

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Торгівельний чат-бот мовою програмування Python»

на здобуття освітнього ступеня бакалавра

зі спеціальності 126 Інформаційні системи та технології

(код, найменування спеціальності)

освітньо-професійної програми Інформаційні системи та технології

(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело*

(підпис)

Єгор ЦАПОВИЧ
Ім'я, Прізвище здобувача

Виконав:
*здобувач вищої освіти,
гр. ІСД-41*

Єгор ЦАПОВИЧ
Ім'я, ПРІЗВИЩЕ

Керівник:
*науковий ступінь,
вчене звання*

Доц. каф. Олег СЕНЬКОВ
Ім'я, ПРІЗВИЩЕ

Рецензент:
*науковий ступінь,
вчене звання*

Ім'я, ПРІЗВИЩЕ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій

Кафедра Інформаційні системи та технології

Ступінь вищої освіти: Бакалавр

Спеціальність: 126 Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедру ІІЗАС

_____ Каміла СТОРЧАК

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Цаповичу Єгору В'ячеславовичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Торгівельний чат-бот мовою програмування Python

керівник кваліфікаційної роботи Олег СЕНЬКОВ, кандидат технічних наук

(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» лютого 2025р. №56

2.Строк подання кваліфікаційної роботи «30» травня 2025 р.

3.Вихідні дані до кваліфікаційної роботи:

Середовище розробки VS code

Месенджер Telegram

Науково-технічна література з питань, мови програмування Python

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

1. Провести аналіз платформ, для інтеграції чат-бота

2. Здійснити аналіз готових рішень, що до створення чат-ботів

3. Обрати технології для розробки бота

4. Розробити архітектуру проекту

5. Провести аналіз отриманих результатів

5.Перелік ілюстративного матеріалу: *презентація*

6.Дата видачі завдання «24» лютого 2025 р

КАЛЕНДАРНИЙ ПЛАН

№ з\п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	15.03.25 – 30.03.25	
2	Аналіз існуючих рішень та платформи для інтеграції бота	01.04.25 – 14.04.25	
3	Обирання технологій	15.04.25 – 22.04.25	
4	Написання коду	23.04.25 – 27.04.25	
5	Інтеграція платіжної системи	28.04.25 – 15.05.25	
6	Тестування розробленої моделі та висновки	16.05.25 – 19.05.25	
7	Розробка демонстраційних матеріалів (презентація)	20.05.25 – 21.05.25	

Здобувач вищої освіти

(підпис)

Єгор ЦАПОВИЧ

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Олег СЕНЬКОВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 63 стор., 43 рис., 5 табл., 20 джерел.

Мета роботи: Показати як створити Telegram бота який буде допомагати продавати товари ,або послуги підприємства.

Об'єкт дослідження: В ролі об'єкта в нас буде виступати сам чат-бот, та як він себе поводить у сфері електронної комерції.

Предмет дослідження: Технологія розробки Telegram бота за допомогою мови програмування Python та сторонніх бібліотек.

Короткий зміст роботи:

У цьому дипломному проєкті розглянуто процес розробки торгівельного Telegram. Основні вимоги до проєкту це розробити готового бота який може: зареєструвати користувача в системі, показувати наявний асортимент товару, оформляти замовлення, та надавати консультацію користувачеві, у режимі реального часу.

В ході розробки, було використано мову програмування Python та бібліотеку Aiogram на якій розроблена логіка нашого проєкту. Зберігання даних такі як дані від користувача, картка товару, номер замовлення, все це зберігається за допомогою наявного компонента в Python - SQLite. В ролі платіжної системи у нас виступить LiqPay. Також буде додана панель адміністратора, що допоможе підтримувати працездатність бота. Окрему увагу приділено архітектурі системи, вибору середовища розробки та аналізу переваг використаних технологій.

Результатом є стабільно працюючий Telegram бот, який може бути впроваджений у реальну роботи бізнесу, для підвищення його працездатності. Розроблене рішення є гнучким та масштабованим, а також має потенціал для подальшого розвитку в перспективі.

КЛЮЧОВІ СЛОВА: TELEGRAM БОТ, PYTHON, AIOGRAM, БАЗАДАНИХ, ПЛАТІЖНА СИСТЕМА, ТЕХНІЧНА ПІДТРИМКА, ІНТЕРФЕЙС КОРИСТУВАЧА, ПАНЕЛЬ АДМІНІСТРАТОРА, ОБРОБКА ЗАМОВЛЕННЯ.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1	
ТЕОРЕТИЧНІ ВІДОМОСТІ ПРЕДМЕТНОЇ ОБЛАСТІ. АКТУАЛЬНІСТЬ БОТІВ ТА МОЖЛИВОСТІ ЇХ РЕАЛІЗАЦІЇ.....	10
1.1 Актуальність ботів	10
1.2 Порівняння платформ для інтеграції бота.....	11
1.3 Основні переваги та недоліки при використанні Telegram ботів	14
1.4.Огляд існуючих конструкторів ботів	16
1.5 Основний функціонал чат-бота	18
РОЗДІЛ 2	
ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ.....	19
2.1 Мова програмування.....	19
2.2 Середовище розробки.....	21
2.3 Бібліотека Python Aiogram	22
2.4 База даних SQLite.....	24
2.5 Створення бота в Telegram.....	25
2.6 Telegram API	26
2.7 Платіжна система LiqPay	27
РОЗДІЛ 3	
РОЗРОБКА ТОРГІВЕЛЬНОГО TELEGRAM БОТА	28
3.1 Архітектура проєкту	28
3.2 Опис основного функціоналу проєкта.....	30
3.3 Тестування та перевірка коректності роботи бота	48
3.4 Оцінка ефективності впроваджень.....	52
ВИСНОВОК.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	58

ВСТУП

У сучасному світі людину важко уявити без смартфона, встановлених додатків, соціальних мереж. З розвитком технологій з'являються все більше речей, які спрощують життя людям, до прикладу візьмемо чат-ботів.

Чат-бот - це робот, який здатний автоматично та швидко відправляти та відповідати на повідомлення користувачів. Вони можуть виконувати чимало різних функцій: від надання необхідних послуг, до виконання простих задач, а розміщуються такі боти в соціальних месенджерах, якими ми часто користуємося - Facebook, Telegram, Viber тощо.

Одна із сфер, у якій часто використовують чат-ботів - це електронна комерція, бо вона стрімко розвивається, зростає попит на інноваційні рішення для автоматизації бізнес процесів. Чат-боти здатні імітувати спілкування з користувачем у режимі реального часу та дозволяють автоматизувати процеси продажу, обслуговування клієнтів та управлінням замовленнями. Наявність та зручність роблять їх невід'ємною частиною сучасних торгових платформ, спрощуючи процес взаємодії між покупцями та продавцями. Цей інструмент, дає змогу зменшити навантаження на персонал, забезпечити цілодобову підтримку та прискорити обробку замовлень.

Робота має на меті продемонструвати створення Telegram чат-бота як інструменту для продажу товару. **У ролі об'єкта дослідження** у нас буде виступати сам чат-бот, а саме - як він функціонує у сфері електронної комерції, зокрема його роль в автоматизації клієнтських комунікацій, обробці запитів і підвищенні ефективності продажів. **Предметом дослідженням** є розробка самого чат-бота з використанням мови програмування Python та вибором бібліотеки Aiogram, проектування його архітектури та основних функцій.

Завдяки методу дослідження ми проаналізуємо популярні месенджери, зокрема Telegram, виявимо їхні сильні та слабкі сторони та зробимо висновок чому саме ми обрали дану платформу. Окрему увагу буде приділено обґрунтуванню актуальності

чат-ботів для бізнесу через статистичні дані, а також розглянемо їхні сильні та слабкі сторони. Крім того буде здійснено огляд технологій, що використовуються при розробці чат-бота, з поясненням зробленого вибору, а також оглянемо сервіси які допоможуть нам створити чат-бота, виявимо їх сильні та слабкі сторони, та з'ясуємо чому саме ми обрали той чи інший вибір.

Робота була апробована на VI Міжнародну науково-технічну конференцію «Сучасний стан та перспективи розвитку IoT», проходила 15 квітня 2025 року. Тезу на тему «Застосування Telegram чат-ботів у сфері e-commerce» була опублікована у збірнику присвяченому цій конференції. Результати дослідження мають як теоретичну, методичну, та і практичну значущість. З теоретичного боку, проєкт демонструє як завдяки одному інструменту, можна збільшити показники продажу, та автоматизацію ручних процесів. З методичного - він може слугувати прикладом для подібних або аналогічних реалізацій. Практична значущість проєкта, полягає у готовності рішення до впровадження, що дасть змогу малим та середнім підприємствам зекономити гроші та час.

РОЗДІЛ 1

ТЕОРЕТИЧНІ ВІДОМОСТІ ПРЕДМЕТНОЇ ОБЛАСТІ. АКТУАЛЬНІСТЬ БОТІВ ТА МОЖЛИВОСТІ ЇХ РЕАЛІЗАЦІЇ

1.1 Актуальність ботів

Ніхто не міг би колись подумати, що спілкування в інтернеті буде займати у людини доволі велику частину його життя. Сучасні бренди вже довгий час використовують таку можливість на користь своєму бізнесу, але здебільшого вони все ж покладаються при спілкуванні з клієнтами на чат-ботів. Що ж таке насправді чат-боти давайте розбиратися.

Якщо ми поглибимось у цю тему то ми з'ясуємо, що це комп'ютерна програма, яка працює на основі машинного навчання та має широку область використання, від імітування спілкування з живою людиною, і до виконання різноманітних завдань. Порівнюючи роботу бота та людини, слід зазначити, що бот швидше реагує на процеси та не втомлюється від однотипних задач. Звісно, він не замінить спілкування з живою людиною, але зможе взяти на себе більшу частину задач.

Згідно зі статистикою за 2023 рік, чат-боти зекономили до 2,5 мільярдів годин робочим підприємствам, а у 2020 році 42 % покупців користувалися послугами цифрових помічників для здійснення покупок. Більше 88 % людей оцінюють свою взаємодію з чат ботами як нейтральну або позитивну, а 62 % людей віддають перевагу роботі з цифровими помічниками клієнтської служби, а не очікування відповіді від людини. Це нам говорить про те, що попит на чат-ботів з кожним роком зростає та не збирається падати [1].

Прогноз прибутку ринку чат-ботів

Мільйон доларів

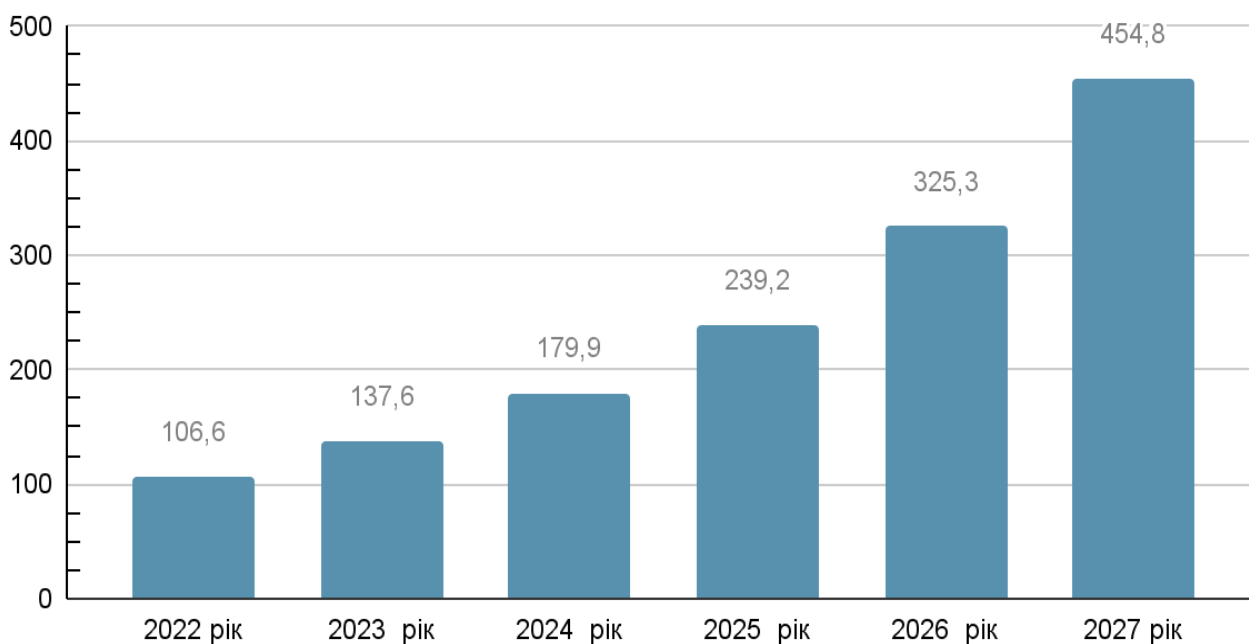


Рис 1.1 - Діаграма прогнозу ринку чат-ботів з 2022 по 2027 роки

На рисунку 1.1 зображено прогнозування ринку чат-ботів. Ресурс media стверджує що прибуток ринку чат ботів до 2027 року зможе вирости у 2 рази [2].

1.2 Порівняння платформ для інтеграції бота

Соціальні мережі відіграють велику роль у нашому повсякденному житті і сучасна людина майже не уявляє свій день без перегляду новин у Telegram або перегляд стрічки в Instagram. Варто лише спробувати провести день без них - і стане одразу зрозуміло, що звичка використання цих додатків стала невід'ємною частиною нашого життя.



Рис 1.2 - Топ найпопулярніших застосунків 2024 року в Україні

На рисунку 1.2 зображено рейтинг мобільних додатків за 2024 рік, які встановлені на телефонах українців. В колонці «Загальний рейтинг популярних застосунків» нас цікавить тільки месенджери, які дозволяють інтегрувати бота на своїй платформі, а саме Telegram, WhatsApp, Viber, Facebook [3].

Для порівняння месенджерів була створена таблиця, в якій перераховані важливі для розробки бота характеристики, що допоможуть визначитися з вибором платформи для інтеграції бота.

Таблиця 1.1

Порівняння соціальних мереж для інтеграції чат-бота

Функціонал	Telegram	WhatsApp	Viber	Facebook
Відкритий API	+	-	-	-
Шифрування, анонімність	+	-	-	-
Автоматизація відповідей	+	+	+	+
Хмарне сховище	+	+	-	+
Підтримка інтеграцій	+	+	+	+

Нижче будуть описані подробиці кожного функціоналу з таблиці 1.1, щоб наочно продемонструвати плюси та мінуси кожного додатка.

Відкритий API надає можливість взаємодіяти з ботом, налаштовувати його більш під гнучкіші задачі, тому відкритість є для нас важливим пунктом при створенні бота. Як ми можемо бачити з таблиці, всі месенджери, окрім Telegram, мають закритий API. У деяких із месенджерів API може бути платним або треба чекати верифікацію від месенджера щоб користуватися ним.

Шифрування - ця функція дозволяє бути впевненим, що твої повідомлення не будуть переглядатися сторонніми особами. Ця функція безумовно є однією із основних переваг Telegram перед іншими месенджерами.

Автоматизація відповідей надає нам можливість створювати сценарії відповідей для бота, що дозволяє автоматизувати обслуговування користувача.

Функція хмарного сховища надає можливість не перейматися за чати або файли, які знаходяться у вас у месенджері, та бути впевненим, що у разі видалення ваші дані не пропадуть з пристрою, а будуть зберігатися на серверах месенджера.

Підтримка інтеграцій дозволить нам підключати нашого бота до зовнішніх сервісів, CRM-систем, платіжних функцій та інших інструментів, які необхідні нам подальшої розробки.

Як ми можемо бачити, з-поміж усіх месенджерів найбільш сильно виділяється Telegram, який безперервно має найбільш гнучкіший функціонал серед своїх конкурентів, а ще одним із головних плюсів є те, що він встановлений майже у кожного українця, що робить просування нашого бота більш швидшим і ефективнішим.

1.3 Основні переваги та недоліки при використанні Telegram ботів

Користувацькі переваги використання Telegram ботів. Першою перевагою є те, що більшість людей вже використовують Telegram, тому встановлення додаткових застосунків не потрібно. Другою перевагою можна виділити миттєву відповідь ботів на запитання, що суттєво покращує досвід користування - особливо у порівнянні зі спілкуванням з людиною-оператором, який не завжди миттєво відповідає на питання. Користувач може звернутися в будь-який час доби до бота та отримати вмиль необхідну інформацію.

Користувацькі недоліки використання Telegram ботів. Однією із головних недоліків Telegram ботів є обмежений кастомний інтерфейс, що не завжди дозволяє створити максимально зручний візуальний досвід. Крім того вся інформація в Telegram відображається у вигляді послідовних повідомлень, що може спричинити плутанину, якщо інформації багато або немає чіткого сценарію взаємодії.

Бізнесові переваги використання Telegram ботів. Для бізнесу Telegram боти є потужним інструментом автоматизації процесів обслуговування клієнтів та

продажу товарів і послуг. Вони допомагають знизити навантаження на персонал, а також забезпечити постійне спілкування з клієнтом. Крім того, завдяки автоматизації повсякденних задач підвищується загальна ефективність бізнес-процесів, а витрати на обслуговування клієнтів зменшуються.

Бізнесові недоліки використання Telegram ботів. Деякі користувачі можуть сприймати повідомлення від бота дуже нав'язливі, що може призвести до блокування бота і, як наслідок, до втрати потенційних клієнтів.

Технічні переваги використання Telegram ботів. З технічної точки зору одним із ключових плюсів Telegram є відкритий API, який дозволяє розробникам реалізовувати широкий спектр функцій та інтеграцій. Бот може взаємодіяти з базами даних, зовнішніми сервісами, CRM-системами та з іншими інструментами. Telegram підтримує хмарне збереження даних, що підвищує надійність зберігання історії повідомлень.

Технічні недоліки використання Telegram ботів. В Telegram ботах відсутня можливість маскувати чутливі дані у повідомленнях, що знижує рівень безпеки при авторизації. Крім того, Telegram не дозволяє створити повноцінний автономний застосунок з власною іконкою та доступом до функцій операційної системи, що обмежує варіативність використання в окремих кейсах. Важливо також пам'ятати про ризик надмірного спаму, який може знижувати загальну ефективність спілкування з користувачем.

З вищесказаного можна зробити висновок, що не зважаючи на недоліки, їхній вплив є незначним у порівнянні з перевагами. Розподілене використання чат-ботів та грамотна розробка сценаріїв взаємодії, дозволяє звести ризики до мінімуму та отримати максимальну віддачу від їх впровадження.

1.4.Огляд існуючих конструкторів ботів

На сьогоднішній день ринок пропонує безліч готових рішень для створення чат-ботів, кожне з яких має свої переваги та недоліки. Але основна маса з них має обмежений функціонал або потребують значних фінансових вкладень. Серед усіх рішень я б виділив би такі найпопулярніші ресурси як: Engati, SendPulse, DialogFlow.

Engati – це інструмент для тих, хто хоче швидко та легко створити свого бота без заглиблення в технічних деталі. Він має простий, але інтуїтивно зрозумілий інтерфейс, а завдяки блок-схемам можна швидко зробити робочий прототип бота. Також цей сервіс надає готові шаблони для різних сфер бізнесу, що значно прискорює процес. Але безкоштовна версія має обмежений функціонал, а для складних завдань, особливо пов'язаних з глибокою обробкою мови, ця платформа може не підійти.

SendPulse - це сервіс, який більш орієнтований на маркетингові задачі. Він дозволяє не лише створювати чат-боти, а й керувати email-розсилками та SMS-повідомленнями. Це непоганий вибір для тих, кому потрібен простий інструмент для комунікації з клієнтами, особливо якщо працюєте з соціальними мережами. Але якщо потрібен бот із більшим функціоналом, то SendPulse може не задовольнити ваші очікування, оскільки його NLP-функціонал досить обмежений.

Dialogflow – це набагато серйозніший інструмент від Google, який дозволяє створювати складних ботів. Він ідеально підходить для тих людей, котрі готові вкласти час у налаштування та мають певні технічні знання. Тут можна реалізувати як текстові, так і голосові боти, а інтеграція з іншими сервісами Google робить його потужним інструментом. Проте безкоштовний тариф має обмеження, які можуть стати перешкодою для масштабування.

Таблиця 1.2

Порівняння конструкторів ботів

Основні функції	Engati	SendPulse	Dialogflow
Простота використання	+	+	-
Підтримка голосових ботів	-	-	+
Готові шаблони	+	+	-
Безкоштовна версія з повним функціями	-	-	-
Потужність NLP-функціоналу	-	-	+
Інтеграція з зовнішніми сервісами	+	+	+
Гнучкість у налаштуванні логіки	-	-	+
Потреба в технічних знаннях	-	-	-
Орієнтація на маркетингові задачі	-	+	-

Згідно з таблицею 1.2, можна зробити висновок, що кожен із цих сервісів має свою задачу під конкретну ціль, що не робить їх поганими варіантами для використання. Проте, під наш список задач та функцій, ці сервіси не підходять, а функціонал у них обмежений у безкоштовною версії, тому було прийнято рішення використовувати мову програмування Python та його бібліотеки.

1.5.Основний функціонал чат-бота

Перш за все, в боті реалізована функція реєстрації користувача. Користувач вводить свої особисті дані: ім'я, прізвище, номер телефону, місце проживання, які зберігаються в базі даних. Надалі ці дані використовуються для оформлення замовлень, що забезпечує персоналізований підхід до клієнта.

Наступна функція, яка реалізована - це каталог товару. Товари згруповані за категоріями (брендами), які користувач обирає за допомогою клавіатури. Кожен товар містить фото, назву, опис, ціну та кнопку «Додати до кошика». Завдяки цьому користувач має змогу ознайомитись із пропозицією магазину без виходу з месенджера. Функціонал кошика дозволяє переглядати додані товари, їхню кількість і загальну суму. Передбачено можливість оплати товару, що супроводжується автоматичним формуванням підсумкового повідомлення із зазначенням дати відправлення, яку бот розраховує залежно від часу замовлення.

Окремо реалізовано панель адміністратора, для користувачів із правами адміністратора доступні функції додавання та видалення товарів. Додавання товару реалізоване поетапно, де послідовно вводяться посилання на фото, назва, опис, ціна та категорія.

Також важливою складовою є функція технічної підтримки. Користувач може надіслати запит на консультацію, який надходить адміністратору. Після підтвердження запиту створюється окремий чат між адміністратором і користувачем, що моделює «живе» спілкування. Передбачено можливість завершення розмови обома сторонами.

РОЗДІЛ 2

ВИБІР ПРОГРАМНИХ ЗАСОБІВ ТА ТЕХНОЛОГІЙ

2.1 Мова програмування

Обравши платформу для створення бота, наступним кроком буде визначити технології, за допомогою яких ми створимо нашого бота.

Першим пунктом вибору програмних засобів буде мова програмування, завдяки якій ми будемо створювати нашого чат-бота.

Чат-бота можна розробляти будь-якою мовою програмування, якою володіє програміст, тому вибір припав на Python. **Python** - одна з найбільш динамічно розвиваних мов програмування у світі. Він є універсальною мовою програмування, яка підходить для розробки будь-чого: від мобільних застосунків і веб сайтів до ігрових рушіїв, систем штучного інтелекту та багато іншого. Однією з основних переваг Python серед інших мов програмування є те, що його синтаксис неймовірно простий. Його легко писати та читати, що значно полегшує вивчення даної мови програмування.

Python ↓	C# ↓
<pre>print("Hello, World")</pre>	<pre>using System; class Hello { static void Main() { Console.WriteLine("Hello, World"); } }</pre>

Рис 2.1 - Порівняння коду програми Hello World між Python та C#

На рисунку 2.1 зображено порівняння між мовою програмування Python та C#. Навіть користувач, який не розбирається у темі програмування, може сказати,

що код зліва виглядає набагато простішим, ніж справа, а це код всього-на-всього вивід повідомлення «**Hello World**» в консоль. Тому не дивно, що багато новачків, які тільки хочуть поринути в світ ІТ, обирають Python як свою першу мову програмування.

Таблиця 2.1

Порівняння основних характеристик Python і C#

Характеристики	Python	C#	Java
Простота синтаксису	Висока	Середня	Середня
Асинхронне програмування	Вбудоване	Потребує додаткових конструкцій	Підтримка через CompletableFuture
Кросплатформеність	Висока	Обмежена .NET-інфраструктурою	Висока
Час розробки	Короткий	Середній	Низький
Ступінь входу	Простий	Середній	Середній
Сфери застосування	Штучний інтелект, автоматизація	Корпоративні рішення, ігрова розробка	Android додатки, фінансова система, серверна розробка

Проаналізувавши таблицю 2.1, та порівнявши декілька мов програмування, можна підсумувати що, на кожній з цих мов можна розробити Telegram бота. Але варто зазначити, що все ж таки для цієї роботи, краще підійде Python. C# більше орієнтований на екосистему .NET, Windows та десктопні рішення, тому він краще підійде для цих задач. Java, хоч і має хорошу підтримку багатопотоковості, але високий рівень складності коду вимагає більше часу для написання навіть простих задач, що ускладнює швидкий старт проєкту. Натомість Python відрізняється своєю

простотою синтаксису та кросплатформеністю, що дозволяє зосередитися безпосередньо на самій логіці проєкту, а не на написанні складного коду. Це й робить його ідеальним варіантом для швидкої реалізації прототипу, або повноцінного сервісу .

2.2 Середовище розробки

Середовище розробки (IDE, Integrated Development Environment) - це програмне забезпечення, яке поєднує в собі текстовий редактор, компілятор або інтерпретатор. Основна мета IDE - підвищити продуктивність розробника за допомогою автоматизації рутинних завдань, підсвічування синтаксису, автозаповнення коду, швидкої навігації проєктом та можливості інтеграції з системами контролю версій.

Існує багато різних IDE, які активно використовуються, але ми поговоримо про самі популярні.

Visual Studio Code — універсальне середовище від Microsoft. Підтримує велику кількість нині популярних мов. Є редактором коду для розробки веб- та хмарних додатків.

Sublime Text — легкий, швидкий і функціональний текстовий редактор із широкими можливостями для розширення. Підходить для роботи з Python, HTML, JavaScript та іншими мовами. Має простий інтерфейс і високу продуктивність навіть на слабких пристроях.

PyCharm — спеціалізоване середовище для розробки на Python, яке забезпечує потужні інструменти для написання, тестування та налагодження коду.

З усіх перелічених редакторів коду, мій вибір пав на Visual Studio Code. Серед усіх готових рішень, він дуже сильно виділяється своїми фішками, до прикладу розберемо декілька з них. Visual Studio Code не вимагає велику кількість ресурсних

систем, що дозволяє його використовувати навіть на недорогих системах. Завдяки великій кількості розширень, будь-який користувач може додати підтримку будь-якої мови програмування чи технології. Окремо для Python доступне офіційне розширення від Microsoft, яке включає підсвічування синтаксису, автодоповнення та багато чого іншого. Гнучкість кастомізації, дозволяє налаштувати майже кожен елемент середовища під себе: від обраної теми, до налаштування клавіш швидкого доступу.

Обравши дане середовище розробки, ми отримали гнучкий, швидкий та зручний інструмент, який не лише забезпечив ефективність при розробці Telegram бота, а й дозволив якісно структурувати проєкт відповідно до сучасних стандарт індустрії.

2.3 Бібліотека Python Aiogram

Бібліотека в програмуванні - це набір функцій які розширюють можливості та функціонал мов програмування. У процесі розробки Telegram бота було прийнято рішення використовувати бібліотеку Aiogram, яка є однією з найпопулярніших рішень для створення асинхронних Telegram ботів на мові програмування Python. **Aiogram** побудована на основі асинхронного програмування, що дозволяє підвищити продуктивність і масштабованість застосунку без залучення багатопоточності.

Перед порівнянням Aiogram з іншими популярними бібліотеками того ж типу слід більш детально поговорити про *асинхронність*. **Асинхронність** - це метод виконання коду, при якому операції, що вимагають часу на виконання (наприклад, запит до сервера, або до бази даних), не блокують виконання іншої частини коду. Це нам говорить про те, що бот може обробляти декілька користувацьких запитів, при цьому не чекати завершення попередніх. На практиці це означає, що бот здатний швидко реагувати на запити декількох користувачів одночасно, не

створюючи зайвих затримок та зависань. Це особливо корисно в умовах великого навантаження на систему.

Ще однією важливою функцією бібліотеки Aiogram є **Finite State Machine (FSM)**. За допомогою цієї системи (або як її ще називають *машиною станів*) можна реалізувати покрокову логіку взаємодії з користувачем, де кожна дія залежить від попереднього кроку. Наприклад, реєстрація користувача в боті, де бот послідовно запитує у користувача ім'я, прізвище, номер телефону тощо. Завдяки FSM можливо реалізувати логічно пов'язані процеси без потреби створювати складні умовні оператори або додаткові бази даних для контролю станів.

Такий підхід дозволяє будувати структуровані сценарії, де покращують підтримку чистого і зрозумілого коду та спрощує масштабування логіки у подальшому.

Таблиця 2.2

Порівняння бібліотек для розробки Telegram ботів

Основні характеристики	Aiogram	Telebot	Python-telegram-bot
Підтримка асинхронності	Так	Ні	Часткова
Вбудований FSM	Так	Ні	Ні
Простота використання	Висока	Висока	Середня
Документація та підтримка	Активна	Частково активна	Дуже активна
Швидкість розробки	Висока	Висока	Середня
Гнучкість	Висока	Середня	Висока

Згідно з таблиці 2.2, можна зробити висновок, що з усіх цих бібліотек які розроблені для створення Telegram ботів, дуже сильно виділяється Aiogram з його функціоналом. Ті можливості та функціонал, який надає ця бібліотека, не стоять навіть поряд з іншими бібліотеками того ж типу. В інших бібліотеках реалізована лише частина функцій, або взагалі відсутні функції, які є в наявності в Aiogram. Тому цілком логічно, що саме ця бібліотека була обрана для реалізації нашого проєкту.

2.4 База даних SQLite

У розробці бота важливу роль відіграє зберігання даних користувачів, товарів, замовлень тощо. Тому для цього проєкту було обрано базу даних SQLite. **SQLite** - це легка, автономна база даних, яка не потребує використання сторонніх серверних програм. Усі дані зберігаються в одному файлі на локальному диску, що спрощує налаштування та використання. Ця база даних підтримує стандарти SQL та дозволяє виконувати всі основні операції.

```
def create_tables():  
    with conn:  
        conn.execute('''  
            CREATE TABLE IF NOT EXISTS users (  
                user_id INTEGER PRIMARY KEY,  
                first_name TEXT,  
                last_name TEXT,  
                phone_number TEXT,  
                city TEXT  
            )  
        ''')
```

Рис 2.2 - Приклад створення бази даних

На рисунку 2.2 показано приклад створення бази даних SQLite. У цьому прикладі ми створюємо базу даних зберігання даних користувача при його

реєстрації в нашому боті. Головною перевагою у використанні даної бази даних у тому, що вона за замовчуванням вбудована в стандарт бібліотек Python, це говорить нам про те, що ніяких додаткових компонентів для встановлення не знадобляться.

2.5 Створення бота в Telegram

Перш ніж почати писати код для бота, спочатку потрібно його зареєструвати у Telegram за допомогою сервісного бота під назвою BotFather. **BotFather** - це бот який є батьком усіх чат-ботів в Telegram, кожен бот який знаходиться в екосистемі Telegram проходив через нього. Для того щоб зареєструвати свого бота, треба у пошукову строку Telegram написати @botfather та перейти до нього, але тут треба бути обережним, бо поширені фейкові боти, які маскуються під інші, тому Telegram додав галочки на деяких ботів, це означає, що модерація телеграм перевірило його, та ним можна сміло користуватися. Щоб створити бота треба звернутися до нього за допомогою командою /newbot, а далі бот автоматично проведе всі етапи реєстрації бота. Після реєстрації ми отримаємо токен (ключ доступу до Telegram Bot Api для управління ботом), та затвердження тегу для доступу у додаток.

Таблиця 2.3

Команди для управління ботом

Команди	Характеристика
/help	Виводить список усіх доступних команд
/setname	Зміна імені бота у додатку
/setuserpic	Встановлення зображення в іконку чат-бота

Продовження таблиці 2.3

/mybots	Надає можливість змінити бота
/setprivacy	Вмикає приватний режим у чаті

Команди які показані на таблиці 2.3 - це набір інструментів у самому телеграмі. Дані команди дозволяють різними способами взаємодіяти з ботом для додаткового налаштування, видалення, зміни даних у разі втрати доступу та багато чого іншого. Таким чином, використання BotFather є обов'язковим елементом без якого не можливо почати роботу з Telegram ботом. Він надає можливості кожному користувачу реєструвати бота офіційно, без сторонніх ресурсів та інструментів, надає можливість початково налаштувати зовнішній вигляд та поведінку бота в месенджері.

2.6 Telegram API

Ще одним найголовнішим компонентом в екосистемі Telegram , який стоїть наряду з BotFather є Telegram API. **Telegram API** - це набір HTTP-запитів, який дозволяє приймати та відправляти повідомлення, обробляти кнопки, медіа, документи, геолокацію та інші типи даних у боті. Усі запити до API мають вигляд посилань, в яких йде протокол передачі гіпертекстових посилань, потім звернення до телеграму, обов'язково має бути вказаний токен бота і назва метода. Відповіді приходять в вигляді JSON-об'єкту, який має текстове поле description, в якому міститься опис результату запиту, а також поле зі значенням ok, що означає про успішність обробки запиту, у разі невдалого виконання запиту це поле буде мати значення false, та можна буде побачити причини помилки.

2.7 Платіжна система LiqPay

Щоб наш торгівельний Telegram бот почав по справжньому продавати товари, необхідно інтегрувати в нього платіжну систему. Серед всіх існуючих сервісів, які приймають платателі було обрано **LiqPay**. Це платіжна система, розроблена ПриватБанком, яка дозволяє інтегрувати прийом оплат на сайтах, мобільних додатках, бота тощо. Платформа підтримує картки Visa/MasterCard, Apple Pay, Google Pay та інші методи, а також надає зручне REST API для підключення до зовнішніх сервісів. Основними перевагами LiqPay є, простота інтеграція, підтримка нашої валюти гривні, швидкість проведення платежів, безпеність, ось чому вона була обрана як основна система платежу в нашому боті.

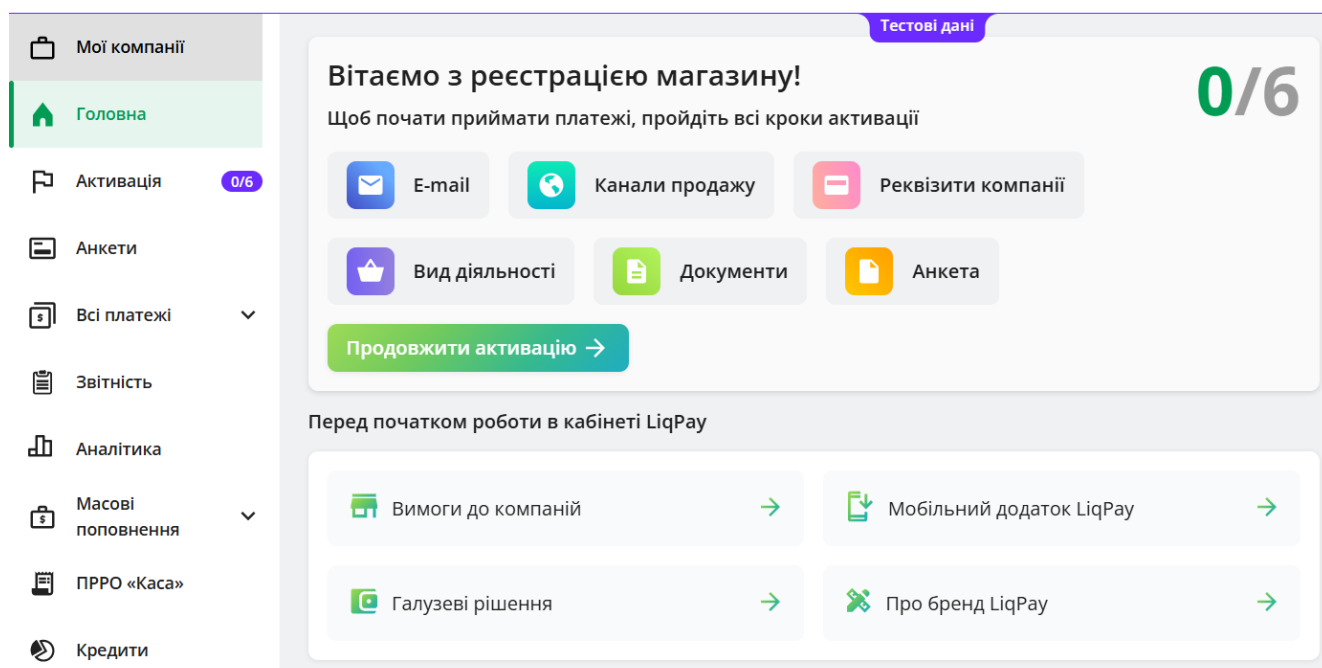


Рис 2.3 Інтерфейс платіжної системи LiqPay

РОЗДІЛ 3

РОЗРОБКА ТОРГІВЕЛЬНОГО TELEGRAM БОТА

3.1 Архітектура проєкту

Архітектура Telegram бота побудована на основі модульної системи, що дозволяє легко та зручно підтримувати код, масштабувати його та використовувати у подальших проєктах. Уся логіка програми розподілена в окремих директоріях, кожна з них відповідає за певну функціональну частину бота: обробники подій, клавіатуру, стани FSM тощо.

Основним скриптом який є точкою входу у код є **main.py**. Цей файл відповідає за запуск бота, ініціалізацію об'єкта диспетчера, підключення хендлерів, підключення FSM станів та запуск системи в режимі long poling (режим при якому постійно отримуємо нові повідомлення від серверів Telegram без затримок).

У файлі **config.py** міститься конфігураційні параметри бота такі як, токен бота, ключі від платіжних систем LiqPay та інші важливі змінні середовища. Цей підхід дозволяє легко змінювати конфігурацію кода, при цьому без необхідної зміни в самій логіці коду.

В модулі **handlers** знаходяться обробники повідомлень і callback запитів - саме тут знаходяться функції які відповідають на дії користувача, натискання кнопок, запит до техпідтримки, оформлення замовлення тощо. Для зручності, кожен з цих обробників був розміщений в окремий файл для зручного модернізування, та додавання нового функціоналу.

У файлі **states.py** реалізовано машину станів (FSM). Завдяки цьому файлу реалізована система покрокових сценаріїв: додавання нового товару в асортимент, реєстрація користувача в боті та стан для технічної підтримки. Бот запам'ятовує,

на якому кроці знаходиться користувач і згідно з цього реагує на його подальші кроки.

Модуль **keyboards** відповідає за клавіатуру в боті, тобто структурований зручний інтерфейс для користувача. Файли з кнопками поділені на два типи: на звичайні (ReplyKeyboardMarkup) та інлайн (InlineKeyboardMarkup), що додає гнучкості в побудові сценаріїв.

У блоці **db** знаходяться файл який створює таблицьки для баз даних та файл, який виконує запити до них. Такий метод дозволяє окремо розділити запити, та створення самих таблицьок, щоб не створювати зайвого навантаження на у читабельність кода, та уникнути дублювання SQL запитів у різних частинах кода.

Окремо реалізовано модуль для платіжної системи **LiqPay**. Вона реалізовано у вигляді API-запитів до серверу системи, створення підписів і формуванням посилань на оплату, які надсилаються користувачу в Telegram.

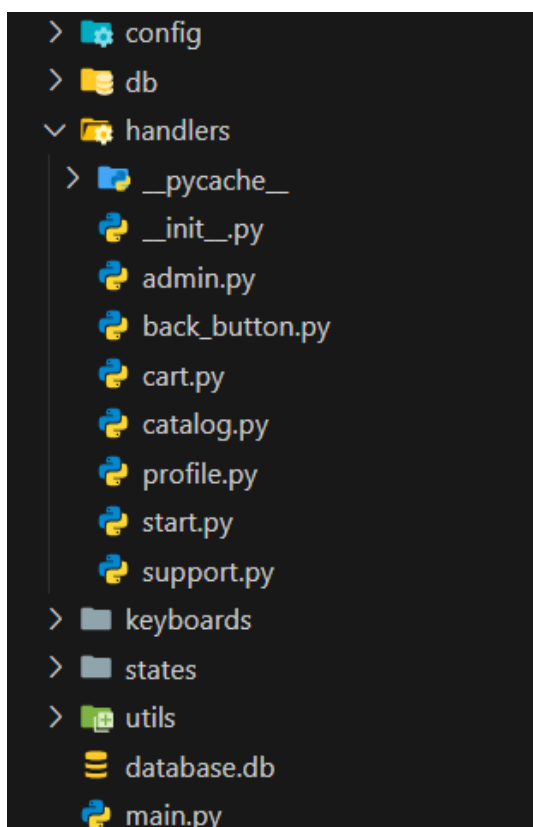


Рис 3.1 - Структура проєкту

Даний підхід такої архітектури демонструє гармонічне поєднання розподілу обов'язків між модулями, що допомагає оновлювати архітектуру проєкта, видаляти застарілий функціонал, або додавати нові функції до проєкту, при цьому сильно не пошкоджуючи у весь код програми.

3.2 Опис основного функціоналу проєкта

Telegram бот, який був створений в рамках даного проєкту, охоплює увесь повний цикл взаємодії з кінцевим користувачем, починаючи від реєстрації користувача в системі бота, вибір продукції за смаком користувача, оформлення товару та його оплати. Дані функції були розроблені з урахуванням зручності, інтуїтивності та практичної користі для кінцевого користувача та адміністратора.

Авторизація користувача в системі

Однією із базових функцій які розроблені в даному Telegram боті, є реєстрація користувача в системі. Реалізація даної частини вкрай необхідна та дуже важлива для цього проєкту, оскільки ця функція дозволяє ідентифікувати кожного користувача окремо, забезпечуючи тим самим персональну взаємодію, оформлення замовлення та подальший аналіз поведінки користувача.

Після першої взаємодії з ботом, якщо обліковий запис користувача ще не зареєстрований у системі, то бот запропонує йому зареєструватися. За допомогою FSM (рис. 3.2) реалізовано покроковий сценарій, у межах якого наш Telegram бот записує послідовно ім'я, прізвище, номер телефону та місце проживання. Усі ці дані зберігаються в базі даних SQLite та надалі використовуються для формування профілю користувача, адресної доставки замовлень та для зв'язку з ним (рис. 3.3). Також у користувача є можливість змінити свої дані, це потрібно для надання адміністрації магазину актуальної інформації, для відправки товару, та для зворотного зв'язку. У такому випадку, після натискання, процес реєстрації починається з початку, і після проходження даного етапу дані оновляються.

```
class Form(StatesGroup):
    first_name = State()
    last_name = State()
    phone_number = State()
    city = State()
```

Рис 3.2 - Стан для реєстрації користувача в Telegram боті

<u>user_id</u>	first_name	last_name	phone_number	city
Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
793264281	Єгор	Цапович	+380989090989	м.Київ

Рис 3.3 - База даних де зберігаються дані користувача

Використання даної системи реєстрації має за собою декілька ключових переваг. По-перше, це підвищує рівень довіри до клієнтів, адміністрація Telegram бота знає що працює з реальним клієнтом. По друге це дозволяє реалізовувати індивідуальні функції, персональний профіль, історія замовлень тощо.

Побудова інтерфейсу

Окремо б хотілося поговорити про користувацький інтерфейс. Перед розробкою бота стояла задача ефективно розробити зручний інтерфейс для користувачів, щоб вони мали приємний користувацький досвід. Тому ключовим рішенням стало використання вбудованих засобів керування ReplyKeyboardMarkup та InlineKeyboardMarkup.

Головне меню побудовано на основі класичної reply-клавіатури, яка автоматично з'являється після запуску бота. Вона містить такі кнопки: «Каталог», «Корзина», «Профіль», «Техпідтримка» та «Про нас».

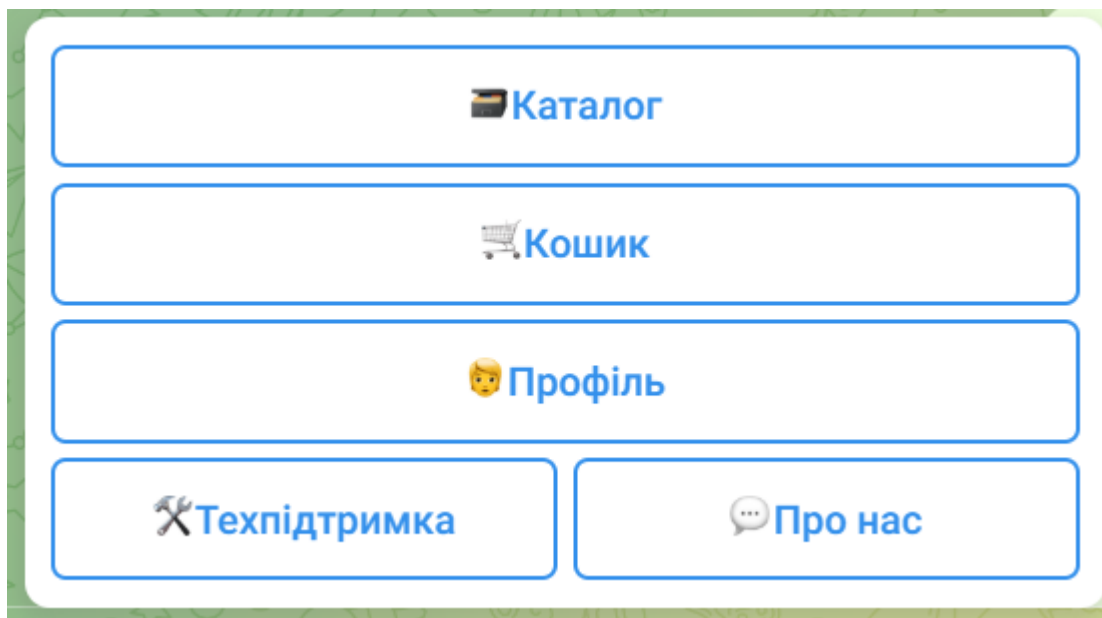


Рис 3.4 - Навігаційна панель з ReplyKeyboardMarkup кнопками

Також в інтерфейсі бота зустрічаються InlineKeyboardMarkup. Це такі кнопки які з'являються після деяких повідомлень. До прикладу візьмемо кнопку «Змінити дані», яка з'являється після натискання кнопки «Профіль». З точки зору звичайного користувача, це дуже зручне, бо кнопка відразу розташована після особистого профілю користувача, і її не треба шукати серед десятки інших кнопок які вбудовані в клавіатуру.

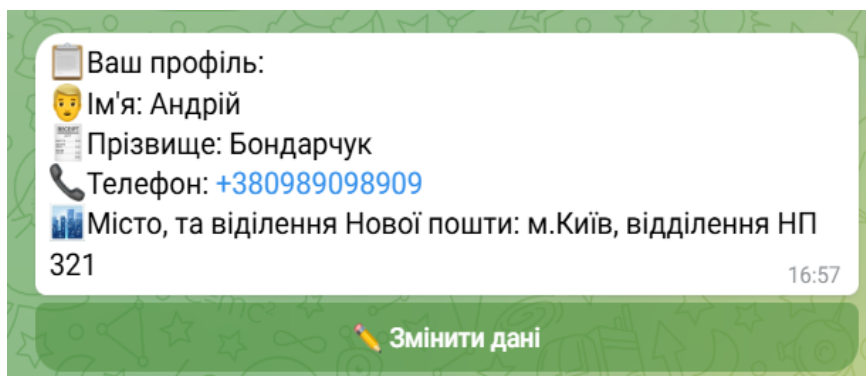


Рис. 3.5 - Повідомлення з прикріпленою InlineKeyboardMarkup

Інтерфейс містить візуальні елементи у вигляді емодзі, які допомагають швидше зорієнтуватися у змісті повідомлень та створюють дружню атмосферу. Всі повідомлення структуровані, поділені на блоки, а ключова інформація, така як ціна чи назва товару, завжди виділена та зрозуміла.

Чому саме були обрані кнопки, а не команди? На мою думку, кнопки мають набагато зручніший підхід у використанні, ніж команди. Поки користувач ознайомиться з усіма командами, які створені в боті, в нього зникне все бажання щось замовити. І навіть якщо він все ж таки ознайомиться з ними, не факт що користувач з може швидко опанувати їх. Він з легкістю може заплутатися, що призведе до втрати потенційного клієнта. Команди значно краще підходять, для використання адміністрації бота або технічному фахівцю, бо їх можна налаштувати на більш гнучкі функції для бота.

Таким чином, користувацький інтерфейс бота забезпечує просту, зручну і логічну навігацію, яка дозволяє користувачеві швидко освоїти роботу з ботом без потреби в додаткових інструкціях.

Функціонал адміністративної панелі

Адміністративна панель реалізована як інтегрована частина інтерфейсу, доступна лише для користувачів, у яких є певний ідентифікатор. У коді

передбачена перевірка ID, що гарантує, що жоден інший користувач не буде отримувати доступу до розширених функцій бота. Це дозволяє безпечно керувати асортиментом товару, при цьому не створюючи окремий інтерфейс, чи сайт для цього.

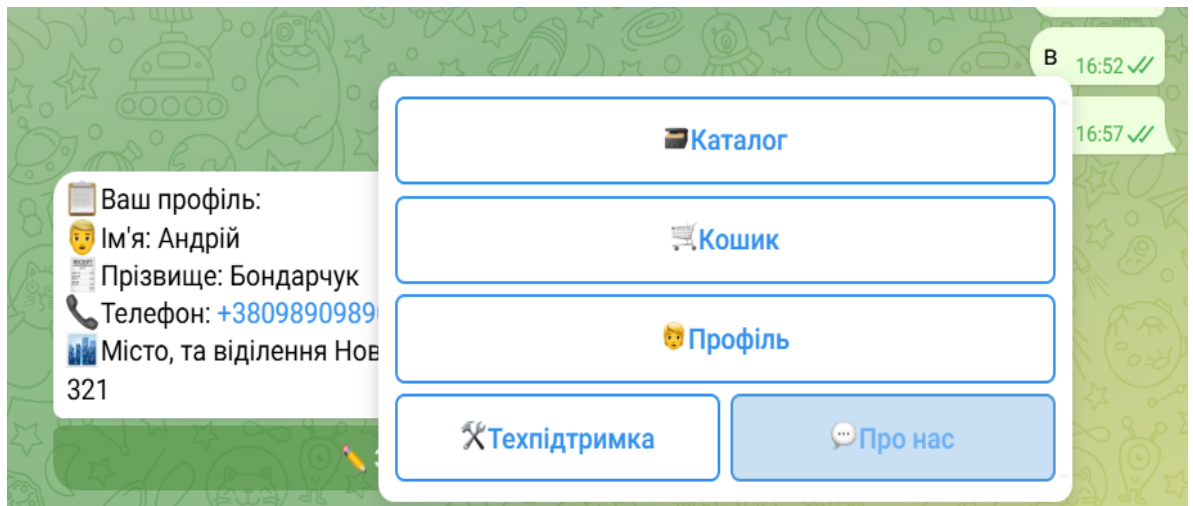


Рис 3.6 - Інтерфейс звичайного користувача

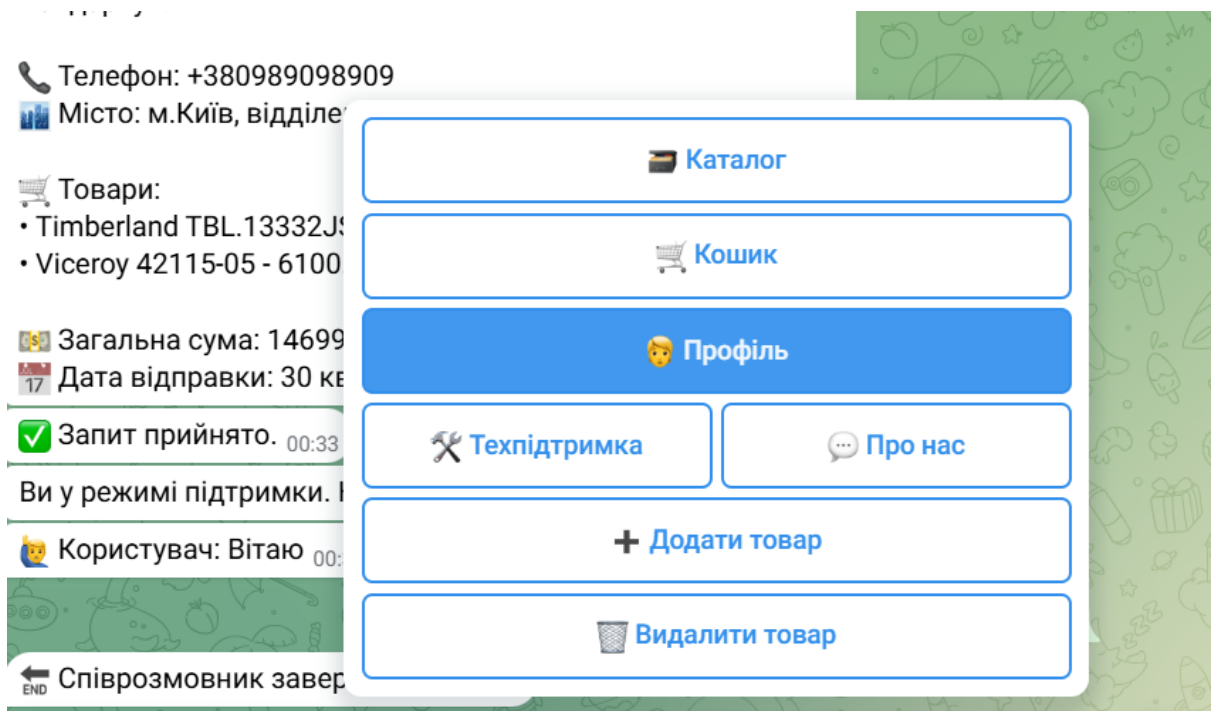


Рис 3.7 - Інтерфейс адміністратора

Як можна побачити на рисунку 3.6 та на рисунку 3.7, користувач та адміністратор мають різні візуальні інтерфейси. Адміністратору надається доступ до кнопок «Додати товар» та «Видалити товар» натомість, як у користувача дані кнопки відсутні.

При натисканні кнопки «Додати товар» запускається покроковий сценарій додавання нового товару до бази за допомогою FSM (рис 3.8), яка в свою чергу, не дозволяє адміністратору пропустити етап заповнення товару, чи ввести неповні дані, що гарантує правильність внесеної інформації.

```
class AddProduct(StatesGroup):  
    photo_url = State()  
    name = State()  
    description = State()  
    price = State()  
    category = State()
```

Рис 3.8 - Стан додавання нового товару

Після того як ми додали товар, адміністратор отримує повідомлення про успішно доданий товар (рис 3.9).

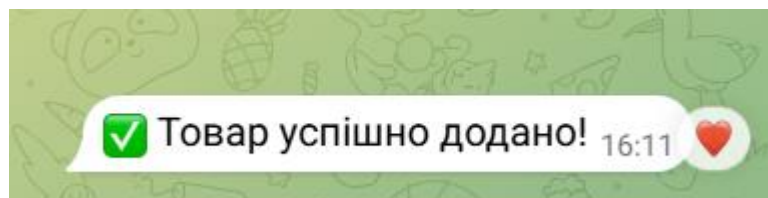


Рис. 3.9 - Інформація про успішно доданий товар

Далі цей товар записується в базу даних, а саме в таблицьку products. Як ми можемо бачити з рисунка 3.10 в таблиці зберігаються усі наші товари які були додано раніше. Товар зберігається за таким принципом: спочатку бот просить нас надати йому посилання на картинку для товару, далі це посилання зберігається в стовпчику photo_url, надалі бот просить надати назву нашого товару, яка в свою чергу буде зберігатися в стовпчику name, далі ми заповнюємо опис товару який зберігається в стовпчику description, заповнюємо стовпчик price, там будуть зберігатися ціна на наш товар та останній стовпчик який ми заповнимо буде category - він відповідає за категорію нашого товару, в нашу випадку це назва виробників годинників.

Таблиця: products						
Filter in any column						
	id	photo_url	name	description	price	category
	Фі...	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1	4	https://watch.ua/images/detailed/35/...	Timberland TBL.13332JSU/02	Стиль: Спорт; Виробник: Timberland; ...	8599.0	Timberland
2	5	https://watch.ua/images/detailed/37/...	Viceroy 42115-05	Стиль: Спорт; Виробник: Viceroy; ...	6100.0	Viceroy
3	6	https://watch.ua/images/thumbnails/...	Casio A159WA-N1	Стиль: Класика; Виробник: CASIO; ...	2400.0	Casio
4	7	https://watch.ua/images/detailed/...	Casio MTP-1221A-1AVEG	Стиль: Класика; Виробник: CASIO; ...	3300.0	Casio
5	11	https://cas-official.com.ua/images/...	Casio GA-2100-1AER	Чоловічий японський годинник ...	6000.0	Casio

Рис. 3.10 - База даних таблиці products

Далі поговоримо про функціонал кнопки «Видалити товар». При натисканні кнопки у нас з'являються всі наші додані товари, до кожного товару прив'язується кнопка «Видалити» (рис 3.11).

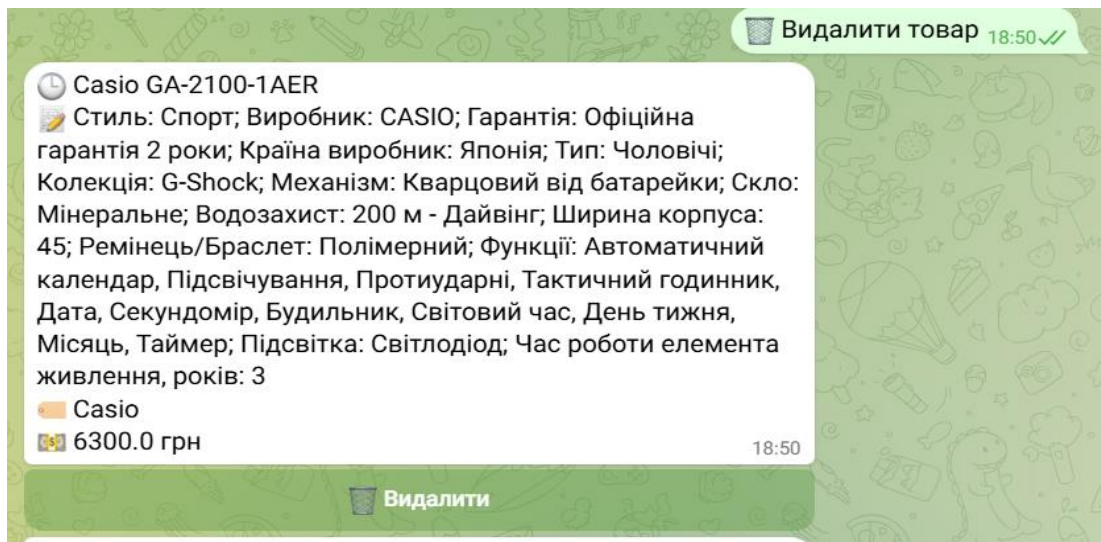


Рис 3.11 - Функціонал кнопки «Видалити товар»

Якщо адміністратор підтверджує видалення товару то після натискання кнопки «Видалити», візьмемо до прикладу годинник «Casio GA-2100-1AER», повідомлення з нашим товаром зникає, а бот надає тимчасове повідомлення, яке повідомляє нас, що товар видалено (рис. 3.12).

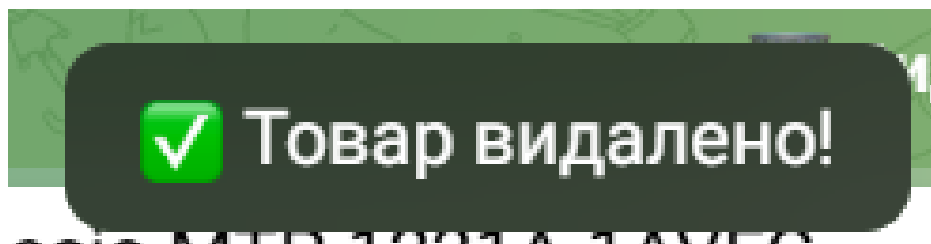


Рис. 3.12 - Інформація про видалення товару

Далі ми оновимо нашу таблицю, та побачити що годинник «Casio GA-2100-1AER» який ми видалили, насправді не зберігається в нашій базі даних

id	photo_url	name	description	price	category
Фі...	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
4	https://watch.ua/images/detailed/35/...	Timberland TBL.13332JSU/02	Стиль: Спорт; Виробник: Timberland; ...	8599.0	Timberland
5	https://watch.ua/images/detailed/37/...	Viceroy 42115-05	Стиль: Спорт; Виробник: Viceroy; ...	6100.0	Viceroy
6	https://watch.ua/images/thumbnails/...	Casio A159WA-N1	Стиль: Класика; Виробник: CASIO; ...	2400.0	Casio
7	https://watch.ua/images/detailed/...	Casio MTP-1221A-1AVEG	Стиль: Класика; Виробник: CASIO; ...	3300.0	Casio

Рис 3.13 - Оновлена табличка products

Ще одна функція яка доступна адміністратору - інформація про оплачений товар. Після того як користувач оплатив товар, адміністратору приходить повідомлення про сплату товару. На рисунку 3.13 ми бачимо таку інформацію: номер замовлення, прізвище та ім'я користувача, його особисті дані номер телефона, місто проживання та адреса поштового відділення, назву товару, загальна сума покупки та дата відправки. Це дозволяє зайвий раз не звертатися до користувача, не запитувати в його що він замов та куди його товар відправили. Ми просто зіб'ємо вже існуючу інформацію про користувача, та реалізуємо зручне повідомлення для адміністратора, якому в свою чергу буде цією інформацією користуватися.

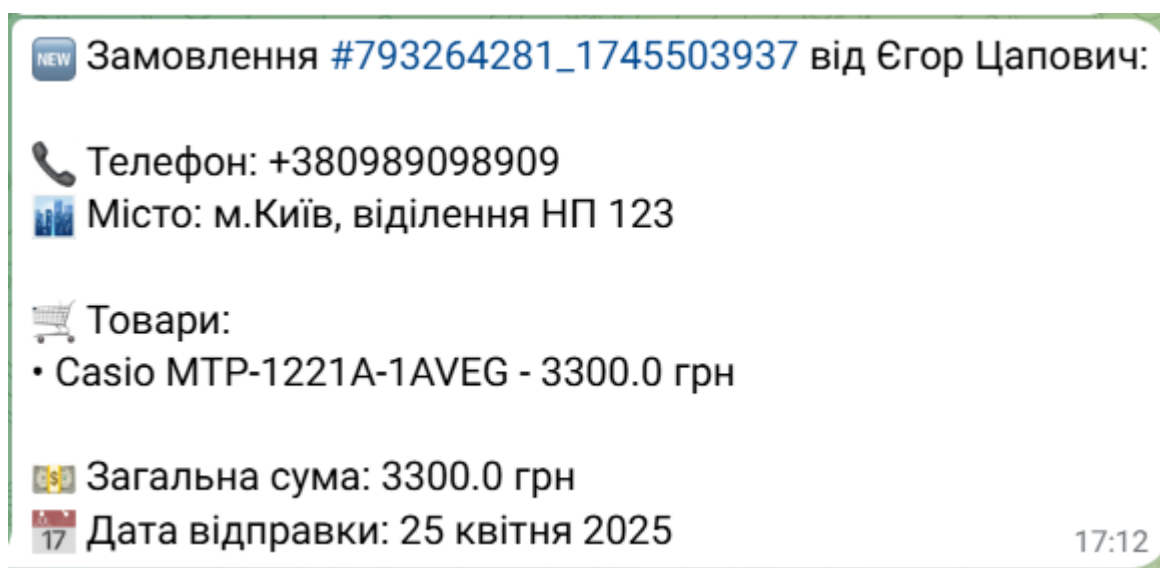


Рис 3.14 - Інформація про замовлений товар

Таким чином, ми реалізували адміністративну панель, яка прихована від звичайних користувачів. Ця панель дозволяє адміністратору який не знайомий з командами SQL легко та ефективно оновлювати асортимент товару, при цьому не використовуючи сторонні сервіси для цього.

Принципи роботи функції техпідтримки.

Тепер розглянемо функціонал за допомогою якого, користувачі зможуть комунікувати з адміністрацією бота. В Telegram боті реалізована функція техпідтримки, за допомогою якої, користувачі зможуть звертатися до адміністрації бота у разі виникнення різних питань.

Інтерфейс цієї функції реалізований максимально просто. Щоб активувати чат з адміністрацією всього-на-всього треба натиснути на кнопку «Техпідтримка», після чого користувач надсилає запит до адміністрації, а у відповідь йому відправляється повідомлення з проханням зачекати, поки адміністратор відповість на запит (рис 3.15).

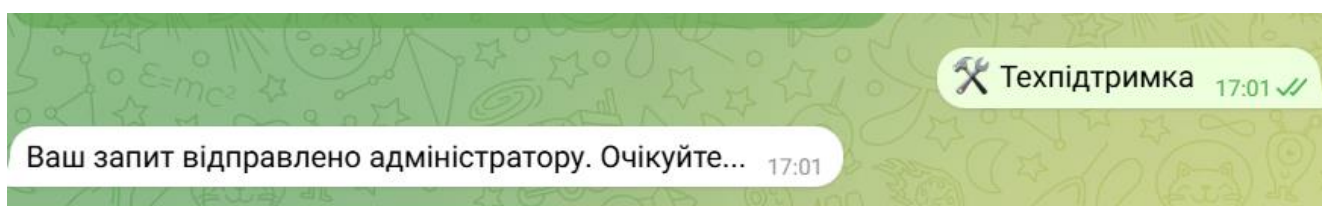


Рис. 3.15 - Відправлення запита до техпідтримки

Після того як користувач надіслав запит, адміністратору отримує повідомлення, що користувач потребує індивідуальної консультації. В цей проміжок часу користувача вносять до порожнього словника `active_chats`, який зберігає інформацію про активні сесії підтримки. Далі в адміністратора має два варіанта: прийняти запит або відхилити його. У випадку відхилення запиту,

адміністратору отримує повідомлення про відхилений запит, а користувач буде повідомлений, що адміністратор зараз недоступний, та попросить його звернутися пізніше. У випадку якщо адміністратор приймає запит, в нього та у користувача відкривається двосторінковий діалог (рис 3.16).

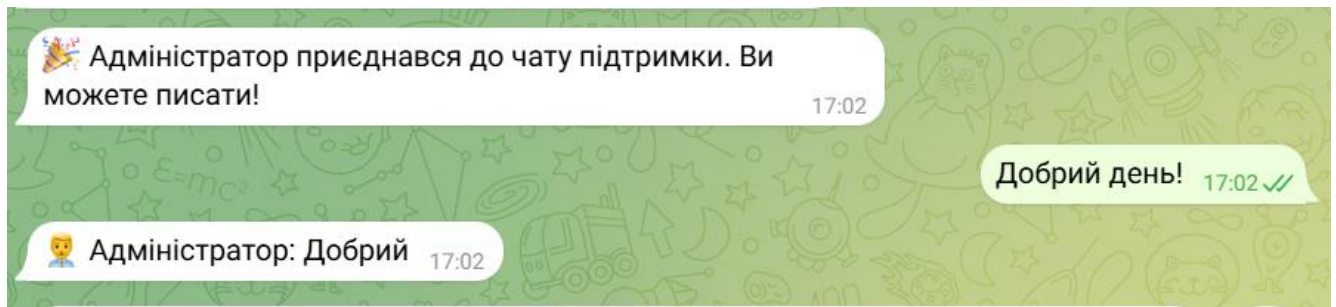


Рис 3.16 - Прийнятий запит на техпідтримку

Уся логіка обміну повідомленнями відбувається в режимі реального часу, при цьому сам бот виступає в ролі посередником між сторонами. Завдяки FSM, підтримка має чіткі стани: активний діалог та очікування відповіді (рис 3.17). Це дозволяє забезпечує контроль над станом користувача, унеможливорює накладення чатів один на одного. В подальшому цю функцію можна масштабувати для кількох адміністраторів.

```
# стан підтримки
class SupportStates(StatesGroup):
    waiting_for_support = State()
    in_support_chat = State()
```

Рис. 3.17 - Стан для технічної підтримки

Після завершення діалогу кожен з учасників має можливість вийти з нього за допомогою кнопки «Завершити розмову» (рис 3.18). Далі обидві сторони діалогу буду повідомленні про завершення сесії (рис 3.19).

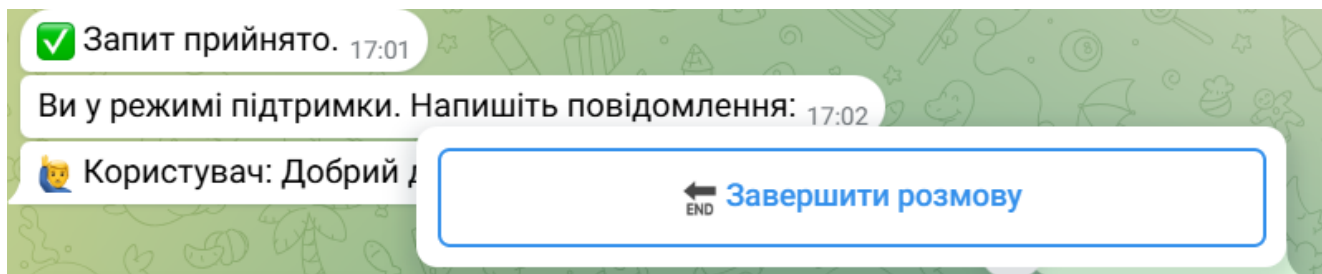


Рис. 3.18 - Кнопка «Завершити розмову»

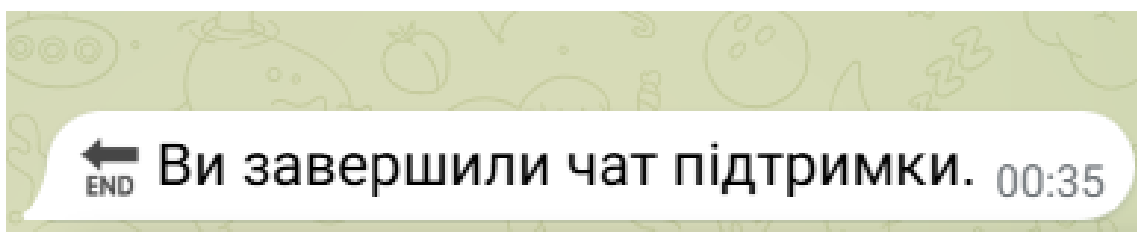


Рис. 3.19 - Повідомлення про успішне завершення діалогу

Таким чином, функція технічної підтримки в Telegram боті забезпечує оперативний та зручний зв'язок між користувачем і адміністратором. Це підвищує якість обслуговування клієнтів та формує позитивний користувацький досвід у нього.

Функціонал та можливості каталогу

Тепер поговоримо про функціонал елемента, який забезпечує взаємодію користувача з асортиментом товарів. Функція каталогу - це зручний спосіб перегляду наявних товарних позицій, які згруповані за категоріями, та можливість додавання їх до кошика без потреби у введенні тексту

Щоб скористатися каталогом, треба відкрити меню навігації та натиснути на кнопку «Каталог».

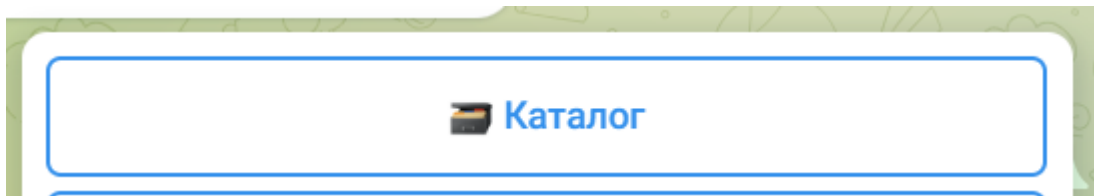


Рис 3.20 - Кнопка «Каталог»

Після цього бот звертається до бази даних і витягує всі унікальні категорії товарів, які знаходяться в таблиці products. Кожен бренд відображається у вигляді окремої кнопки (рис 3.21). Це дозволяє реалізувати фільтрацію кожного товару, що спрощує навігацію для користувача в пошуку конкретного товару або бренду.

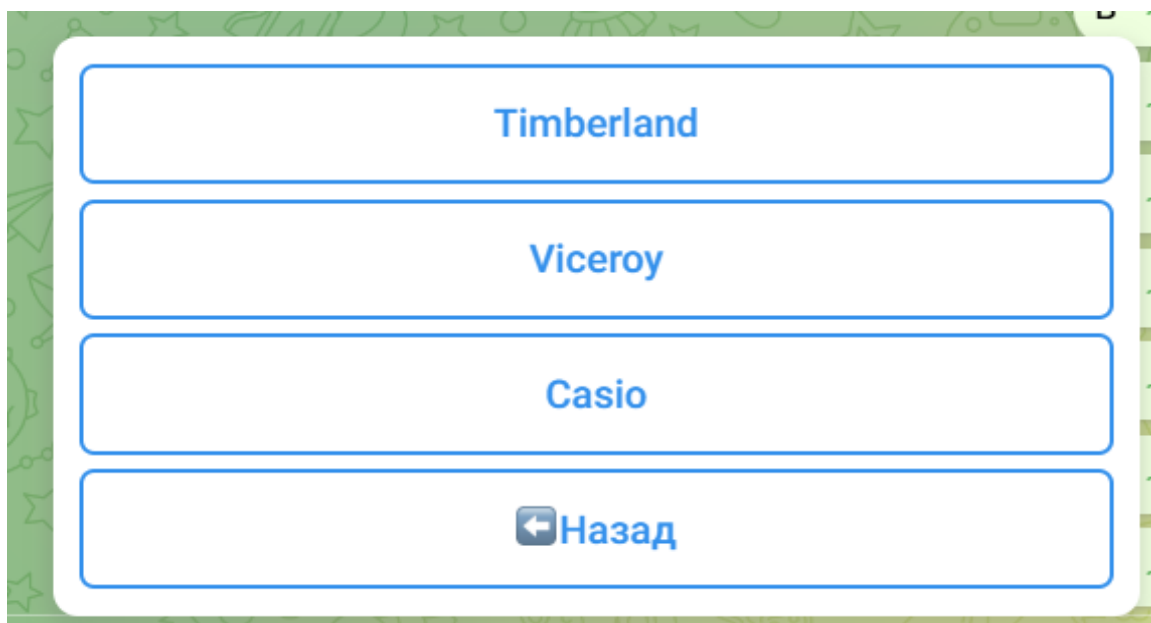


Рис. 3.21 - Кнопки категорії товарів

Після вибору категорії бот показує список товарів, які належать до відповідного бренду. Кожний товар представлений у вигляді повідомлення з фотографією товару, назвою, описом і ціною. Крім того, поруч з кожною карткою

товару розміщується кнопка «Додати в кошик», яка дозволяє відразу помістити товар в кошик, де він буде тимчасово зберігатися.

Така реалізація дозволяє повноцінно замінити класичний інтернет магазин, зберігаючи зручність, швидкість і візуальну привабливість. Завдяки кнопкам, користувачу не потрібно вводити велику кількість команд - увесь процес з ботом зводиться до декількох натискань.

Формування кошика та оплата замовлення

Формування функціоналу кошика і оплати замовлення є центральною частиною бізнес-логіки нашого бота, оскільки саме ці дії будуть забезпечувати завершення взаємодії користувача з магазином та конвертацію вибраних товарів у замовлення. Тож давайте розглянемо їхню логіку детальніше.

Після того, як користувач обрав товар з каталогу, та натиснув на кнопку «Додати в кошик», бот записує відповідний товар в базу даних - у таблицю cart.



The image shows a screenshot of a database interface. At the top, there is a label 'Таблиця:' followed by a dropdown menu showing 'cart'. Below this, there are two columns: 'user_id' and 'product_id'. Each column has a 'Фільтр' (Filter) button below it. The first row of data shows '1' in the 'user_id' column and '4' in the 'product_id' column.

	user_id	product_id
1	79326...	4

Рис. 3.22 - База даних таблиці cart

На рисунку 3.22 ми можемо бачити, що в таблиці є два стовпця: `user_id`, `product_id`. Кожен вибраний товар прив'язується до `user_id`, тобто до id аккаунта Telegram, що дозволяє зберігати індивідуальний кошик для кожного користувача окремо без змішування даних. В кожного товар є свій особистий id завдяки якому дуже легко відстежувати товар та зберігати ці дані `product_id`.



Рис. 3.23 - Функціонал кнопки «Кошик»

На рисунку 3.23 зображений функціонал кнопки «Кошик». Після натискання на неї, Telegram бот відправляє нам повідомлення в якому міститься інформація: назва товару, ціна та якщо позицій товару декілька, то підсумовується загальна сума кошика. Далі у користувач має доступ до деяких кнопок: «Оформити замовлення», «Очистити кошик» (рис. 3.24).

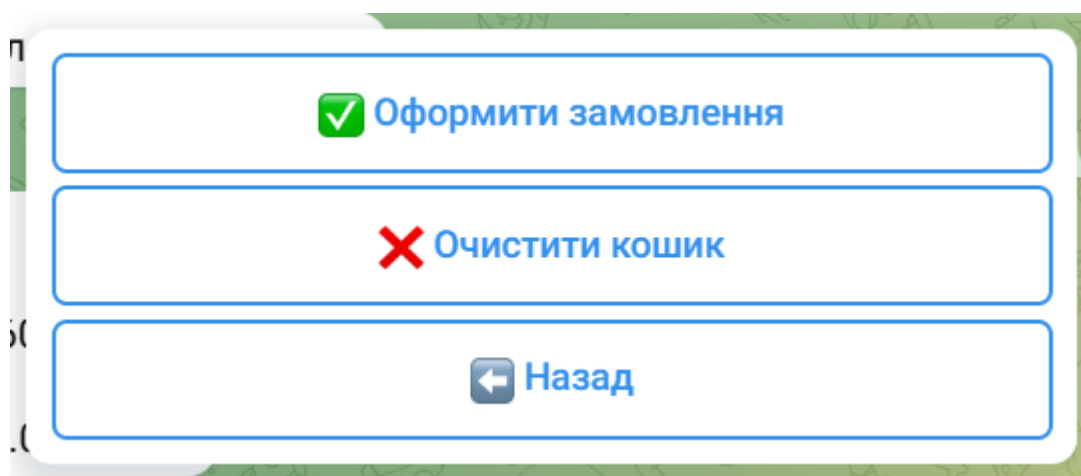


Рис 3.24 - Кнопки «Оформити замовлення» та «Очистити кошик»

Кнопка «Очистити кошик» відповідає за повне очищення кошика користувача. В такому разі товар з бази даних видаляється, а користувачу надсилається повідомлення, що його кошик повністю порожній. У разі натискання кнопки «Оформити замовлення» бот створює унікальний ідентифікатор

замовлення, зберігає дані про обрані товари, суму та статус замовлення в таблиці orders. Вся ця інформація надається користувачу у вигляді повідомлення та прикріпленою кнопкою «Оплатити замовлення» (рис 3.25).

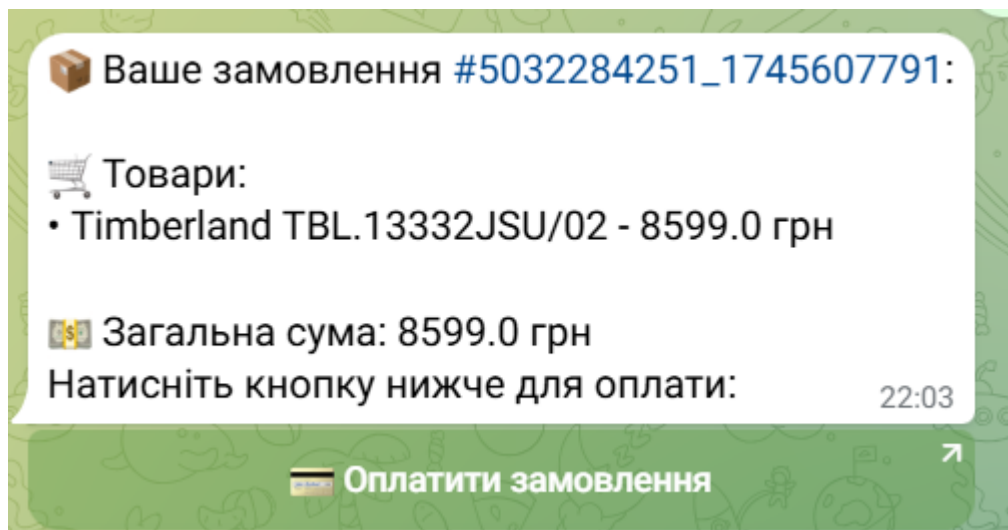
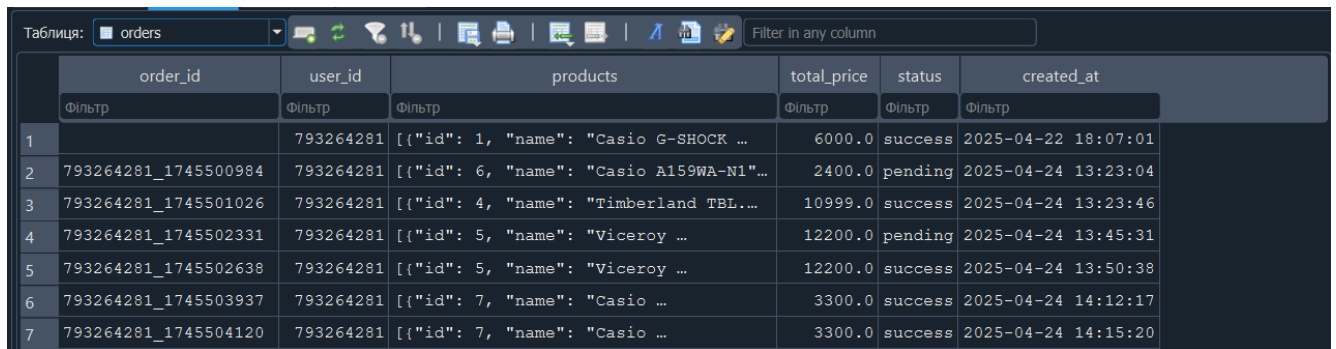


Рис. 3.25 - Структура повідомлення для оплати замовлення.

Після натискання кнопки бот автоматично генерує платіжне посилання через інтеграцію з платіжною системою LiqPay. Алгоритм передбачає формування платіжного запиту у форматі JSON-запиту, кодування його в Base64, створення криптографічного підпису, і передачу даних через офіційний API LiqPay.

Після переходу на сайт користувач повинен оплатити товар та після успішної оплати повернутися знову до бота. Далі користувач отримує повідомлення від бота, з успішною оплатою та датою відправки цього товару. Вся ця інформація про оплачений товар зберігається в таблиці orders (рис 3.28).



	order_id	user_id	products	total_price	status	created_at
	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр	Фільтр
1		793264281	[{"id": 1, "name": "Casio G-SHOCK ...	6000.0	success	2025-04-22 18:07:01
2	793264281_1745500984	793264281	[{"id": 6, "name": "Casio A159WA-N1"...	2400.0	pending	2025-04-24 13:23:04
3	793264281_1745501026	793264281	[{"id": 4, "name": "Timberland TBL. ...	10999.0	success	2025-04-24 13:23:46
4	793264281_1745502331	793264281	[{"id": 5, "name": "Viceroy ...	12200.0	pending	2025-04-24 13:45:31
5	793264281_1745502638	793264281	[{"id": 5, "name": "Viceroy ...	12200.0	success	2025-04-24 13:50:38
6	793264281_1745503937	793264281	[{"id": 7, "name": "Casio ...	3300.0	success	2025-04-24 14:12:17
7	793264281_1745504120	793264281	[{"id": 7, "name": "Casio ...	3300.0	success	2025-04-24 14:15:20

Рис. 3.28 - База даних таблицьки orders

Ця таблиця має шість стовпчиків: order_id, user_id, products, total_price, status, created_at. Перший стовпчик призначений для зберігання унікального ідентифікатора замовлення. Другий потрібен для впізнання користувача, щоб не було плутанини з замовленнями. Третій стовпчик потрібен, для зберігання короткої інформація про товар: назва товару, ціна, id цього товару, та ціну. Четверний відповідає за фінальну суму товарів, тобто якщо в стовпчику products вказані декілька товарів, то їхня спільна сума буде зберігатися тут. П'ятий стовпчик потрібен для надання статусу замовлення, якщо згенерувалося посилання на товар, але по якійсь причині він ще не оплачений, то його статус буде pending, як тільки користувач оплатить товар, статус негайно зміниться на success. Останній стовпець зберігає дату замовлення, це потрібно для того щоб користувач знав коли він оплатив товар, та коли йому його відправлять.

Завдяки цьому процесу закінчується повний цикл: від вибору товарів, до підтвердження оплати. Уся логіка майже реалізована у межах Telegram, що робить роботу з ботом простою, зручною та ефективною.

3.3 Тестування та перевірка коректності роботи бота

Після розробки основного функціоналу, наступним кроком для кожного продукту є його тестування для перевірки його працездатності, стабільності та відповідностей до поставлених вимог. Тестування Telegram бота проводилося поетапно, приділяючи особу увагу перевірки основних його функцій, взаємодію його з користувачем, а також цілісність збережених даних.

Першим кроком етапом тестування було протестувати, як бот взагалі реагує на наших клієнтів. Було застосовано кілька сценаріїв. Перший сценарій, перевірити як бот реагує на користувача який не зареєстрований в системі бота, а другий сценарій, перевірити як бот реагує на користувача зареєстрованого в системі.

У першому випадку наш бот реагує так: коли не зареєстрований користувача намагається додати товар у свій кошик, бот його повідомляє, що він спочатку повинен зареєструватися. У випадку якщо не зареєстрований користувач буде намагатися зв'язатися з адміністратором, то він буде попереджений про те, що він не зареєстрований. У разі натискання користувача на кошик, буде виконаний аналогічний алгоритм дій.

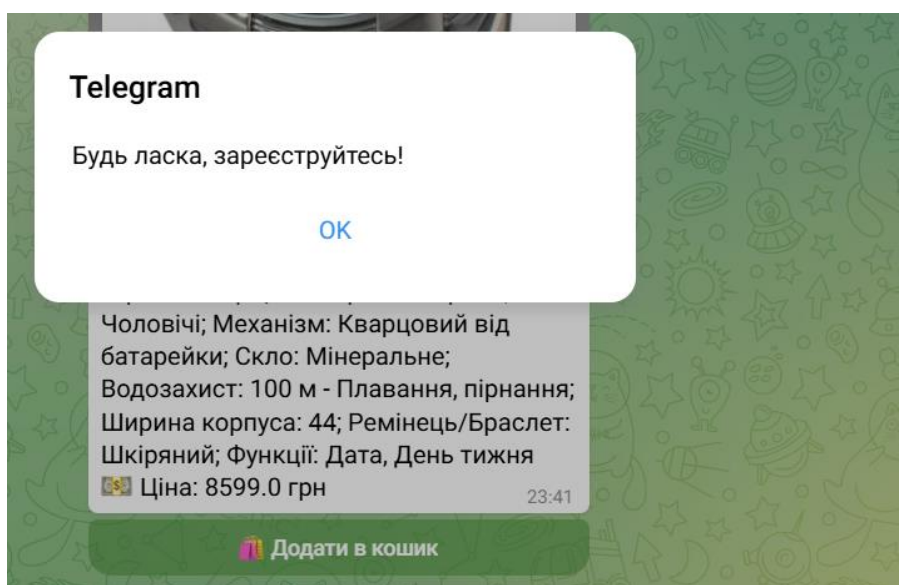


Рис. 3.29 - Не зареєстрований користувач натискає на кнопку «Додати в кошик»



Рис. 3.30 - Не зареєстрований користувач натискає на кнопку «Техпідтримка»

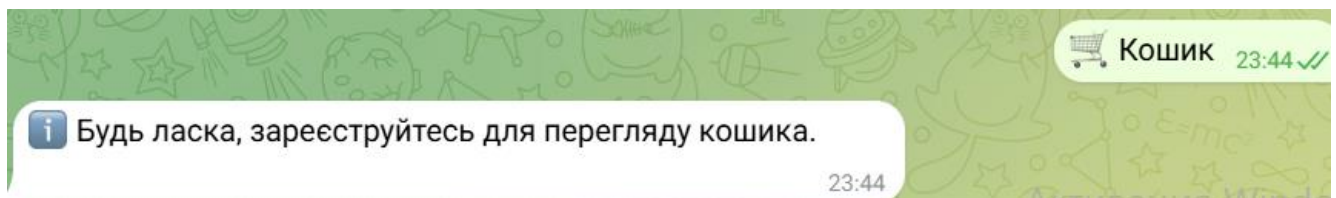


Рис. 3.31 - Не зареєстрований користувач натискає на кнопку «Кошик»

Далі ми перевіряємо функціонал зареєстрованого користувача. Почнемо з кнопки «Профіль». Це нам потрібно для того, щоб з'ясувати чи зберігає наш бот дані користувача. Ми перевірили кнопку «Змінити дані». Як наслідок перевірок в нас все коректно працює. Бот зберігає дані зареєстрованого користувача в базі даних, при натисканні «Змінити дані» бот запускає заново процес реєстрації. При введенні некоректного формату номеру телефону наш бот повідомляє про це користувача, та попросить ввести коректний формат. І як фінал цього, бот зберігає змінені дані в базу, та оновлює профіль користувача.

5032284251	Євген	Конбх	+380989098909	м.Київ, відділення НП 321
------------	-------	-------	---------------	---------------------------

Рис. 3.32 - Збереженні дані користувача до зміни даних

5032284251	Андрій	Бондарчук	+380989098909	м.Київ, відділення НП 321
------------	--------	-----------	---------------	---------------------------

Рис. 3.33 - Змінені дані користувача

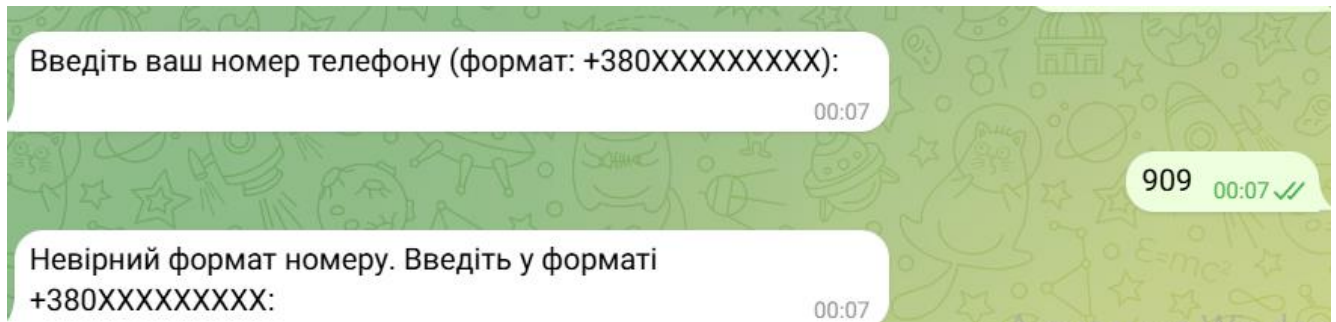


Рис. 3.34 - Повідомлення про некоректний формат номеру телефона

Наступним етапом було перевірити коректність роботи каталогу. Було протестовано вивід категорій, відображення товарів із правильними фото, описами та цінами, а також функція додавання товарів до кошика. Функціонал кошика було протестовано за різними умовами: додавання товару, очищення всього кошика, оформлення замовлення. Бот коректно рахував загальну суму замовлення та переходив до етапу оплати через LiqPay. Процес інтеграції було проведено в тестовому режимі. Система коректно приймала платежі, перевірила статус замовлення, а бот оновлював статус автоматично в базі даних.

793264281_1746627246	793264281	[{"id": 11, "name": "Casio ...	6000.0	pending	2025-05-07 14:14:06
----------------------	-----------	--------------------------------	--------	---------	---------------------

Рис. 3.35 - Статус не оплаченого замовлення

793264281_1746627246	793264281	[{"id": 11, "name": "Casio ...	6000.0	success	2025-05-07 14:14:06
----------------------	-----------	--------------------------------	--------	---------	---------------------

Рис. 3.36 - Статус оплаченого замовлення

Панель адміністратора також була протестована на працездатність. За підсумками тестування, адміністративна панель працює коректно та ніяких проблем не було виявлено. Доступ до панелі мав тільки адміністратор, користувач не зміг скористатися нею. Додавання та видалення товару теж коректно працює.

Доданий товар відображається в базі даних, та показується в каталозі, а видалений товар зникає з бази даних та більше ніде не з'являється.

Окрему увагу ми приділили функція технічної підтримки. В ході тестування користувач зміг надіслати запит, адміністратор зумів прийняти або відхилити його, а сама сесія спілкування завершувалася належним чином із повідомленням про завершення.

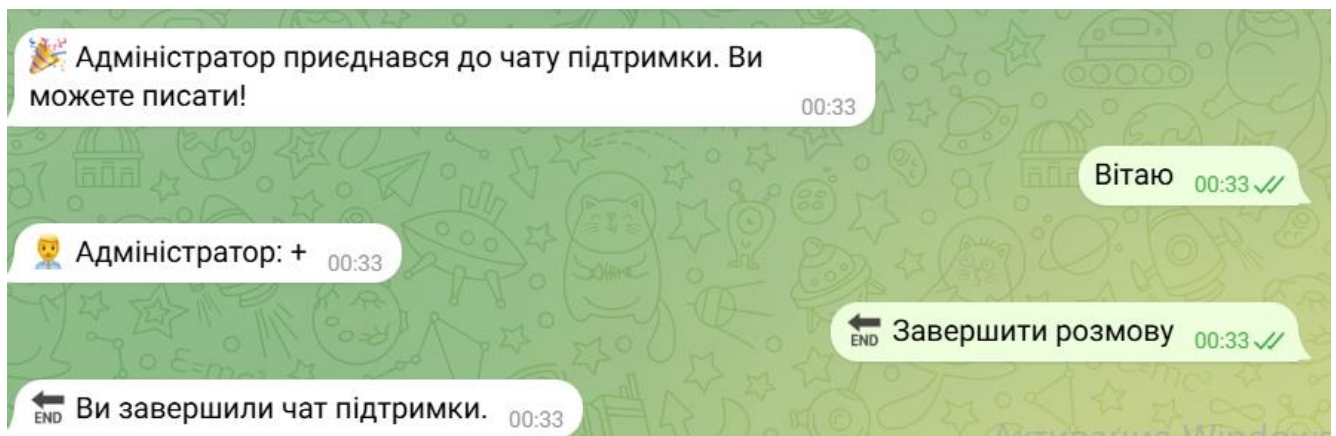


Рис. 3.37 - Діалог з боку користувача

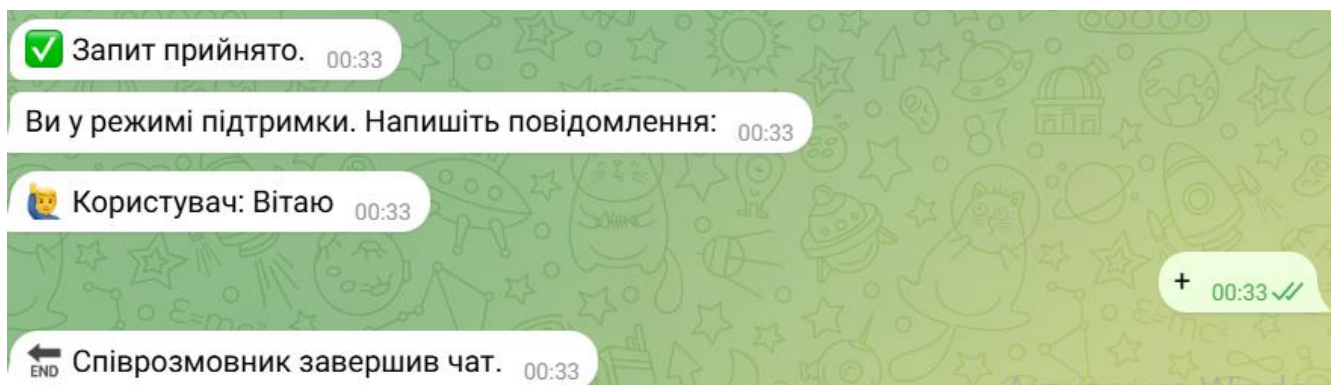


Рис. 3.38 - Діалог з боку адміністратора

Таким чином було протестовано основні функції бота в максимально наближених умовах реального користування. Бот адекватно реагував на дії користувача, правильно обробляв виняткові ситуації та гарантував цілісність збережених даних.

3.4 Оцінка ефективності впроваджень

У процесі створення Telegram бота стояла чітка мета - розробити повноцінне рішення, яке б могло полегшити роботу малого та середнього бізнесу: від обробки замовлення до керування товаром та спілкування з клієнтами.

Перш за все, бот кардинально змінив взаємодію користувача з магазинами. Тепер не треба заходити на сторонні сайти чи завантажувати додатки - усе доступно в самому Telegram. Клієнт натискає в боті на каталог, обирає товар, додає його в корзину та оформлює його за декілька хвилин. Сам інтерфейс інтуїтивно зрозумілий, головний функціонал оформлений за допомогою кнопок, що дозволяє навіть недосвідченим користувачам комфортно користуватися сервісом.

Для адміністратора бот також полегшує роботу. Тепер йому не треба збирати вручну дані про замовлення, зберігати дані клієнта в окремих файлах чи чатах. Уся інформація зберігається в базі даних бота, а керування товаром оформлено через окремі кнопки: видалення та додавання товару в кілька кліків без сторонніх сервісів.

Одним з найголовнішим результатом впровадження стало інтегрування платіжної системи LiqPay. Це дозволило автоматизувати процес оплати, а й зняти додаткове навантаження з адміністратора щодо підтвердженні замовлення. Коли оплачується замовлення бот автоматично перевіряє статус оплати та змінює його, а вже потім надсилає повідомлення адміністратору. Цей процес усунув потребу в ручному режимі перевіряти чеки та виписки з банку.

Технічна підтримка - ще один елемент який довів свою ефективність. Раніше система листування потребувала окремого листування в сервісі чи на сторонніх платформах. Зараз достатньо всього натиснути на одну кнопку, при цьому не виходячи з інтерфейсу бота, для миттєвого спілкування між клієнтом та продавцем.

Загалом ефективність бота проявляється на декількох рівнях. Перше - це скорочення часу взаємодії з клієнтами. Друге - це зменшення кількості помилок під

час оплати за обробці замовлення. І, що важливо бот зменшує технічний бар'єр для малого та середнього бізнесу - все працює в знайомому інтерфейсі Telegram.

Таким чином цей бот виконав поставлені йому задачі: зменшив час на взаємодію з клієнтами, автоматизував продажі, зменшив рутину, і залишився простим у використанні як для клієнтів, так і для адміністраторів. Саму в простоті та ефективності полягає його цінність.

ВИСНОВОК

У результаті виконання дипломної роботи, було створено Telegram бота, який замінює повноцінно онлайн магазин. Він реалізує повний цикл обслуговування клієнта - від ознайомлення товару до його оплати. Цей проєкт не обмежується лише демонстраційним функціоналом - його одразу можна впроваджувати в роботу малого та середнього бізнесу без потреби у великих фінансових та технічних вкладень.

В процесі розробки було використано сучасні технології: мова програмування Python, бібліотека Aiogram, платіжна система LiqPay, Telegram API, а також система зберігання даних SQLite. Це дозволило досягти гнучкості, стабільності та простоти підтримки коду.

Було розроблені базові функції для користування: каталог товару, кошик, технічна підтримка тощо. Також додали адміністративну панель, яка допомагає адміністратору підтримувати коректну роботу бота. Інтерфейс бота максимально інтуїтивний, що допомагає його освоїтись навіть менш досвідченим користувачам в інтернеті. Тестування показали, що система працює стабільно, не потребує складної інфраструктури та легко масштабується.

Варто підкреслити, що Telegram як платформа має великий потенціал. Сотні користувачів вже звикли до цих ботів, що відкриває широкі можливості для масштабування таких рішень у майбутньому.

Загалом, проєкт виконав своє головне завдання - дешеве рішення, яке допомагає без клопоту взаємодіяти з клієнтами, при цьому приймати мінімальну участь в його підтримці.

Перспективи розвитку проєкту

Хоч наш бот вже готовий до повноцінної роботи, але варто зазначити що він має великий потенціал для подальшого розвитку. До прикладу можна реалізувати інтеграцію з Google таблицями чи іншими подібними сервісами. Це нам дозволить отримувати зручно інформацію про кількість замовлених товарів за місяць, щоб можна було підвищити загальну кількість продажів, завдяки аналізу популярних товарів, та позбавлення інших які не користуються попитом серед клієнтів. Як варіант можна інтегрувати інші системи оплати такі як Google Pay, Apple Pay тощо.

Не поганим варіантом буде додати окремі вкладки з популярними чи акційними товарами, які допоможуть в швидкому темпі заманити до нас потенційних клієнтів.

У майбутньому наш бот може стати однією частин мультиканального сервісу - наприклад одночасно обслуговувати клієнтів з різних месенджерів через єдину базу.

Таким чином, створений Telegram бот - це не лише готовий інструмент, а перспективна платформа для подальшого масштабування. Його легко підтримувати розвивати та адаптувати до нових бізнес потреб.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1.BEST Chatbot Statistics for 2025 | Master of Code Global. *Master of Code Global*. URL: https://masterofcode.com/blog/chatbot-statistics?utm_source=chatgpt.com .
- 2.Techlabs M. Can Chatbots Help Reduce Customer Service Costs by 30%?. *Medium*. URL: <https://chatbotsmagazine.com/how-with-the-help-of-chatbots-customer-service-costs-could-be-reduced-up-to-30-b9266a369945> .
- 3.ТОП-20 найпопулярніших мобільних застосунків України у 2024 році. *UA.NEWS*. URL: <https://ua.news/ua/technologies/top-20-najpopulyarnishyh-mobilnyh-zastosunkiv-ukrayiny-u-2024-rotsi> .
- 4.Telegram – нова ера у спілкуванні. *Telegram*. URL: <https://telegram.org/?setln=uk>.
- 5.Telegram APIs. *Telegram APIs*. URL: <https://core.telegram.org/api>.
- 6.Welcome to Python.org. *Python.org*. URL: <https://www.python.org/>.
- 7.Python 3.13 documentation. *Python documentation*. URL: <https://docs.python.org/3.13/>.
- 8.Enterprise-grade AI-native Customer eXperience platform for revenue growth | Engati. *Enterprise-grade AI-native Customer eXperience platform for revenue growth | Engati*. URL: <https://www.engati.com/>.
- 9.Сервіс email, sms, web push розсилки. *SendPulse*. URL: <https://sendpulse.ua/>.
- 10.Dialogflow. *Dialogflow*. URL: <https://dialogflow.cloud.google.com/#/getStarted> (date of access: 08.05.2025).
- 11.Microsoft. Visual Studio Code - Code Editing. Redefined. *Visual Studio Code - Code Editing. Redefined*. URL: <https://code.visualstudio.com/>.
- 12.Sublime Text - the sophisticated text editor for code, markup and prose. *Sublime Text - Text Editing, Done Right*. URL: <https://www.sublimetext.com/>.

- 13.JetBrains. PyCharm: The only Python IDE you need. *JetBrains*. URL: <https://www.jetbrains.com/pycharm/>.
- 14.aiogram 3.20.0.post0 documentation. *aiogram 3.20.0.post0 documentation*. URL: <https://docs.aiogram.dev/en/v3.20.0.post0/>.
- 15.SQLite Documentation. *SQLite Home Page*. URL: <https://www.sqlite.org/docs.html>.
- 16.*LiqPay Documentation*. URL: <https://www.liqpay.ua/doc>.
- 17.Shaw Z. Learn Python the Hard Way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code. Pearson Education, Limited.
- 18.Matthes E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press, Incorporated, 2015. 560 p.
- 19.Barry P. Head First Python: A Brain-Friendly Guide. O'Reilly Media, 2016. 624 p.
- 20.Automate the Boring Stuff with Python: Practical Programming for Total Beginners. 2nd ed. No Starch Press, 2019. 592 p.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інформаційних систем та технологій

КВАЛІКАЦІЙНА РОБОТА

на тему:

«Торгівельний чат-бот мовою програмування Python»

на здобуття освітнього ступеню бакалавра
зі спеціальності 126 Інформаційні системи та технології
освітньо – професійної програми Інформаційні системи та технології

Виконав: Єгор Цапович, ІСД-41

Науковий керівник роботи:

Олег Сеньков

Актуальність теми: У сучасному світі автоматизація бізнес-процесів є ключовим фактором ефективної комунікації з клієнтами та підвищення продажів. Одним з найбільш перспективних інструментів автоматизації у сфері електронної комерції є чат-боти, які можуть замінити людину у побутових речах: відповідати на питання, надавати консультацію, обробляти замовлення, приймати оплату.

Об'єкт дослідження: чат-бот який інтегрований у сферу електронної комерції

Предмет дослідження: технологія розробки Telegram бота за допомогою мови програмування Python та сторонніх бібліотек.

Мета дослідження: створити Telegram бота який буде допомагати продавати товари, або послуги підприємства.

Завдання дослідження:

- провести аналіз платформ для інтеграції бота
- провести аналіз готових рішень
- обрати технології для розробки бота
- розробити архітектуру проекту
- провести аналіз отриманого результату

ВИКОРИСТАННІ ТЕХНОЛОГІЇ

При розробці бота були обрані наступні компоненти:

- 1) Мова програмування Python
- 2) Бібліотека Aiogram
- 3) Месенджер Telegram
- 4) База даних SQLite
- 5) Платіжна система LiqPay



Python



Aiogram



Telegram



SQLite

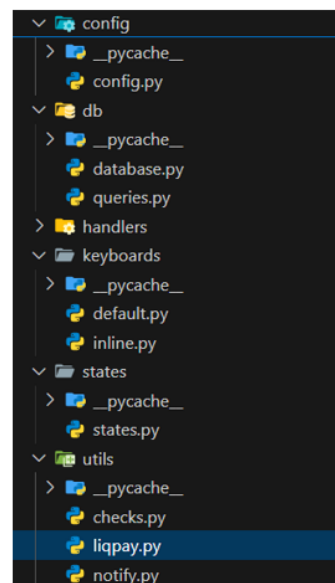


LiqPay

3

АРХІТЕКТУРА ПРОЄКТУ

Проект складається з модульної архітектури, що дозволяє легко додавати або оновлювати функціонал.



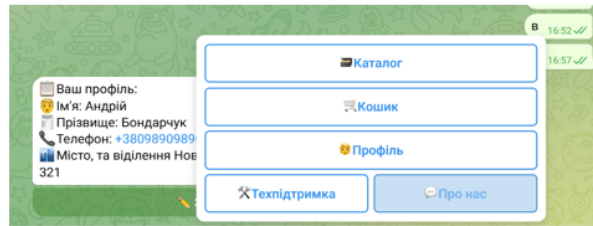
Архітектура проєкту

4

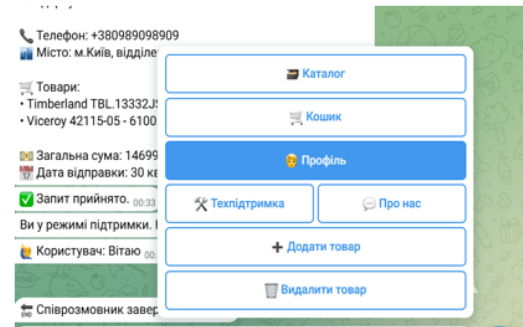
ВІЗУАЛЬНИЙ ІНТЕРФЕЙС

Інтерфейс бота представлений у вигляді кнопок .

Окремі персони які занесені до списку адміністратора, мають набагато більший функціонал.



Інтерфейс користувача



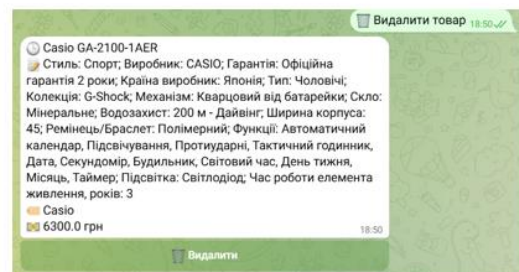
Інтерфейс адміністратора

5

ФУНКЦІЇ АДМІНІСТРАТОРА



Функціонал кнопки «Додати товар»



Функціонал кнопки «Видалити товар»

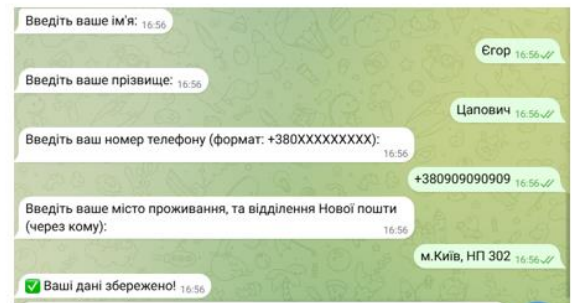
6

ФУНКЦІОНАЛ КНОПКИ «ПРОФІЛЬ»

Після натискання на кнопку профіль, бот відправляє повідомлення з персональними даними користувача. Під повідомлення підкріплена кнопка яка дозволяє оновлювати персональні данні клієнта.



Функціонал кнопки «Профіль»

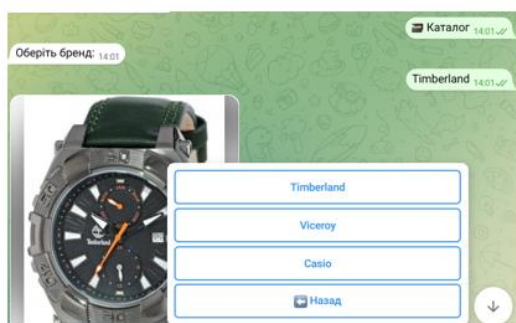


Функціонал кнопки «Змінити дані»

7

ФУНКЦІОНАЛ КНОПКИ «КАТАЛОГ»

При натисканні на кнопку «Каталог» з'являється мені де пропонують обрати бренд товару. Після натискання на бренд користувачу прийдуть повідомлення з усіма наявними товарами цього бренду. Під кожним товаром є кнопка яка дозволяє додати бажаний товар до кошика.



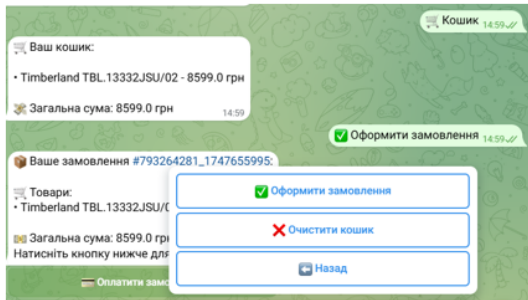
Функціонал кнопки «Каталог»



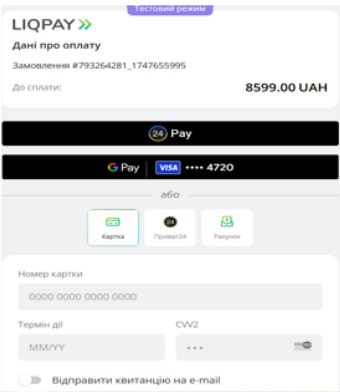
Картка товару

8

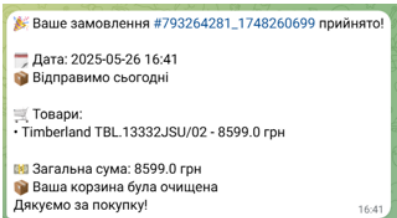
ФУНКЦІОНАЛ КНОПКИ «КОШИК»



Функціонал кнопки «Кошик»

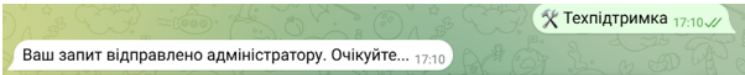


Інтерфейс платіжної системи

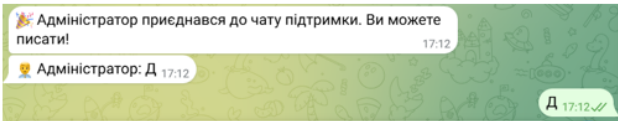


Інформація про успішне замовлення

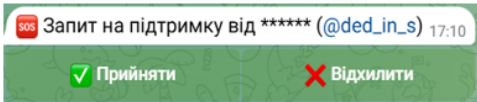
ФУНКЦІОНАЛ КНОПКИ «ТЕХПІДТРИМКА»



Запит користувача до адміністратора



Діалог між адміністратором та користувачем



Запит адміністратора від користувача

ВИСНОВОК

В ходів виконання роботи, було створено чат-бота, який може продавати товари, тим самим замінюючи повноцінно магазин та персонал. Було використано технології як: Python, Aiogram, Telegram, SQLite та LiqPay. Реалізовано функції такі як: кошик, каталог, товару, оформлення та оплата товару, функція техпідтримки та профіль користувача.

Представлений приклад роботи повністю готовий, з можливістю подальшого розвитку, шляхом додавання додаткових функцій.

11

Апробація результатів дослідження

VI Міжнародну науково-технічну конференцію «Сучасний стан та перспективи розвитку IoT», проходила 15 квітня 2025 року. Тезу на тему «Застосування Telegram чат-ботів у сфері e-commerce» була опублікована у збірнику присвяченому цій конференції.

ДЯКУЮ ЗА УВАГУ!!