Міністерство освіти і науки України
Сумський державний університет
Навчально-науковий інститут бізнесу, економіки та менеджменту
Кафедра економічної кібернетики

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему «Автоматизована система обліку клієнтів та замовлень онлайн магазину на базі Telegram бота»

Виконав студент 4 курсу, групи ЕК-81а
(номер курсу) (шифр групи)
Спеціальності 051«Економіка («Економічна кібернетика»)»
Півень А.В.
(прізвище, ініціали студента)
Керівник д.е.н., професор Кузьменко О.В.
(посада, науковий ступінь, прізвище, ініціали)

Суми – 2022 рік

РЕФЕРАТ

Кваліфікаційної роботи бакалавра на тему

«АВТОМАТИЗОВАНА СИСТЕМА ОБЛІКУ КЛІЄНТІВ ТА ЗАМОВЛЕНЬ ОНЛАЙН МАГАЗИНУ НА БАЗІ TELEGRAM БОТА»

Студент Півень Артем Вадимович

Об’єктом дослідження **є** створений чат-бот «Techno_Market», що виконує функцію підбору та замовлення електронної техніки.

Предметом дослідження **є** інтернет технології в електронному бізнесі та особливості їх застосування при створенні telegram-бота.

Метою дипломної роботи є розробка Telegram-бота для є-бізнесу з функцією підбору по категоріям електронної техніки для користувачів магазину.

Завданнями роботи є:

- дослідити теоретичні основи чат-боту telegram;
- визначити всі переваги та недоліки боту;
- створити технічне завдання для створення системи
- дослідити мову програмування для створення бота;

За результатами роботи була створена автоматизована система обліку клієнтів та замовлень онлайн на базі Telegram бота

Випускна робота містить 54 сторінки , 5 таблиць , 15 рисунків, список використаних джерел з 22 найменувань, 2 додатки.

Рік виконання випускної роботи 2022

Рік захисту роботи 2022

Міністерство освіти і науки України

Сумський державний університет

Навчально-науковий інститут бізнесу, економіки та менеджменту

Кафедра економічної кібернетики

ЗАТВЕРДЖУЮ
Завідувач
кафедри
д.е.н.,
професор
_____О.В.
Кузьменко“__”_____2022р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
БАКАЛАВРА
спеціальність 051 Економіка (Економічна кібернетика)студенту 4 курсу, групи ЕК-81а

Півень Артем Вадимович
(прізвище, ім’я, по батькові студента)

1. Тема роботи Автоматизована система обліку клієнтів та замовлень онлайн магазину на базі telegram бота
2. Затверджена наказом по університету від «09.05.2022» року № 0324-VI
3. Термін подання студентом закінченої роботи «9» червня 2022року
4. Мета кваліфікаційної роботи розробка Telegram-бота для є-бізнесу з функцією підбору по категоріям електронної техніки для користувачів магазину.
5. Об’єкт дослідження створений чат-бот «Techno_Market», що виконує функцію підбору та замовлення електронної техніки
6. Предмет дослідження інтернет технології в електронному бізнесі та особливості їх застосування при створенні telegram-бота
7. Кваліфікаційна робота виконується на матеріалах відкритих даних технічної документації

8. Орієнтовний план кваліфікаційної роботи, терміни подання розділів керівникові та зміст завдань для виконання поставленої мети

Розділ 1. Дослідження та аналіз чат ботів – 09.06.2022

(назва – термін подання)

У розділі 1. 1.1 Загальна характеристика чат-ботів, 1.2 Основні недоліки та переваги при використанні телеграм ботів, 1.3 Технічне завдання до створення системи

(зміст конкретних завдань до розділу, які має виконати студент)

Розділ 2. Реалізація системи обліку клієнтів та замовлень онлайн магазину на базі Telegram бота – 09.06.2022

(назвал – термін подання)

У розділі 2. 2.1 Стек технологій який використовувався при розробці чат-бота, 2.2 Описання структури і функціоналу Telegram-бота, 2.3 Оцінка очікуваного ефекту від впровадження системи автоматизації.

(зміст конкретних завдань до розділу, які повинен виконати студент)

9. Консультації з роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Кузьменко О.В., д.е.н., професор	04.04.2022	05.04.2022
2	Кузьменко О.В., д.е.н., професор	04.04.2022	05.04.2022

10. Дата видачі завдання: «04» квітня 2022року

Керівник кваліфікаційної роботи _____ О.В. Кузьменко
(підпис) (ініціали, прізвище)

Завдання до виконання одержав _____ А.В. Півень
(підпис) (ініціали, прізвище)

ЗМІСТ

ВСТУП

Актуальність. Сьогодні технології не стоять на місці: розумні будинки, електромобілі, хмарні технології, 3D-друк, з’являються нові технології та тенденції, деякі з яких можуть серйозно вплинути на всю IT-індустрію. Відносно нещодавно всі говорять про програми, але нещодавня розробка ботів або скорочено «чат-робот» віщує світле майбутнє для комунікацій та маркетингу. Так дійсно ж чат-боти необхідні в сьогоденні?

Насамперед, це дуже спрощує роботу для бізнесу. Саме за допомогою ботів можна скоротити витрати та мінімізувати навантаження на працівників, що надає їм виконувати понад 80% завдань, Оскільки роботи стають розумнішими, завдання стануть складнішими, і роботи отримають можливість вчитися, виконуючи більш складні завдання. Ринкові умови, що постійно змінюються, висока швидкість прийняття рішень, необхідність багатозадачності та зниження ризиків в управлінні активами вимагає сучасного підходу до організації бізнесу. Вирішення дедалі складніших внутрішніх і зовнішніх середовищ підприємств — повна автоматизація бізнес-процесів. Це вивільняє цінні ресурси для стратегічного планування та централізованого управління ключовими сферами бізнесу. У зв’язку зі збільшенням обсягу інформації в організаційних інформаційних системах (ІС) виникає потреба в автоматизації інформаційних процесів, прискоренні та використанні більш складних методів обробки. Автоматизація бізнесу – це часткова або повна трансформація стереотипних операцій і бізнес-завдань під управлінням спеціалізованих інформаційних систем або складного апаратного та програмного забезпечення. Таким чином, можна вивільнити людські та фінансові ресурси для підвищення продуктивності та ефективності стратегічного управління.

Метою дипломної роботи є розробка Telegram-бота для є-бізнесу з функцією підбору по категоріям електронної техніки для користувачів магазину.

Для досягнення поставленої мети необхідно виконати такі завдання:

- дослідити теоретичні основи чат-боту telegram;
- визначити всі переваги та недоліки боту;
- створити технічне завдання для створення системи
- дослідити мову програмування для створення бота;

Об’єктом дослідження **є** створений чат-бот «Techno_Market», що виконує функцію підбору та замовлення електронної техніки.

Предметом дослідження **є** інтернет технології в електронному бізнесі та особливості їх застосування при створенні telegram-бота.

Результати дипломної роботи можуть використовуватися у майбутньому для створення власного інтернет-магазину.

Методи дослідження**.** Технології створення чат-боту за допомогою мови програмування Python на основі бібліотеки aiogram.

РОЗДІЛ І. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ЧАТ-БОТІВ

1.1 Загальна характеристика чат-ботів

Термін бот походить від слова «робот». Скорочена назва вказує на суть концепції - спеціальної утиліти, призначеної для виконання операцій за заданим алгоритмом і через людино-машинний інтерфейс. Хто такі боти в Інтернеті? Це своєрідні симулятори рухів живої людини, і часом вони настільки складні, що їх неможливо відрізнити від реальних користувачів. Після отримання запиту програма працює за заздалегідь розробленим сценарієм, а її правила залишаються незмінними. Інтелектуальні роботи призначені для виконання монотонних і рекурсивних робіт. У порівнянні з людиною, він швидше реагує на процеси і не втомлюється від одноманітних рухів. Використання роботів необхідно для виконання таких видів операцій: однотипні, що чергуються один з одним, швидкі - людина зазвичай не впорається з їх виконанням. Програма нового століття з багатьма корисними функціями та допомагає людям у їхній роботі.

Якщо використувати платформи, які користуються ботами, то саме великий відсоток стосується мессенджерів. Що таке месенджер? У більшості випадків поняття месенджер відноситься до програми чи мобільного додатка, в якому можливо обмінюватися повідомлення з іншими користувачами. Месенджерами почали користуватися ще в 90-х роках, але їхня популярність проявила себе лише в епоху пост-ПК. Швидке зростання мобільних пристроїв зробило їх доступними для кожної людини. На сьогоднішній день у багатьох людей є смартфон під рукою[6].

Найпопулярнішими месенджерами в Україні є WhatsApp, Viber та Telegram. Створення чат-ботів роспочалася тоді, як Telegram відкрив платформу для створення певних облікових записів, обробки та відправки інформації. На данний час більша частина ботів базується на Telegram. Зручність та простота використання чат-бота полягає в тому, що вони в легко доступні в каожному мессенджері та їх легко знайти.

Чат-боти - це спеціальні акаунти, за якими не закріплена будь-яка людина, а повідомлення, надіслані з них або на них, обробляються зовнішньою системою. Крім того, для користувача спілкування з ботом виглядає як просте листування з справжньою людиною.

Чат-бот — це розумна програма, яка живе у месенджерах та виконує різні функції.

Функції чат-бота:

1) Підтримка клієнтів

Чат-бот допоможе замінити незручний FAQ на сайті, який іноді не одночасно можна побачити, чи зможе відповісти на типові питання клієнта.

Бот може працювати 24 години на добу та розвантажить ваших співробітників.

2) Клієнтський сервіс

Чат-бот завдяки своїй функціональності надає користувачам робити різного роду придбання, вибір товару. В роздрібній торгівлі, з постійним розширенням асортименту, складніше шукати конкретних товарів. Після невеликого аналізу бот зрозуміє, що цікавить клієнта і відправить пряме посилання.

3) Маркетинг

Чат-бот - це ще один маркетинговий інструмент, який допоможе розповсюджувати контент, підтримувати лояльність клієнтів та збирати аналітику. За допомогою нього можна робити розсилки, інформувати клієнтів про акції, збирати коментарі про товари або послуги, якість обслуговування.

4) Робота всередині компанії

Чат-боти допомагають оптимізувати такі процеси як: бронювання переговорів, інформування працівників про дати відпустки, розклад корпоративного транспорту, терміни зарплати та багато іншого.

5) Рекрутинг

6) Функціональність просунутих ботів - це первинний збір інформації про кандидатів. На основі співбесіди з чат-ботом менеджер з персоналу вирішує, яких кандидатів слід запросити на живу співбесіду, хто має пройти тестове завдання, а кому слід відмовити.

Існує кілька варіантів класифікації чат-ботів, але проаналізувавши їх усі, виділимо два види: бізнес-класифікація чат-бот додатків та класифікація чат-ботів за технічним типом.

Чат-бот у контексті Telegram — це стороння програма, доступна за системою Telegram. Користувач взаємодіє з ботом, надсилаючи боту повідомлення, команди та вбудовані запити. Для керування ботом потрібно спеціальний API, який використовує протокол HTTPS. За допомогою Telegram-бота можна:

1. Отримувати спеціальні сповіщення та новини.

2. Інтеграція зі сторонніми сервісами: Gmail, Wiki, YouTube, Github тощо. 3. Отримати платіж від користувача.

4. Створювати корисні утиліти

5. Розробка ігор на основі технології HTML

Telegram-бот - це спеціальний обліковий запис з точки зору месенджера Користувачі можуть взаємодіяти двома способами:

1. Відкрийте чат з ботом або додайте його до групи та надішліть Текст або команда в полі за допомогою введення повідомлення.

2. Відкрийте будь-який чат і введіть «@bot name» у поле введення повідомлення. Ось як вони працюють з вбудованими ботами[1].

1.2 Основні недоліки та переваги при використанні телеграм ботів.

Недоліки ботів

При використанні ботом було виявлено, що при авторизації у вашому сервісі всередині бота неможливо замаскувати пароль, в результаті чого він залишається в листуванні. Доводиться рекомендувати видалити його вручну.

Telegram намагається синхронізувати історію повідомлень на всіх пристроях. Однак іноді при синхронізації пропадають кнопки, якими керується робота. Для цього передбачена команда перезапуску бота (/start).

Як було сказано вище, вся інформація виводиться у вигляді повідомлень, які не заміняють старі, а скроляться нагору в міру появи нових. Це нагадує старі комп'ютерні інтерфейси з «лістингами» та «скролінгами». У програмі, як ми всі звикли, стара інформація (наприклад, напис або картинка) просто замінюється на нову. У боті картинка з'являється внизу у вигляді нового повідомлення, а стара переміщається нагору.

У програмі можна натискати активні елементи інтерфейсу (картинки, кнопки). У роботі активні кнопки розташовані тільки внизу, на місці клавіатури. До такого способу взаємодії слід звикнути.

Все, що робить кнопка - це відправляє боту текст, написаний на ній. Це накладає певні обмеження розробки бота.

Telegram не має вбудованого магазину ботів, тому виявлення вашого бота користувачами практично повністю лежить на ваших плечах. У разі це питання неактуальне, т.к. бот використовується операторами за вказівкою адміністратора чат-центру.

Бот «живе» всередині Telegram, отже, неможливо відокремити повідомлення від бота та інших адресатів. У бота немає власної іконки, як у програми, відсутній доступ до різних функцій операційної системи,

наприклад, до розташування користувача, рекламних платформ, in-app платежів і т.п[3].

Що стосується безпеки, то всередині самого месенджера боти не можуть заподіяти щось без втручання користувача, тому що це майже як, облікові записи користувачів, які лише керуються алгоритмічно. Як і звичайні користувачі, нудних ботів можна повністю видалити с можливість заблокувати назавжди. Найважливіше те, щоб бот не ініціював розмову – користувач сам перший надсилає йому певний запит, тому бот не має турбувати користувача без дозволу на це.

Основною перевагою чат-ботів Telegram є те, що збільшують продажі. Завдяки їм у короткі терміни можна отримати цільову аудиторію. При правильному використанні бота він допоможе у продажах та надасть необхідну інформацію покупцю.

Як показує практика, використання чат-бота для підтримки клієнтів може зменшити близько 40% часу консультантів в онлайн-чатах. Також, аналізуючи статистиску, можно дізнатися, що більшість звернень до чат-боту приходить в неробочий час. Безумовно, великою перевагою є те, що чат-бот може надавати відповідь відразу декільком користувач. В середньому час відповіді варіюється близько 30 секунд і до 1 хвилини.

Великою перевагою чат-ботів є кросплатформність. Готового робота не складно адаптувати до інших платформ. Тому можливо одразу замовити бота для Telegram, Facebook Messenger та WhatsApp, не розробляючи їх окремо.

Також однією з переваг є оптимізація ресурсів. Чат-боти економлять ресурси компанії. Таким чином можна автоматизувати щоденний процес «відповіді на найчастіші запитання». Проаналізуйте, які запити найчастіше надходять до вашої підтримки. Якщо ви можете підготувати єдині відповіді на ці запити - напишіть їх у чат-бот. Виграють усі: для бізнесу це можливість звільнити працівника від рутини та використати його енергію та час на

важливіші завдання, а для клієнта — шанс швидко закрити свою проблему. Підвищити залучення користувачів. За допомогою чат-ботів компанія може швидше надавати заявникам необхідну інформацію про свою продукцію. Попитом на своєчасну переробку є вже «нагрітий» лід, який, швидше за все, стане замовником. Тому переконайтеся, що в сценарії бота немає тупиків: навіть якщо бот не може відповісти на конкретне запитання, він повинен проінструктувати клієнтів, як вирішити їх проблему[8].

Генерація лідів є також перевагою, тому що завдяки добре структурованим питанням у боті ви можете керувати клієнтами від першого контакту до розміщення замовлення. Крім того, за допомогою ботів компанії можуть створювати опитування та отримувати відгуки від клієнтів. Потім на основі отриманої інформації зробіть більш точні та персоналізовані повідомлення на різних етапах продажу. У програмах, які створюють чат-ботів, легко відстежувати кількісні показники обміну повідомленнями та залучення користувачів.

Виходячи з перерахованого та розглянутого вище можна зробити висновок, що популярність чат-ботів набирає обертів та компанії, що прагнуть до збільшення ефективності свого бізнесу, все частіше замовляють розробникам нових чат-ботів.

1.3 Технічне завдання до створення системи

Головним завданням є створення інформаційного чат-боту в мережі Telegram. Насамперед потрібно створити привітальне повідомлення, після цього, за допомогою програмного коду створити кнопки для подальшої навігації користувача в системі. С початку повинні бути створені три основних кнопки:

– Графік роботи безпосередньо самого оффлайн магазину;

– Каталог усіх товарів в наявності;

– Місцезнаходження магазину в разі самовивозу товару з магазину

Наступним этапом буде створення розділів з видами усіх товарів з точним описом товару, а саме: назва, фото товару, технічні характеристики, ціна. Після виведення усіх товарів клієнту, під кожним товаром буде створена inline-кнопка «Придбати» для переходу саме у процес оформлення та придбання товару. Після оформлення замовлення клієнту в особисті повідомлення бот відправить усі введені дані клієнтом.

Для модераторів боту потрібно створити окремий простір для додавання, редагування, видалення усіх товарів та прийом замовлень від клієнтів, за допомогою наперед створених команд. Також для власника та модераторів чат-боту повинна бути створена база даних на базі програмного забезпечення Sqlite з вмістом усіх даних про опис товарів, та кількість замовлень зроблених користувачами

Для реалізації звичайно потрібно визначитися з мовою програмування, адже без коду запрограмувати бота не є можливим. За результатом роботи бот має виконувати такі функції:

– Імітувати спілкування з користувачами в режимі реального часу;

– Використовувати текст для пошуку інформації;

– Пошук потрібного товару за короткий проміжок часу;

– Швидке оформлення та покупка товару;

Цей робот може бути цікавий не тільки з точки зору реалізації, але і в в плані подальшого вдосконалення. Бот повинен мати зручний інструмент пошуку та відображення необхідної інформації. Дана розробка буде корисна кожному користувачу мережі, якому потрібно швидко та зручно придбати електронну техніку без зайвих дзвінків, особливо це корисно тим, хто має під рукою смартфон, а не стаціонарний комп’ютер або ноутбук, адже не всі сайти відомих магазинів мають мобільну версію сайту, і тому звичайний користувач має деякі незручності із замовленням товару, а завдяки цьому Telegram боту ці проблеми вирішуються.

РОЗДІЛ ІІ. РЕАЛІЗАЦІЯ СИСТЕМИ ОБЛІКУ КЛІЄНТІВ ТА ЗАМОВЛЕНЬ ОНЛАЙН МАГАЗИНУ НА БАЗІ TELEGRAM БОТА

2.1 Стек технологій який використовувався при розробці чат-бота

У сучасному світі дуже багато зав'язано на ІТ-технологіях, практично в будь-якій компанії працівники використовують різні програми для ефективної та якісної роботи, світ не стоїть на місці, а розвивається.

Для написання чат-бота необхідно вибрати мову програмування від якої залежатиме швидкість написання коду, можливості мови тощо.

Python — це потужна, проста в освоєнні мова програмування. Ця мова має ефективні високорівневі структури даних і простий, але ефективний підхід до об’єктно-орієнтованого програмування. Елегантний синтаксис і динамічне введення Python, а також його інтерпретація, роблять його ідеальною мовою для написання сценаріїв і швидкої розробки програм у багатьох областях на більшості платформ[9].

Переваги Python:

- відкритий розвиток;

- легко навчатися, особливо на початковому етапі;

- Синтаксичні особливості спонукають програмістів писати читабельний код; - Надає інструменти для швидкого прототипування та динамічної семантики; - Має велику громаду з позитивним ставленням до новачків;

- Багато корисних бібліотек і мовних розширень можна легко використовувати у ваших проектах завдяки надзвичайно однорідному механізму імпорту та програмному інтерфейсу;

- модульний механізм добре продуманий і простий у використанні;

- абсолютно все в Python є об’єктом у сенсі ORP, але об’єктно-орієнтований підхід не нав’язується програмісту.

Недоліки Python:

- підтримка багатопоточності виявилася не дуже вдалою;

- у порівнянні з іншими мовами програмування загального призначення, такими як Java, Python не створює багато якісних програмних проектів;

- відсутність комерційної підтримки інструментів розробки (хоча це змінюється з часом);

- обмежені початкові кошти для використання бази даних;

- тести показують нижчу продуктивність Python порівняно з основними віртуальними машинами Java, що робить мову відомою як повільну.

Завдяки популярності платформ обміну миттєвими повідомленнями та досягненнями в AI Chatbots зазнавали вибухонебезпечного зростання зі швидкістю 80% підприємств, які бажають використовувати боти Чата до 2020 року. Це створило відмінну можливість для позаштатних розробників Python, оскільки є велика потреба у розвитку як прості, так і складні роботи чату [10].

Однією з найпопулярніших платформ обміну повідомленнями є телеграма зі звітом 200 мільйонів щомісячних користувачів. Telegram надає чудове API для роботів у чаті, що дозволяє користувачеві не тільки спілкуватися за допомогою текстових повідомлень, але й мультимедійного контенту із зображеннями та відео та багатим контентом з HTML та JavaScript. API можна навіть використовувати для керування покупками у телеграмі.

Python чудово підходить для створення ботів Telegram, на основі використання API Telegram Bot створена асинхронна структура айограм для розробки чат-ботів. Фреймворк aiogram був обраний тому, що він простий у використанні та повністю асинхронний [16]. Створено з використанням стандартної бібліотеки Python asyncio та асинхронної веб-фреймворку aiohttp.

Найважливішими особливостями є те, що він витримує дуже велике навантаження при використанні невеликих машинних ресурсів (фізичних чи віртуальних), має вбудований кінцевий автомат для проектування більш складних чат-ботів, підтримує функціональність вебхуку Telegram API (здатний робити запити у відповідь на оновлення)[11] .

Технонологія Aiogram - це технологія, яка використовувалась під час створення чат-боту. Досить простий і повністю асинхронний фреймворк для API Telegram Bot, написаний на Python 3.7 з стандартнї бібліотеки «asyncio» та іншого асинхронного веб-фреймворку «aiohttp» [2]. Його найголовнішою особливістю, є можливість витримувати дуже великі навантаження, при мінімальних ресурсах. Ця бібліотека вважається найкращою в розробці ботів на Python [12].

Середовище розробки Pycharm

Універсальний редактор коду для мови Python під назвою PyCharm призначений для швидкої та ефективної розробки програмного забезпечення. Цей редактор має в наявності для використання багато інструментів, які дозволять вам зосередитися на написанні саме бізнес-логіки застосунку без рутинних речей: налаштування віртуального середовища, підключення до системи контролю версій, установка зовнішних бібліотек тощо. Використовувана остання версію редактора для розробки - 1.34.

До можливостей цього редактора коду належать:

1. Безкоштовний текстовий редактор з відкритим вихідним кодом.

2. Файлова система проекту для полегшення переходу між компонентами;

3. Адекватні можливості тестування та налагодження;

4. Зовнішні ресурси для підтримки впровадження програмного продукту;

5. Підтримує базові VCS та доступні інструменти, що сприяють полегшенню у використанні цих систем;

6. Можливість розробляти та встановлювати власні плагіни для розширень функціоналу [22].

Sqlite

База даних (БД) — це набір даних і структур із певними характеристиками та зв`язками. Об’єднування даних в єдину базу даних дозволяє варіювати групування інформації. При створенні веб-додатків часто користуються реляційні бази даних, в яких зберігається вся необхідна інформація у вигляді таблиць і посилань між таблицями. Мова SQL використовується для доступу до інформації та керування нею, також дозволяє будувати запити до бази даних для запису, оновлення, видалення, отримання та структурування даних [20].

СУБД вирішує такі завдання:

1. Управління операцією з даними.
2. Підтримування цілісності бази даних (правильність і послідовність).
3. Резервне копіювання та відновлення бази даних.
4. Організації отримують доступ до даних одночасно.
5. Захист даних.

SQLite — це реляційна СУБД, написана на вбудованій мові C безпосередньо в програмі.

Особливості SQLite:

1. Реалізує важливі частини стандарту SQL-92.
2. У комплекті з кількома мовами програмування, включаючи Node.js.
3. Зберігає всю базу даних у доступному для використання міжплатформному файлі, локально максимальний розмір якого становить 281 ТБ.
4. Початкова конфігурація не потрібна.
5. Жодних зовнішніх залежностей, усі коди до 350 КБ.
6. Простий і зрозумілий API[21]

Telegram bot API – це HTTP-інтерфейс для роботи з ботами телеграм. Кожен бот – це обліковий запис, створений спеціально для автоматизації оброблення та надсилання повідомлень.

Документація Telegram Bot API підкреслює дві найбільш протилежні версії, як отримати оновлення:

– а) періодичні запити;
– б) встановлення веб-хуків[13].

Вхідні оновлення зберігаються на сервері доки вони не обробляться, але не більше 24 годин. У відповідь отримуємо об’єкт Update, який реалізовано в JSON, незалежно від методу отримання оновлень. Перший і найпростіший варіант — періодичне опитування серверів Telegram для отримання нової інформації. Відкривається з'єднання на нетривалий час і всі оновлення відразу відправляються боту, все це здійснюється через зв’язок Long Polling. Цей спосіб простий, але не дуже надійний [15].

Веб-хуки працюють трохи інакше. Якщо повідомлення надійшло в чат, поді сам Telegram говорить про це, і саме в цьому проявляють себе вебхуки. Тому відпадає необхідність періодично опитувати сервер, таким чином зникають причини помилок пошукових систем. Однак для цієї можливості необхідно платити, встановивши на свій пристрій повноцінний веб-сервер на якому планується запустити пошукових ботів. На рисунку 2.1 представлена схема роботи чат-бота Telegram.

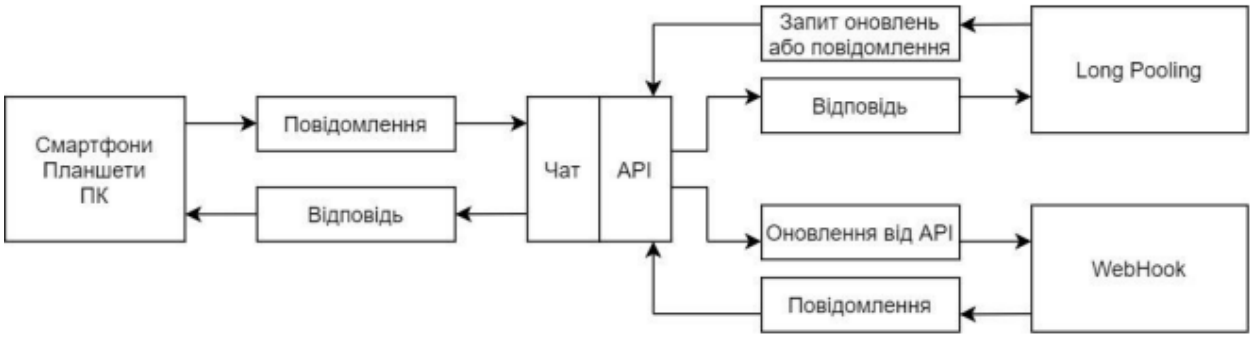

Рисунок 2.1 – Схема роботи чат-бота Telegram

2.2. Описання структури і функціоналу Telegram-бота.

Спершу в пошуковій стрічці Telegram потрібно знайти BotFather – чат-бот, за допомогою якого можна створити власного бота та надати йому всі необхідні параметри та функції. BotFather надає можливість дати ім'я створеному боту, створити опис, та надати список усіх команд, які доступні всим користувачам. Потрібно вказати правильну інформацію, для того, щоб користувачам було просто та зручно використовувати телеграм-бота. Під час створення видається унікальний токен, який необхідний для зв'язку з ботом та їхньої взаємодії.

Відправляємо команду /newbot і вводимо ім'я нашого бота та його нік, якому його шукатимуть інші користувачі. Надалі всі налаштування, крім нікнейму, можна буде поміняти. У разі потреби також можна отримати новий унікальний ключ (рис.2.2).

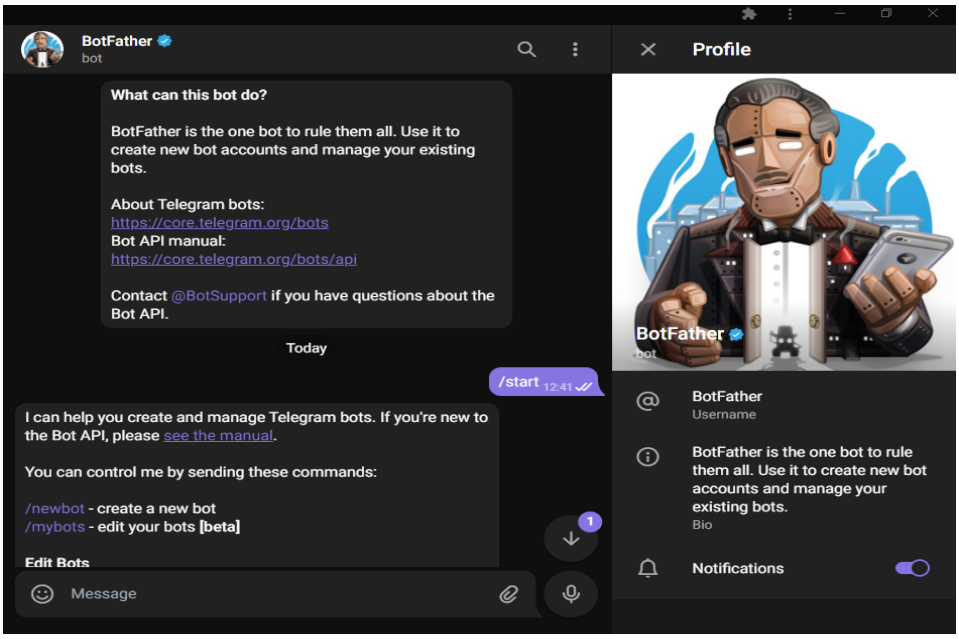

Рисунок 2.2 – BotFather

Далі можна додати опис бота, який користувач отримуватиме спочатку діалогу, надати команди, які допоможуть відправляти повідомлення швидше та завантажити боту зображення для спрощення пошуку.

Таблиця 2.1 – Доступні команди для змін чат-ботів

Команда	Опис
/setname	Надає зміни зв’язані з ім’ям
/setdescriptio n	Задає текст, який буде відображатися при першій взаємодії з ботом
/setabouttext	Надає опис чат-боту
/setuserpic	Завантажується зображення для бота
/setcommand	Створення списку команд

s	
/deletebot	Видалення боту

Окрім основних команд, існують інші команди, які зображені у таблиці 2.2.

Таблиця 2.2. – Доступні команди для додаткового налаштування чат-бота

Команда	Опис
/token	Надає отриманий раніше токен
/revoke	Видаляє токен
/setinline	Вмикає або вимикає спробу визвати чат-бота з іншого чату
/setinlinegeo	Функція місцезнаходження
/setinlinefeedback	Інформація про кількість команд
/setjoingroup	Можливість додавання до інших чатів
/setprivacy	Функція конфіденційності

Після отримання токена та первинного налаштування можна починати розробку чат-бота.

Основний принцип роботи складається с API, конкретно в виді файла json с категоріями та продукції товару, який збирається через програмне забезпечення Python, та в клавіатурі генеруються категорії. При натисканні на певну категорію збираються конкретні товарі та завантажуються в Telegram з посиланням на конкретний товар [17].

Структура чат-боту поділяється на декілька аспектів, а саме: клієнтська частина, модераторська частина, а також база даних. Клієнтська частина вміщує привітальне повідомлення, три активних кнопки (“Каталог товарів”,”Місцезнаходження”,”Графік роботи”). Вміст коду створення клієнтської частини представлено в Додатку А.

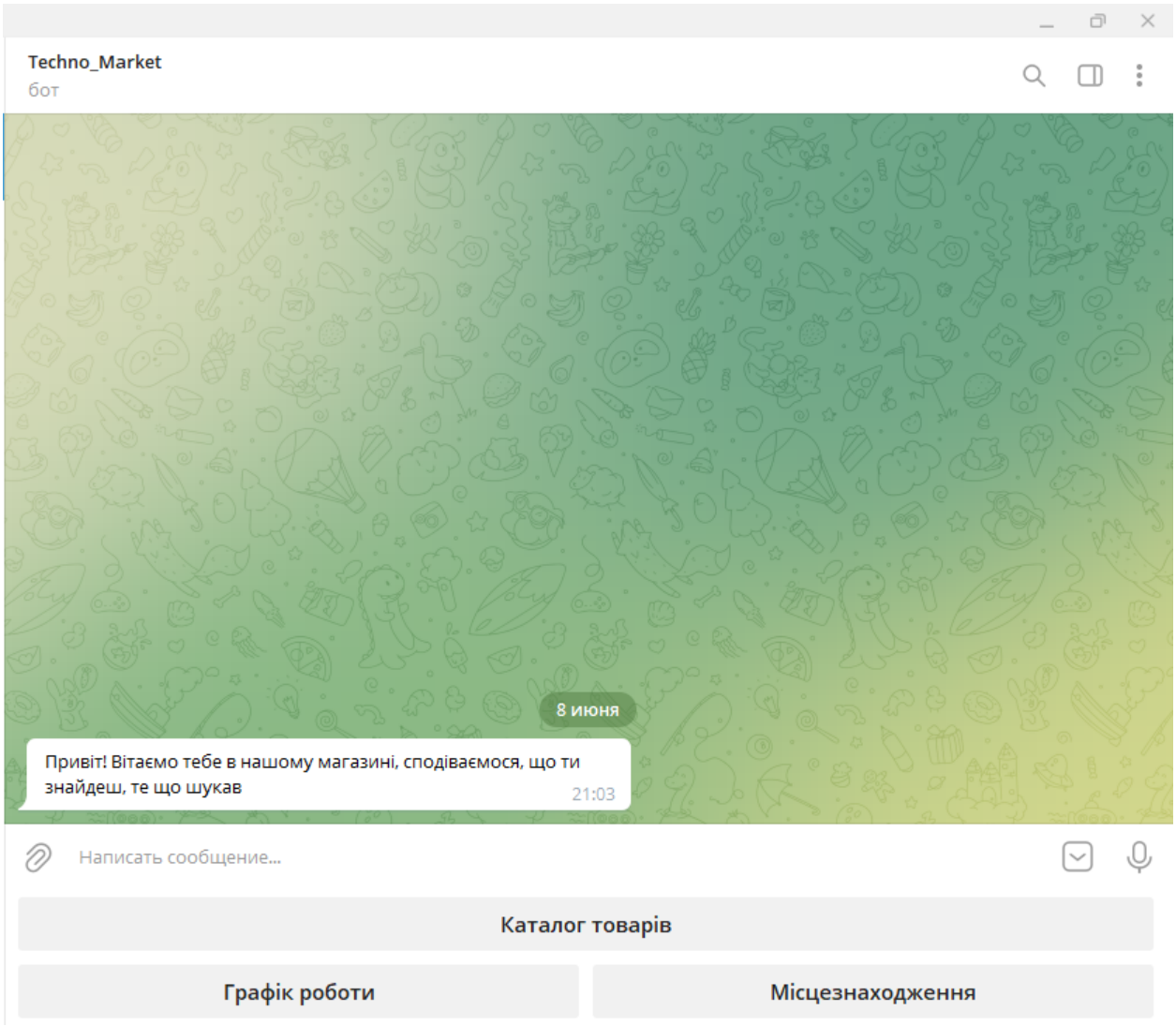

Рисунок 2.3 – Структура клієнтської частини

Модераторська частина складається з активних кнопок, зв’язаних з додаванням всього контенту в бота, та тісно пов’язана з базою даних, яка виконує роль зберігання усіх змін всередині системи. Вміст коду створення модераторської частини представлено в Додатку Б.

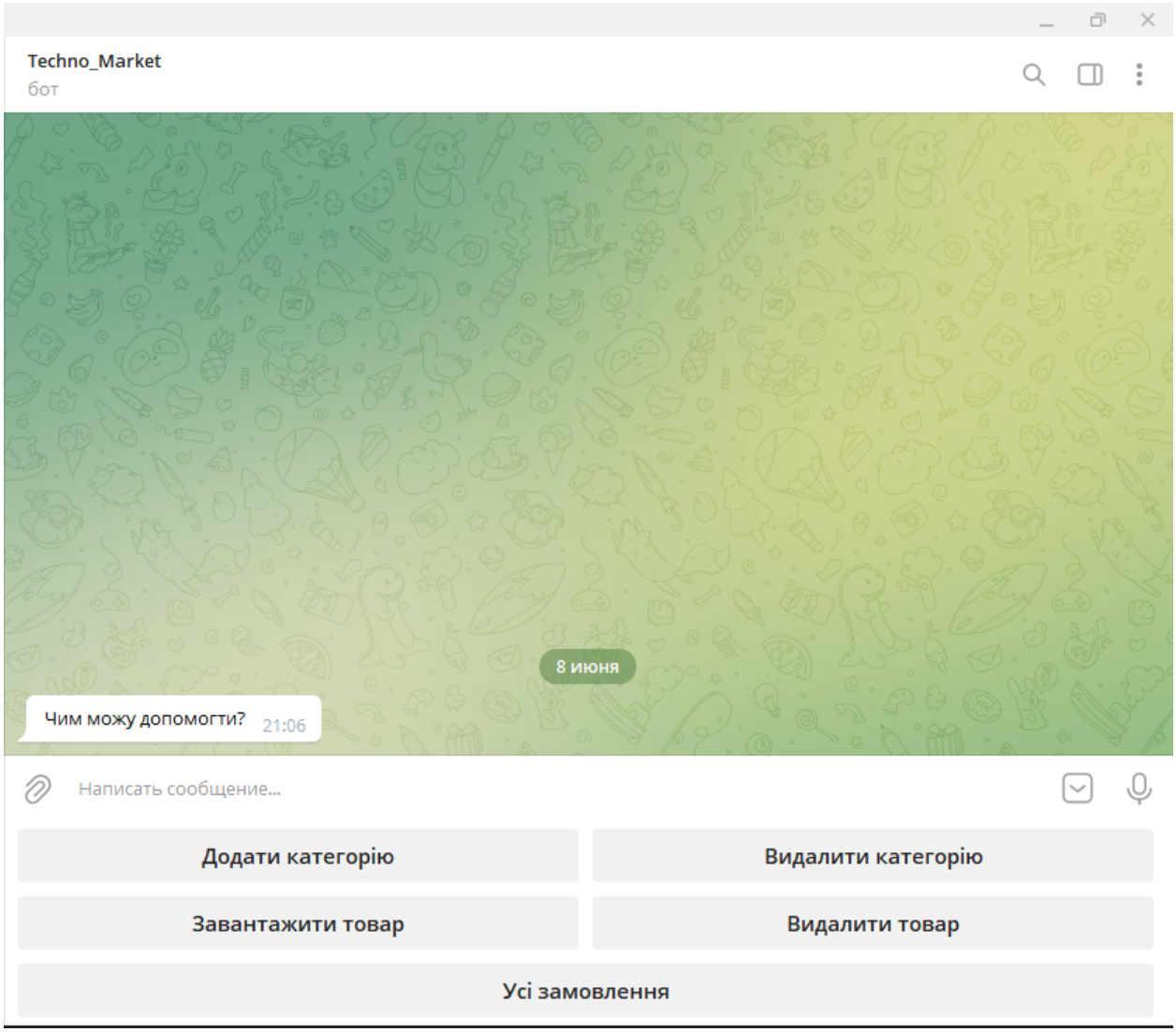

Рисунок 2.4 – Структура модераторської частини

База даних складається з трьох таблиць:

– categories – вміщує в себе інформацію про додані категорії;

– menu – містить усю інформацію про доданий модератором товар, та всі зміни, які проводяться всередині чат-боту;

– orders – зберігає інформацію про зроблені користувачем замовлення.

Таблиця «categories» зберігає в собі інформацію про всі створені категорії вбудованого в Telegram бота магазину, а саме присвоєний ідентифікаційний номер категорії та її назву.

Таблиця «menu» зберігає інформацію про людину, яка обирає товар, назву товара, його ціну, в якій категорії він знаходиться та характеристики товару.

Таблиця «orders» зберігає інформацію про особистий ідентифікаційний номер користувача, його персональні дані, назву придбаного товару, ціну та адресу доставки товару.

Повна структура створеної бази даних показана на рисунку 2.5

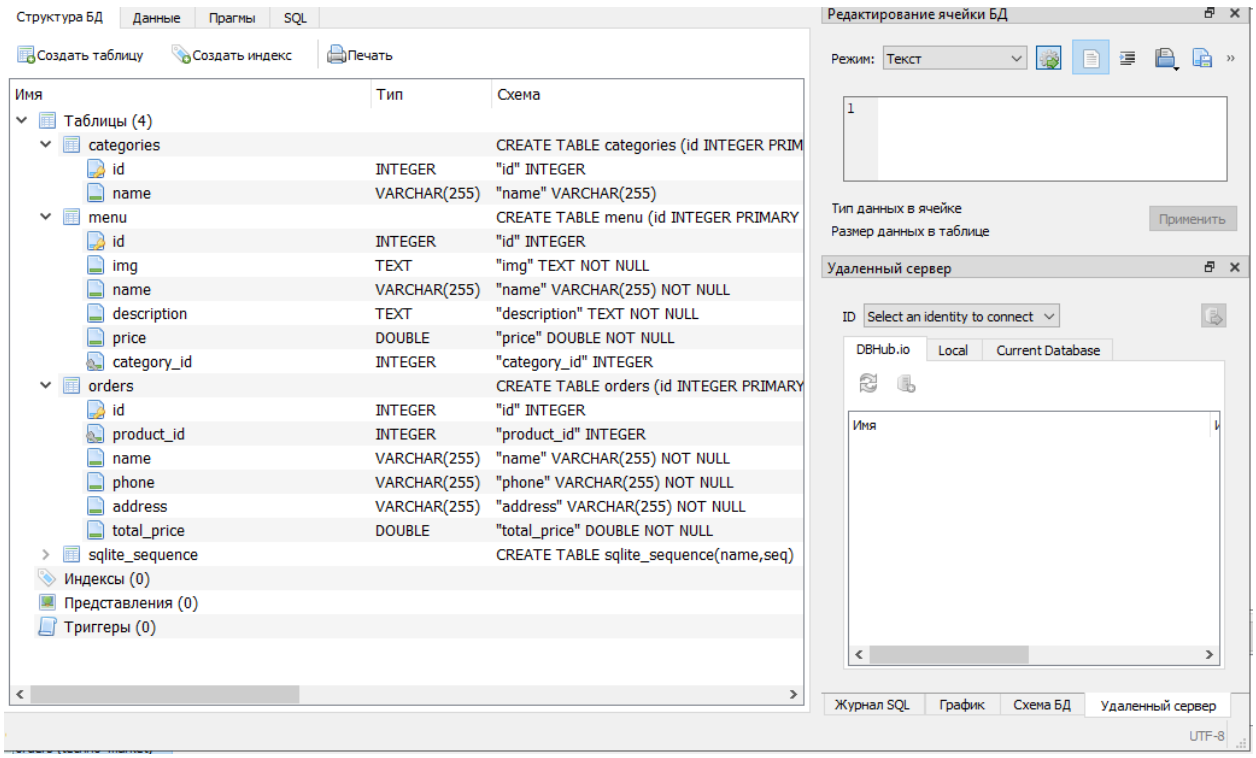

Рисунок 2.5 – Структура створеної БД

Приклад створення таблиці «orders» наведено на рисунку 2.6.

```
CREATE TABLE orders (
    id          INTEGER       PRIMARY KEY AUTOINCREMENT,
    product_id  INTEGER       REFERENCES menu (id),
    name        VARCHAR (255) NOT NULL,
    phone       VARCHAR (255) NOT NULL,
    address     VARCHAR (255) NOT NULL,
    total_price DOUBLE        NOT NULL
);
```

Рисунок 2.6 – Приклад створення таблиці «orders»

Клієнтська частина

Початок роботи з ботом починається з команди «/start». Після цього програма повинна відповісти письмовим текстом або надіслати голосове повідомлення.

Ця функція реалізована за допомогою бібліотеки Aiogram,(рис.2.7).

```
async def command_start(message: types.message):
    try:
        await bot.send_message(message.from_user.id, "Привіт! Вітаємо тебе в нашому магазині, сподіваємося, що ти знайдеш, те що шукав", reply_m
        await message.delete()
```

Рис. 2.7 – Приклад перехоплювача

У відповідь на команду «/start» бот видає наперед задане повідомлення клієнту або простому користувачу. Далі чат-бот пропонує скористуватися своїм функціоналом. Після того, як користувач вибере необхідну для себе опцію, перехоплювач прийме цей запит на опрацювання

Після опрацювання ботом запиту, користувач вибирає собі необхідну категорію товару, наприклад ноутбуки. Чат-бот видає необхідну інформацію про всі ноутбуки, які знаходяться або не знаходяться в наявності, також коротку інформацію про модель, та ціну конкретного товару. Функція в коді зображена на рисунку 2.8 [5].

```python
async def shop_menu_command(message: types.message):
    categories = await sqlite_db.sql_get_category()
    for cat in categories:
        if fuzz.ratio(message.text.lower(), cat[0].lower()) > 70:
            cat_id = await sqlite_db.sql_get_id_category(message.text)
            products = await sqlite_db.sql_get_product_by_category(cat_id[0][0])
            if len(products) == 0:
                await bot.send_message(message.from_user.id, "В даній категорії відсутні товари",
                                       reply_markup=kb_client)
            for ret in products:
                await bot.send_photo(message.from_user.id, ret[1],
                                     f'{ret[2]}\nТехнічні характеристики: {ret[3]}\nЦіна {ret[4]}',
                                     reply_markup=InlineKeyboardMarkup().\
                                     add(
                                         InlineKeyboardButton(f'Придбати', callback_data=f'buy {ret[0]}')))
    if fuzz.ratio(message.text.lower(), 'Повернутися'.lower()) > 70:
        await bot.send_message(message.from_user.id, "Вітаю з покупкой", reply_markup=kb_client)
    if fuzz.ratio(message.text.lower(), 'Графік роботи'.lower()) > 70:
        await bot.send_message(message.from_user.id, "Пн-Пт з 8:00 до 18:00, Сб-Нд з 10:00 до 17:00")
    if fuzz.ratio(message.text.lower(), 'Місцезнаходження'.lower()) > 70:
        await bot.send_message(message.from_user.id, "вул. Незалежна 24")
    if fuzz.ratio(message.text.lower(), 'Каталог товарів'.lower()) > 70:
        # await sqlite_db.sql_read(message)
        await bot.send_message(message.from_user.id, "Оберіть категорію товару",
                               reply_markup=await category_kb(categories))
```

Рис. 2.8 – Програмний код

Приклад отриманої інформації для користувача зображений на рисунку 2.9.

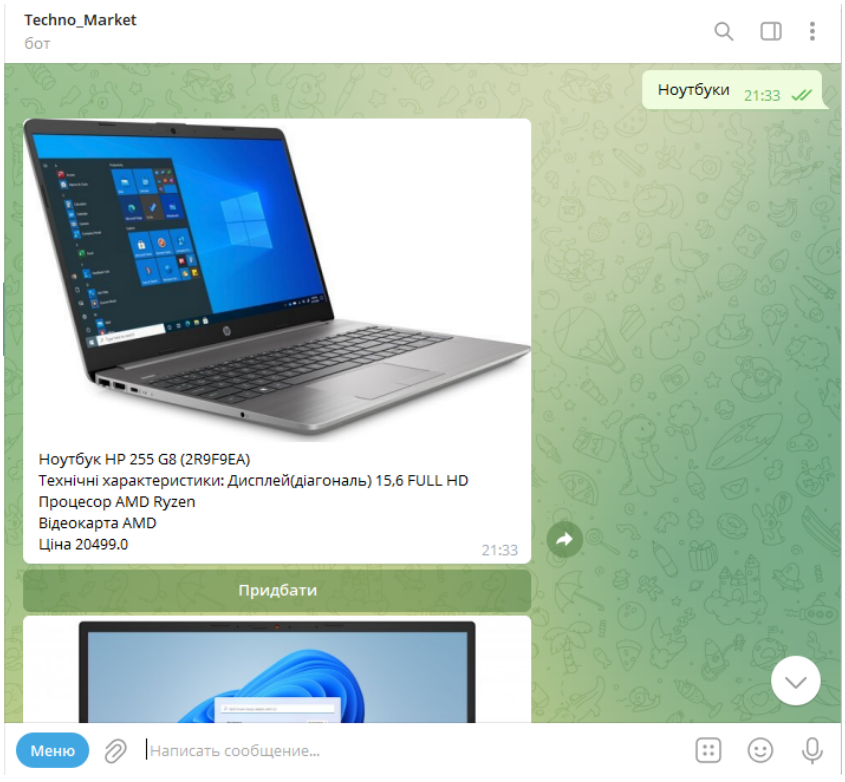

Рис. 2.9 – Інформація по конкретному запиту користувача

Після вибору клієнтом конкретного виду товара, клієнт натискає inline кнопку «Придбати». Після цього запиту бот відправить повідомлення, завдяки якому почнеться оформлення товару, приклад такого замовлення зображено на рисунку 2.10.

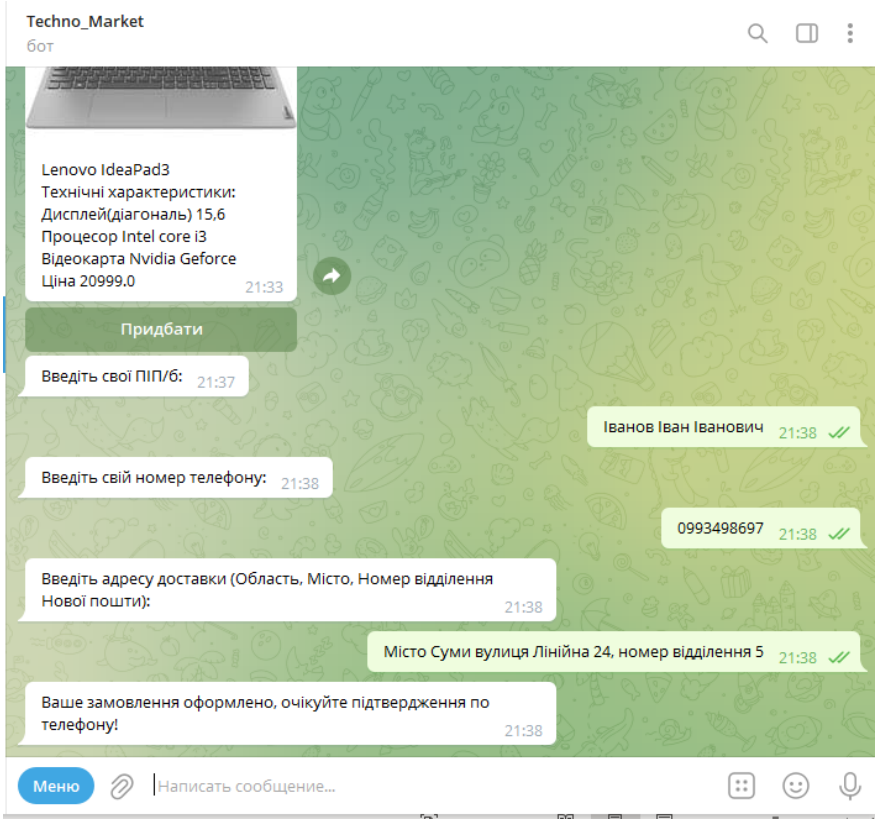

Рисунок 2.10 – Приклад оформлення замовлення

Після збору всіх даних, вони автоматично передаються в базу даних, та модератор по натисканню кнопки побачить це замовлення і зв’яжеться з клієнтом.

Також клієнт має змогу через активну кнопку дізнатися про Графік роботи безпосередньо оффлайн магазину для самовивозу конкретного товару(рис. 2.11)

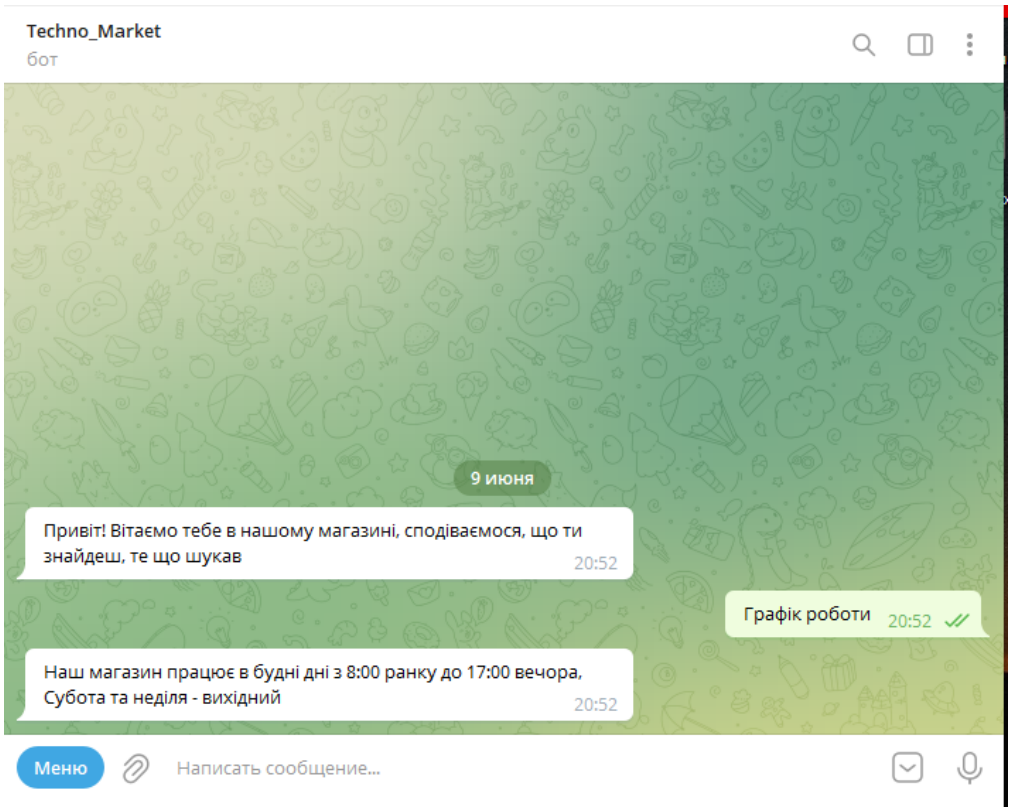

Рисунок 2.11 – Графік роботи магазину

Користувач має можливість запитати в бота про точне місцезнаходження магазину, для того, щоб не блукати по місту в пошуках свого товару(рис. 2.12).

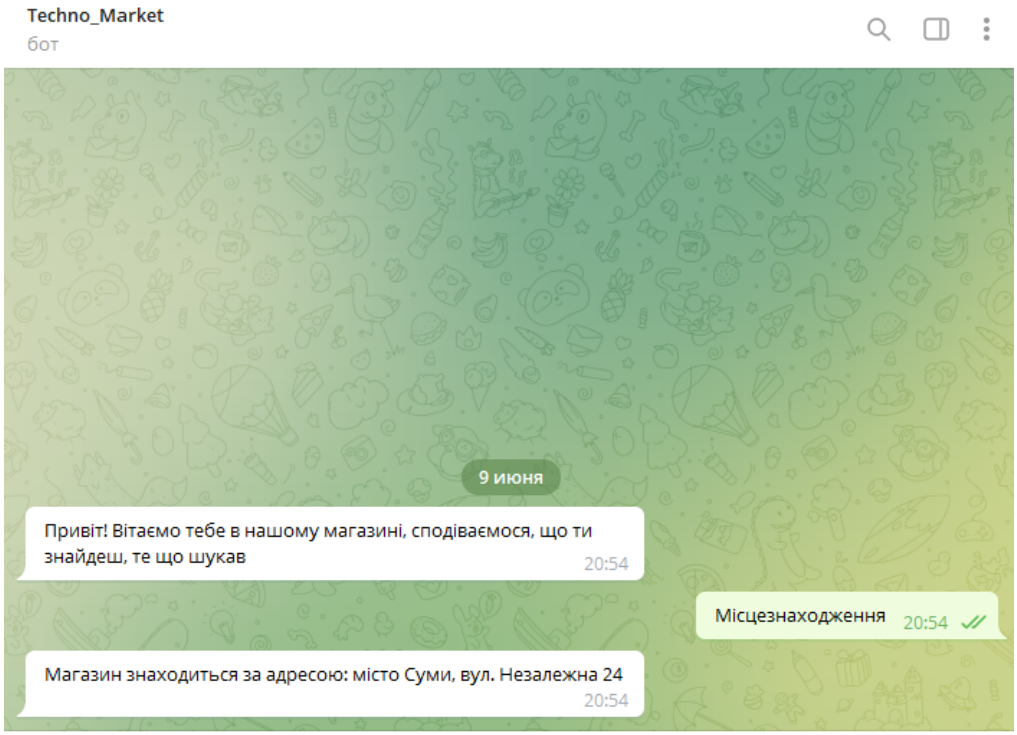

Рисунок 2.12 – Місцезнаходження магазину

Нижче наведені ключі API, завдяки яким збирались та систематизувались дані.

Таблиця 2.3 – Ключі API

Id
Product id
Name
Phone
Address
Total_price

Модераторська частина

Для повного доступа до чат-боту розроблена спеціальна прихована команда, за допомогою якої можна створювати або видаляти категорії, завантажувати та видаляти товар, передивитися усі замовлення клієнтів. Частина коду створення команди показана на рисунку 2.13

```
# dp.message_handler(commands=['moderator'], is_chat_admin=True)
async def make_changes_command(message: types.Message):
    global ID
    ID = message.from_user.id
    await bot.send_message(message.from_user.id, 'Чим можу допомогти?', reply_markup=admin_kb.button_case_admin)
    await message.delete()
```

Рисунок 2.13 – Створення спеціальної команди модератора

Після введення в окремій групі с ботом цеї команди, бот реагує на це відповіддю в особисті повідомлення, та дає доступ до активних кнопок, які доступні тільки модератору. Нижче наведено приклад взаємодії модератора з чат-ботом(рис. 2.14)

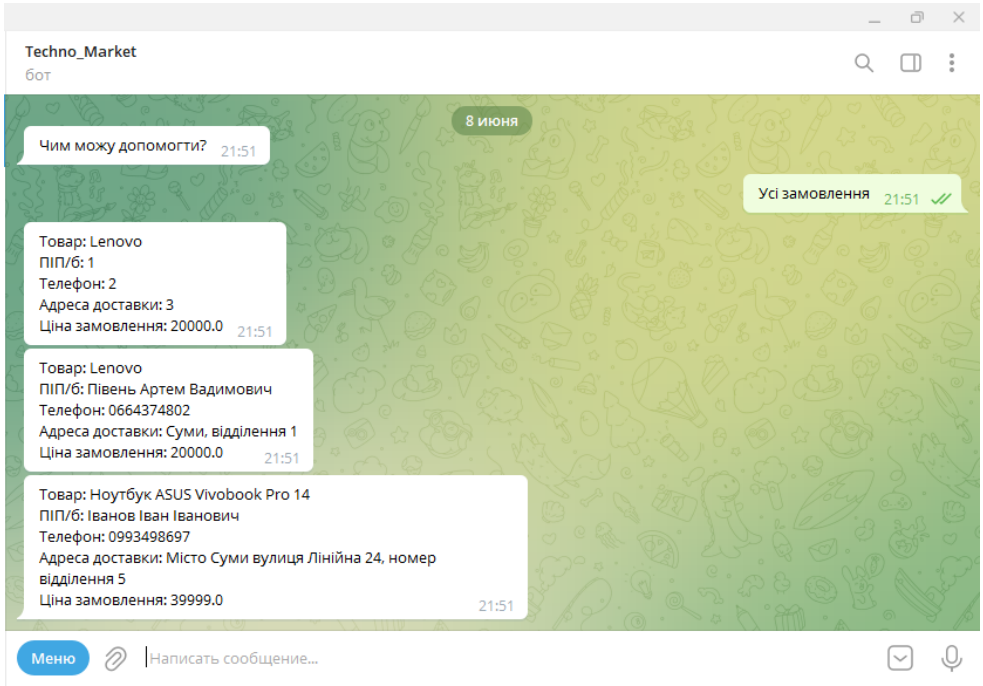

Рисунок 2.14 – Результат виконаної команди

Виходячи з сформованих даних та всіх можливих варіантів використання чат-боту, розроблена Use Case діаграма [14]. Вона представлена на рисунку 2.15.

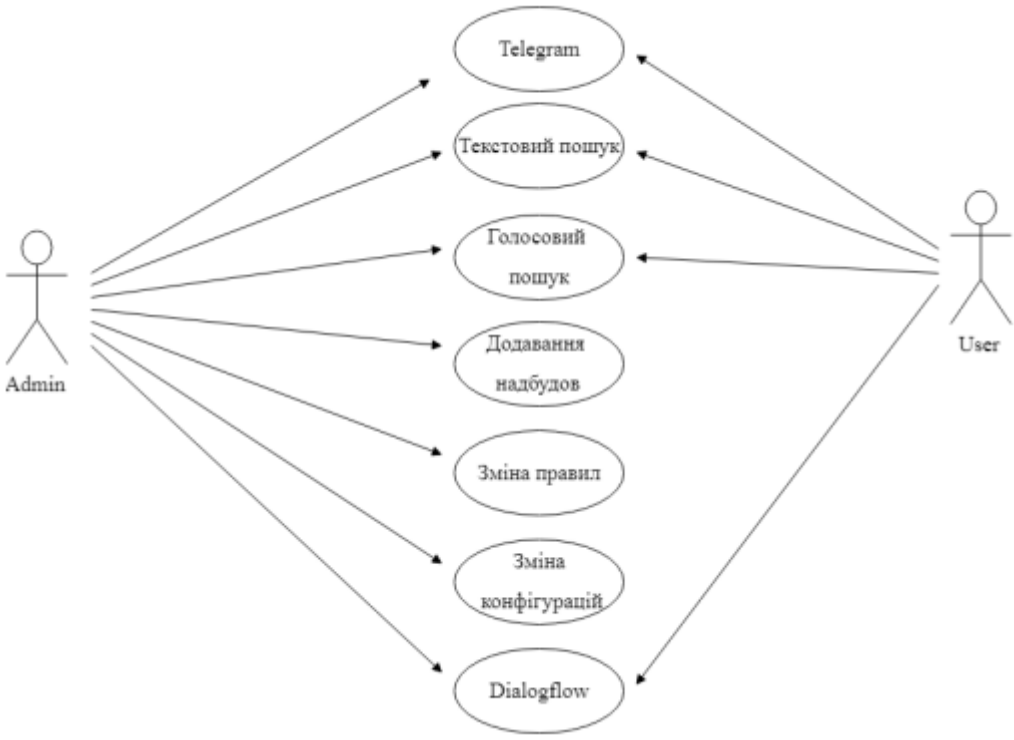

Рисунок 2.15 – Діаграма використання бота

2.3 Оцінка очікуваного ефекту від впровадження системи автоматизації

Ефективність будь якої автоматизованої системи завжди визначається, як відношення сукупних витрат до результата роботи системи. У таких систем у грошовому вираженні зазвичай визначається ефективність у вигляді основних показників:

1) економічний ефект за рік;

2) термін рентабельності капітальних витрат, які задіяні в розробку та впровадження системи;

3) розрахунковий коефіцієнт ефективності капітальних витрат.

Для розрахунку очікуваного ефекту необхідно розрахувати всю суму капітальних витрат, які включають в себе витрати на програмне та апаратне забезпечення, проектування, програмування, впровадження та налагодження системи. Отже, капітальні витрати (*К*) розраховуються за формулою 2.1:

$$K = K_1 + K_2 + K_3 + K_4 + K_5 + K_6, \quad (2.1)$$

де K_1 – витрати на проектування системи;

K_2 – витрати на програмування системи;

K_3 – витрати на впровадження системи;

K_4 – витрати на налагодження та тестування системи;

K_5 – витрати на програмне забезпечення;

K_6 – витрати на апаратне забезпечення;

Витрати К1 – К4 відображаються у заробітній платі програміста. Для визначення суми цих витрат необхідно зробити прогнози щодо витрат робочого часу та потім помножити на погодинну оплату праці.

Очікувані витрати робочого часу на витрати К1 – К4:

– витрати на проектування (аналіз предметної області) – 10 годин;

– витрати на програмування системи – 35 годин;

– витрати на впровадження системи – 35 годин;

– витрати на налагодження та тестування системи – 15 годин;

Отже, підсумовуючи витрати робочого часу становлять 95 годин Витрати на оплату праці програмістів розраховуються за формулою 2.2:

$$K_1 = \sum_{k=1}^{k} N_k \cdot r_k \cdot T_k \qquad (2.2)$$

де N_k - кількість розробників, чол.;

r_k - годинна заробітна плата розробника, грн.;

T_k - трудомісткість розробки (кількість витраченого часу), год.;

Проаналізувавши сайти з надання роботи [19], було визначего середню заробітну плату програміста в розмірі близько 25.000 гривень. Тоді при 5-денному робочому дні по 8 годин за день, годинна заробітна плата за розроблення програми буде становити 149 грн/год.

Витрати на програмне забезпечення створення додатку наведені у таблиці 2.3.

Таблиця 2.3 – Витрати на програмне забезпечення

Програмні продукти	Вартість, грн
PyCharm All products pack	29594
Разом	29594

Далі необхідно розрахувати витрати на апаратне забезпечення. У таблиці 2.4 наведено перелік необхідного устаткування.

Таблиця 2.4 – Витрати на апаратне забезпечення

Устаткування	Вартість, грн
Хостинг для бази даних та телеграм боту	10800
Разом	10800

Отже, капітальні витрати становлять :

$K = 95 * 149 + 29594 + 10800 = 54\,549$ (грн)

Для розрахунку очікуваного економічного ефекту необхідно розрахувати суму річної економії, яка розраховується за формулою 2.3:

$$S = Вб - Вп \text{ ,} \tag{2.3}$$

де *Вб* – приведені до одного року витрати до впровадження системи;

Вп – приведені до одного року витрати після впровадження системи.

Вб будуть складати річні витрати програміста, виходячи з його річної заробітної плати, скорегованої на відсоток роботи, яка відводиться на безпосереднє створення коду та тестування бота та, короткострокової оренди хостинга для боту та бази даних. Дані витрати наведені у формулі 2.4

$$Вб = Взп + Вштр \text{ ,} \tag{2.4}$$

де *Взп* – це витрати заробітної плати програміста;

Вштр – витрати на оренду хостинга та та програмного забезпечення щомісячно за підвищенною ціною.

Отже, витрати до впровадження системи складають:

Вб = 14155 + 50717 = 64872(грн)

Звідси, розмір річної економії складе:

$S = 64872 - 54549 = 10323$ (грн)

Тепер можна розрахувати річний економічний ефект та термін окупності системи.

Річний економічний ефект від розробки і упровадження системи визначається, як різниця між сумою річної економії (річний приріст прибутку) і нормативним значенням повернення одноразових витрат на впровадження системи (формула 2.5):

$$E_y = S - K \cdot r_n, \tag{2.5}$$

де E_y - річний економічний ефект;

S – річна економія, яку отримає підприємство в результаті впровадження системи;

K – капітальні вкладення, які було витрачено в результаті впровадження системи;

r_n – нормативний коефіцієнт окупності капітальних вкладень, узятий для конкретної галузі (для даного прикладу r_n = 0,14).

Сума річного економічного ефекту складе:

$\mathrm{E_y} = \ 10323 - 50717 * 0{,}14 = 3223$ (грн)

Розрахуємо коефіцієнт економічної ефективності капітальних витрат (формула 2.6):

$$R_{ce} = \frac{S}{C} \tag{2.6}$$

Підставивши відповідни значення у формулу 3.6 отримаємо коефіцієнт ефективності капітальних витрат:

$R_{ce} = \frac{10323}{50717} = 0.203$

Розрахуємо термін окупності витрат на впровадження проекту (формула 2.7):

$$P_p = \frac{C}{S} = \frac{1}{R_{ce}} \tag{2.7}$$

Підставивши відповідні значення у формулу отримаємо:

$P_p = \frac{1}{0.203} = 4.9$ (роки).

Аналізуючи дані показники можна сказати, що система окупиться в період приблизно п’яти років і буде приносити економічний ефект.

ВИСНОВОК

В ході виконання дипломної роботи були вирішені наступні питання:

– досліджено основне поняття чат-боту, його функцій, переваги та недоліки;

– розглянуто та проаналізовано мову програмування Python для створення Telegram-бота;

– створено та налаштовано роботу чат-бота;

– проведення тестування функціональних можливостей бота;

– проведено оцінку очікуваного ефекту від впровадження системи автоматизації.

Результатом роботи є створений Telegram-bot «Techno_Market», який буде допомогати користувачам вибирати якісний товар за дуже короткий час. Це збереже час користувачам при виборі певного гаджету та автоматизує роботу магазинів без великих на це трудомістких затрат.

Розроблений Telegram-bot має свій ряд особливостей:

– зручний та простий інтерфейс;

– ніяких команд, весь функціонал за допомогою кнопок

– робота в режимі 24/7.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Все, про що повинен знати розробник Телеграм-ботів [Електронний ресурс]. URL: https://habr.com/ru/post/543676/

2. Ласкаво просимо до документації aiogram! [Електронний ресурс]. URL: https://docs.aiogram.dev/en/latest/

3. Чат-бот для бізнесу: плюси й мінуси впровадження роботів [Електронний ресурс]. URL: https://nikopolnews.net/chat-bot-dlya-biznesu-pljusi-j-minusi-vprovadzhennya-robotiv/

4. Використання мови Python для розробки науково-технічного програмного забезпечення [Електронний ресурс]. URL: https://dou.ua/lenta/articles/python-for-science

5. [AIOGram] Урок 2. Наводимо порядки та додаємо фільтри [Електронний ресурс]. URL: https://geekstand.top/development/aiogram-urok-2-navodim-porjadki-i-dobavljaem-filtry/

6. Все про чат-боти: переваги, типи та схема роботи. [Електронний ресурс]. URL: https://www.interkassa.com/blog/vse-o-chat-botah-preimushchestva-tipy-i-shema-raboty/

7. Створення Telegram-бота. Основи. [Електронний ресурс]. URL: https://wibe.team/sozdanie-bota-v-telegram

8. Документація по створенню Telegram-ботів [Електронний ресурс]. URL: https://core.telegram.org/bots

9. Python is a programming language, [Електронний ресурс]. URL: https://www.python.org

10. Посібник з мови програмування Python. [Електронний ресурс]. URL: https://metanit.com/python/tutorial

11. Python Telegram Bot’s documentation [Електронний ресурс]. URL: https://python-telegram-bot.readthedocs.io/en/stable

12. Асинхронний Telegram бот мовою Python 3 з використанням бібліотеки aiogram [Електронний ресурс]. URL: https://opensourcelibs.com/lib/aiogram-lessons

13. python-telegram-bot [Електронний ресурс] – Режим доступу: https://github.com/python-telegram-bot/python-telegram-bot

14. Use-case diagrams [Електронний ресурс]. – Режим доступу до ресурсу: https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagrams-use-case

15. pyTelegramBotAPI [Електронний ресурс]-Режим доступу: https://github.com/eternnoir/pyTelegramBotAPI

16. Створюємо Telegram бота на Python. Частина 1[Електронний ресурс] - Режим доступу: https://codeguida.com/post/410

17. Welcome to Python Telegram Bot's documentation [Електронний ресурс] - Режим доступу: https://python-telegram-bot.readthedocs.io/en/stable

18. Learn to build your first bot in Telegram with Python n [Електронний ресурс] - Режим доступу: https://medium.freecodecamp.org/learn-to-build-yourfirst-bot-in-telegram-with-python-4c99526765e4

19. Rabota.ua [Електронний ресурс] – Режим доступу: https://rabota.ua/ua/zapros/%25D0%25BF%25D1%2580%25D0%25BE%25D0%25B3%25D1%2580%25D0%25B0%25D0%25BC%25D1%2596%25D1%2581%25D1%2582/%D1%81%D1%83%D0%BC%D1%8B

20. What is SQLite? [Електронний ресурс].– Режим доступу до ресурсу: https://www.sqlite.org/index.html

21. Що таке SQLite [Електронний ресурс].– Режим доступу до ресурсу: https://metanit.com/sql/sqlite/1.1.php

22. PyCharm: the Python IDE for Professional Developers by JetBrains [Електронний ресурс] — Режим доступу: https://www.jetbrains.com/pycharm

ДОДАТОК А

SUMMARY

Piven A.V. Automated accounting system for customers and orders online store based on Telegram bot. Bachelor`s thesis. Sumy State University, Sumy, 2022.

Today, technology is not standing still: smart homes, electric cars, cloud technology, 3D printing, new technologies and trends are emerging, some of which could seriously affect the entire IT industry. Relatively recently, everyone is talking about programs, but the recent development of bots, or chat robot for short, portends a bright future for communications and marketing. So are chatbots really needed today?

First of all, it greatly simplifies the work for business. It is with the help of bots that you can reduce costs and minimize the burden on employees, which allows them to perform more than 80% of tasks. Constantly changing market conditions, high speed of decision-making, the need for multitasking and reducing risks in asset management require a modern approach to business organization. The solution to increasingly complex internal and external environments of enterprises - full automation of business processes. This frees up valuable resources for strategic planning and centralized management of key business areas. Due to the increase in the amount of information in organizational information systems (IS), there is a need to automate information processes, speed up and use more complex processing methods. Business automation is the partial or complete transformation of stereotypical operations and business tasks under the control of specialized information systems or complex hardware and software. In this way, human and financial resources can be freed up to increase the productivity and efficiency of strategic management.

Keywords: chatbot, python, aiogram, automation.

АНОТАЦІЯ

Півень А.В. Автоматизована система обліку клієнтів та замовлень онлайн магазину на базі Telegram бота. Кваліфікаційна робота бакалавра. Сумський державний університет, Суми, 2022 рік.

Сьогодні технології не стоять на місці: розумні будинки, електромобілі, хмарні технології, 3D-друк, з’являються нові технології та тенденції, деякі з яких можуть серйозно вплинути на всю IT-індустрію. Відносно нещодавно всі говорять про програми, але нещодавня розробка ботів або скорочено «чат-робот» віщує світле майбутнє для комунікацій та маркетингу. Так дійсно ж чат-боти необхідні в сьогоденні?

Насамперед, це дуже спрощує роботу для бізнесу. Саме за допомогою ботів можна скоротити витрати та мінімізувати навантаження на працівників, що надає їм виконувати понад 80% завдань, Оскільки роботи стають розумнішими, завдання стануть складнішими, і роботи отримають можливість вчитися, виконуючи більш складні завдання. Ринкові умови, що постійно змінюються, висока швидкість прийняття рішень, необхідність багатозадачності та зниження ризиків в управлінні активами вимагає сучасного підходу до організації бізнесу. Вирішення дедалі складніших внутрішніх і зовнішніх середовищ підприємств — повна автоматизація бізнес-процесів. Це вивільняє цінні ресурси для стратегічного планування та централізованого управління ключовими сферами бізнесу. У зв’язку зі збільшенням обсягу інформації в організаційних інформаційних системах (ІС) виникає потреба в автоматизації інформаційних процесів, прискоренні та використанні більш складних методів обробки. Автоматизація бізнесу – це часткова або повна трансформація стереотипних операцій і бізнес-завдань під управлінням спеціалізованих інформаційних систем або складного апаратного та програмного забезпечення. Таким чином, можна вивільнити людські та

фінансові ресурси для підвищення продуктивності та ефективності стратегічного управління.

Ключові слова: чат-бот, python, aiogram, автоматизація

ДОДАТОК Б

Лістинг коду файла client.py

```
from aiogram import types, Dispatcher

from create_bot import bot, dp

from keyboards import kb_client

from aiogram.types import ReplyKeyboardRemove, ReplyKeyboardMarkup, InlineKeyboardMarkup, InlineKeyboardButton

from data_base import sqlite_db

import fuzzywuzzy

from fuzzywuzzy import fuzz

from fuzzywuzzy import process

from aiogram.dispatcher import FSMContext

from aiogram.dispatcher.filters.state import State, StatesGroup

from aiogram.dispatcher.filters import Text



class FSMOrder(StatesGroup):

    product_id = State()

    name = State()

    phone = State()

    address = State()
```

```
# dp.message_handler(commands=["start", "help"])
async def command_start(message: types.message):
    try:
        await bot.send_message(message.from_user.id, "Привіт! Вітаємо тебе в нашому магазині, сподіваємося, що ти знайдеш, те що шукав", reply_markup=kb_client)
        await message.delete()
    except:
        await message.reply("Розмова з ботом через ЛС:https://t.me/Tech_no_Mar_ket_Bot")


async def category_kb(categories):
    button_category = ReplyKeyboardMarkup(resize_keyboard=True)
    for cat in categories:
        button_category.add(cat[0])
    button_category.add("Повернутися")
    return button_category


async def shop_menu_command(message: types.message):
    categories = await sqlite_db.sql_get_category()
    for cat in categories:
```

```
    if fuzz.ratio(message.text.lower(), cat[0].lower()) > 70:
        cat_id = await sqlite_db.sql_get_id_category(message.text)
        products = await sqlite_db.sql_get_product_by_category(cat_id[0][0])
        if len(products) == 0:
            await bot.send_message(message.from_user.id, "В даній категорії відсутні товари",
                                   reply_markup=kb_client)
        for ret in products:
            await bot.send_photo(message.from_user.id, ret[1],
                                 f'{ret[2]}\nТехнічні характеристики: {ret[3]}\nЦіна {ret[4]}',
                                 reply_markup=InlineKeyboardMarkup(). \
                                 add(
                                     InlineKeyboardButton(f'Придбати', callback_data=f'buy {ret[0]}')))
    if fuzz.ratio(message.text.lower(), 'Повернутися'.lower()) > 70:
        await bot.send_message(message.from_user.id, "Вітаю з покупкой", reply_markup=kb_client)
    if fuzz.ratio(message.text.lower(), 'Графік роботи'.lower()) > 70:
        await bot.send_message(message.from_user.id, "Наш магазин працює в будні дні з 8:00 ранку до 17:00 вечора, Субота та неділя - вихідний")
    if fuzz.ratio(message.text.lower(), 'Місцезнаходження'.lower()) > 70:
        await bot.send_message(message.from_user.id, "Магазин знаходиться за адресою: місто Суми, вул. Незалежна 24")
```

```
    if fuzz.ratio(message.text.lower(), 'Каталог товарів'.lower()) > 70:
        # await sqlite_db.sql_read(message)
        await bot.send_message(message.from_user.id, "Оберіть категорію товару",
                               reply_markup=await category_kb(categories))


@dp.callback_query_handler(Text(startswith='buy '), state=None)
async def buy_callback_run(message: types.Message, state: FSMContext):
    if fuzz.ratio(message.data.lower(), 'buy '.lower()) > 70:
        product_price = await sqlite_db.sql_get_product_by_id(int(message.data.replace('buy ', '')))
        async with state.proxy() as data:
            data['product_id'] = int(message.data.replace('buy ', ''))
            data['total_price'] = float(product_price[0][4])
        await FSMOrder.name.set()
        await bot.send_message(message.from_user.id, "Введіть свої ПІП/б:")


@dp.message_handler(state=FSMOrder.name)
async def load_name(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['name'] = message.text
    await FSMOrder.next()
```

```
    await bot.send_message(message.from_user.id, "Введіть свій номер телефону:")


@dp.message_handler(state=FSMOrder.phone)
async def load_phone(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['phone'] = message.text
    await FSMOrder.next()
    await bot.send_message(message.from_user.id, "Введіть адресу доставки (Область, Місто, Номер відділення Нової пошти):")


@dp.message_handler(state=FSMOrder.address)
async def load_address(message: types.Message, state: FSMContext):
    async with state.proxy() as data:
        data['address'] = message.text
    await sqlite_db.sql_add_order(state)
    await state.finish()
    await bot.send_message(message.from_user.id, "Ваше замовлення оформлено, очікуйте підтвердження по телефону!")
```

```
def register_handlers_client(dp: Dispatcher):
    dp.register_message_handler(command_start, commands=["start", "help"])
    dp.register_message_handler(shop_menu_command)
```

ДОДАТОК В

Лістинг коду файлу admin.py

```
from aiogram.dispatcher import FSMContext

from aiogram.dispatcher.filters.state import State, StatesGroup

from aiogram import types, Dispatcher

from aiogram.dispatcher.filters import Text

from create_bot import dp, bot

from data_base import sqlite_db

from keyboards import admin_kb

from aiogram.types import InlineKeyboardMarkup, InlineKeyboardButton,
ReplyKeyboardMarkup, ReplyKeyboardRemove

import fuzzywuzzy

from fuzzywuzzy import fuzz

from fuzzywuzzy import process

ID = None



class FSMAdmin(StatesGroup):

    photo = State()

    name = State()

    description = State()

    category_id = State()

    price = State()
```

```
class FSMAdminCategory(StatesGroup):
    name = State()




# dp.message_handler(commands=['moderator'], is_chat_admin=True)
async def make_changes_command(message: types.Message):
    global ID
    ID = message.from_user.id
    await bot.send_message(message.from_user.id, 'Чим можу допомогти?',
reply_markup=admin_kb.button_case_admin)
    await message.delete()




# dp.message_handler(commands="Завантажити", state=None)
async def cm_start(message: types.Message):
    if message.from_user.id == ID:
        if message.text == 'Завантажити товар':
            await FSMAdmin.photo.set()
```

```
        await message.reply("Завантаж фото",
reply_markup=ReplyKeyboardRemove())
    if message.text == 'Додати категорію':
        await FSMAdminCategory.name.set()
        await message.reply("Додайте назву",
reply_markup=ReplyKeyboardRemove())
    if message.text == 'Видалити товар':
        read = await sqlite_db.sql_delete_command()
        for ret in read:
            await bot.send_photo(message.from_user.id, ret[1], f'{ret[2]}\nОпис:
{ret[3]}\nЦіна {ret[4]} ')
            await bot.send_message(message.from_user.id, text='^^^',
reply_markup=InlineKeyboardMarkup(). \
                                add(InlineKeyboardButton(f'Видалити {ret[2]}',
callback_data=f'del {ret[0]}')))
    if message.text == 'Видалити категорію':
        read_cat = await sqlite_db.sql_read_category()
        for ret in read_cat:
            await bot.send_message(message.from_user.id, f'Категорія: {ret[1]}')
            await bot.send_message(message.from_user.id, text='^^^',
reply_markup=InlineKeyboardMarkup(). \
                                add(InlineKeyboardButton(f'Видалити {ret[1]}',
callback_data=f'del_cat {ret[0]}')))
    if message.text == 'Усі замовлення':
```

```
        orders = await sqlite_db.sql_get_orders()
        if len(orders) == 0:
            await bot.send_message(message.from_user.id, 'Замовлень немає',
reply_markup=admin_kb)
        else:
            for order in orders:
                product_name = await sqlite_db.sql_get_product_by_id(order[1])
                user_name = order[2]
                user_phone = order[3]
                user_address = order[4]
                user_total_price = order[5]
                await bot.send_message(message.from_user.id, f"Товар:
{product_name[0][2]}\nПІП/б: {user_name}\nТелефон: {user_phone}\nАдреса
доставки: {user_address}\nЦіна замовлення: {user_total_price}")


async def load_category(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['name'] = message.text
        await sqlite_db.sql_add_category(state)
        await state.finish()
        await message.reply("Категорія додана")
```

```
# dp.message_handler(state="*", commands='Скасувати')
# dp.message_handler(Text(equals='Скасувати', ignore_case=True),state="*")
async def cancel_handler(message: types.Message, state=FSMContext):
    if message.from_user.id == ID:
        current_state = await state.get_state()
        if current_state is None:
            return
        await state.finish()
        await message.reply('OK')




# dp.message_handler(content_types=['photo']. state=FSMAdmin.photo)
async def load_photo(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['photo'] = message.photo[0].file_id
        await FSMAdmin.next()
```

```
        await message.reply("Тепер введи назву")



# dp.message_handler(state=FSMAdmin.name)
async def load_name(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['name'] = message.text
        await FSMAdmin.next()
        await message.reply('Введи необхідний опис товару')



# dp.message_handler(state=FSMAdmin.description)
async def load_description(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['description'] = message.text
        await FSMAdmin.next()
        categories = await sqlite_db.sql_get_category()
        button_category = ReplyKeyboardMarkup(resize_keyboard=True)
        for cat in categories:
```

```
            button_category.add(cat[0])
        await message.reply("Оберіть категорію товару на клавіатурі",
reply_markup=button_category)




async def load_category_id(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        cat_id = await sqlite_db.sql_get_id_category(message.text)
        async with state.proxy() as data:
            data['category_id'] = int(cat_id[0][0])
        await FSMAdmin.next()
        await message.reply("Введи ціну товару",
reply_markup=ReplyKeyboardRemove())




# dp.message_handler(state=FSMAdmin.price)
async def load_price(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['price'] = float(message.text)
```

```
        await sqlite_db.sql_add_command(state)
        await state.finish()
        await message.reply("Товар додано")


@dp.callback_query_handler(Text(startswith='del_cat '))
@dp.callback_query_handler(Text(startswith='del '))
async def del_callback_run(callback_query: types.CallbackQuery):
    if fuzz.ratio(callback_query.data.lower(), 'del '.lower()) > 70:
        await sqlite_db.sql_delete_command(callback_query.data.replace('del ', ''))
        await callback_query.answer(text=f'Товар видалений.', show_alert=True)
    if fuzz.ratio(callback_query.data.lower(), 'del_cat '.lower()) > 70:
        await sqlite_db.sql_delete_command_cat(callback_query.data.replace('del_cat
', ''))
        await callback_query.answer(text=f'Категорія видалена.', show_alert=True)


def register_handlers_admin(dp: Dispatcher):
    dp.register_message_handler(make_changes_command,
commands=['moderator'], is_chat_admin=True)
    dp.register_message_handler(cm_start, state=None)
```

```
    dp.register_message_handler(cancel_handler, state="*",
commands='Скасувати')

    dp.register_message_handler(cancel_handler, Text(equals='Скасувати',
ignore_case=True), state="*")

    dp.register_message_handler(load_photo, content_types=['photo'],
state=FSMAdmin.photo)

    dp.register_message_handler(load_name, state=FSMAdmin.name)

    dp.register_message_handler(load_category, state=FSMAdminCategory.name)

    dp.register_message_handler(load_description, state=FSMAdmin.description)

    dp.register_message_handler(load_category_id, state=FSMAdmin.category_id)

    dp.register_message_handler(load_price, state=FSMAdmin.price)
```