МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: **«РОЗРОБКА TELEGRAM БОТА ДЛЯ ПОДОРОЖЕЙ»**

Виконав: студент 2 курсу, групи 8.1263
спеціальності 126 Інформаційні системи та технології
(шифр і назва спеціальності)
освітньої програми Інформаційні системи та штучний інтелект
(назва освітньої програми)
Д. В. Строков
(ініціали та прізвище)
Керівник доцент кафедри комп'ютерних наук, к.т.н Добровольський Г.А
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент завідувач кафедри фундаментальної та прикладної математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти магістр

Спеціальність 126 інформаційні системи та технології
(шифр і назва)

Освітня програма Інформаційні системи та штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Шило Г.М.
(підпис)

" 01 " 03 2024 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Строкову Денису Віталійовичу
(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка Telegram бота для подорожей

керівник роботи Добровольський Геннадій Анатолійович, к.т.н, доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 01 » березня 2024 року № 385-с

2. Строк подання студентом роботи 27.11.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Основні теоретичні відомості.
3. Розробка Telegram бота для подорожей

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 01.03.2024

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	15.09.2024	
2.	Збір вихідних даних.	17.09.2024	
3.	Обробка методичних та теоретичних	21.09.2024	
	джерел.		
4.	Розробка першого та другого розділу.	24.09.2024	
5.	Розробка третього розділу.	31.09.2024	
6.	Оформлення та нормоконтроль	29.12.2024	
	кваліфікаційної роботи магістра.		
7.	Захист кваліфікаційної роботи.	12.12.2024	

Студент _____ Д.В. Строков
(підпис) (ініціали та прізвище)

Керівник роботи _____ Г.А. Добровольский
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г.Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка Telegram бота для подорожей»: 48 с., 19 рис., 8 джерел.

ГЕОЛОКАЦІЯ, ІНТЕГРАЦІЯ API, МОВА ПРОГРАМУВАННЯ, ПЛАНУВАННЯ ПОДОРОЖЕЙ, ЧАТ-БОТ, TELEGRAM.

Об’єкт дослідження – процес розробки telegram бота для подорожей.

Дослідження зосереджене на кількох важливих аспектах створення програмного забезпечення, таких як вибір технологій, проектування, інтеграція з зовнішніми API для пошуку туристичних атракцій, реалізація інтуїтивно зрозумілого користувацького інтерфейсу.

Мета роботи: розробка Telegram бота для подорожей та дослідження процесу створення такого типу програмного забезпечення. Основні завдання роботи включають вибір відповідних технологій розробки, проектування архітектури бота, інтеграцію з зовнішніми API для отримання актуальної інформації, реалізацію функції маршрутизації, а також тестування програмного забезпечення. Результатом роботи буде функціональна та готова до використання Telegram бота для подорожей.

Метод дослідження – аналіз літературних джерел, проектування та розробка Telegram бота.

Для написання коду було використано інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів середовища Visual Studio.

SUMMARY

Master’s Qualifying Theses «Development of a Telegram Bot for Travel»: 48 pages, 19 figures, 8 sources.

GEOLOCATION, API INTEGRATION, PROGRAMMING LANGUAGE, PLANNING TRIPS, CHAT-BOT, TELEGRAM.

Object of the study – the process of developing a Telegram bot for traveling. The study focuses on several important aspects of software creation, such as the choice of technologies, design, integration with external APIs for searching for tourist attractions, implementation of an intuitive user interface.

Aim of the study: to develop a Telegram bot for traveling and analyze the process of creating such software. The main tasks include selecting appropriate development technologies, designing the bot’s architecture, integrating external APIs to retrieve up-to-date information, implementing routing functionality, and testing the software. The outcome of the work will be a functional and ready-to-use Telegram bot for traveling.

Method of research – analysis of literature sources, design and development of the Telegram bot.

The integrated software development environment and a number of other tools of the Visual Studio environment were used to write the code.

ЗМІСТ

Завдання на кваліфікаційну роботу…….. 2
Реферат 4
Summary 5
Вступ 7
1 Огляд існуючих технологій 9
1.1 Платформи для розробки ботів 9
1.2 Вибір типу програмного продукту 12
1.3 Основи розробки Telegram ботів 15
1.4 Огляд існуючих аналогів 17
2 Проект програмного продукту. 19
2.1 Постановка задачі 19
2.2 Опис роботи Telegram бота для подорожей 20
2.3 Варіанти використання та сценарії використання 24
3 Розробка Telegram бота для подорожей 26
3.1 Створення Telegram бота для подорожей у BotFather Telegram 26
3.2 Вибір мови програмування 29
3.3 Розробка Telegram бота для подорожей 30
3.4 Приклад роботи програмного продукту 34
Висновки 41
Перелік посилань 48

ВСТУП

Розробка Telegram ботів – сучасний та перспективний напрямок у сфері програмного забезпечення, який охоплює широкий спектр завдань: від автоматизації бізнес-процесів до створення інтерактивних сервісів для користувачів. Завдяки популярності месенджера Telegram, його відкритому API та зручності інтеграції, боти активно використовуються в таких сферах, як електронна комерція, навчання, туризм та розваги.

Створення Telegram ботів може бути реалізовано як окремими розробниками, так і командами, які спеціалізуються на розробці програмного забезпечення. У процесі розробки важливо враховувати потреби кінцевих користувачів, особливості функціонування API Telegram, інтеграцію зі сторонніми сервісами, а також забезпечення зручності та надійності використання бота.

Програмісти, які займаються створенням ботів, повинні мати навички роботи з мовами програмування, знати особливості архітектури месенджерів і розуміти принципи інтеграції з іншими системами. Їх завдання – не лише написання коду, а й забезпечення безперебійного функціонування продукту.

Метою даної кваліфікаційної роботи є розробка Telegram бота для подорожей, який дозволить користувачам зручно планувати маршрути, отримувати інформацію про туристичні об’єкти та користуватися іншими корисними функціями.

Для досягнення мети необхідно вирішити такі задачі:

– аналіз існуючих рішень: вивчення наявних Telegram ботів для подорожей, їх функціональності, дизайну та зручності використання;
– проєктування архітектури: визначення структури бота, модулів та алгоритмів для його функціонування;

- інтеграція зі сторонніми API: підключення до сервісів для отримання даних про маршрути, туристичні об’єкти, розклади транспорту тощо;
- розробка функціоналу: реалізація основних можливостей бота, таких як пошук інформації, створення маршрутів і взаємодія з користувачем;
- тестування та оптимізація: перевірка функціональності бота, виявлення та виправлення помилок, удосконалення продуктивності.

Структурно робота складається з трьох розділів.

Перший розділ містить аналіз існуючих технологій для розробки Telegram ботів, зокрема огляд платформ для створення ботів, вибір типу програмного продукту та основи розробки Telegram ботів. Також розглянуто огляд існуючих аналогів, щоб виявити переваги та недоліки існуючих рішень.

Другий розділ містить постановку задачі для розробки Telegram бота для подорожей, опис роботи бота. Також тут наведені варіанти використання бота та можливі сценарії використання для користувачів.

Третій розділ присвячений процесу розробки Telegram бота для подорожей. У ньому детально описано етапи створення бота в BotFather, вибір мови програмування для розробки, а також безпосередньо розробка бота та приклади його роботи.

1 ОГЛЯД ІСНУЮЧИХ ТЕХНОЛОГІЙ

1.1 Платформи для розробки ботів

На ринку існує багато платформ для створення ботів, кожна зі своїми перевагами та обмеженнями. Серед них варто відзначити такі популярні рішення, як Facebook Messenger, Viber, WhatsApp і Telegram.

Давайте детальніше розглянемо ці варіанти, щоб зрозуміти, чому Telegram виділяється з натовпу та стає все популярнішим серед розробників ботів.

Першим аналогом є Facebook Messenger Platform: Facebook надає API для створення ботів, які можуть взаємодіяти з користувачами через Messenger.

Переваги:

– доступ до глобальної аудиторії Facebook, яка складає мільярди користувачів;
– інтеграція з іншими продуктами Facebook, такими як Instagram та WhatsApp;
– підтримка мультимедіа та інтерактивних елементів.

Недоліки:

– складна система налаштування: розробникам потрібна реєстрація додатків через Facebook Developers;
– строгі правила платформи обмежують спосіб використання ботів (наприклад, обов’язкове дотримання вікна взаємодії протягом 24 годин);
– залежність від політики Facebook, яка може змінюватися.

Другим аналогом є Viber Bots: Viber дозволяє створювати ботів для взаємодії з користувачами через свій додаток. Це популярне рішення для локальних ринків, наприклад, України.

Переваги:

– простий у налаштуванні, особливо для малого бізнесу;

– зручність використання в регіонах, де Viber має велику аудиторію (Україна, Східна Європа, Близький Схід);
– інтеграція з локальними сервісами.

Недоліки:

– менша аудиторія у порівнянні з Telegram;
– обмежені можливості налаштування функціоналу;
– немає такого широкого спектру API функцій, як у Telegram.

Третім аналогом є WhatsApp Business API: дозволяє створювати ботів для взаємодії з користувачами WhatsApp на основі офіційного API. Це рішення орієнтоване на бізнеси, які хочуть автоматизувати спілкування з клієнтами, підтримувати замовлення або обробляти запити через популярний месенджер.

Переваги:

– широка аудиторія через WhatsApp, одну з найбільших платформ для обміну повідомленнями в світі;
– можливість інтеграції з CRM-системами для бізнес потреб;
– інтерфейс з підтримкою мультимедійних повідомлень.

Недоліки:

– складніша система налаштування та доступу до API порівняно з Telegram;
– необхідність підтвердження бізнес-акаунту, що може зайняти час;
– платні функції для деяких видів взаємодії, зокрема для великих компаній.

Четвертим аналогом є WeChat Official Accounts Platform: дозволяє створювати ботів для взаємодії з користувачами WeChat. Це рішення ідеально підходить для бізнесів, орієнтованих на китайський ринок, які прагнуть автоматизувати обслуговування клієнтів, надсилати сповіщення, інтегрувати платежі через WeChat Pay або створювати інтерактивні сервіси на основі міні-програм. WeChat надає офіційний API для розробки ботів з можливістю глибокої інтеграції в екосистему платформи.

Telegram має ряд вагомих переваг, які роблять його незамінною платформою для розробників ботів:

Широка аудиторія.

Telegram швидко набрав популярності у світі. Це робить його ідеальним місцем для розробки ботів, що можуть охопити глобальну аудиторію.

Простота налаштування через BotFather.

На відміну від конкурентів, реєстрація бота через BotFather займає лише кілька хвилин. Більше того, розробники отримують простий і зрозумілий API для роботи.

Багатий функціонал.

Telegram API дозволяє створювати складні боти з такими можливостями:

– інтерактивні клавіатури;
– надсилання мультимедіа, документів, геолокації;
– інтеграція з іншими сервісами через вебхуки;
– гнучкість використання.

Telegram дозволяє запускати ботів на будь-яких пристроях і операційних системах без обмежень.

Відкритість та безкоштовність.

Telegram не вимагає плати за використання API, що робить його доступним навіть для початківців. Більше того, відсутність реклами в платформі дозволяє зберігати фокус на продуктивності бота.

Широкі можливості кастомізації.

Розробники можуть створювати унікальний функціонал, який обмежений на інших платформах.

Швидка інтеграція з іншими технологіями.

Telegram боти можуть інтегруватися з базами даних, сторонніми API, платіжними системами (наприклад, Stripe, PayPal) та навіть з іншими месенджерами.

Telegram з BotFather – найбільш гнучка та зручна платформа для розробників ботів. Її доступність, широкий спектр функцій і легкість

налаштування роблять її відмінною від конкурентів, таких як Facebook Messenger, WhatsApp чи Viber. Telegram є ідеальним вибором як для новачків, які тільки починають вивчати створення ботів, так і для досвідчених розробників, які хочуть створювати складні інтеграційні рішення для бізнесу або широкого кола користувачів.

1.2 Вибір типу програмного продукту

При виборі типу програмного продукту для автоматизації процесів, взаємодії з користувачами чи надання послуг, важливо врахувати безліч факторів, таких як зручність, ефективність, масштабованість, а також доступність для кінцевого користувача. Одним із найпопулярніших і найбільш ефективних рішень є Telegram платформа. Ось детальніше про те, чому саме Telegram бот є відмінним вибором:

- популярність Telegram: один з найпопулярніших месенджерів у світі. Це забезпечує величезну аудиторію для ботів. Користувачі не потребують додаткових програм для взаємодії з ботами достатньо лише мати Telegram на своєму пристрої, що робить платформу доступною для більшості людей;
- безпека та конфіденційність: забезпечує високий рівень безпеки завдяки використанню шифрування за замовчуванням для всіх чатів. Це важливий аспект, особливо якщо бот має обробляти чутливу або конфіденційну інформацію, наприклад, для банківських послуг, бронювання квитків або обробки замовлень. Відсутність стороннього доступу до даних користувачів є важливою перевагою для багатьох компаній і організацій;
- інтеграція та розвиток: Telegram надає безкоштовне API, що дозволяє розробникам легко створювати боти без значних витрат часу та ресурсів. За допомогою BotFather офіційного інструмента для

створення ботів можна швидко налаштувати і запустити бота без потреби в складних програмних знаннях. Розробка ботів не вимагає значних фінансових вкладень, що робить цей продукт доступним навіть для стартапів або малого бізнесу;

– мультимедійність і інтерактивність: Telegram боти підтримують різноманітні формати контенту, включаючи текст, зображення, відео, аудіо, кнопки для вибору варіантів, опитування та інтерактивні меню. Це дозволяє створювати гнучкі й багатофункціональні боти, які можуть не тільки надавати інформацію, а й здійснювати складні взаємодії з користувачами. Наприклад, бот може здійснювати бронювання, надавати рекомендації або проводити опитування, що значно покращує взаємодію з користувачем;
– додаткове програмне забезпечення: Telegram боти не потребують установки або оновлень на стороні користувача, оскільки вони працюють безпосередньо в месенджері. Це дуже зручно як для розробників, так і для користувачів, оскільки відсутні додаткові бар’єри для початку взаємодії з ботом;
– підтримка багатьох платформ і пристроїв: Telegram доступний на різних пристроях, таких як смартфони, планшети і ПК, а також підтримує всі основні операційні системи (iOS, Android, Windows, macOS, Linux). Це забезпечує однакову взаємодію для користувачів незалежно від того, яке пристрій вони використовують, а також дозволяє компаніям охоплювати більш широкий ринок;
– підтримка багатомовності та локалізація: Telegram має підтримку багатьох мов, що дозволяє розробляти боти, які можуть працювати з різними аудиторіями по всьому світу. Це важливо для компаній, які мають міжнародну клієнтську базу, оскільки бот може надавати послуги кількома мовами, що покращує досвід взаємодії з користувачем і забезпечує більшу доступність для різних культурних груп;

- масштабованість: боти можуть бути налаштовані і розширені під різні вимоги користувача або розробника. Вони легко масштабуються, дозволяючи впроваджувати нові функції або змінювати існуючі за допомогою простих оновлень. Наприклад, якщо бот розширює свій функціонал, розробник може легко додати нові інтерактивні елементи, оновити інформацію або змінити алгоритми роботи без необхідності оновлювати саму програму або створювати нові версії додатка;
- вартість розробки та використання: боти не потребують платних ліцензій або значних витрат на інфраструктуру. Telegram надає всі необхідні інструменти для розробки безкоштовно, що робить цю платформу ідеальною для стартапів, малого бізнесу та організацій, які хочуть мінімізувати витрати на технології.

Вибір бота Telegram як типу програмного продукту для автоматизації взаємодії з користувачами та надання послуг є хорошим рішенням, враховуючи численні переваги, які пропонує цей інструмент.

Telegram – одна з найпопулярніших комунікаційних платформ, що забезпечує доступ до широкої аудиторії без встановлення додаткових програм, робить взаємодію з ботами зручним і доступним для користувачів.

Завдяки високій безпеці, можливості інтеграції в різні системи та простоті використання Telegram дозволяє швидко впроваджувати нові функції без значних фінансових витрат.

Підтримуючи мультимедіа та інтерактивність, можна створювати багатофункціональні рішення, які значно підвищують ефективність взаємодії з користувачем. Розширюваність бота також дозволяє вам адаптувати бота до конкретних бізнес-потреб. Ще одна важлива перевага полягає в тому, що користувачі можуть отримати доступ до нього з різних пристроїв і операційних систем. Це робить Telegram універсальним для різних цільових груп. Локалізація та багатомовність також є важливими факторами, які дозволяють нам працювати на міжнародних ринках.

Усі ці функції роблять Telegram ідеальним вибором для компаній, які хочуть автоматизувати процеси, покращити взаємодію з клієнтами та зменшити витрати на розробку та підтримку програмних продуктів.

1.3 Основи розробки Telegram ботів

Розробка Telegram ботів не вимагає складних налаштувань, і основи роботи з ними легко освоїти. Основна ідея полягає в тому, що бот працює через Telegram API і дозволяє автоматизувати різні процеси: від обробки повідомлень до інтеграції зі сторонніми сервісами.

Глобальна структура розробки Telegram бота:

– ідея та концепція: чи це буде інструмент для бізнесу, наприклад, для обробки замовлень? Або це сервіс для персональної автоматизації, як бот для нагадувань? Можливо, це розважальний бот із міні-іграми чи інтерактивними історіями? Перший етап включає створення концепції, визначення основних сценаріїв використання та аудиторії;
– вибір платформи: Telegram пропонує дуже зручну платформу для створення ботів завдяки своєму відкритому API;
– реєстрація: кожен бот починається з реєстрації через BotFather – офіційний бот Telegram для керування іншими ботами. Він виконує роль адміністратора та інструмента для розробників. Створення нового бота та отримання унікального токена. Налаштування параметрів, таких як опис, аватар, список команд. Інструмент для керування API токенами.

Bot Father є офіційним ботом Telegram, який діє як «менеджер» для вашого боту. Це обов’язковий елемент для кожного розробника, оскільки дозволяє:

– реєстрацію бота: за допомогою команди /newbot можна створити нового бота;

– отримання токену: після створення бота BotFather надає унікальний токен, що використовується для підключення вашого програмного забезпечення до Telegram API;
– налаштування: BotFather дозволяє змінити ім’я бота, його опис, аватар, а також налаштувати інтерактивні команди;
– інтеграції: через BotFather можете підключити вашого бота до вебхуків або включити режим інтерактивних кнопок.

Переваги Telegram ботів:

– доступність: боти працюють на всіх пристроях, де є Telegram;
– масштабованість: Telegram бот може обслуговувати необмежену кількість користувачів одночасно;
– інтеграція: можливість підключення до інших сервісів через API для розширення функціоналу;
– персоналізація: Боти можуть адаптуватися до потреб користувачів, зберігаючи інформацію про попередні взаємодії.

Взаємодію між людиною та Telegram ботом можна описати так:

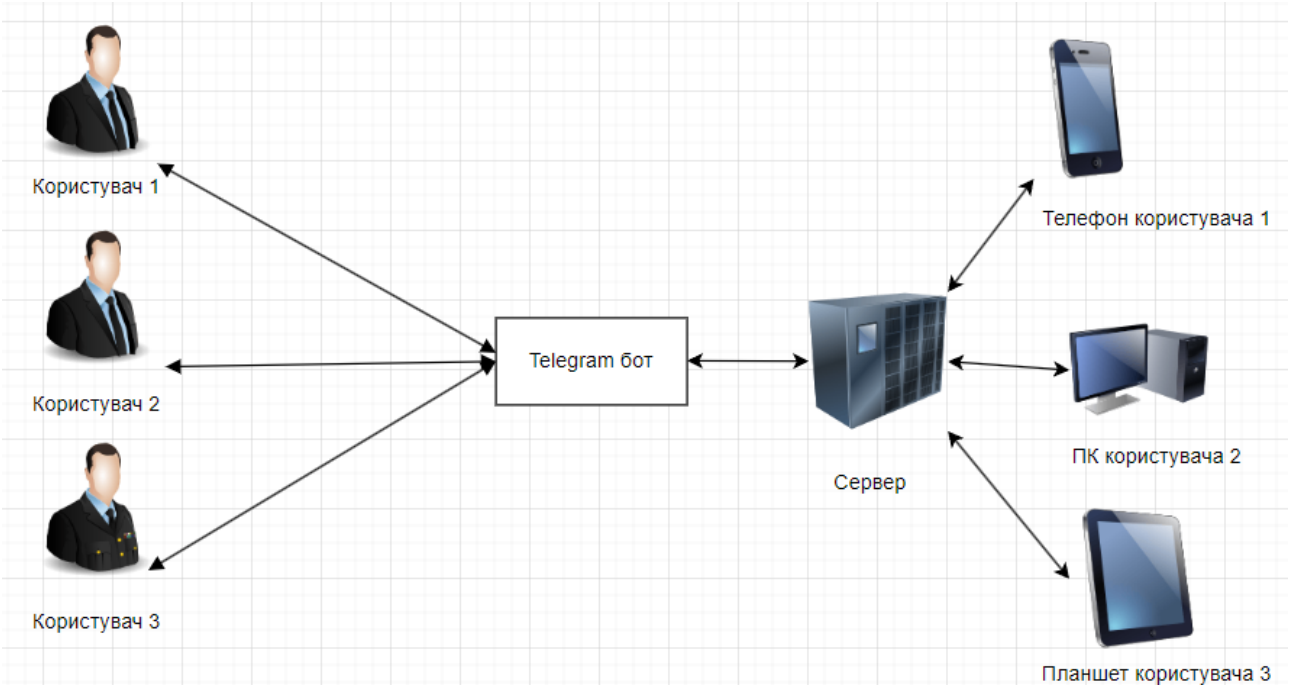


Рисунок 1.1 – Взаємодія між людиною та Telegram ботом

Узагальнюючи, не існує чіткої основи для розподілу ботів. Причина у тому, що вони зазвичай поєднують декілька функцій і успішно виконують багато завдань одночасно.

1.4 Огляд існуючих аналогів

У світі ботів для подорожей існує ряд платформ, які надають подібні функціональності до Telegram бота. Вони дозволяють користувачам отримувати інформацію, планувати поїздки, здійснювати бронювання та отримувати рекомендації за допомогою автоматичних відповідей або інтерактивних можливостей.

Першим аналогом є OpenTripBot – основною метою є допомога користувачам у знаходженні цікавої інформації про місця для відвідування.

Переваги:

– бот надає можливість користувачам інтерактивно отримувати інформацію про місця;
– відповіді включають текстові повідомлення, зображення та карти для зручності користувачів;
– відображає відстань від користувача до місця;
– зручний інтерфейс.

Недоліки:

– немає можливості обрати декілька місць одразу;
– відображає тільки місцезнаходження місць без додаткової інформації, наприклад: як добратися до цього місця;
– підтримка тільки двох мов: англійська, російська.

Другим аналогом є Kayak Bot – це популярний сервіс для планування подорожей. Він дозволяє користувачам шукати авіаквитки, готелі, автомобілі для оренди та бронювати їх через месенджер Facebook.

Переваги:

– включає не лише пошук авіаквитків та готелів, але й оренду автомобілів;
– зберігає попередню інформацію розмови та використовує історію пошуку для персоналізації;
– перенаправляє користувача на офіційні сторінки для оренди, пошуку авіаквитків та готелів.

Недоліки:

– потрібна реєстрація користувача на платформі Facebook;
– підтримує тільки одну мову: англійська;
– як і будь-який сервіс, що залежить від сторонніх постачальників (авіакомпаній, готелів), бот може відображати не зовсім актуальні або точні ціни та пропозиції, що може призвести до непорозумінь або втрати користувачами певних вигод.

Третім аналогом є AI Travel Bot- який використовує штучний інтелект для надання допомоги в плануванні подорожей і пропонує користувачам зручний спосіб знайти варіанти для подорожей, бронювання, а також отримати персоналізовані поради та рекомендації.

Переваги:

– бот може допомогти знаходити дешеві авіаквитки, автобуси чи потяги, порівнюючи ціни на основі популярних сервісів, таких як Kayak;
– інтеграція з платформами для бронювання готелів дозволяє користувачам знаходити найкращі варіанти для зупинки;
– надаються новини та важлива інформація щодо безпеки, зміни візових вимог чи інших подій.

Недоліки:

– використовує дані від сторонніх постачальників, тому точність і актуальність пропозицій можуть варіюватися;
– в подальшому можливі витратити на функції ШІ.

2 ПРОЕКТ ПРОГРАМНОГО ПРОДУКТУ

2.1 Постановка задачі

Створення Telegram бота для подорожей, який допомагатиме користувачам планувати поїздки, відкривати цікаві місця поруч, прокладати маршрути, а також залишати й переглядати відгуки. Основна мета – надати користувачам швидкий доступ до інформації, зручний інструмент взаємодії.

Функціональні вимоги:

а) пошук цікавих місць:

– визначення туристичних атракцій поблизу за допомогою API Google Places;

– можливість обирати кілька місць для створення маршруту;

б) побудова маршрутів:

– інтеграція з Google Maps для створення маршруту між обраними точками;

– генерація посилань для перегляду маршрутів у Google Maps;

в) відгуки:

– користувачі можуть залишати свої відгуки після подорожі;

– відгуки доступні для перегляду іншим користувачам;

– модерація відгуків через спеціальні команди;

г) інтерактивний інтерфейс:

– кнопки для взаємодії (надсилання геолокації, обрання місць);

– інтуїтивно зрозуміле меню для користувачів;

д) адаптація до платформ:

– бот працює однаково на смартфонах, планшетах;

– не потребує встановлення додаткових програм.

2.2 Опис роботи Telegram бота для подорожей

Алгоритм роботи Telegram бота для подорожей має наступний вигляд:

а) старт бота:

– користувач надсилає команду /start.

Відповідь бота:

– виводить вітальне повідомлення;

– пропонує надіслати геолокацію або перезавантажити бота;

– відображає кнопку для перегляду відгуків.

б) перезавантаження:

– користувач натискає кнопку «Перезавантажити бота»;

– отримання вітального повідомлення.

в) отримання геолокації:

– користувач надсилає своє місцезнаходження;

– при помилці користувач отримає повідомлення.

Відповідь бота:

– зберігає координати;

– викликає функцію, яка надсилає запит до Google Places API;

– користувач отримує список туристичних місць поблизу;

– відображає знайдені місця, як кнопки для вибору, додаючи кнопку «Готово».

г) вибір місць:

– користувач натискає кнопки з назвами місць;

Відповідь бота:

– додає вибрані місце;

– оновлює кнопки, відмічаючи обрані місця;

– продовжує приймати вибір місць до натискання кнопки «Готово»;

д) побудова маршруту:

– користувач натискає кнопку «Готово».

Відповідь бота:

– перевіряє кількість вибраних місць: якщо менше двох, надсилає попередження. Якщо два або більше, то отримує деталі обраних місць. Після цього формує маршрут у Google Maps зі стартовою, кінцевою точкою та проміжними місцями;

е) додавання відгуків:

– користувач натискає кнопку «Залишити відгук».

Відповідь бота:

– просить користувача написати свій відгук;

– зберігає відгук разом з іменем користувача;

– дякує за залишення відгука;

ж) перегляд відгуків:

– користувач натискає кнопку «Ось що люди думають про мене».

Відповідь бота:

– перевіряє чи є збережені відгуки: якщо таких відгуків немає, то виводить повідомлення, що відгуки відсутні. Якщо є, відображає всі відгуки;

з) обробка помилок:

– якщо користувач намагається виконати дію без достатніх даних (наприклад, немає вибраних місць або відсутня геолокація), бот надсилає відповідні повідомлення з попередженнями.

Цей алгоритм забезпечує простий та інтуїтивний сценарій взаємодії користувача з ботом, дозволяючи швидко знайти цікаві місця, спланувати маршрут та залишити або переглянути відгуки.

Діаграма, що відображає алгоритм роботи Telegram бота для подорожей. Вона структуровано відображає основні етапи й логіку взаємодії користувача з ботом (див. рис. 2.1).

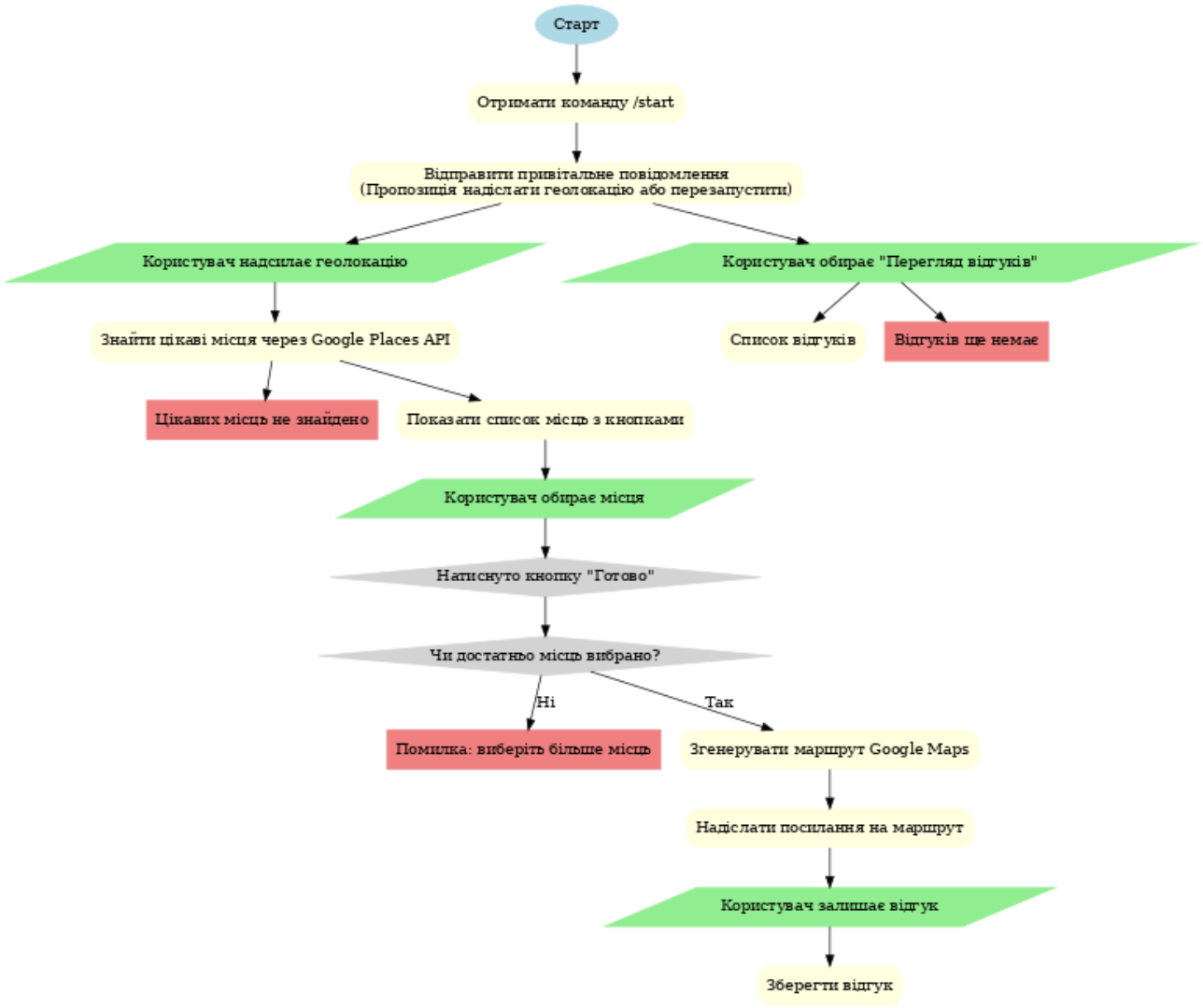


Рисунок 2.1 – Алгоритм роботи Telegram бота для подорожей

Пропонований алгоритм Telegram бота для подорожей відображає чіткий та послідовний процес взаємодії з користувачами, спрямований на забезпечення ефективного та зручного планування подорожей.

Пошук туристичних місць поблизу користувача покрокова схема:

а) початкова взаємодія з ботом:

– користувач натискає кнопку "🔑 Надіслати геолокацію";

– бот запитує геолокацію користувача через кнопку у повідомленні.

б) надсилання геолокації користвучаем:

– користувач надсилає свою геолокацію;

– бот отримує координати (широта та довгота) від користувача;

– обробляє інформацію для подальшої передачі;

в) пошук місць через Google Places API:
   – бот викликає API Google Places, передаючи координати користувача для пошуку туристичних місць поруч, передаються такі дані: координати користувача, радіус до 5000 метрів;
   – Google Places API повертає список місць, таких як музеї, парки, пам’ятники тощо;

г) виведення результатів:
   – бот формує клавіатуру з кнопками для кожного знайденого місця;
   – користувач вибирає місця, що його цікавлять;

д) оновлення вибору:
   – якщо користувач вибрав місце, бот додає його до списку обраних атракцій і оновлює клавіатуру;
   – бот чекає подальших виборів або натискання кнопки "✅ Готово";

Покрокова схема створення маршруту та генерація посилання:

а) вибір місць для маршруту:
   – користувач вибирає кілька місць із результатів пошуку (наприклад, музеї, парки);
   – користувач натискає кнопку "✅ Готово", коли завершив вибір;

б) перевірка вибору:
   – бот перевіряє, чи вибрано хоча б два місця;
   – якщо вибрано менше ніж два місця, бот повідомляє, що потрібно вибрати хоча б два місця;

в) формування маршруту:
   – бот створює список координат (широта, довгота) для кожного вибраного місця;
   – бот формує посилання на Google Maps для маршруту з усіма точками;

г) надсилання посилання на маршрут:
   – бот відправляє користувачеві посилання на маршрут на Google Maps;

– користувач може побачити маршрут та натискати на нього, щоб відкрити в Google Maps.

Інтеграція з Google Places API дозволяє надавати актуальну інформацію про довколишні пам’ятки, а можливість збирати та відображати відгуки допомагає підвищити довіру та залучення користувачів.

2.3 Варіанти використання та сценарії використання

У цьому розділі розглянемо основні варіанти використання та сценарії використання для бота, що дозволяє знаходити туристичні атракції поблизу та створювати маршрути. Це дозволить зрозуміти, як користувачі можуть взаємодіяти з ботом у різних ситуаціях, а також що саме вони можуть отримати в результаті кожної дії.

Варіанти використання:

а) користувач може отримати список цікавих місць поблизу, в межах радіусу 5000 метрів, наприклад: під час подорожі або на відпочинку в новому місті;

б) користувач може створити маршрут для відвідування кількох атракцій в певній послідовності. Це дозволяє спланувати подорож, включаючи основні цікаві місця;

в) користувач може поділитися своїми враженнями від подорожі або відвідування конкретних місць, залишаючи відгуки про бот або туристичні місця.

Сценарії використання:

г) користувач подорожує містом і хоче знайти цікаві місця поблизу:

– користувач відкриває бота та натискає кнопку "📍 Надіслати геолокацію";

– бот отримує координати користувача та надсилає запит до Google Places API для пошуку туристичних атракцій поруч;

- користувач переглядає список місць і вибирає атракції, що його цікавлять;
- користувач натискає "✅ Готово" після вибору місць;
- бот створює маршрут і надає посилання на нього через Google Maps;
- користувач використовує посилання для навігації та планування маршруту;

д) після відвідування кількох місць, користувач хоче поділитися своїми враженнями про місця або про самого бота:

- користувач натискає кнопку "📝 Залишити відгук";
- бот просить ввести відгук;
- користувач пише відгук і надсилає його;
- бот підтверджує отримання відгуку та повідомляє, що він буде доданий до списку;

е) користувач хоче перезавантажити взаємодію з ботом і почати новий процес пошуку:

- користувач натискає кнопку "🔄 Перезавантажити бота";
- бот запитує геолокацію та повторно починає процес пошуку цікавих місць;

ж) у випадку, якщо система не знаходить жодних туристичних місць поруч:

- користувач намагається знайти місця, але API не дає жодних результатів;
- бот відправляє повідомлення з повідомленням про помилку: «На жаль, не вдалося знайти цікаві місця поруч»;
- користувач може спробувати ще раз або перезапустити процес.

Ці варіанти та сценарії використання покривають основні випадки, коли користувач може взаємодіяти з ботом. Вони допомагають зрозуміти, як бот буде використовуватися на практиці для пошуку цікавих місць та створення маршрутів.

3 РОЗРОБКА TELEGRAM БОТА ДЛЯ ПОДОРОЖЕЙ

3.1 Створення Telegram бота для подорожей у BotFather Telegram

Щоб створити Telegram-бота, першим кроком є використання офіційного сервісу Telegram під назвою BotFather. Цей сервіс є інструментом, який дозволяє створювати, налаштовувати та керувати ботами.

Після того, як знайдемо BotFather, потрібно натиснути кнопку «Старт», отримаємо повідомлення (див. рис. 3.1).

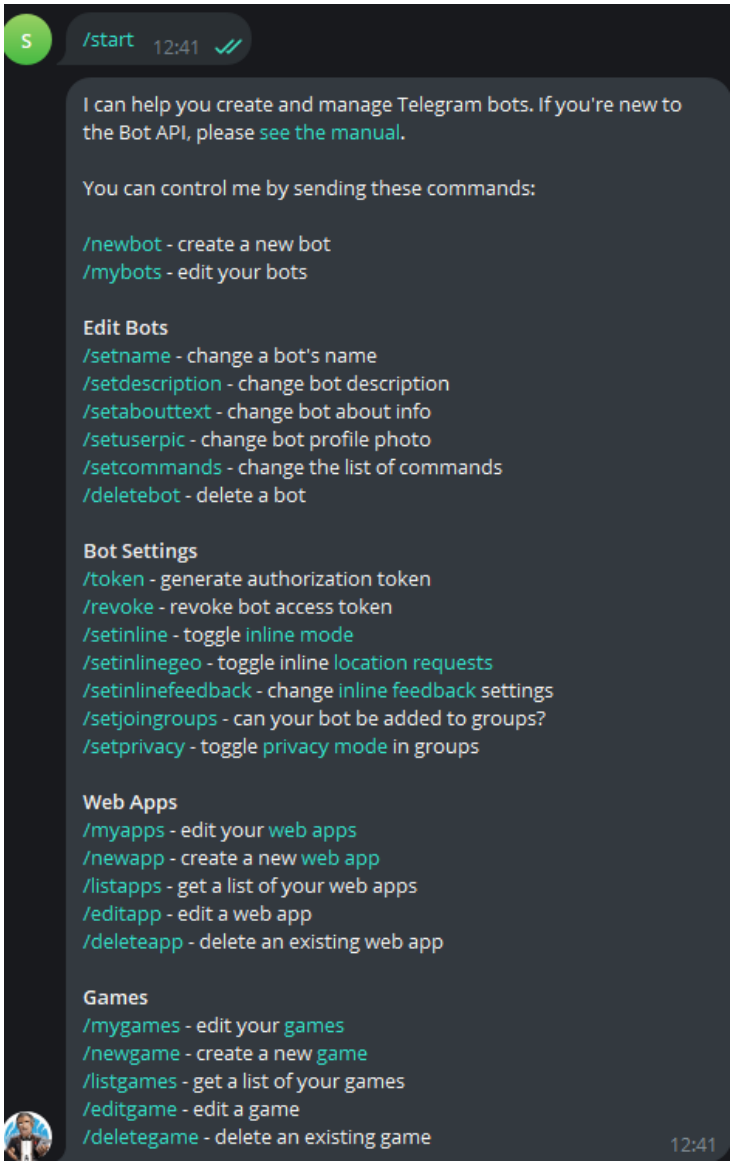


Рисунок 3.1 – Початок роботи з BotFather

Обираємо команду «Створити нового бота», бот відповість, що потрібно створити ім’я бота та унікальний ідентифікатор для нашого бота. Після надання потрібної інформації, отримаємо токен доступу (див. рис. 3.2).

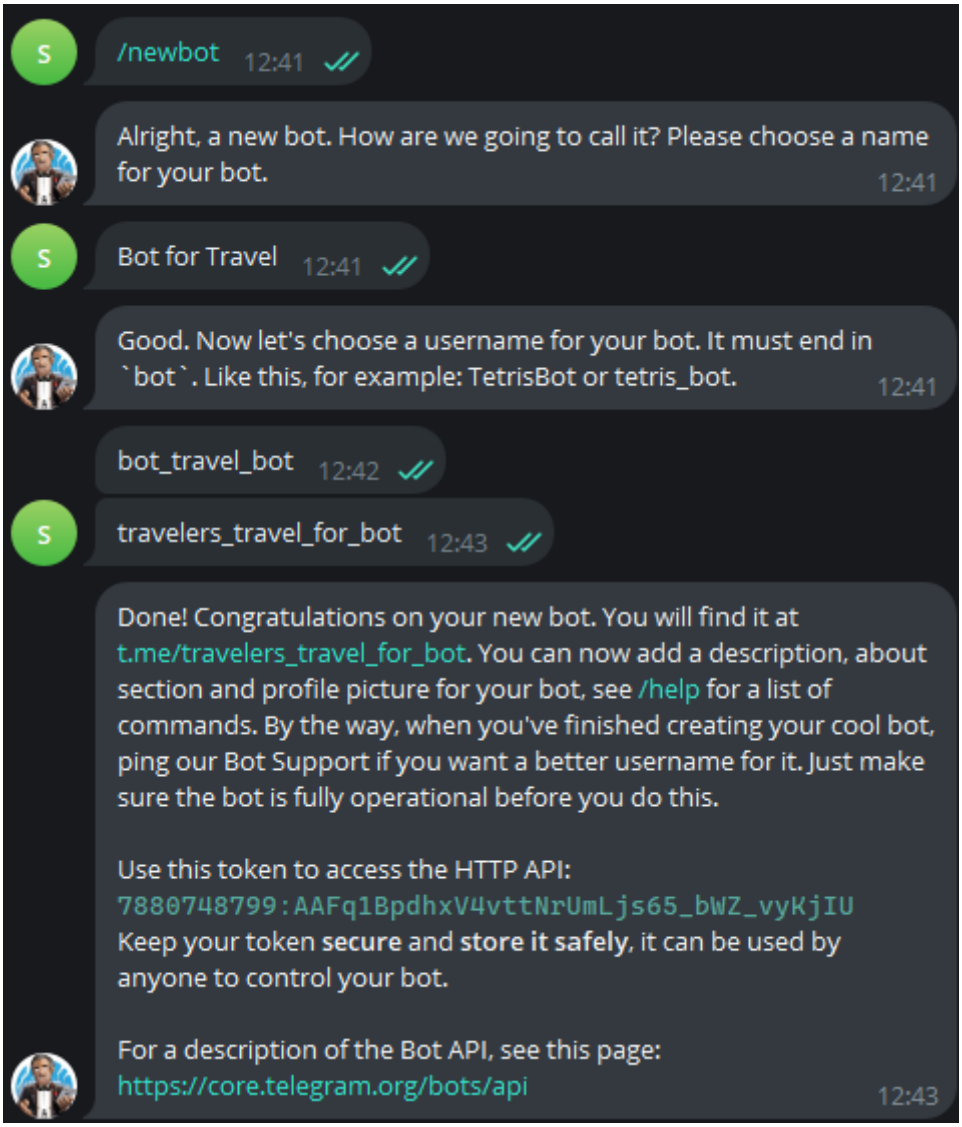


Рисунок 3.2 – Створення ім’я, ідентифікатора бота та токен доступу

Опис бота – це не лише інформаційний блок, а важливий елемент, який сприяє покращенню досвіду користувача, забезпечує чіткість взаємодії і допомагає ефективно використовувати бота (див. рис. 3.3).

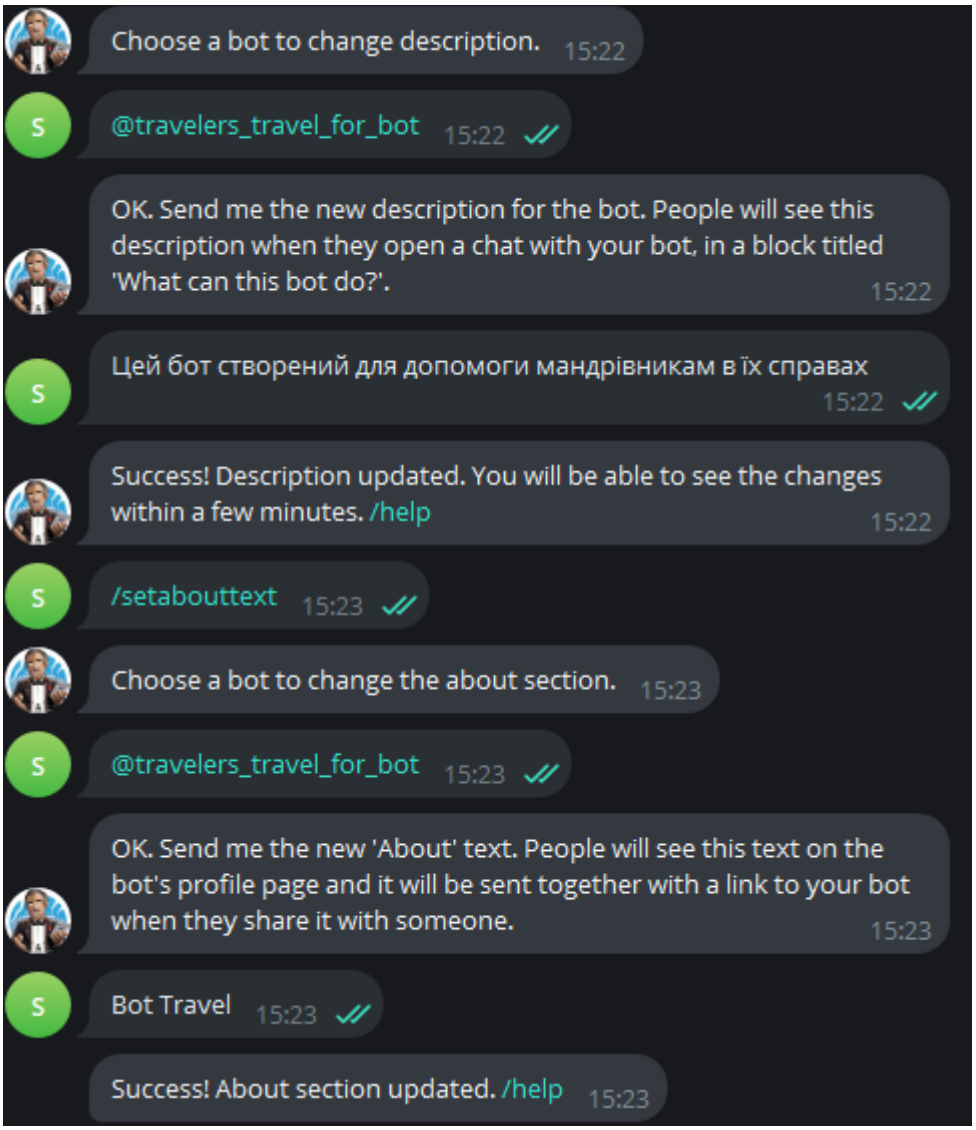


Рисунок 3.3 – Додавання опису бота

Аватар для бота – це не лише елемент візуального дизайну, а й важливий інструмент для взаємодії з користувачами. Вона може допомогти створити унікальний образ вашого бота, підвищити довіру користувачів, забезпечити професійний вигляд і покращити досвід користувача, що важливо для будь-якого бота (див. рис. 3.4).

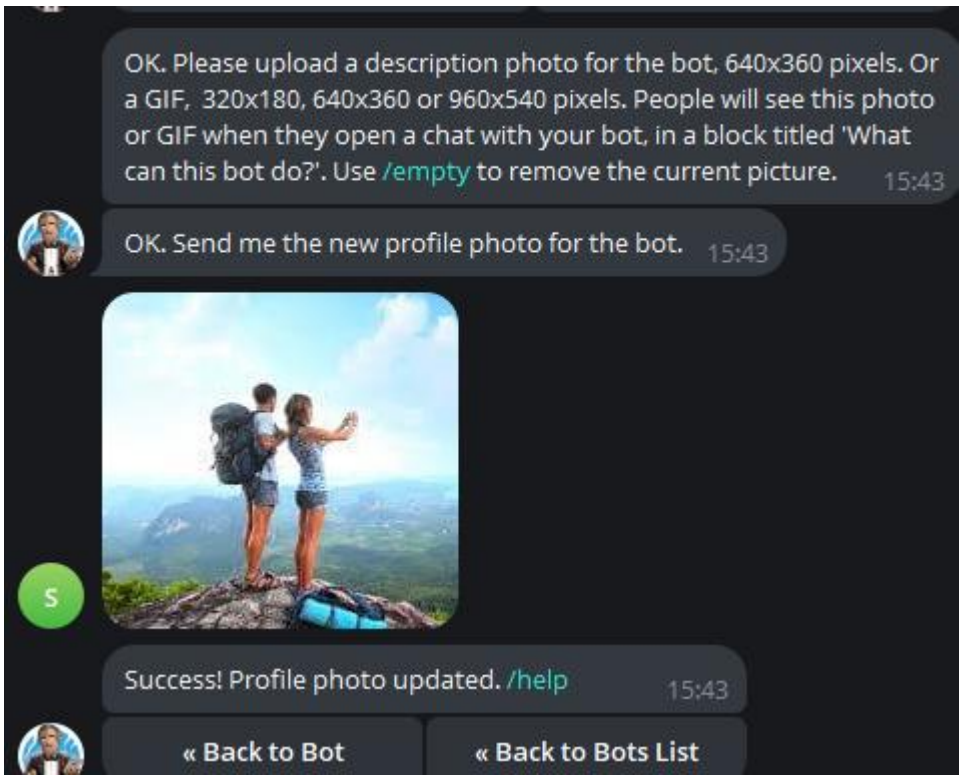


Рисунок 3.4 – Додавання аватару до бота

BotFather надає користувачам покрокові інструкції щодо створення нового бота, включаючи генерацію унікального токена доступу для подальшої інтеграції та розробки.

Завдяки цьому сервісу налаштування ключових параметрів, таких як ім’я, опис та інші характеристики бота, стає легким і зручним процесом.

3.2 Вибір мови програмування

Для розробки Telegram бота для подорожей використовувалася мова програмування Python.

Python – популярна мова програмування, відома своєю простотою, універсальністю та великою кількістю бібліотек. Python використовується в багатьох сферах, від веб-розробки до наукових досліджень до штучного інтелекту та автоматизації завдань. Його кросплатформенність дозволяє запускати бота на будь-якому пристрої, а активна спільнота забезпечує доступ до великої кількості навчальних матеріалів та прикладів. Python надає

розробникам можливість швидко створювати прототипи та легко адаптувати код до нових вимог.

Особливості мови програмування Python:

а) легкість у вивченні та використанні: Python має чіткий і легкий для розуміння синтаксис, який дозволяє розробникам швидко писати код. Це робить Python ідеальним вибором для початківців;

б) широка екосистема бібліотек і фреймворків: Python підтримує величезну кількість бібліотек для різних задач, таких як веб-розробка (Django, Flask), обробка даних (Pandas, NumPy), машинне навчання (TensorFlow, scikit-learn) та багато інших. У нашому випадку, для створення Telegram-бота використовуються такі бібліотеки, як requests для роботи з API;

в) кросплатформеність: Python працює на всіх популярних операційних системах (Windows, macOS, Linux). Це дозволяє розробникам запускати свій код на будь-якій платформі, не вносячи значних змін у код.

Python підходить для створення ботів Telegram завдяки своїй простоті, потужним бібліотекам і широким можливостям інтеграції з різними API.

У нашому випадку Python об’єднує всі компоненти, необхідні для створення бота, який взаємодіє з користувачами, обробляє їхні запити, отримує дані за допомогою зовнішніх служб і надає корисні функції.

3.3 Розробка Telegram бота для подорожей

Архітектура цього Telegram бота створена для ефективної взаємодії з користувачами та інтеграції із зовнішніми сервісами.

Telegram API.

Telegram API з його допомогою можна обробляти команди користувачів, отримувати їх геолокаційні дані, створювати кастомізовані кнопки для зручної навігації та надсилати інтерактивні повідомлення.

Google Places API.

Цей API застосовується для отримання інформації про туристичні місця поблизу користувача. Використовується метод nearbysearch для пошуку цікавих локацій за координатами та метод details для отримання детальної інформації про вибрані місця.

Бібліотека requests.

Це популярна бібліотека Python, яка використовується для надсилання HTTP-запитів. Це спрощує роботу з REST API, обробку відповідей із серверів і взаємодію з веб-сервісами.

Inline кнопки Telegram.

Це дозволяє зручно організувати навігацію по бот-інтерфейсу, а також взаємодіяти з користувачами без необхідності обробляти текстові команди вручну. Наприклад, кнопки для вибору місць або для підтвердження дій. Це дозволяє користувачам легко вибирати цікаві місця, переглядати відгуки та взаємодіяти з ботом без необхідності вводити текстові команди.

Ці технології обрані завдяки своїй простоті, ефективності та здатності створювати ботів Telegram, які взаємодіють з користувачами, використовують географічну інформацію та працюють з API для надання послуг для пошуку пам’яток і створення маршрутів.

Перейдемо до написання коду для Telegram бота. На самому початку потрібно підключити усі необхідні бібліотеки, API та деякі налаштування для роботи (див. рис. 3.5).

Властивості:

– requests – бібліотека для роботи з HTTP-запитами. Використовується для взаємодії з API Google Places та іншими веб-сервісами;
– telebot – бібліотека для роботи з Telegram API, що дозволяє створювати бота;

```
import requests
import telebot
from telebot.types import ReplyKeyboardMarkup, KeyboardButton,
InlineKeyboardButton, InlineKeyboardMarkup

PLACES_API_KEY = 'AIzaSyCVWIYvZTrkxyiA4VKH_S0fY8cLxNjlU2U'
MODERATOR_ID = 7573998428
bot = telebot.TeleBot("7880748799:AAFq1BpdhxV4vttNrUmLjs65_bWZ_vyKjIU")

selected_attractions = []
user_location = {}
user_reviews = []
```

Рисунок 3.5 – Підключення необхідних бібліотек та API

– PLACES_API_KEY – ключ API для Google Places, що дає змогу використовувати сервіси Google для пошуку туристичних місць;
– MODERATOR_ID – ID адміністратора бота, який має права видаляти відгуки користувачів;
– bot – створюється об'єкт Telegram бота, використовуючи токен;
– selected_attractions, user_location, user_reviews – списки для зберігання вибраних місць, геолокації користувачів і відгуків користувачів відповідно.

Створення вітального повідомлення для користувача (див. рис 3.6).

Властивості:

– @bot.message_handler(commands=['start']) – декоратор, який вказує на обробку команди /start. Коли користувач надсилає команду /start, бот виконує функцію start(message);
– location_button = KeyboardButton(text="🔑 Надіслати геолокацію", request_location=True) – створення кнопки для відправки геолокації користувачем. Кнопка запитує у користувача доступ до його місцезнаходження за допомогою Telegram API. Користувач може натискати на неї, щоб поділитися своєю локацією;

```
@bot.message_handler(commands=['start'])
def start(message):
    location_button = KeyboardButton(
        text="📍 Надіслати геолокацію",
        request_location=True
    )
    restart_button = KeyboardButton(
        text="🔄 Перезавантажити бота"
    )
    reply_markup = ReplyKeyboardMarkup(
        resize_keyboard=True
    ).add(location_button, restart_button)
    reviews_button = InlineKeyboardButton("📄 Ось що люди думають про мене",
callback_data="show_reviews")
    review_markup = InlineKeyboardMarkup().add(reviews_button)
    bot.send_message( message.chat.id,
        (
                "Привіт! 👋 Я твій особистий гід по місту 🗺. "
                "Разом ми відкриємо найцікавіші куточки поруч! 😎 " "Надішли
                своє місцезнаходження, і я підберу для тебе найкращі маршрути!
✨\n\n"),
        reply_markup=reply_markup)
    bot.send_message(
        message.chat.id,
        "Що люди кажуть про мене? 😁 Натисни кнопку нижче, щоб дізнатися!",
        reply_markup=review_markup
    )
```

Рисунок 3.6 – Обробка команди «Старт»

– restart_button = KeyboardButton(text="🔄 Перезапустити бота") – створення кнопки для перезапуску бота. Користувач може натискати на неї, щоб знову почати взаємодію з ботом;
– reply_markup=ReplyKeyboardMarkup(resize_keyboard=True).add(locatio n_button, restart_button) – створення клавіатури для користувача;
– resize_keyboard=True дозволяє автоматично налаштувати розмір клавіатури під пристрій користувача;
– reviews_button = InlineKeyboardButton("📄 Ось що люди думають про мене", callback_data="show_reviews") – створення інлайн кнопки для перегляду відгуків про бота. При натисканні цієї кнопки бот викликає функцію, яка показує відгуки;
– review_markup = InlineKeyboardMarkup().add(reviews_button) – створення інлайн клавіатури, що містить кнопку для перегляду відгуків;

– bot.send_message(message.chat.id, ...) – надсилання привітального повідомлення користувачеві. У цьому повідомленні бот пояснює, що він є особистим гідом, готовим допомогти знайти цікаві місця поруч і створити маршрути. Також додається клавіатура для вибору геолокації та перезапуску бота;
– bot.send_message(message.chat.id, "Що люди кажуть про мене? 😄 Натисни кнопку нижче, щоб дізнатися!", reply_markup=review_markup) – надсилання іншого повідомлення користувачеві з кнопкою для перегляду відгуків про бота.

Цей фрагмент коду обробляє отриману від користувача геолокацію та знаходить туристичні місця поблизу, використовуючи Google Places API. Коли користувач надсилає свою геолокацію, бот виконує пошук цікавих локацій та відправляє користувачу кнопки для вибору місць. Кожна кнопка відповідає певному місцю, і користувач може вибрати ті, що йому цікаві. Бот також додає кнопку "Готово" для завершення вибору (див. рис. 3.7).

Властивості:

– @bot.message_handler(content_types=[‘location’]) – цей декоратор вказує, що функція location(message) буде викликатися, коли користувач надішле своє місцезнаходження (геолокацію). Це дозволяє боту отримувати координати користувача для подальшого пошуку цікавих місць;
– latitude = message.location.latitude та longitude = message.location.longitude – з отриманого повідомлення, яке містить геолокацію, бот витягує широту (latitude) та довготу (longitude), щоб використовувати їх для пошуку місць поблизу;
– user_location[message.chat.id] = message.location – збереження геолокації користувача в словник user_location, де ключем є ID чату, а значенням – геолокація. Це дозволяє зберігати дані для кожного користувача окремо;

```
@bot.message_handler(content_types=['location'])
def location(message):
    latitude = message.location.latitude
    longitude = message.location.longitude
    user_location[message.chat.id] = message.location
    places = find_attractions(latitude, longitude)

    if not places:
        bot.send_message(
            message.chat.id,
            "😕 На жаль, не вдалося знайти цікаві місця поруч."
        )
        return

    keyboard = []
    for place in places:
        name = place['name']
        place_id = place['place_id']
        button = InlineKeyboardButton(
            name, callback_data=place_id
        )
        keyboard.append([button])

    keyboard.append([InlineKeyboardButton("✅ Готово", callback_data="done")])
    reply_markup = InlineKeyboardMarkup(keyboard)
    bot.send_message(
        message.chat.id,
        (
            "🏘 Ось цікаві місця поруч із вами! "
            "🚶‍♂🌍 Виберіть ті, що вас зацікавили, і давайте вирушимо у
подорож!"
        ),
        reply_markup=reply_markup
    )
```

Рисунок 3.7 – Функція обробки та надання інформації за геолокацією користувача

– places = find_attractions(latitude, longitude) – виклик функції find_attractions(), яка здійснює запит до Google Places API для пошуку туристичних місць поблизу отриманих координат (широти та довготи);
– if not places: – перевірка, чи вдалося знайти місця поблизу. Якщо місця не знайдені (наприклад, API не повернув жодних результатів), бот відправляє повідомлення про те, що не вдалося знайти цікаві місця;
– keyboard = [] – ініціалізація порожнього списку keyboard, який буде містити кнопки для кожного знайденого місця;
– for place in places: – цикл по всіх знайдених місцях. Для кожного місця;
– name = place[‘name’] – отримання назви місця;

– place_id = place[‘place_id’] – отримання унікального ідентифікатора місця;
– button = InlineKeyboardButton(name, callback_data=place_id) – створення кнопки з назвою місця та унікальним ідентифікатором як callback data. Це дозволяє ботові зберігати дані для кожної кнопки;
– keyboard.append([button]) – додавання кнопки до клавіатури;
– keyboard.append([InlineKeyboardButton("✅ Готово", callback_data="done")]) – додавання кнопки "Готово", яка дозволяє користувачу завершити вибір місць;
– reply_markup = InlineKeyboardMarkup(keyboard) – створення об’єкта InlineKeyboardMarkup, що містить всі кнопки, які бот має надіслати користувачу;
– bot.send_message(message.chat.id, ...) – надсилання повідомлення користувачу, де вказується, що бот знайшов цікаві місця поруч, і пропонується вибір таких місць за допомогою кнопок.

Цей фрагмент коду відповідає за пошук туристичних місць поблизу вказаних координат (широта та довгота) за допомогою Google Places API. Функція find_attractions формує URL для запиту до Google Places API, де зазначаються координати користувача, радіус пошуку (5000 м) та тип місць, які потрібно знайти – в даному випадку це туристичні атракції. Далі функція надсилає HTTP-запит і повертає результати у вигляді списку (див. рис. 3.8).

```
def find_attractions(lat, lon):
    url = (
        f'https://maps.googleapis.com/maps/api/place/nearbysearch/json?'

f'location={lat},{lon}&radius=5000&type=tourist_attraction&key={PLACES_API_KEY}'
    )

    response = requests.get(url)

    if response.status_code == 200:
        data = response.json()
        return data.get('results', [])
    return None
```

Рисунок 3.8 – Функція пошуку туристичних місць поблизу користувача

Властивості:

– lat, lon: координати місцезнаходження користувача;
– PLACES_API_KEY: ключ API для доступу до сервісів Google Places;
– url: строка, що формує запит до Google Places API з параметрами локації та типу пошуку;
– requests.get(url): надсилає HTTP GET-запит до API;
– response.json(): перетворює відповідь сервера у формат JSON;
– data.get('results', []): отримує список результатів пошуку або порожній список, якщо результатів немає.

Цей фрагмент коду обробляє події, пов'язані з натисканням кнопок в бота. Використовується метод callback_query_handler для того, щоб реагувати на натискання Inline кнопок, які були відправлені користувачу раніше. Кожен callback обробляється через перевірку значення call.data, що передається разом з подією. У залежності від вибору користувача, виконується відповідна дія (див. рис. 3.9).

```
@bot.callback_query_handler(func=lambda call: True)
def button_click(call):
    if call.data == "done":
        done(call.message.chat.id)
    elif call.data == "leave_review":
        bot.send_message(call.message.chat.id, "📝 Напишіть свій відгук нижче:")
        bot.register_next_step_handler(call.message, receive_review)
    elif call.data == "show_reviews":
        show_reviews(call.message.chat.id)
    else:
        selected_attractions.append(call.data)
        update_attraction_keyboard(call.message, selected_attractions)
        bot.answer_callback_query(call.id, text=f"Ви вибрали: {call.data}")
```

Рисунок 3.9 – Функція обробки натискання кнопки

Властивості:

– call.data: це значення, яке містить дані, пов'язані з натиснутою кнопкою (наприклад, done, leave_review, або place_id);
– done(call.message.chat.id): викликається, коли користувач натискає кнопку "Готово", завершуючи вибір локацій;
– bot.send_message(call.message.chat.id, "📝 Напишіть свій відгук нижче:"): просить користувача залишити відгук;

– bot.register_next_step_handler(call.message, receive_review): реєструє наступний крок для обробки відгуку користувача;
– show_reviews(call.message.chat.id): викликається для показу списку відгуків користувачів;
– selected_attractions.append(call.data): додає вибрану локацію в список обраних місць;
– update_attraction_keyboard(call.message,selected_attractions): оновлює клавіатуру, щоб показати оновлений список вибраних місць;

Ця функція done виконується після того, як користувач натискає кнопку "✅ Готово", що сигналізує про завершення вибору локацій для маршруту (див. рис. 3.10).

```
def done(chat_id):
    if not selected_attractions:
        bot.send_message(chat_id, "⚠ Ви не вибрали жодного місця.")
        return
    locations = []
    for place_id in selected_attractions:
        place_details = get_place_details(place_id)
        if place_details:
            lat = place_details['geometry']['location']['lat']
            lon = place_details['geometry']['location']['lng']
            locations.append((lat, lon))
    if len(locations) < 2:
        bot.send_message(chat_id, "⚠ Потрібно вибрати хоча б два місця для
побудови маршруту.")
        return
    origin = locations[0]
    destination = locations[-1]
    waypoints = locations[1:-1]
    waypoints_str = "|".join([f"{lat},{lon}" for lat, lon in waypoints])
    google_maps_url = (
        f"https://www.google.com/maps/dir/?api=1"
        f"&origin={origin[0]},{origin[1]}"
        f"&destination={destination[0]},{destination[1]}"
        f"&waypoints={waypoints_str}"
    )
    bot.send_message(chat_id, f"🗺 Ваш маршрут по вибраним
місцям:\n{google_maps_url}")
    leave_review_button = InlineKeyboardButton("📝 Залишити відгук",
callback_data="leave_review")
    review_markup = InlineKeyboardMarkup().add(leave_review_button)
    bot.send_message(chat_id, "🚀🌍 Приятної подорожі!",
reply_markup=review_markup)
    selected_attractions.clear()
```

Рисунок 3.10 – Створення маршруту та генерація посилання до цього маршруту

Властивості:

- if not selected_attractions:: Якщо користувач не вибрав жодного місця, бот надсилає повідомлення з попередженням, що жодне місце не було обрано, і функція завершиться;
- bot.send_message(chat_id, "⚠ Ви не вибрали жодного місця."): Повідомлення для користувача, коли не вибрано жодного місця;
- locations = []: Створюється порожній список для збереження координат обраних місць;
- Для кожного place_id, що міститься у списку обраних місць, викликається функція get_place_details(place_id), яка повертає деталі про місце;
- lat = place_details[‘geometry’][‘location’][‘lat’]: Отримуються координати місця (широта);
- lon = place_details[‘geometry’][‘location’][‘lng’]: Отримуються координати місця (довгота);
- locations.append((lat, lon)): Координати кожного місця додаються до списку locations;
- if len(locations) < 2: Якщо кількість обраних місць менше двох, бот не може побудувати маршрут;
- bot.send_message(chat_id, "⚠ Потрібно вибрати хоча б два місця для побудови маршруту."): Повідомлення користувачеві, якщо обрано менше двох місць;
- origin = locations[0]: Перше місце у списку стає точкою початку маршруту;
- destination = locations[-1]: Останнє місце у списку стає кінцевою точкою маршруту;
- waypoints = locations[1:-1]: Усі інші місця між початком і кінцем є проміжними точками маршруту;
- waypoints_str = "|".join([f"{lat},{lon}" for lat, lon in waypoints]): Формує рядок з координатами проміжних точок для Google Maps;

– google_maps_url = f"https://www.google.com/maps/dir/?api=1&origin={origin[0]},{origin[1]}&destination={destination[0]},{destination[1]}&waypoints={waypoints_str}": Створюється URL для побудови маршруту в Google Maps;
– bot.send_message(chat_id, f"🗺 Ваш маршрут по вибраним місцях:\n{google_maps_url}"): Бот надсилає користувачу URL з маршрутом, де він може переглянути деталі маршруту на Google Maps.

Цей фрагмент коду виводить відгуки про бота (див. рис. 3.11).

```
def receive_review(message):
    username = message.from_user.username or message.from_user.first_name
    user_reviews.append((username, message.text))
    bot.send_message(message.chat.id, "💬 Дякуємо за ваш відгук!")


def show_reviews(chat_id):
    if not user_reviews:
        bot.send_message(chat_id, "😕 Відгуків поки немає.")
    else:
        reviews_text = "\n\n".join([f"💬 @{username}: {review}" for username,
review in user_reviews])
        bot.send_message(chat_id, f"Ось що люди думають про
мене:\n\n{reviews_text}")
```

Рисунок 3.11 – Функція перегляду відгуків

Властивості:

– user_reviews: Список, що зберігає відгуки користувачів у вигляді кортежів (username, review);
– message.text: Текст відгуку користувача;
– message.from_user.username or message.from_user.first_name: Перевірка на наявність імені користувача (username) або використання його першого імені;
– bot.send_message(): Відправка повідомлення користувачеві після отримання відгуку або при відображенні всіх відгуків;
– Форматування відгуків: Виведення всіх відгуків в один список, кожен відгук оформлений з ім’ям користувача та текстом відгуку.

3.4 Приклад роботи програмного продукту

Тестування проводилося на такому обладнанні: телефон Xiaomi Redmi Note 10 Pro, версія Android 13 TKQ1.221013.002, версія MIUI Global 14.0.8, версія Telegram для Android v11.4.2 (5469) store bundled arm64-v8a.

Після відправки команди /start, бот привітає користувача та розповість деяку інформацію для чого він створений (див. рис. 3.12).



Рисунок 3.12 – Вітальне повідомлення від бота

Після натискання інтерактивної кнопки «Надіслати геолокацію», від бота надійте повідомлення про усі місця поблизу користувача. (див. рис. 3.13).

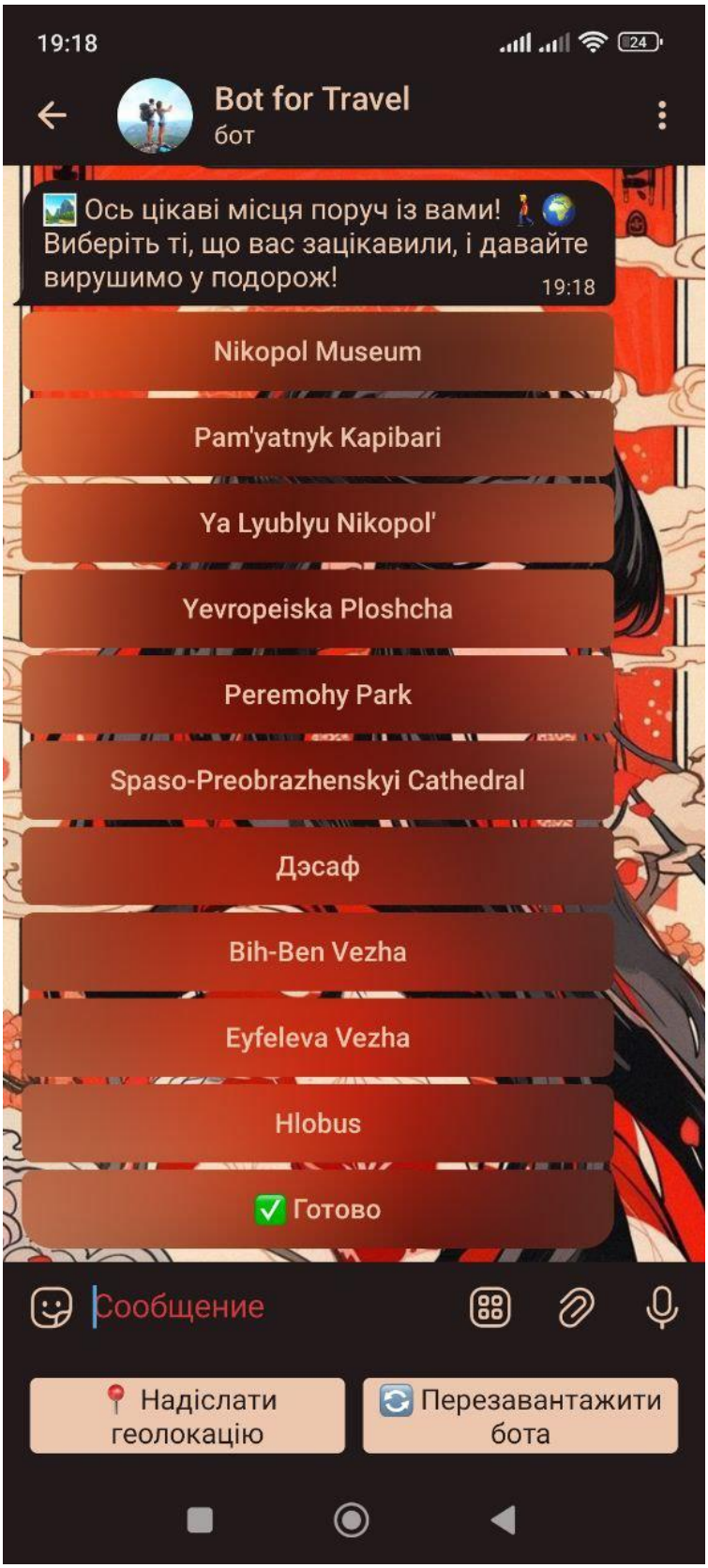


Рисунок 3.13 – Результат дії кнопки «Надіслати геолокацію»

Користувач вибирає потрібні місця, які хоче відвідати та після натискання кнопки «Готово», бот відправить посилання на маршрут (див. рис. 3.14).

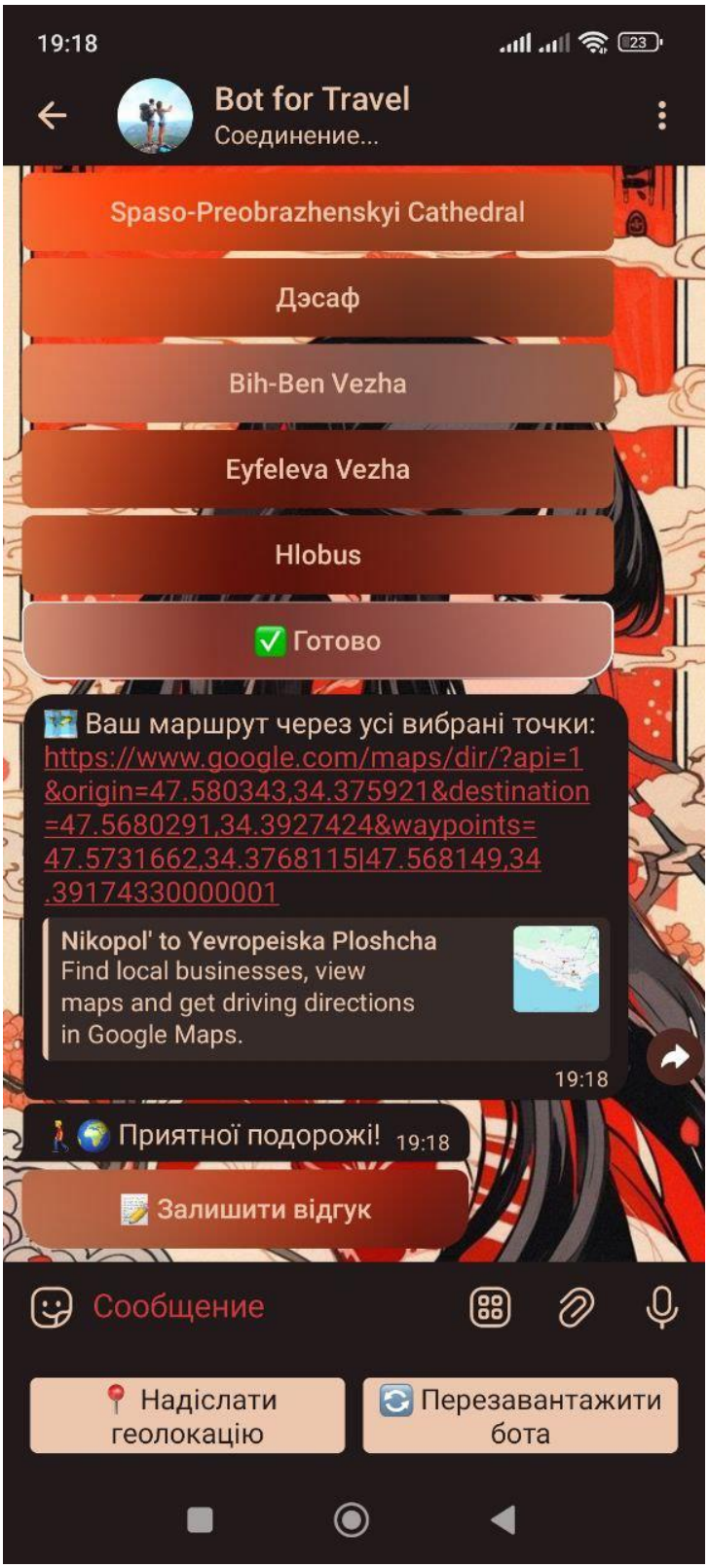


Рисунок 3.14 – Після вибору цікавих місць та натискання кнопки «Готово», отримаємо такий маршрут

Після переходу по посиланню на маршрут, користувачеві відображається його маршрут (див. рис. 3.15).

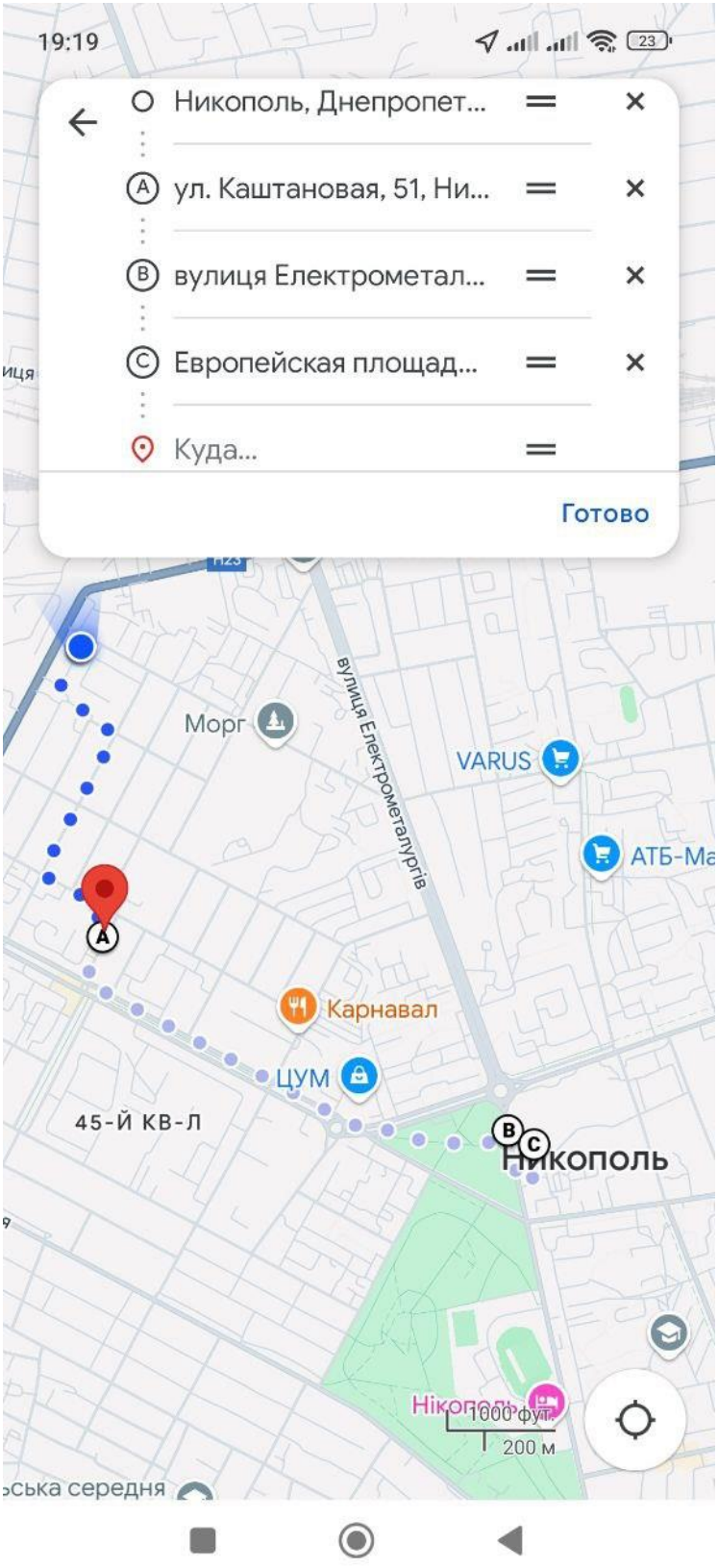


Рисунок 3.15 – Маршрут від бота

Коли користувач отримає посилання на маршрут, бот надсилає йому повідомлення, що користувач може залишити відгук (див. рис. 3.16).

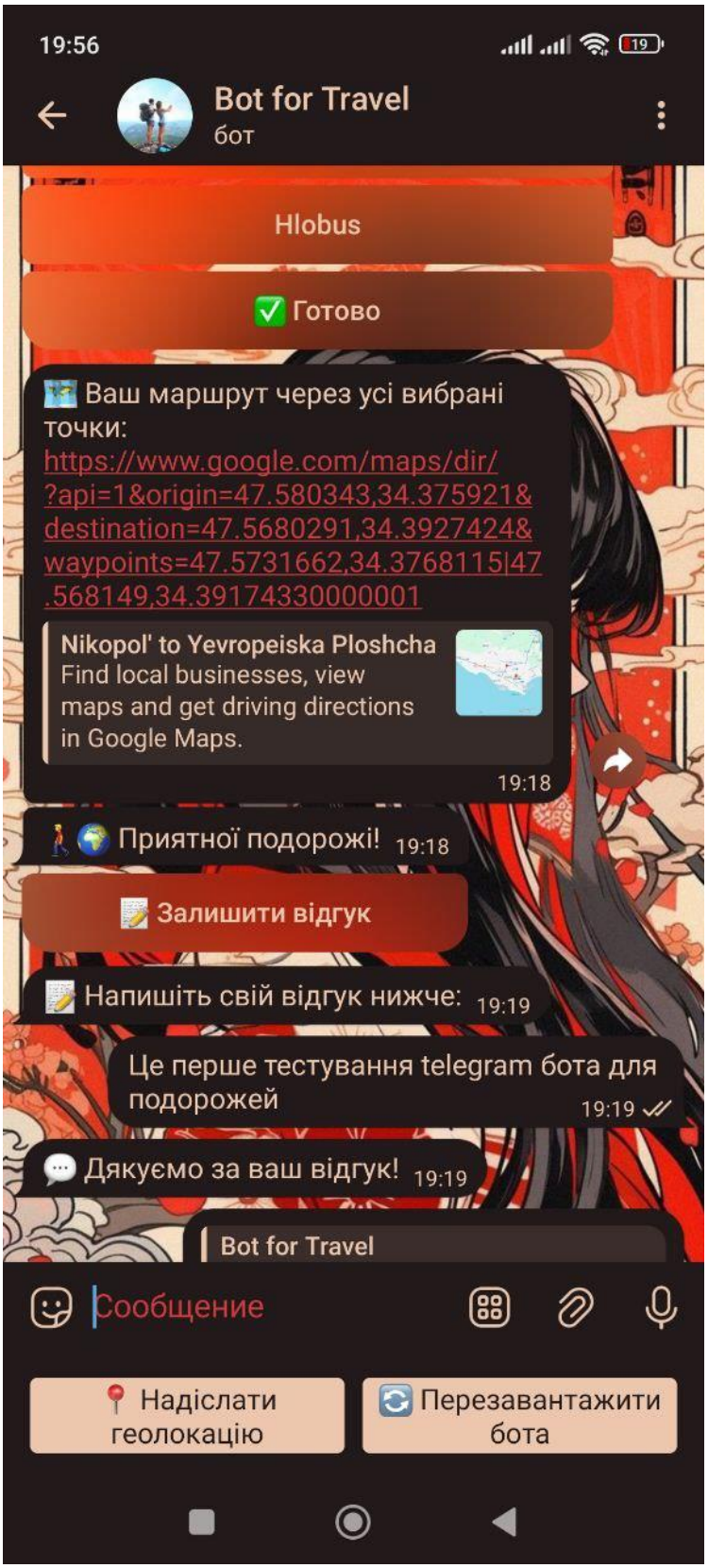


Рисунок 3.16 – Написання відгука

Після того, як користувач залишить свій відгук або інший користувач надається можливість перегляду відгуків (див. рис. 3.17).

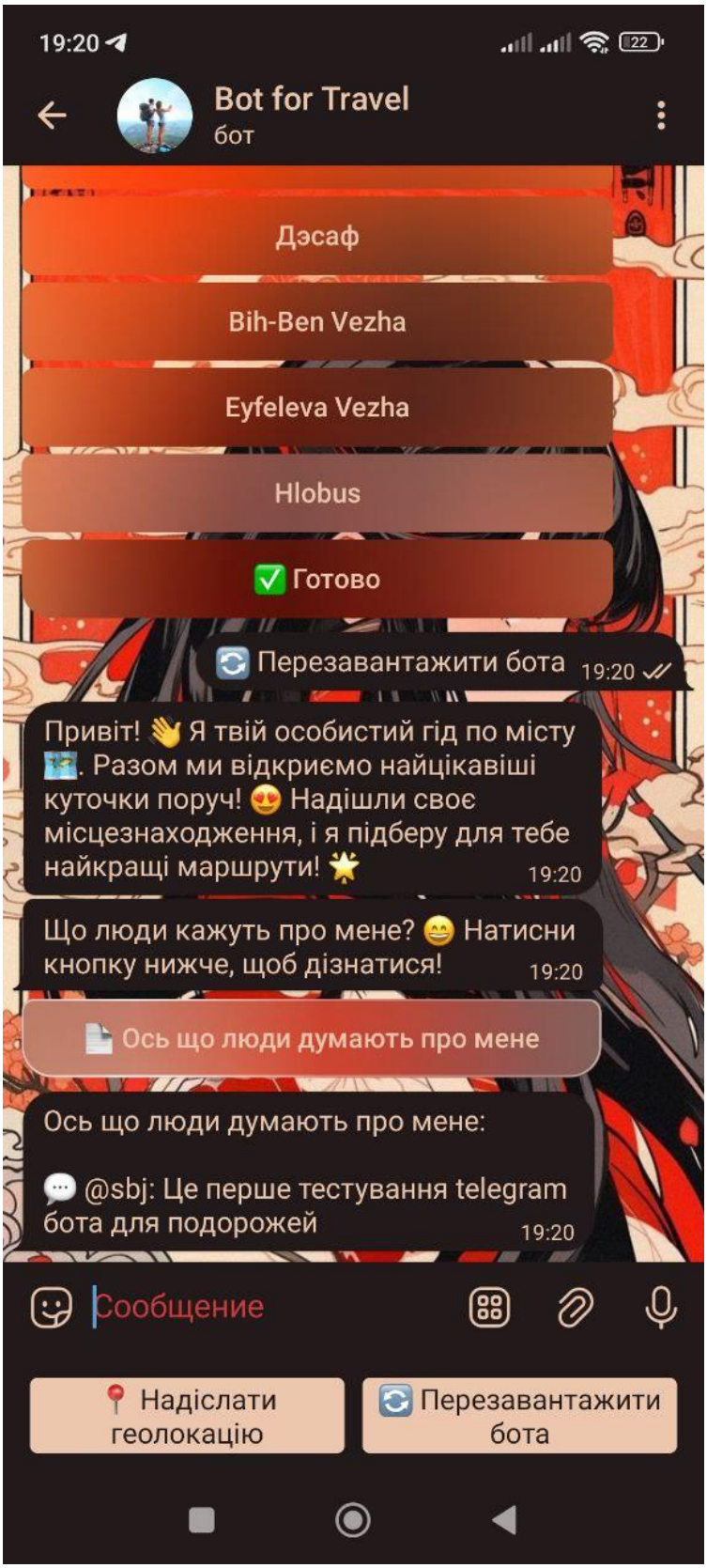


Рисунок 3.17 – Перегляд відгуків

ВИСНОВКИ

У даній кваліфікаційній роботі магістра було розглянуто тему розробки Telegram бота для подорожей, що дозволяє користувачам знаходити туристичні локації поруч, формувати маршрути та залишати відгуки. Процес розробки Telegram бота для подорожей продемонстрував значущість інтеграції різноманітних API та ефективного використання геолокаційних даних для створення зручного і функціонального інструменту.

Використання таких технологій, як Google Maps API і Telegram Bot API, дозволяє забезпечити високий рівень користувацької взаємодії та задоволеності. Важливим аспектом є також інтерактивність інтерфейсу, що дозволяє користувачам легко вибирати туристичні локації та формувати маршрути, що робить процес планування подорожей більш зручним та ефективним. Крім того, можливість залишати відгуки дозволяє збирати цінний зворотний зв’язок, що підвищує функціональність бота в майбутньому.

Розробка подібних рішень допомагає не тільки покращити досвід користувачів, але й забезпечити постійну актуальність і доступність інформації, що є ключовим для успішної роботи таких інструментів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Python. URL : https://www.python.org (дата звернення: 15.09.2024).

2. Telegram API. URL : https://core.telegram.org (дата звернення: 16.09.2024).

3. BotFather Telegram. URL : https://t.me/botfather (дата звернення: 18.09.2024).

4. Google Maps Platform. URL : https://developers.google.com/maps?hl=en (дата звернення: 25.09.2024).

5. Документація про використання Google Directions API. URL : https://developers.google.com/maps/documentation/directions/overview?hl=en (дата звернення: 28.09.2024).

6. Бібліотека requests. URL : https://pypi.org/project/requests/ (дата звернення: 30.09.2024).

7. Бібліотека telebot. URL : https://pypi.org/project/pyTelegramBotAPI/ (дата звернення: 04.10.2024).

8. Документація про використання Google Places API. URL : https://developers.google.com/maps/documentation/places/web-service?hl=en (дата звернення: 11.10.2024).