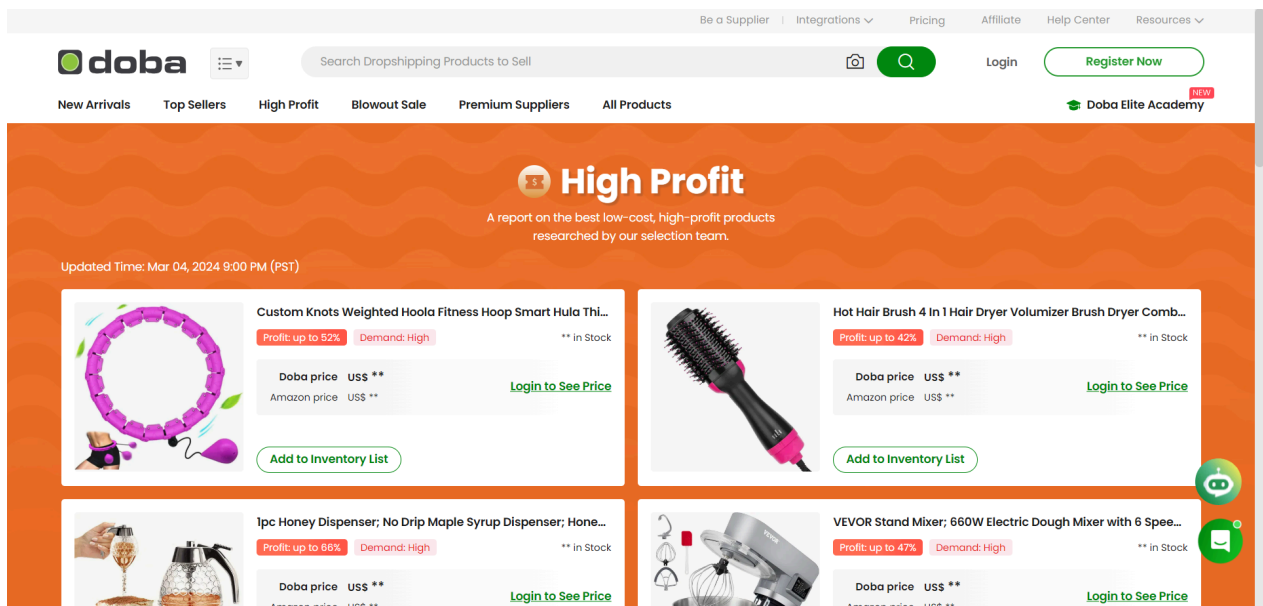


Рисунок 1.3 Сайт Doba

- 4) Використання маркетплейсів (реалізація інтернет-магазину на базі маркетплейсу, такого як Amazon, eBay, Etsy тощо). У цьому випадку продавці реєструються на маркетплейсі та продають свої товари через нього, використовуючи його інфраструктуру та аудиторію.

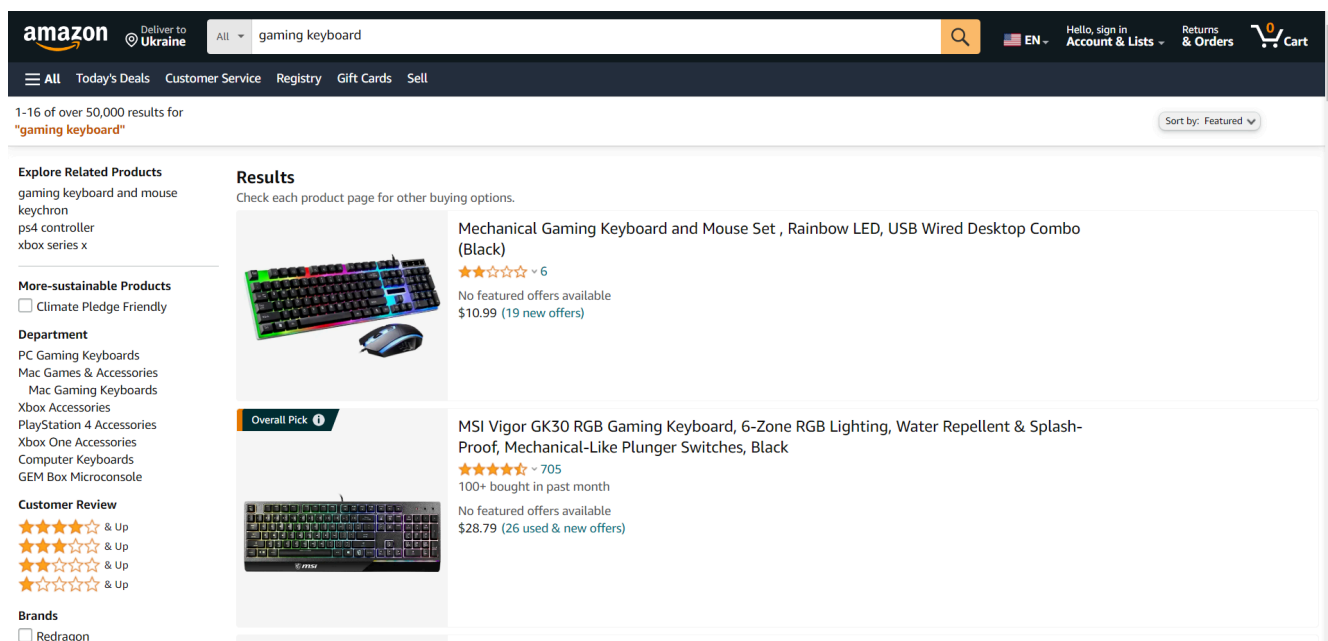


Рисунок 1.4 Сайт Amazon

Приклад покрокової взаємодії користувача з онлайн-магазином:

1. Клієнт обирає товар в каталозі онлайн з пристрою.
2. Заявка з сервера надходить адміністратору, або в автоматизовану систему обробки замовлень.
3. Інформацію про наявність товару на складі звіряють з базою даних або відправляється запит постачальнику про терміни доставки продукції. Потім інформація передається покупцеві.
4. Клієнт оплачує покупку через електронну платіжну систему.
5. Покупець отримує повідомлення про успішно здійснену операцію.
6. Заявка передається на склад, де готують посилку покупцеві.
7. Працює логістика, замовлення вирушає до клієнта.
8. Замовник отримує лист на електронну пошту, або СМС про те, що товар в дорозі.

9. Покупець отримує замовлення, слідує повідомлення продавцю на закриття угоди з комерційного та бухгалтерського обліку.
10. Замовник пише відгук. Продавець реагує (відповідає).
11. Ведеться і показується статистика відгуків/подяк.

Опис предмету дослідження

Складові та безпека доступу Telegram

Telegram – месенджер, що здобув популярність завдяки своїй простоті, зручності, безпеці та швидкості роботи. Наразі він налічує більше 1 млрд користувачів, що робить його однією з найпопулярніших платформ для спілкування.

Чат-бот – це автоматизована програма, яка може спілкуватися з користувачами та виконувати різні завдання, наприклад, приймати замовлення, надавати інформацію, відповідати на запитання клієнтів та допомагати їм.

Чат-бота саме в месенджері Telegram було обрано через його велику аудиторію, надійність та доступність, оскільки він підтримується на всіх пристроях (телефон, планшет, комп'ютер), і безкоштовність (безкоштовне створення та використання бота в Telegram).

API – це набір готових класів, процедур, функцій, структур і констант, що надаються додатком (бібліотекою, сервісом) для використання в зовнішніх програмних продуктах.

API-ключ Telegram – це унікальний ідентифікатор, який використовується для автентифікації та авторизації доступу до Telegram API. Telegram API - це набір програмних інтерфейсів, які дозволяють розробникам створювати програми та сервіси, які взаємодіють з Telegram. Телеграм API використовується розробниками для створення ботів, аплікацій та інших сервісів, які взаємодіють з месенджером Telegram.

Безпека доступу до бота в Telegram є важливим аспектом, оскільки вона стосується як захисту даних користувачів, так і захисту самого бота від несанкціонованого доступу або зловмисних дій. Ось кілька рекомендацій для забезпечення безпеки доступу до бота в Telegram:

1. Безпечне зберігання токенів

Зберігання токена бота в безпечному місці та уникання його публікації.

Токен повинен бути збережений у змінних середовища або в захищених конфігураційних файлах.

2. Використання двофакторної автентифікації (2FA)

Використання двофакторної автентифікації для акаунту, який створює і керує ботом. Це додасть додатковий рівень безпеки.

3. Логування і моніторинг

Зберігання логування активності бота та регулярний моніторинг їх для виявлення підозрілої активності.

4. Оновлення та патчі

Регулярне оновлення серверу, бібліотеки та інших компонентів системи для захисту від відомих вразливостей.

5. Обмеження прав доступу

Забезпечення того, щоб ваш бот мав мінімально необхідні права для виконання своїх функцій. Не рекомендовано надавати зайві дозволи.

6. Фільтрація вхідних повідомлень

Реалізація фільтрації вхідних повідомлень для уникнення зловмисного коду або спаму. Використання регулярних виразів або інших методів для перевірки вхідних даних.

7. Використання чатів

Уникання використання ботів у чатах з високим рівнем ризику, особливо тих, де вони можуть бути використані для фішингових атак або інших шахрайських дій.

Щоб отримати унікальний API-ключ бота для можливості реалізації програми, потрібно звернутись до головного бота, який має назву BotFather, створити нового бота з ім'ям і отримати ідентифікатор. Нижче наведено приклад взаємодії користувача з BotFather:

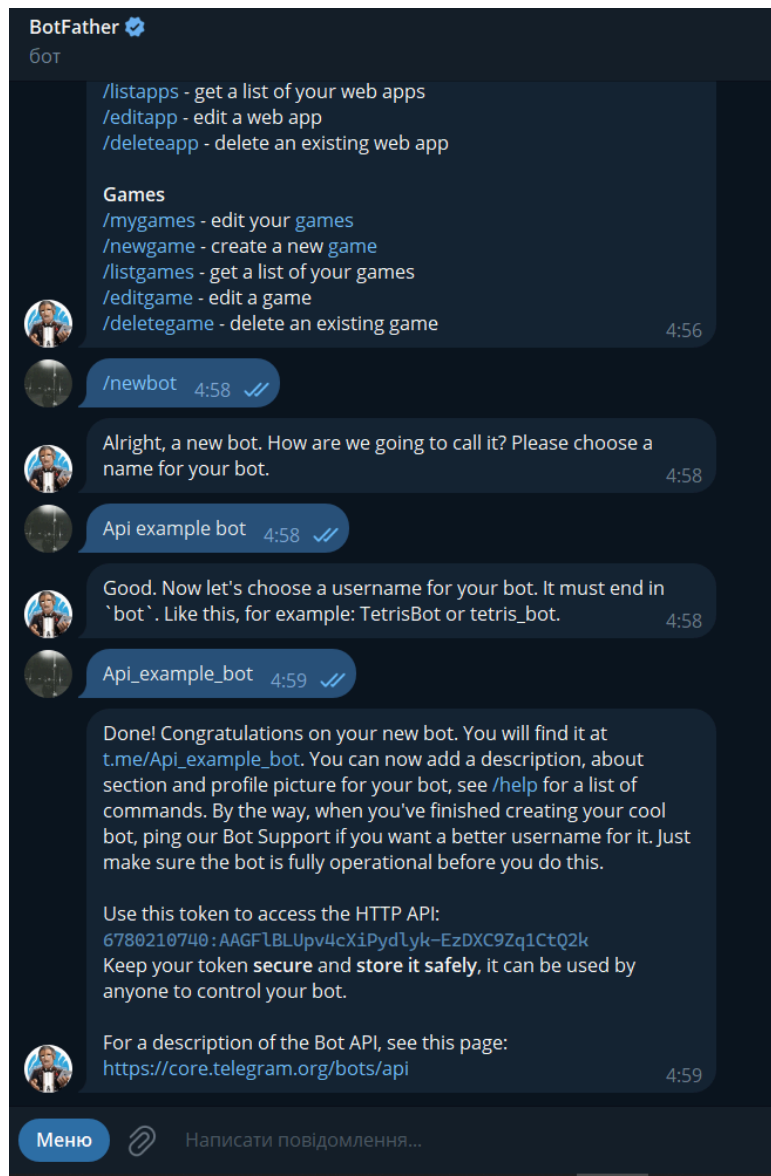


Рисунок 1.5 Отримання API токена в BotFather

Вибір БД і їх безпека

База даних (БД) – це організована структура, призначена для зберігання, зміни й обробки взаємопов’язаної інформації, переважно великих обсягів. Бази даних активно використовують для динамічних сайтів зі значними обсягами даних – часто це інтернет-магазини, портали, корпоративні сайти.

БД виконує функцію зберігання інформації про товар, такої як ціна, кількість, опис. В даній роботі використовуватиметься БД aiosqlite, яка є асинхронною, оскільки вся програма написана за допомогою асинхронного фреймворку aiogram. За допомогою цих інструментів бот буде більш ефективним у використанні ресурсів і зможе обробляти більше запитів одночасно. База даних aiosqlite дозволяє

виконувати операції з базою даних асинхронно, що покращує продуктивність вашого бота та дозволяє легко взаємодіяти з базою даних під час обробки багатьох запитів одночасно.

Безпека баз даних – це комплекс заходів, які використовуються для захисту систем управління базами даних від кібератак і несанкціонованого використання, а також для створення і підтримки їх конфіденційності, цілісності та доступності. Програми безпеки баз даних призначені для захисту від несанкціонованого використання, пошкодження і вторгнення даних в бази даних, всієї системи управління даними і кожного додатка.

Захист безпеки баз даних включає в себе: дані в базі даних; систему управління базами даних всі пов'язані з нею додатки; фізичний та/або віртуальний сервер бази даних та апаратне забезпечення, що лежить в його основі; обчислювальна та/або мережева інфраструктура, яка використовується для доступу до бази даних.

Безпека баз даних – це складний і об'ємний проект, який включає в себе всі аспекти технологій і практик інформаційної безпеки. Доступність і корисність бази даних можуть стати вразливими для кібератак.

SQL-ін'єкція або SQLi – вразливість, яка дозволяє атакуючому використовувати фрагмент шкідливого коду мовою структурованих запитів (SQL) для маніпулювання базою даних та отримання доступу до потенційно цінної інформації. Атаки на основі таких вразливостей – одні з найпоширеніших і найнебезпечніших: вони можуть бути націлені на будь-який веб-додаток або веб-сайт, які взаємодіють з базою даних SQL (а переважна більшість баз даних реалізована саме на SQL).

Успішно проведена атака з SQL-ін'єкцією може взагалі себе не проявляти. Проте іноді можна помітити такі симптоми:

- отримання надлишкового числа запитів за короткий проміжок часу.

Наприклад, масовий потік електронних листів від форми зворотного зв'язку веб-сайту.

- рекламні блоки, які перенаправляють користувача на підозрілі веб-сайти.
- дивні спливаючі вікна та повідомлення про помилки.

Залежно від способу отримання доступу до даних бекенд-сервера та потенційних масштабів шкоди SQL-ін'єкції можна розділити на три категорії:

1) Внутрішньосмугова атака (In-band SQLi)

Це найпростіший вид атаки для зловмисників, тому що для реалізації атаки та збору результатів використовується той самий канал зв'язку. Цей тип SQLi-атак поділяють на два підвиди:

- атака на основі помилок (Error-based SQLi). За такої атаки дії зловмисника призводять до того, що база даних генерує повідомлення про помилку. На основі отриманих повідомлень про помилки зловмисник намагається сформулювати уявлення про інфраструктуру бази даних.
- атака з урахуванням об'єднання (Union-based SQLi). Атакуючий отримує необхідні дані шляхом об'єднання кількох інструкцій SELECT в єдину відповідь HTTP за допомогою SQL-оператора UNION.

2) Інференційна атака (Inferential SQLi, також відома як «сліпа SQL-ін'єкція»)

При таких атаках зловмисники вивчають відповіді та поведінку сервера після надсилання наборів даних, щоб дізнатися більше про структуру бази даних. При цьому жодні записи з бази даних веб-сайту не передаються зловмиснику, і він не бачить їх у тому ж каналі зв'язку, як у випадку внутрішньосмугової атаки (це і пояснюється назва «сліпа SQL-ін'єкція»). Такі атаки поділяють на два підвиди:

- сліпа атака, заснована на часі (Time-based SQLi). Атакуючі направляють SQL-запит до бази даних, змушуючи її зробити затримку кілька секунд, як підтвердить чи спростує отриманий запит.

- бульова сліпа атака (Boolean SQLi). Атакуючі роблять SQL-запит до бази даних, очікуючи отримати результат у вигляді ствердної чи негативної відповіді.

3) Позасмугова атака (Out-of-band SQLi)

Така атака відбувається у двох випадках:

- коли атакуючі не можуть провести атаку та зібрати дані через один і той же канал зв'язку;
- коли сервер працює надто повільно чи нестабільно, щоб досягти потрібного результату.

Успішна SQLi-атака може завдати серйозної шкоди бізнесу. SQL-ін'єкція може призвести до таких наслідків:

- розкриття конфіденційних даних. Атакуючі можуть отримати конфіденційну інформацію, що зберігається на SQL-сервері;
- компрометація цілісності даних. Зловмисники можуть редагувати або видалити інформацію у системі;
- порушення приватності користувачів. Залежно від того, які дані зберігаються на SQL-сервері, атака може призвести до розкриття конфіденційних даних користувача – адрес, номерів телефонів і відомостей банківських карт;
- отримання зловмисниками адміністративного доступу до системи. Якщо у користувача бази даних є привілеї адміністратора, за допомогою шкідливого коду атакуючий може отримати доступ до системи;
- отримання зловмисниками загальних прав доступу до системи. Якщо для перевірки імен користувачів та паролів застосовуються занадто прості SQL-команди, атакуючий зможе отримати доступ до вашої системи, навіть не маючи облікових даних користувача. Після цього зловмисник зможе

дістатися конфіденційної інформації та змінити її, створивши великі проблеми для бізнесу.

Аналіз актуальності

З усе більшим використанням інформаційних систем у житті людей, а саме у сферах торгівлі, маркетингу та обслуговування підвищується попит, а слідом і рівень конкуренції. Тому задля ефективної взаємодії з потенційними та існуючими користувачами потрібне впровадження нових технологій, автоматизація процесів і аналіз даних.

Згідно зі статистикою, у 2019 році частка інтернет-торгівлі у світовому товарообороті роздрібних продажів складала 14%, у 2020 році вона оцінювалася більше ніж в 17%, а вже у 2021 частка інтернет-комерції склала приблизно 19.5%.

Серед причин стрімкого зростання популярності онлайн-магазинів варто згадати:

- високий рівень організації. В умовах активної конкуренції сучасним віртуальним магазинам доводиться докладати зусилля, щоб утриматися на плаву. Тому сайти торгових точок характеризуються простим і доступним інтерфейсом, наявністю зручних систем навігації та пошуку, великою кількістю корисної інформації;
- підвищений рівень комфорту. Робити покупки, не виходячи з дому – давня мрія затятих супротивників тривалого шопінгу в традиційному його поданні. Замовити будь-який товар в інтернет-магазині можна всього в кілька кліків, при цьому практично не відволікаючись від важливих справ;
- можливість економії часу. Як уже згадувалося вище, багато користувачів люблять електронні торгові майданчики саме завдяки економії вільного часу. Як показує практика, кілька годин, які зазвичай йдуть на забіг по магазинах, можна витратити з більшою користю;
- індивідуальний підхід до клієнтів, консультації та інформаційне обслуговування. Більшість поважаючих себе інтернет-магазинів забезпечує злагоджений зворотній зв'язок зі споживачами. Здійснюється він через спеціальні форми на сайтах, гарячі лінії або ж за допомогою обміну електронними повідомленнями;

- можливість порівняння вподобаних позицій і цін на них. Практично у всіх сучасних віртуальних магазинах використовуються системи фільтрів, які й дозволяють сортувати товари за певними параметрами, виключати зайві варіанти та проводити порівняння відразу декількох позицій каталогу.

Стан вже існуючих рішень

На даний момент існують боти в Telegram від різних інтернет-магазинів, таких як Rozetka, Allo, Lamoda та ін.

Ці боти дозволяють користувачам:

- переглядати каталог товарів;
- шукати товари за назвою, категорією, ціною;
- отримувати інформацію про товари;
- робити замовлення;
- оплачувати замовлення;
- відстежувати замовлення;
- отримувати підтримку.

Дані рішення є функціонуючими та зручними, але з постійним покращенням функціоналу самої платформи, вдосконаленням можливостей мов програмування для створення програми та обробки взаємодії користувача з нею, можливі оновлення та модернізації.

При виборі телеграм-бота інтернет-магазину слід врахувати такі фактори:

- якість бота: Бот повинен бути зручним у використанні, мати чіткий інтерфейс і швидко відповідати на запити;
- функціональність бота: Бот повинен пропонувати всі необхідні вам функції, такі як можливість переглядати каталог товарів, робити замовлення та отримувати підтримку;

- асортимент товарів: Бот повинен пропонувати широкий асортимент товарів, які вас цікавлять;
- ціни: Бот повинен пропонувати конкурентні ціни на товари.

Визначення цілей дослідження та постановка задачі

Метою досліджень в роботі є створення автоматизованої взаємодії інформаційної системи інтернет-магазину з потенційними та існуючими користувачами.

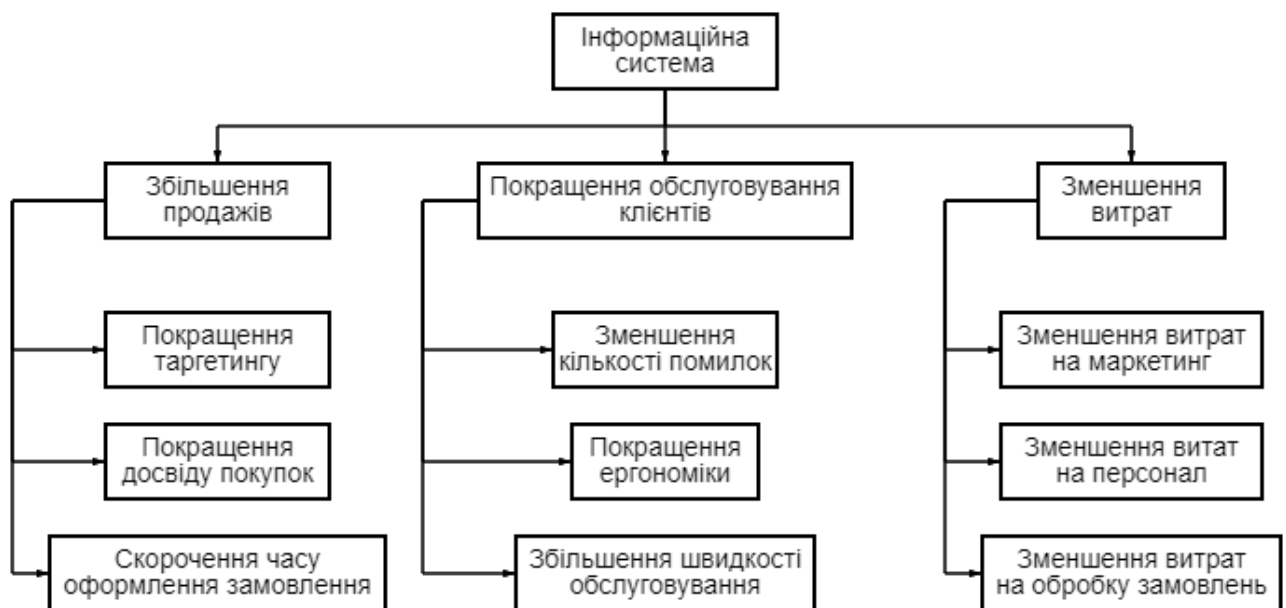


Рис. 1.6 Дерево цілей автоматизованої інформаційної системи інтернет-магазину

Реалізація мети здійснюється вирішенням наступних основних завдань:

- створення Telegram бота;
- створення бази даних;

– підключення програми до бази даних.

Об'єктом дослідження роботи є інтернет-магазини, їх створення та використання.

Предметом дослідження є методи створення бота для взаємодії користувача з інтернет-магазином за допомогою мови програмування Python та його фреймворків і бібліотек.

ПРОЄКТУВАННЯ КЛІЄНТ-СЕРВЕРНОГО ДОДАТКУ

Telegram як платформа для розміщення бота

Ведення інтернет-магазину шляхом створення бота в месенджері має такі переваги як:

1. Зручність для користувачів

- **Доступність:** Telegram є популярною платформою з великою кількістю користувачів у всьому світі. Встановлений на багатьох мобільних пристроях, він забезпечує легкий доступ до магазину без необхідності встановлювати додаткові додатки.
- **Простота використання:** Інтерфейс бота може бути інтуїтивно зрозумілим і зручним для користувачів, що дозволяє їм швидко знаходити товари, робити замовлення та отримувати підтримку.

2. Автоматизація процесів

- **Автоматичні відповіді:** Бот може автоматично відповідати на запити користувачів, що зменшує навантаження на службу підтримки та покращує користувацький досвід.
- **Обробка замовлень:** Бот може автоматично приймати і обробляти замовлення, оновлювати статуси замовлень і надсилати користувачам повідомлення про зміни в реальному часі.

3. Ефективність і економія

- **Зниження витрат:** Використання бота може знизити витрати на обслуговування клієнтів та управління замовленнями, оскільки багато процесів автоматизуються.
- **Швидка розробка:** Розробка бота для Telegram може бути швидшою та дешевшою порівняно з розробкою окремого мобільного додатку чи веб-сайту.
- **Безкоштовне розміщення бота в месенджері,** що є чудовою перевагою зменшуючи витрати.

4. Підвищення залученості користувачів

- Push-сповіщення: Бот може надсилати користувачам миттєві повідомлення про акції, нові товари чи інші важливі події, що допомагає підтримувати високий рівень залученості.
- Інтерактивність: Інтерактивні можливості Telegram, такі як кнопки, опитування та інші елементи інтерфейсу, можуть зробити взаємодію з ботом цікавішою для користувачів.

5. Інтеграція з іншими системами

- Платіжні системи: Бот може бути інтегрований з різними платіжними системами, що дозволяє користувачам здійснювати оплату безпосередньо через Telegram.
- CRM та ERP системи: Інтеграція з CRM та ERP системами дозволяє автоматично синхронізувати дані про замовлення та клієнтів, що спрощує управління бізнесом.

6. Безпека та конфіденційність

- Безпечні комунікації: Telegram забезпечує високий рівень безпеки і шифрування повідомлень, що важливо для захисту особистих даних користувачів.
- Контроль доступу: Бот може бути налаштований так, щоб обмежити доступ до певних функцій або даних тільки для авторизованих користувачів.

7. Масштабованість та адаптивність

- Легка масштабованість: Систему можна легко масштабувати в залежності від зростання бізнесу, додавати нові функції та інтеграції без значних витрат.
- Адаптивність до змін: Оскільки бот є програмним рішенням, його можна швидко адаптувати до змін у бізнес-процесах або до нових вимог ринку.

Мова програмування Python

Python – це високорівнева, інтерпретована мова програмування з відкритим вихідним кодом, яка широко використовується в галузі програмування. Особливістю Python є простота та лаконічність синтаксису, що робить її привабливою для початківців, а також широкі можливості для розробки великих та складних проектів.

Python підтримує об'єктно-орієнтований та функціональний підходи програмування, що дозволяє програмістам розробляти програми з використанням різних парадигм. Python має вбудовану підтримку для роботи з рядками, списками, словниками, кортежами, множинами та іншими типами даних. Python також має багато стандартних бібліотек, що значно полегшують розробку програм.

Ще однією особливістю Python є його платформонезалежність. Python може працювати на різних операційних системах, таких як Windows, MacOS та Linux. Python має велику спільноту користувачів та розробників, що сприяє розвитку нових ідей та розширенню можливостей мови.

Python широко використовується для розробки веб-додатків, наукових досліджень, обробки даних, машинного навчання, розробки ігор та багатьох інших задач. Багато відомих компаній, таких як Google, Facebook, Dropbox, NASA та інші, використовують Python у своїх проектах.

Переваги Python

1. Легкість вивчення та використання:

- Простий синтаксис: Python має простий та зрозумілий синтаксис, який нагадує англійську мову, що робить його легким для вивчення навіть для новачків.
- Читабельність коду: Завдяки ясному та лаконічному синтаксису, код на Python легко читати і розуміти, що полегшує підтримку і розробку.

2. Велика стандартна бібліотека:

- Широкий набір модулів: Python має багату стандартну бібліотеку, яка включає модулі для роботи з інтернетом, маніпуляції з файлами, обробки тексту, виконання математичних обчислень і багато іншого.
- Вбудовані утиліти: Бібліотека Python пропонує багато вбудованих утиліт, які дозволяють вирішувати різні задачі без необхідності встановлення додаткового програмного забезпечення.

3. Крос-платформність:

- Портативність: Python є крос-платформною мовою, що означає, що код, написаний на Python, може виконуватися на різних операційних системах, таких як Windows, macOS, Linux тощо.

4. Розширюваність та інтеграція:

- Сумісність з іншими мовами: Python можна інтегрувати з мовами C, C++, Java, та іншими, що дозволяє використовувати Python для створення додатків на різних платформах.
- Велика кількість бібліотек та фреймворків: Python має багатий набір сторонніх бібліотек і фреймворків, які допомагають вирішувати специфічні задачі, такі як веб-розробка (Django, Flask), наукові обчислення (NumPy, SciPy), машинне навчання (TensorFlow, scikit-learn) і багато інших.

5. Спільнота та підтримка:

- Велика та активна спільнота: Python має велику і активну спільноту розробників, яка постійно створює нові інструменти, бібліотеки та ресурси для навчання.

- Документація та ресурси: Є багато офіційної та неофіційної документації, підручників, курсів і форумів, де можна знайти допомогу та поради щодо вирішення різних проблем.

Застосування Python

1. Веб-розробка:

- Фреймворки: Django, Flask, Pyramid - це популярні веб-фреймворки на Python, які дозволяють швидко і ефективно створювати веб-додатки та сервіси.
- Скрипти та автоматика: Python часто використовується для автоматизації рутинних задач, управління веб-серверами і написання скриптів для веб-скрапінгу.

2. Наука про дані та аналітика:

- Бібліотеки: NumPy, pandas, Matplotlib, Seaborn - це популярні бібліотеки для аналізу даних, візуалізації та наукових обчислень.
- Машинне навчання: Scikit-learn, TensorFlow, Keras - це фреймворки та бібліотеки, які широко використовуються для створення моделей машинного навчання та штучного інтелекту.

3. Автоматизація та скрипти:

- Системне адміністрування: Python часто використовується для написання скриптів, які автоматизують завдання системного адміністрування, наприклад, управління файлами, налаштування мережі та моніторинг системи.

- Автоматизація задач: Завдяки своїй простоті та універсальності, Python підходить для автоматизації широкого спектру завдань, від рутинних офісних процесів до складних наукових обчислень.

4. Розробка ігор:

- Ігрові движки та бібліотеки: Pygame - це популярна бібліотека для створення 2D-ігор, яка забезпечує легкий доступ до графіки та звуку.
- Прототипування: Python часто використовується для швидкого прототипування ігор та створення ігрових концепцій.

5. Розробка програмного забезпечення:

- Десктопні додатки: Бібліотеки, такі як Tkinter, PyQt, Kivy, дозволяють створювати крос-платформні графічні інтерфейси для десктопних додатків.
- Мобільні додатки: Фреймворк Kivy дозволяє створювати мобільні додатки для Android та iOS.

База даних Sqlite

SQLite – це компактна, вбудована система керування реляційними базами даних (СУБД). Вона відрізняється від багатьох інших СУБД, таких як MySQL або PostgreSQL, тим, що вона вбудовується безпосередньо в додаток, не вимагаючи окремого сервера баз даних.

SQLite – це зручне і компактне рішення для зберігання та управління даними. Особливо вона зручна в мобільних додатках, де немає необхідності в складних серверах. SQLite дає змогу розробникам ефективно керувати даними, водночас скорочуючи та забезпечуючи їхню цілісність, зберігаючи водночас простоту використання та низькі накладні витрати.

Основні характеристики Sqlite:

1. Легкість. SQLite дуже компактна і легка у використанні. Не вимагає великих ресурсів системи і може працювати на практично всіх платформах, включаючи мобільні пристрої та вбудовані системи;
2. Інтеграція. SQLite інтегрується безпосередньо в додаток. Файл бази даних SQLite зберігається на диску, і додаток працює з ним безпосередньо;
3. Відкритий вихідний код. SQLite поширюється як бібліотека з відкритим вихідним кодом (open source). Можна змінювати вихідний код відповідно до ваших потреб;
4. Підтримка SQL. SQLite підтримує повний набір SQL-команд. Можна виконувати операції вставки, вибірки, оновлення та видалення даних з використанням SQL-запитів;
5. Низькі накладні витрати на адміністрування. Оскільки SQLite вбудовується в застосунок і не потребує окремого сервера, адміністрування бази даних зводиться до мінімуму;
6. Підтримка транзакцій. SQLite підтримує транзакції, що гарантує цілісність даних і забезпечує атомарність операцій;
7. Переносимість. SQLite-база даних може бути перенесена з однієї платформи на іншу без зміни коду програми;
8. Широке поширення. SQLite широко використовується в різних галузях, включно з мобільними додатками (Android і iOS), настільними додатками, вбудованими системами, браузером та багато іншого.

Бібліотеки та фреймворки Python

Бібліотеки та фреймворки є невід'ємною частиною розробки програмного забезпечення та дозволяють розширювати можливості мови програмування. У данному випадку, використання різноманітних бібліотек дозволило значно полегшити та прискорити розробку проекту, а також реалізувати деякі складні функції та алгоритми. Далі будуть наведені описи та призначення використаних бібліотек, а також їхній внесок у реалізацію проекту.

Бібліотека **os** в Python надає численні функції для взаємодії з операційною системою. Вона дозволяє виконувати широкий спектр операцій, таких як робота з файловою системою, змінними середовища, процесами тощо.

dotenv та **load_dotenv** є частинами пакета `python-dotenv`, який використовується для завантаження змінних середовища з файлу «.env» у змінні середовища. Це зручно для налаштування конфігурації додатку, особливо коли потрібно зберігати конфіденційні дані, такі як ключі API, паролі та інші конфігураційні параметри. Завдяки цьому реалізується безпека доступу до телеграм-бота, API ключ якого знаходиться у файлі «.env».

Бібліотека **logging** в Python використовується для реєстрації (логування) подій, помилок та повідомлень у програмах. Вона надає широкий спектр можливостей для запису різних рівнів логів (наприклад, DEBUG, INFO, WARNING, ERROR, CRITICAL) та їх зберігання у різних місцях (консоль, файли, мережеві сервіси тощо).

Aiogram є популярною бібліотекою для створення Telegram-ботів на Python. Вона забезпечує асинхронну взаємодію з API Telegram, що дозволяє створювати швидкі та ефективні боти. Основними характеристиками та перевагами бібліотеки `aiogram`:

1. Асинхронність:

- використовує `asyncio`, що дозволяє створювати високопродуктивні боти, здатні обробляти велику кількість запитів одночасно.

- Підтримка асинхронних функцій робить aiogram дуже ефективним у використанні ресурсів.

2. Легкість використання:

- простий та інтуїтивно зрозумілий синтаксис, що полегшує процес розробки.
- Відмінно підходить як для новачків, так і для досвідчених розробників.

3. Широкий функціонал:

- підтримка всіх методів API Telegram, включаючи роботу з повідомленнями, файлами, користувачами, групами і каналами.
- Можливість налаштовувати інлайн-кнопки, меню, опитування та інші елементи інтерфейсу.

4. Модульність:

- можливість створювати модулі для розділення логіки бота на різні частини, що полегшує управління великими проектами.
- Вбудована підтримка обробників для різних типів подій, таких як повідомлення, команди, кнопки тощо.

5. Документація та підтримка спільноти:

- відмінна документація з великою кількістю прикладів і підказок.
- Активна спільнота користувачів, яка може допомогти у разі виникнення питань або проблем.

Основні компоненти

1. Dispatcher: Основний компонент для реєстрації обробників подій. Дозволяє організувати обробку різних типів повідомлень і подій.

2. **Executor:** Компонент для запуску бота. Може працювати як у звичайному режимі, так і у режимі вебхуків.
3. **Middlewares:** Механізм для додавання додаткової логіки перед або після обробки подій, що дозволяє легко інтегрувати такі речі, як логування, авторизація, обробка помилок тощо.
4. **Filters:** Фільтри дозволяють відбирати повідомлення за різними критеріями, такими як тип повідомлення, команди, текст, регулярні вирази тощо.

Діаграма послідовності

Для побудови діаграми послідовності Telegram-бота з базою даних можна виділити такі основні компоненти та етапи взаємодії:

- **Користувач:** Взаємодіє з ботом через Telegram.
- **Telegram Бот:** Використовує API Telegram для обробки запитів і відправки відповідей користувачу.
- **Сервер/Додаток:** Логіка бота, що обробляє запити, приймає рішення і взаємодіє з базою даних.
- **База Даних:** Зберігає інформацію про користувачів, замовлення, товари тощо.

Нижче наведена діаграма, яка ілюструє цей зв'язок.

Опис діаграми

1. **Запит користувача:** Користувач надсилає повідомлення або команду до Telegram-бота.
2. **Отримання повідомлення ботом:** Telegram API передає повідомлення на сервер, де працює бот.
3. **Зв'язок з базою даних:** Для виконання запиту бот звертається до бази даних для зчитування або запису даних.
4. **Отримання даних з бази:** База даних повертає необхідну інформацію бот.

- Відправка відповіді користувачу: Бот формує відповідь і надсилає її через Telegram API користувачу.

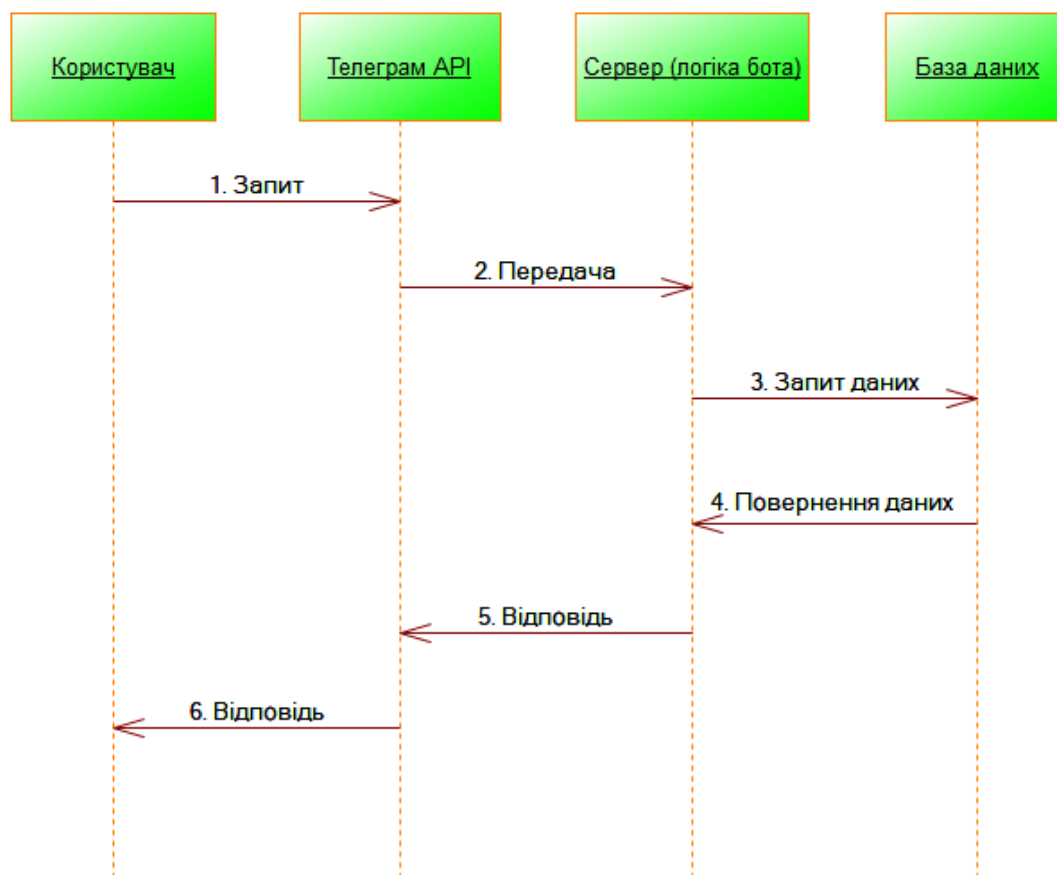


Рис. 2.1 Діаграма послідовності

РЕАЛІЗАЦІЯ ПРОЄКТУ. ОПИС ПРОГРАМНОГО ПРОДУКТУ

Взаємодія з інформаційною системою

Розділення функціоналу між користувачами та адміністраторами є важливим для забезпечення зручності, безпеки та ефективності роботи телеграм-бота для інтернет-магазину. Це дозволяє кожній групі користувачів виконувати свої завдання з максимальною продуктивністю, мінімізуючи можливість помилок та забезпечуючи якісне обслуговування кінцевих клієнтів.

Користувачі – це кінцеві клієнти, які взаємодіють з ботом для здійснення покупок. Їм потрібен простий та інтуїтивний інтерфейс для виконання таких дій:

- Вибір категорії товару: Користувачі повинні мати можливість вибрати між різними категоріями товарів (наприклад, чоловіче чи жіноче взуття).
- Вибір моделі та розміру: Після вибору категорії користувачі можуть обрати конкретну модель та відповідний розмір.
- Вибір кількості товару: Користувачі можуть вказати необхідну кількість обраного товару.
- Отримання номера замовлення: Після оформлення замовлення користувачі отримують підтвердження з номером замовлення.

Адміністратори – це співробітники магазину, які відповідають за підтримку та управління каталогом товарів. Їм необхідний розширений функціонал для виконання наступних завдань:

- Перегляд товарів: Адміністратори повинні мати можливість переглядати наявні товари, щоб контролювати актуальність інформації та стежити за запасами.

- Додавання товарів: Для поповнення асортименту адміністратори можуть додавати нові товари, включаючи деталі про них (назва, опис, ціна, розміри).
- Видалення товарів: Якщо певний товар більше не доступний або його потрібно виключити з каталогу, адміністратори можуть видалити його.

Причини необхідні для розділення функціоналу:

- Простота та зручність: Користувачі не повинні мати доступу до адміністративних функцій, щоб не перевантажувати інтерфейс і не створювати можливостей для помилок.
- Безпека: Адміністративні функції повинні бути захищені від доступу звичайних користувачів. Це важливо для запобігання випадкових або зловмисних змін у каталозі товарів.
- Ефективність управління: Адміністратори повинні мати спеціалізований набір інструментів, який дозволяє їм ефективно керувати каталогом товарів та підтримувати його актуальність.

Було створено Use case діаграму, яку зображено нижче (рис. 3.1), для того щоб переглянути можливості Користувача та Адміністратора у взаємодії з інтернет-магазином

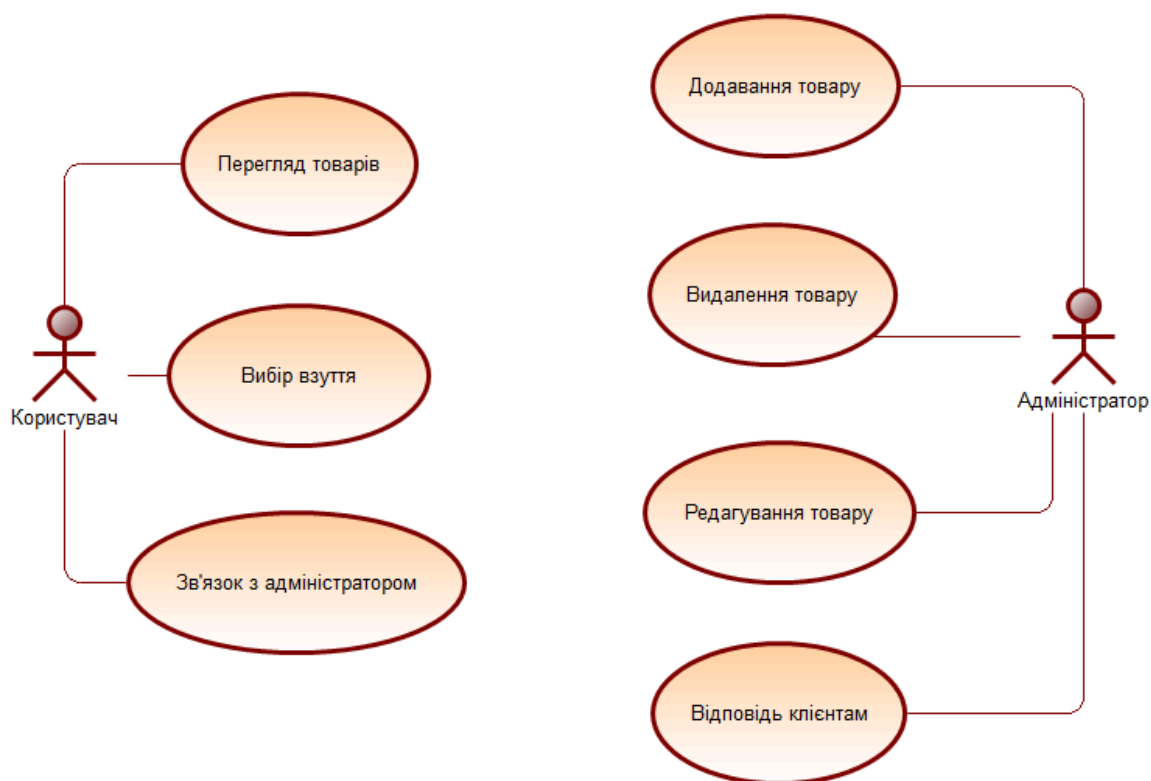


Рис. 3.1 Use case діаграма для Користувача та Адміністратора

Для адміністратора та користувача розроблений окремий функціонал, далі наведені скріншоти початку взаємодії (це відбувається за командою /start) звичайного користувача та адміністратора з ботом:

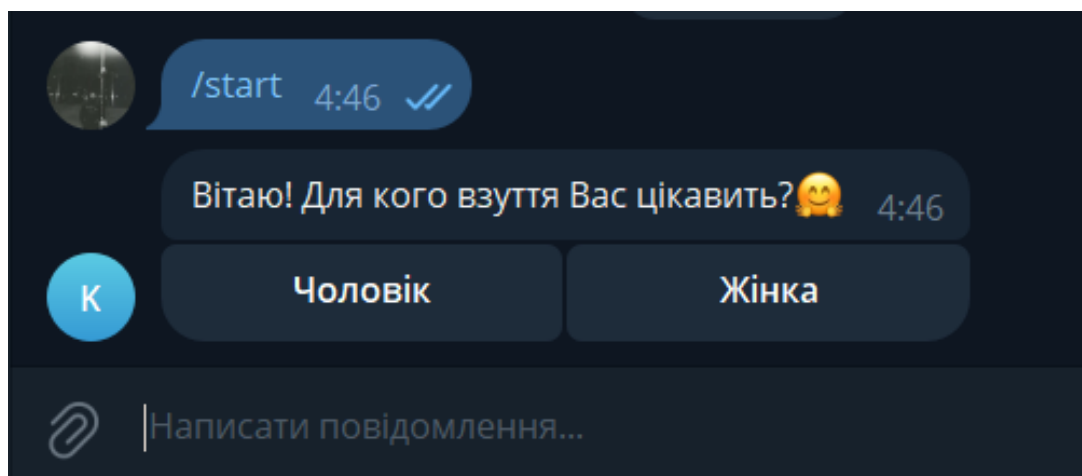


Рис. 3.2 Початок роботи користувача з ботом

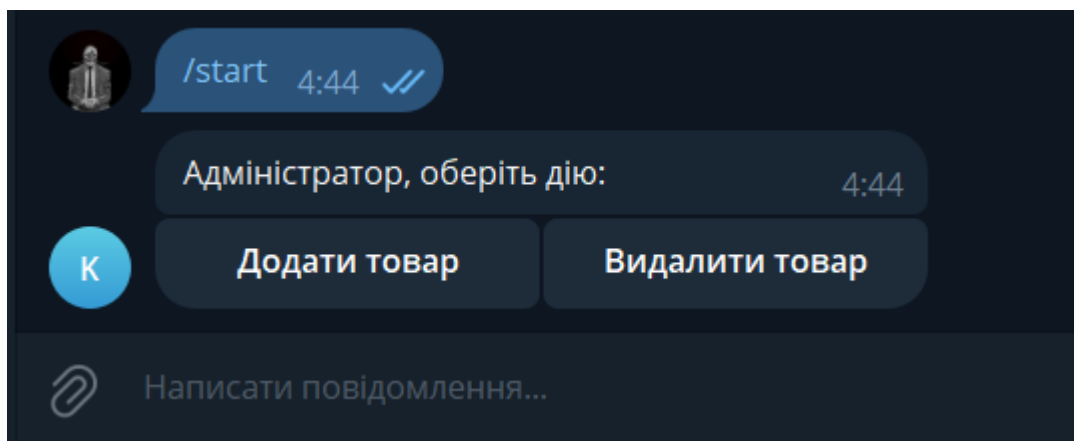


Рис. 3.3 Початок роботи адміністратора з ботом

Реалізація функціоналу бота для адміністратора. FSM

Після команди старт при виборі адміністратором кнопки «Додати товар», є чітко визначена послідовність його дій для додавання товару, тобто не виконавши першого етапу, перехід до другого реалізований бути не може. У випадку нашого бота, для заповнення бази даних потрібно пройти такі етапи:

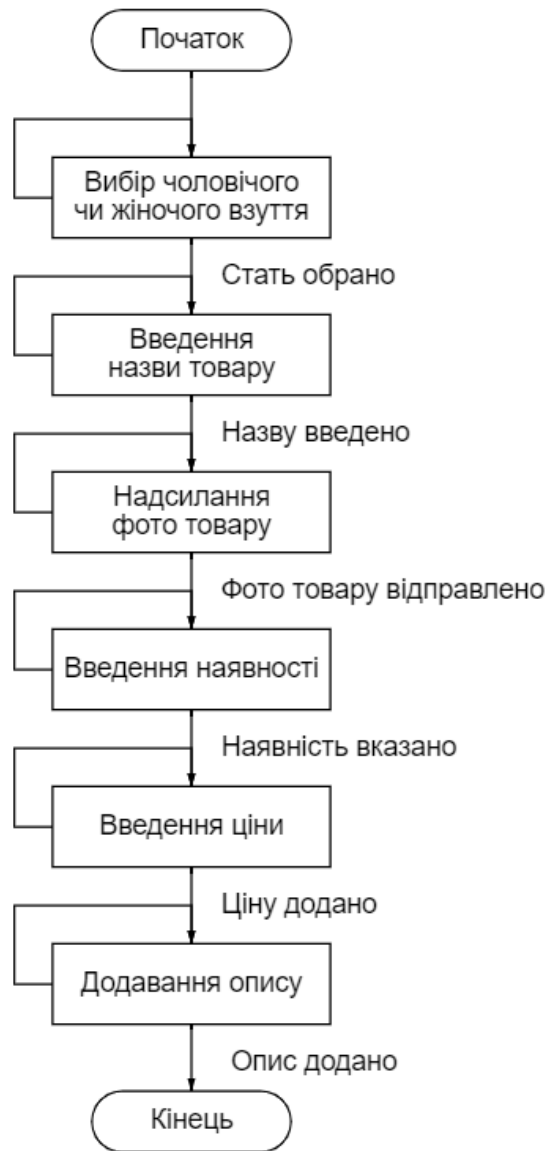


Рис. 3.4 Почерговий перехід до різних станів додавання товару

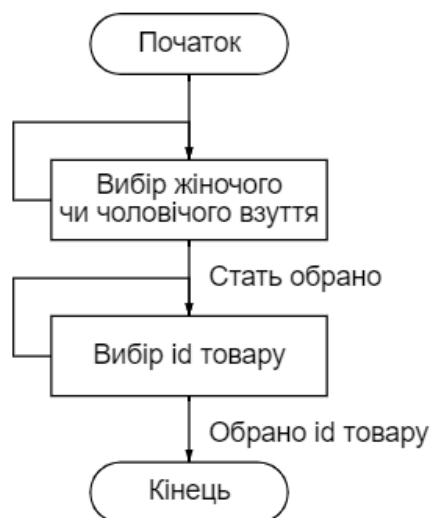


Рис. 3.5 Почерговий перехід до різних станів видалення товару

Для того щоб реалізувати покрокове заповнення даних було використано "Кінцевий автомат" (скорочено FSM), який також називають "автоматом" або "скінченним автоматом" – це абстрактний автомат, який складається з множини станів (включаючи початковий стан і один або декілька кінцевих станів), множини вхідних подій, множини вихідних подій і функції переходу з одного стану в інший. Функція переходу приймає на вхід поточний стан і вхідну подію і повертає новий набір вихідних подій і наступний (новий) стан. Деякі стани використовуються як "термінальні стани". Робота FSM починається зі спеціального стану, який називається початковим станом, проходить через переходи в залежності від вхідних даних до різних станів і зазвичай закінчується в термінальному або кінцевому станах. Стан, який позначає успішне виконання операції, називається станом прийняття.

Детермінований кінцевий автомат або акцепторний детермінований кінцевий автомат – це п'ятірка $(\Sigma, S, s_0, \delta, F)$, де

Σ – вхідний алфавіт (скінченна, непорожня множина символів).

S – скінченна, непорожня множина станів.

s_0 – початковий стан, елемент S .

δ – функція переходу зі стану в стан: $\delta : S \times \Sigma \rightarrow S$ (у недетермінованому скінченному автоматі це буде $\delta : S \times \Sigma \rightarrow \wp(S)$, тобто δ повертає множину станів).

$(\wp(S))$ – це потужність множини S

F – множина кінцевих станів, (можливо, порожня) підмножина S .

Формальна специфікація скінченних автоматів (FSM) пояснює та обмежує поведінку програми. Машини станів підтримують написання самоописуваного коду. Вони допомагають виявляти логічні помилки і, якщо щось йде не так, надають точні повідомлення про винятки.

```

1  from aiogram.dispatcher.filters.state import State, StatesGroup
2
3  class Add_Item(StatesGroup):
4      set_title = State()
5      set_photo = State()
6      set_availability = State()
7      set_price = State()
8      set_description = State()
9      set_genders = State()

```

Рис. 3.6 Визначення станів для додавання товару

```

12  class Admin(StatesGroup):
13      set_id = State()
14      set_gender = State()

```

Рис. 3.7 Визначення станів для видалення товару

Додавання товару до БД

Перед початком взаємодії користувачів з системою, адміністратор повинен заповнити даними БД. Для початку було створено такі стовпці: id, title (назва товару), photo (зображення товару), availability (наявність моделі), price (ціна моделі) і description (опис товару).

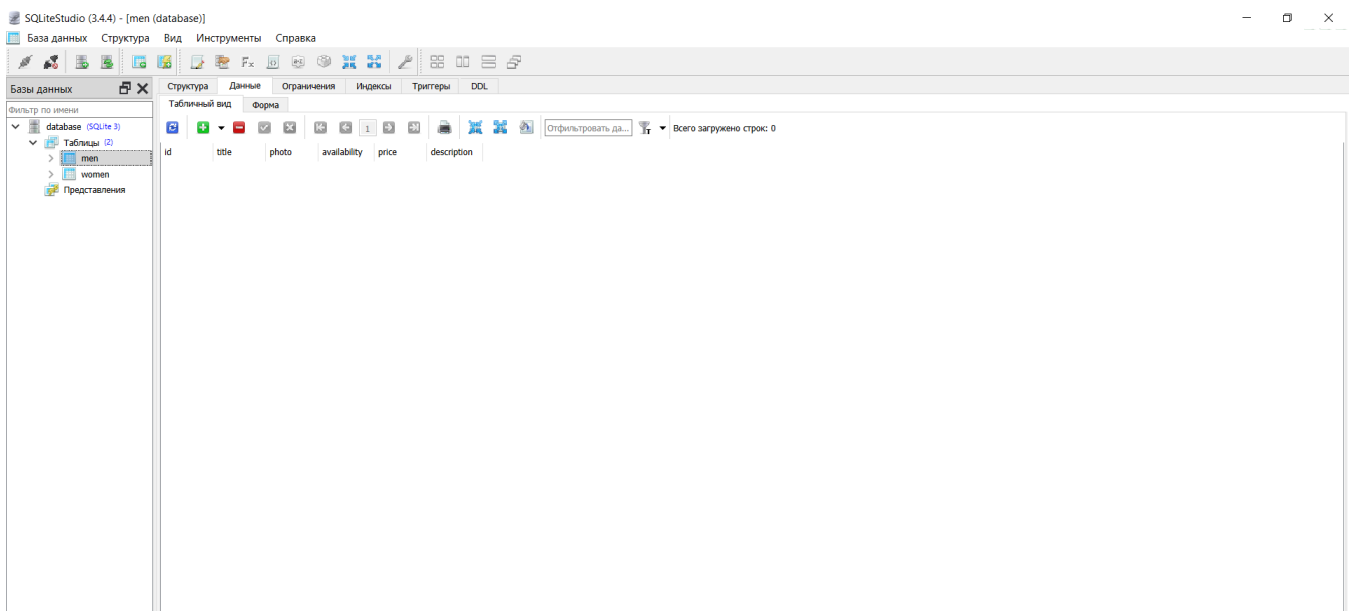


Рис. 3.8 Початковий стан БД

Далі буде зображено покроковий алгоритм додавання товару адміністратором:

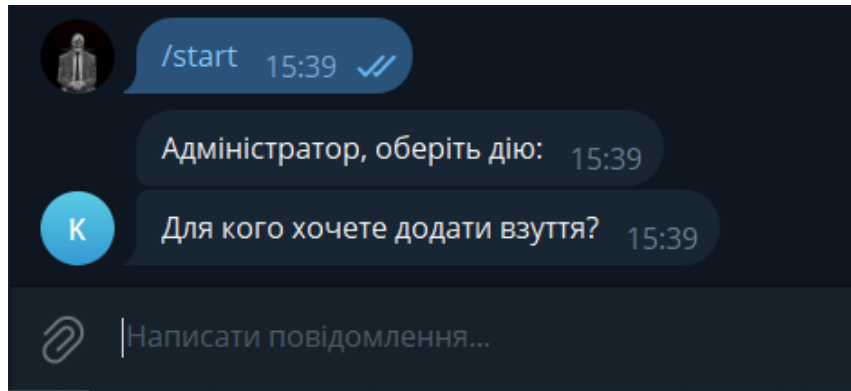


Рис. 3.9 Вибір статі для таблиці

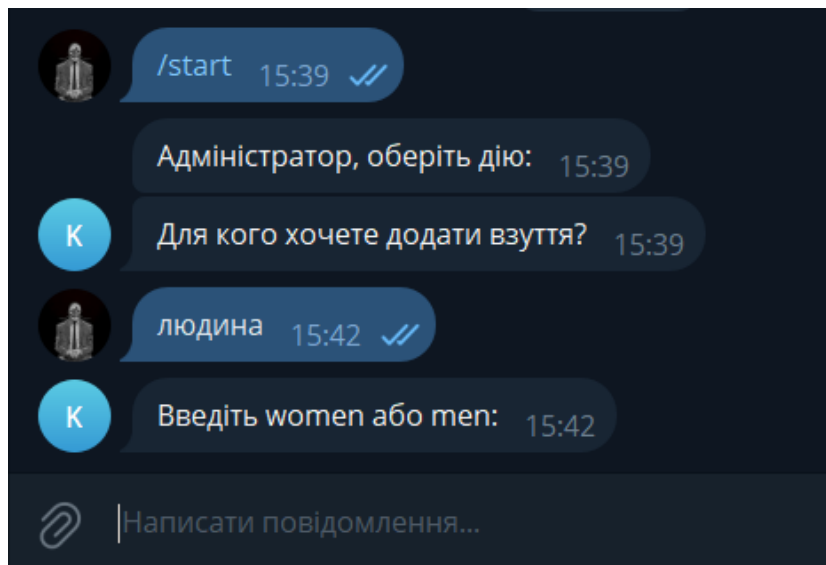


Рис. 3.10 Виникнення винятку



Рис. 3.11 Запит на отримання назви товару

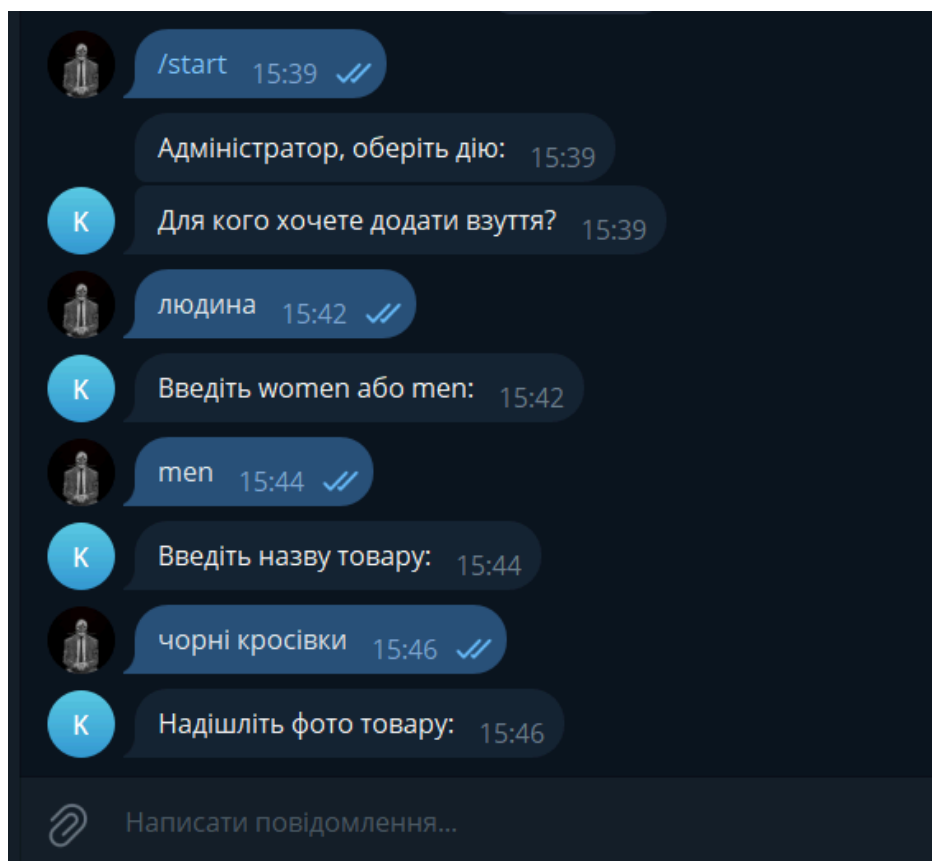


Рис. 3.12 Очікування фото товару



Рис. 3.13 Введення наявності товару



Рис. 3.14 Додавання ціни моделі



Рис. 3.15 Введення опису товару



Рис. 3.16 Успішне додавання товару

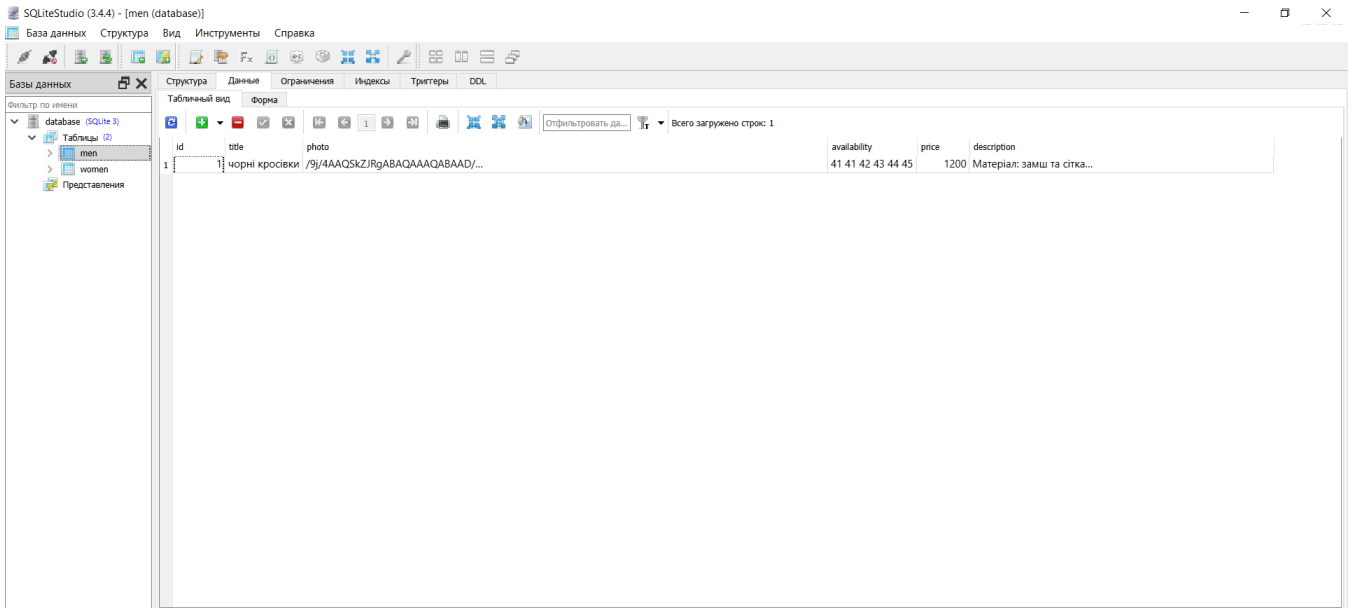


Рис. 3.17 Відображення в БД доданого товару

Додамо ще один товар до БД для його подальшого видалення адміністратором

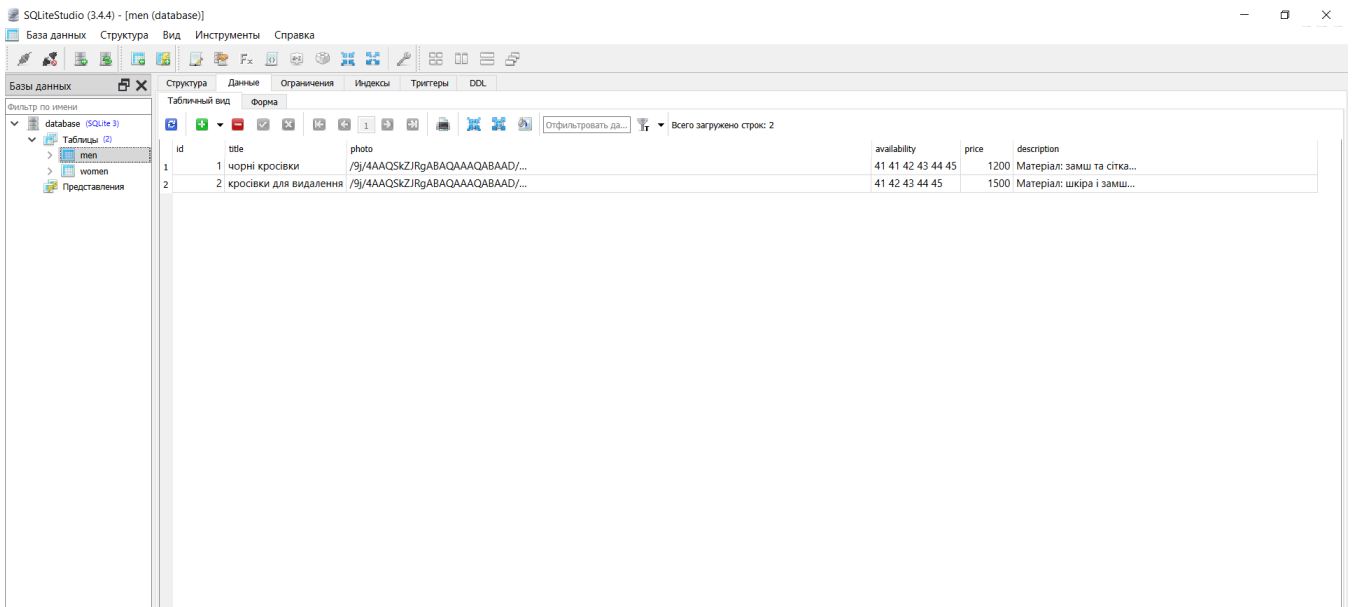


Рис. 3.18 БД з двома доданими товарами

Видалення товару з БД

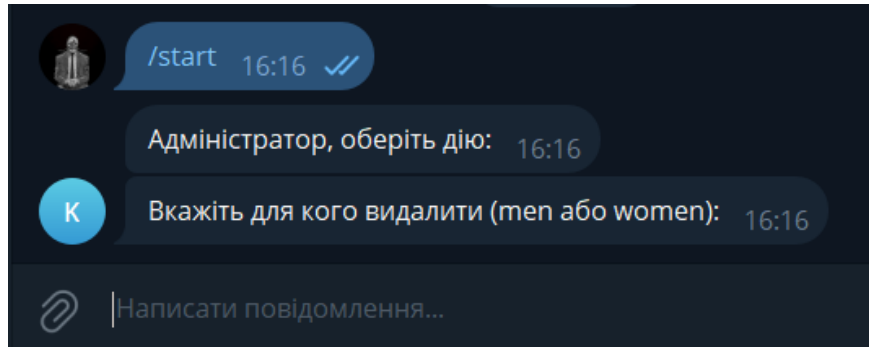


Рис. 3.19 Вибір статі для таблиці

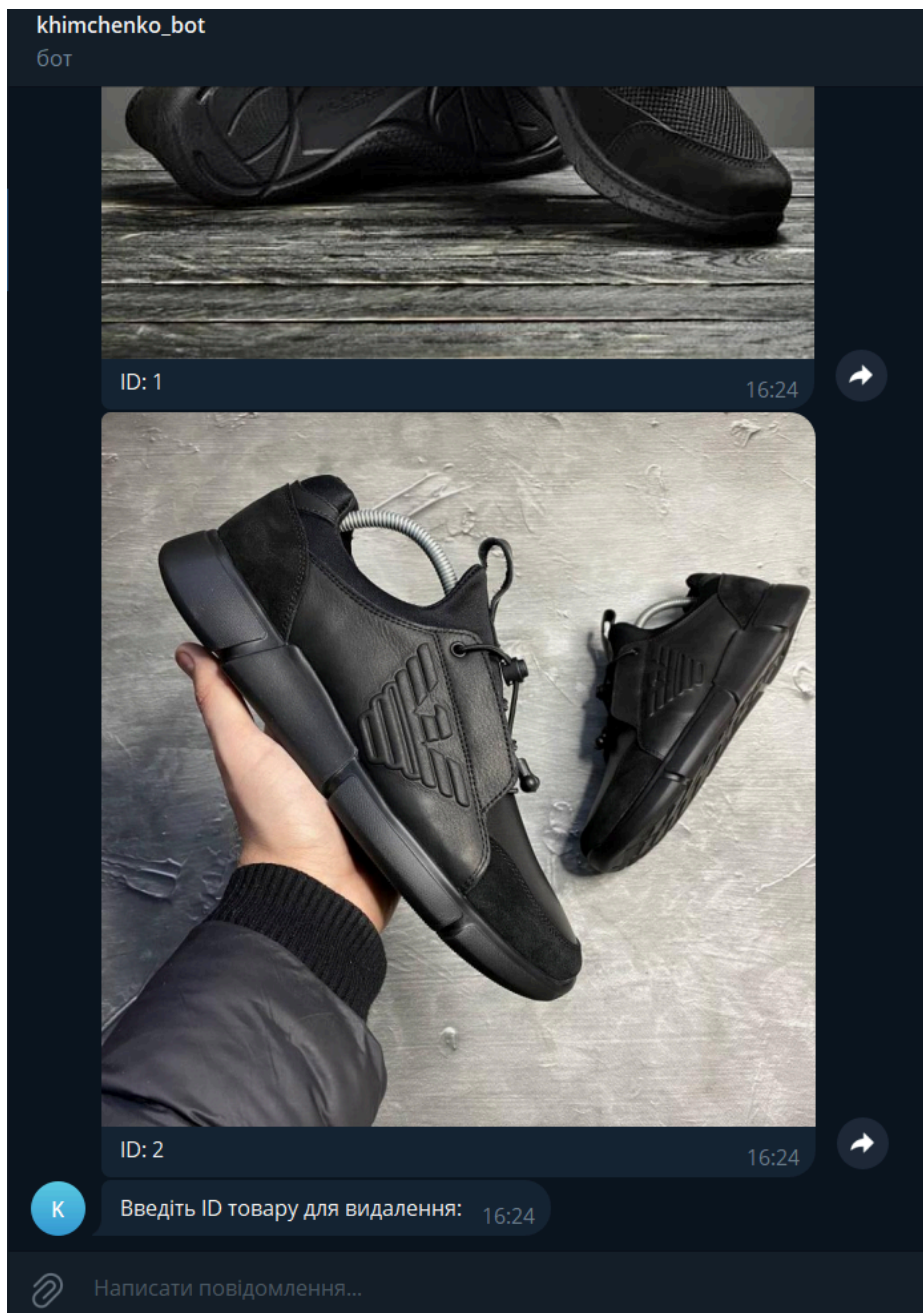


Рис. 3.20 Вибір id товару для видалення

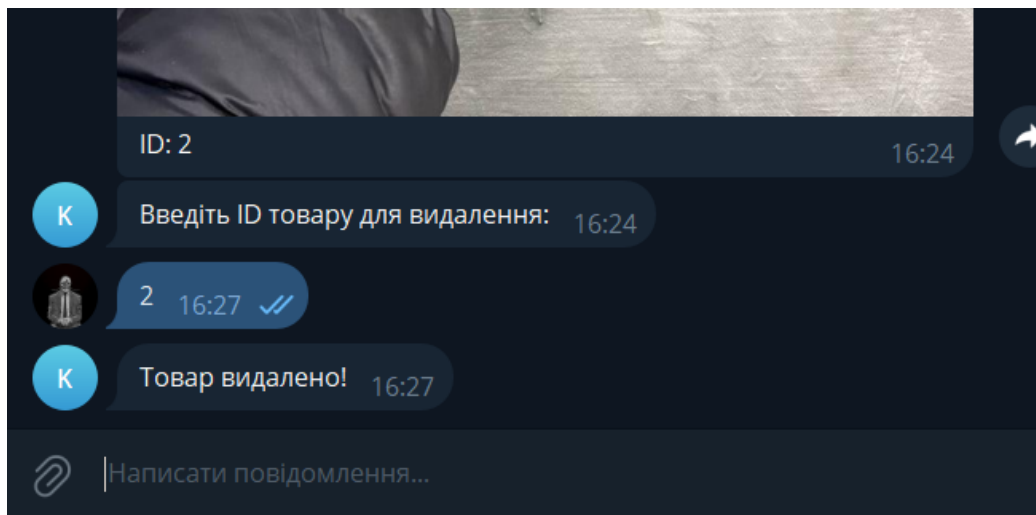


Рис. 3.21 Успішне видалення товару

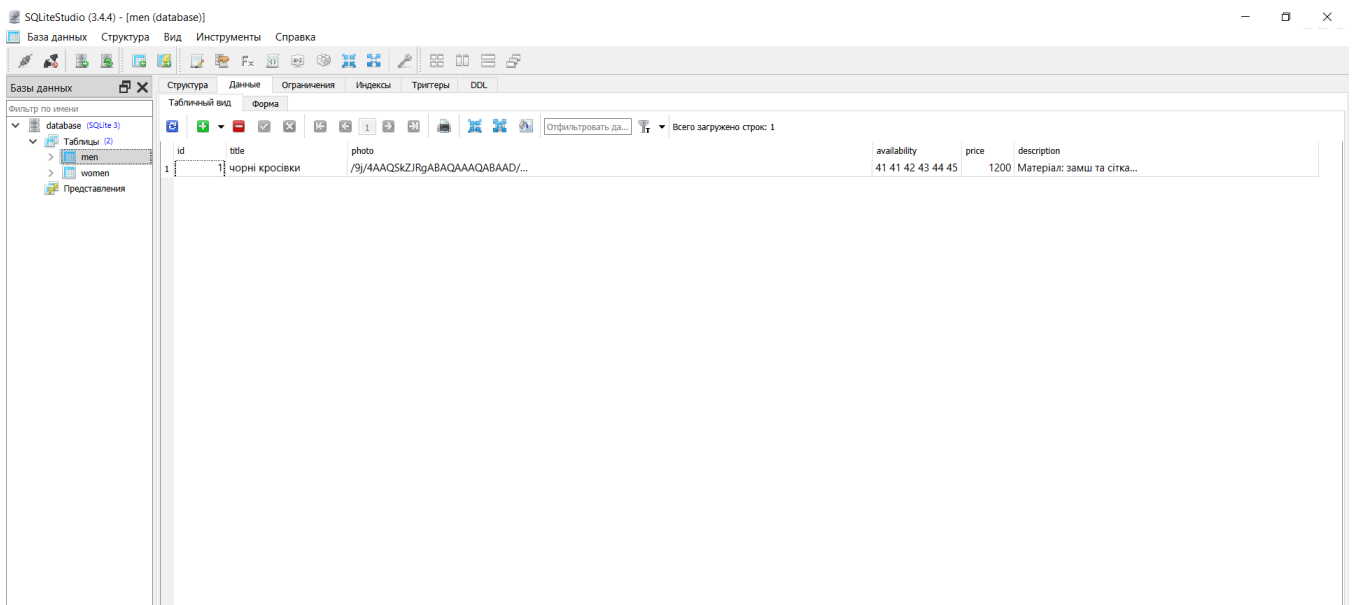


Рис. 3.22 Таблица чоловічого взуття в БД

Взаємодія користувача з ботом

Після того як користувач обирає для кого взуття його цікавить, бот надає каталог товарів з детальною інформацією, після чого бот направляє контакти менеджера для подальшого оформлення замовлення.

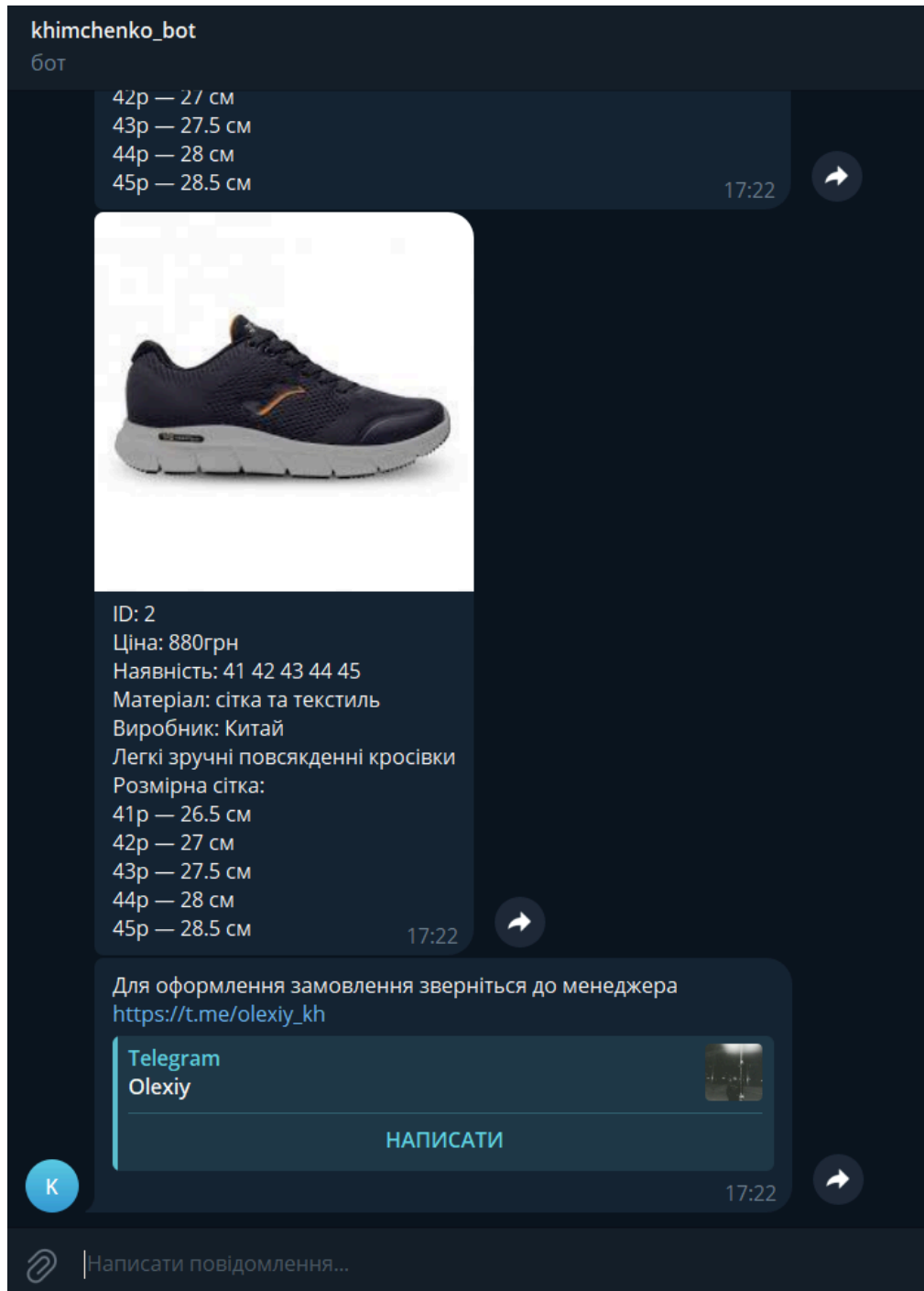


Рис. 3.23 Каталог товарів та контакти менеджера

ВИСНОВКИ

В результаті виконання атестаційної роботи було розроблено автоматизовану інформаційну систему для взаємодії з користувачами та адміністраторами використовуючи мову програмування Python, фреймворк aiogram і базу даних aiosqlite.

Отже, в результаті виконання атестаційної випускної роботи було проведено:

- аналіз предметної області;
- комплексне проектування системи;
- розробка програмного забезпечення.

У процесі розробки вдалось розробити систему для зручного додавання та видалення товару адміністратором і перегляду всіх товарів користувачами з подальшою можливістю звернутись до менеджера за замовленням.

Дана система не є абсолютно завершеною, зважаючи на досвід використання користувачів та подальші можливі оновлення бібліотек, технологій та фреймворків, будуть проводитись покращення функціоналу та зручності користування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://interkassa.com/blog/shho-take-elektronna-komerciya-e-commerce-dlya-pochatkivciv>
2. <https://wezom.com.ua/ua/blog/elektronna-komerciya>
3. <https://buklib.net/books/25202/>
4. <https://project-seo.net/blog-uk/elektronna-komercija-istoria-fakty-trendy/>
5. <https://remonline.ua/blog/how-to-open-online-store-in-2022/>
6. https://zn.ua/ukr/novosti-kompaniy/kilkist-internet-magaziniv-strimko-zrostaye-osnovni-prichini-yih-populyarnosti-297316_.html
7. <https://datalabsua.com/en/database-security/>
8. <https://ukeywaf.com/baza/shho-take-sql-inyekcija/>
9. <https://foxminded.ua/prohramuvannia-baz-danykh-na-sqlite/>
10. <https://medium.com/swlh/implementing-finite-state-machine-as-python-language-native-construct-cb56e3c46215>
11. <https://python-course.eu/applications-python/finite-state-machine.php>