

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ, КОМП'ЮТЕРНОЇ
ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА ПРИКЛАДНОЇ ІНФОРМАТИКИ**

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

_____ Гамаюн В.П.
(підпис) (ПІБ)

“ _____ ” _____ 2021р.

**ДИПЛОМНИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)**

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ “БАКАЛАВР”

Тема: Telegram бот для самоосвіти

Виконавець: _____ Мурадов Ігор Емільович
(підпис) (ПІБ)

Керівник: _____ Толстікова Олена Володимирівна
(підпис) (ПІБ)

Нормоконтролер: _____ Боровик Володимир Миколайович
(підпис) (ПІБ)

Київ 2021

ВСТУП

Кожна розумна людина прагне до саморозвитку. Але сучасний швидкий темп життя потребує від людини адаптації до суспільного ритму. Отже необхідно отримувати нову інформацію з нових джерел щонайшвидше, не витрачаючи час на пошуки шляхів для розвитку.

У дипломному проекті буде розглянуто питання актуальності месенджерів, необхідності створення чат-ботів, процесу створення Telegram бота для самоосвіти.

Згідно особливостей даного напрямку, необхідно виконати аналіз популярності сучасних месенджерів та запропонувати рішення для найактуальнішого з них. Також одним з завдань розглянути необхідні модулі та функції для створення корисного та простого у використанні чат-бота.

Для забезпечення користувача Telegram ботом для самоосвіти необхідно розглянути наступні питання:

- Аналіз популярності месенджерів – для того аби виокремити для себе кількість потенційних користувачів;
- Актуальність та прогресивність месенджерів – розробка має вестись на передовій платформі для залучення більшої аудиторії;
- Методи і підходи до розробки – чат-бот має виконувати заплановані функції, але не має бути надто складним для користувача;
- Аналіз необхідних засобів реалізації – розуміння актуальності та функціоналу обраних засобів реалізації;

РОЗДІЛ 1

TELEGRAM ТА ЙОГО ПОПУЛЯРНІСТЬ.

АКТУАЛЬНІСТЬ TELEGRAM БОТІВ

1.1. Системи Миттєвого Обміну Повідомленнями. Telegram.

Системи миттєвого обміну повідомленнями (СМОП) – спеціальна телекомунікаційна служба для обміну повідомленнями між пристроями користувачів. Обмін можливий як через комп'ютерні мережі (в тому числі інтернет), так і безпосередньо між пристроями.

Для користування даним видом комунікації необхідна клієнтська програма. Клієнтську програму СМОП називають месенджером.

На відміну від електронної пошти спілкування у месенджерах іде в реальному часі. Тобто повідомлення не зберігається на поштової скриньці, а надходить одразу до адресата. Також месенджери підтримують можливість обміну повідомленнями між більш ніж двома співрозмовниками (конференція, чат).

Система миттєвого обміну повідомленнями працює за деяким протоколом. Протоколи бувають безсерверні або серверні. У безсерверних протоколах (FChat, NASSI, UChat) повідомлення передаються безпосередньо від одного співрозмовника іншому. Але все таки серверні протоколи є найпоширенішими. В них месенджери не працюють самостійно, а підключаються до мережі обміну повідомленнями, яку і іменують сервером.

В свою чергу Telegram є безкоштовним кросплатформним клауд-месенджером. Підтримує обмін не лише текстовими повідомленнями, але ще й ще медіафайлами, голосовими та відеоповідомленнями. Завдяки своєму шифруванню почав набувати свою популярність.

1.2 Telegram. Аналіз актуальності серед месенджерів

Детально розглянемо сучасні месенджери. Останні роки вони знову набирають свою популярність. Імовірна причина росту кількості користувачів – зміни в сфері мобільного Інтернету: більші швидкості, нижчі ціни, та ширше поширення смартфонів.

На графіку (рисунок 1.1) представлено кількість активних користувачів у найвідоміших месенджерах станом на січень 2021 року, за даними statista.com. Як ми бачимо одноосібним лідером є WhatsApp із загальною кількістю близько 2 мільярдів осіб. Слідом іде Facebook Messenger із 1.3 мільярдами активних користувачів за місяць. Telegram в свою чергу посідає 5 місце за кількістю активних користувачів за місяць з кількістю у 500 мільйонів користувачів.

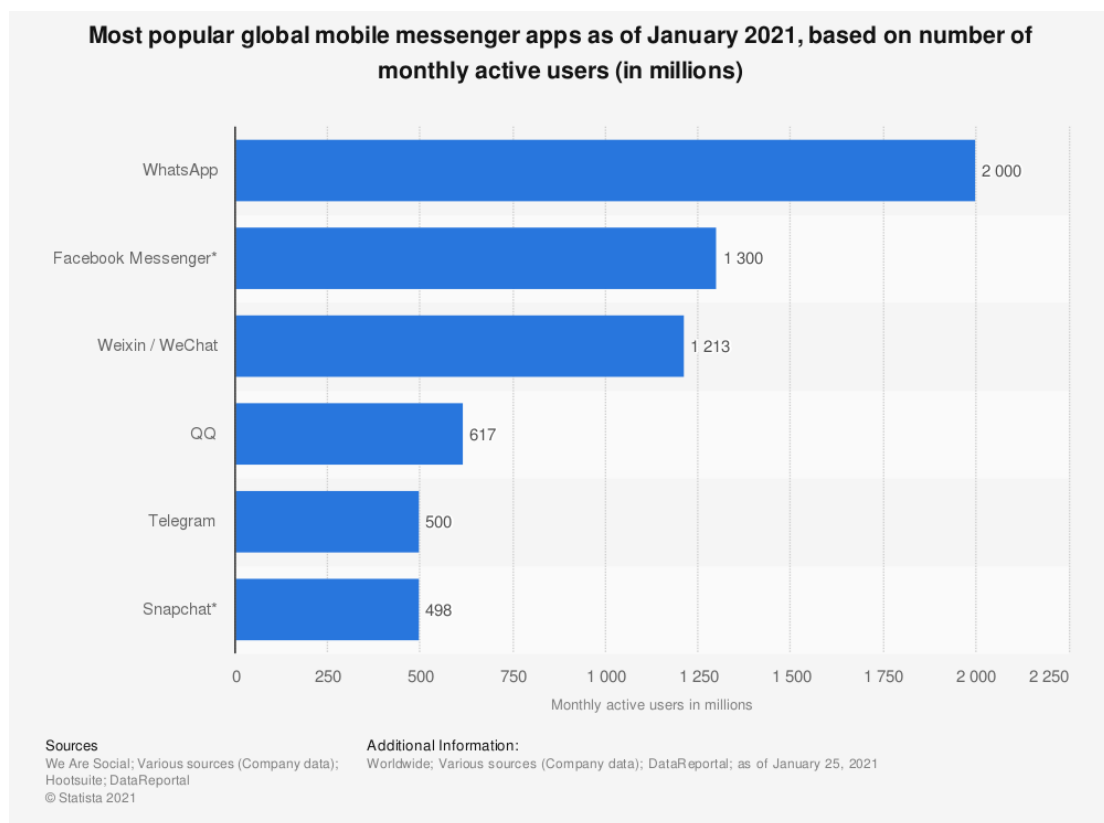


Рис. 1.1. Кількість активних користувачів у найпопулярніших Месенджерах, млн

Очевидно, що всі ці месенджери різні, проте вони мають певну низку спільних рис, що і стали запорукою їх успіху. Практично кожен з цих месенджерів має зручний, сучасний та зрозумілий користувацький інтерфейс. Також усі вони підтримують можливість спілкування у групових чатах, створення та використання ботів. Також деякі з них дають змогу створення каналів, які повторюють частково функціонал групових чатів, але лише особи з певним рівнем доступу мають змогу публікувати там повідомлення.

Проте для того аби аналізувати всі переваги та недоліки наведених раніше систем миттєвого обміну повідомленнями, потрібно також описати певні їх відмінності. Для прикладу порівняємо WhatsApp, як найпопулярніший у світі месенджер, та Telegram, як один з найбільш швидкозростаючий месенджер.

Далі наведено графік росту кількості активних юзерів месенджера WhatsApp за період з квітня 2013 по березень 2020 (рисунок 1.2). За цей час аудиторія збільшилась в 10 разів із 200 мільйонів до 2 мільярдів. Тобто можемо зробити висновок, що приблизно кожна 4 людина на планеті користується WhatsApp щомісяця.

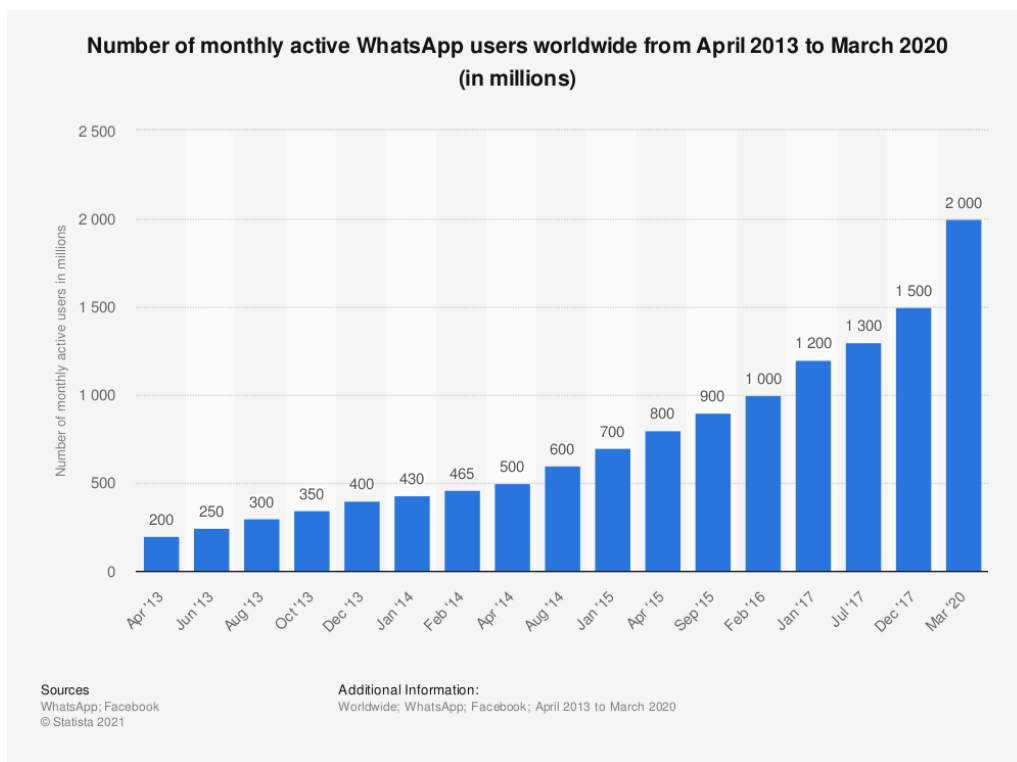


Рис. 1.2. Кількість активних користувачів WhatsApp за місяць, млн

Telegram – месенджер, який набуває популярності по всьому світу. В період із лютого 2016 по квітень 2020 кількість активних користувачів зросла в 4 рази (рисунок 1.3.). За той же період у WhatsApp цей показник зріс вдвічі. Telegram сміливо можна називати одним з найбільш стрімко зростаючих масенджерів. Станом на квітень 2021 він встиг пересягнути відмітку у 500 мільйонів активних користувачів.

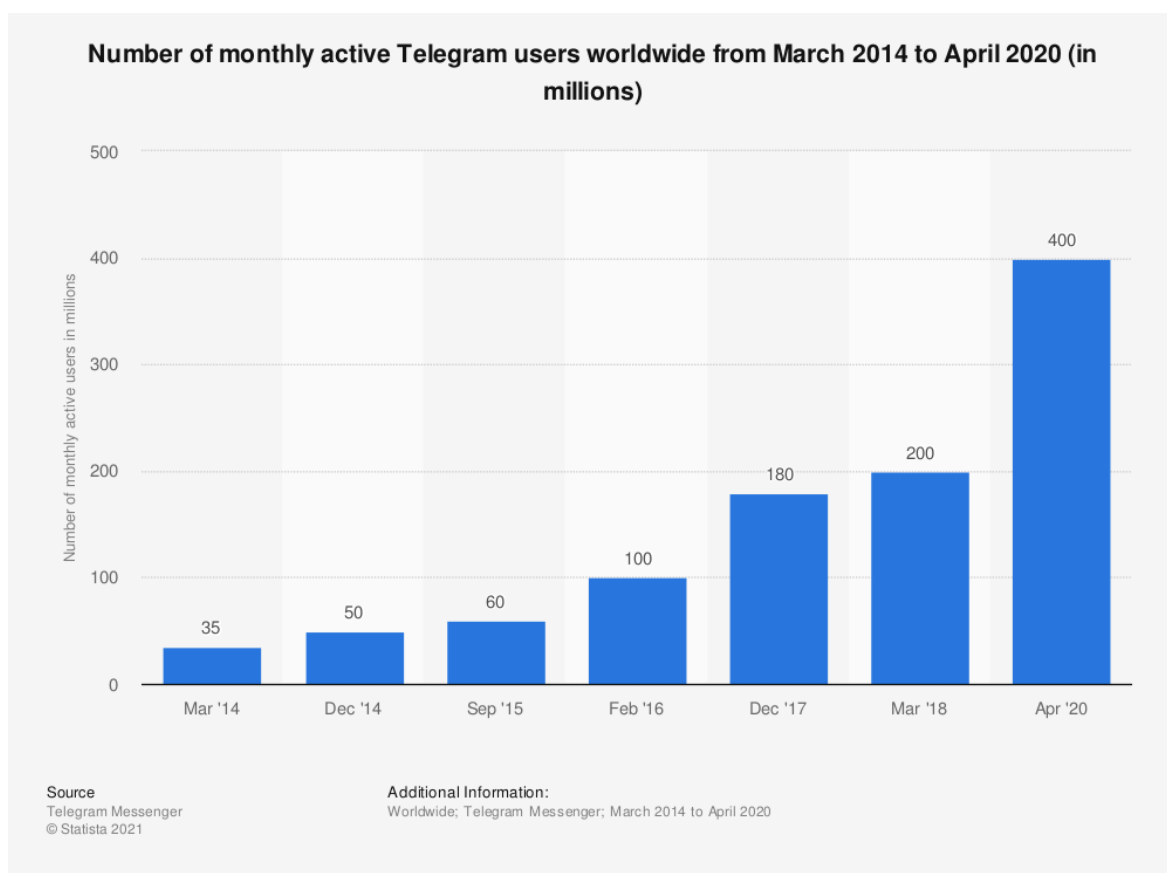


Рис. 1.3. Кількість активних користувачів Telegram за місяць, млн

Усі, наведені СМОП, так чи інакше, підтримують інтеграцію із програмами ботами. Але, наприклад Snapchat та QQ Mobile підтримують так звану сіру інтеграцію, тобто вони не забороняють, але і не створюють особливих умов для співпраці своїх месенджерів з користувацькими ботами. Також подібні варіанти інтеграції не регулюють використання сервісів програмами під виглядом користувачів.

Усі інші підтримують інтеграцію ботів на рівні платформи, що означає всебічну підтримку їх створення за допомогою надання прикладного програмного інтерфейсу (API) та докладної документації, щодо правил використання платформи.

Важливо розуміти, що існує велика кількість різноманітних ботів, які так чи інакше, беруть участь у життях людей прямо зараз. При відвідуванні сайтів, боти-помічники допомагають користувачу в вирішенні проблем чи вибором товару. Також при виникненні більш складних питань бот може направити користувача до спеціаліста з відповідного питання. В деяких магазинах на додачу існують боти, що повідомляють користувача про новинки, розсилаючи повідомлення у месенджери, спрощуючи процес пошуку та замовлення цікавих для користувача товарів та послуг, дозволяючи при цьому навіть не покидати сам месенджер.

Як показав огляд сучасних СМОП, вони є важливою ланкою сучасного спілкування у мережі. Цим зумовлюється популярність створення ботів для різних сфер, аби спростити взаємодію користувача з кінцевим продуктом. Звісно існує велика кількість месенджерів, але чому саме Telegram? Telegram є одним з найпоширеніших СМОП в Україні, також темпи його популяризації зумовлюють ріст кількості нових користувачів. Окремо хочеться зазначити, що взаємодія розробника та СМОП, одна з найбільш зручних, саме в Telegram. Всі ці фактори роблять розробку свого бота саме для цієї системи одною з найперспективніших.



Рис.1.4. Логотип месенджера Telegram

1.3 Сучасні методи створення телеграм ботів

1.3.1. Класифікація ботів для СМОП

Боти класифікуються за декількома параметрами. За способом отримання інформації від користувача вони поділяються на:

- Текстові – які обробляють інформацію від користувача у текстовій формі та обробляють її для виділення команд;
- Голосові – які перетворюють людське мовлення на текст, а після цього його аналізують

Текстові боти є простішими у розробці та швидшими, за рахунок того, що не виникає затримки в процесі обробки інформації. Голосові боти в свою чергу аналізують отримані команди повільніше, проте дана затримка, з кожним етапом розвитку технологій в розробці ботів, зменшується дозволяючи пришвидшувати алгоритми обробки голосової інформації. Що позитивно впливає на досвід користувача, щодо взаємодії з голосовими ботами. Адже голосові помічники є зручнішими у багатьох випадках, оскільки вони можуть бути повноцінними співрозмовниками, синтезуючи відповідь у вигляді аудіо.

За призначенням чат-ботів поділяють на:

- ЧБ для розмов на широкий спектр тем;
- Орієнтованих на певну тему;

Перші призначені для діалогу із користувачем на абстрактні теми та без певної чіткої мети. Другі орієнтовані на вирішення певних буденних проблем та завдань засобами звичайної мови. Набільш поширені серед ботів саме другий тип, що направлені на вирішення певних проблем, певної людини та досягнення поставленої мети. Про такий бот йтиметься далі у дослідженні.

За типом взаємодії ботів можна поділити на:

- Примітивні боти
- ЧБ, які здатні виокремити команди із повідомлення
- ЧБ, які здатні розуміти контекст діалогу

1.3.2. Огляд ЧБ з відкритим кодом

Основними взаємодіючими елементами є користувач, платформа месенджера Telegram, сервер Bot API та сервер бота. При надсиланні повідомлення користувачем воно потрапляє на сервер месенджера, де і зберігається до моменту обробки його ботом. При потраплянні повідомлення на сервер відбувається, на обробку якої бот підписується за допомогою веб-хуків. Веб-хук у розробці - це спосіб доповнення або зміни поведінки веб-сторінки чи веб-додатка, за допомогою зворотних викликів. Повідомлення тоді потрапляє на сервер, на якому розміщено ЧБ. Бот обробляє повідомлення, можливо звертається до сторонніх API для отримання додаткової інформації та формує відповідь користувачу. Сформована відповідь передається на сервер Телеграм і зберігається там поки користувач не прочитає її.

1.3.3. Приклад архітектури серверної частини бота

Розглянемо узагальнений приклад серверної архітектури бота на (рисунок 1.5)

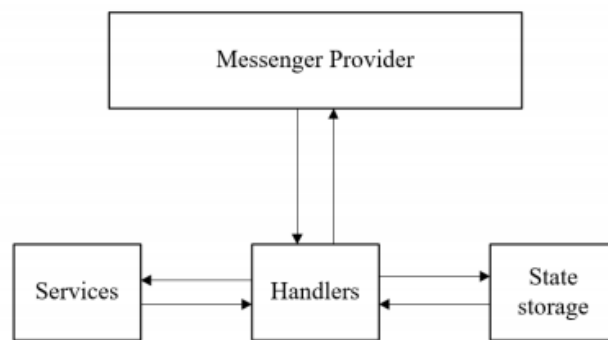


Рис.1.5 Узагальнений приклад серверної архітектури бота

У даному випадку бот складається із чотирьох основних частин:

- Провайдер месенджера (Messenger Provider);
- Обробники команд (Handlers);
- Сервіси (Services);
- Контейнер стану (State storage).

Провайдер месенджера обгортає Telegram Bot API і забезпечує універсальний інтерфейс для прийому, обробки та відправки повідомлень. Провайдер месенджера визначає, що хоче користувач, і спрямовує його до правильного обробника. Мета цієї частини забезпечити шар абстракції над API месенджерів. Тобто, якщо необхідно буде підтримувати більше СМОП або перейти на використання іншого месенджера то єдиний код, який потрібно буде змінити це ця частина.

1.4 Приклади чат ботів в різних сферах життя

1.4.1 Фінанси - @PrivatBankBot

Офіційний бот «ПриватБанку», за допомогою якого можна відправляти і отримувати гроші на банківську карту. Використовується бот в індивідуальних і групових чатах через вбудовані запити. Щоб відправити або отримати гроші через Telegram, досить в будь-якому чаті набрати @PrivatBankBot і суму платежу. Але перед цим необхідно пройти авторизацію, надавши боту свій номер телефону і підтвердивши платіжну карту. Транзакції за допомогою бота застосовні тільки для власників карток «ПриватБанку». Всі перекази підтверджуються через SMS-повідомлення.



Рис. 1.6 Приклад роботи @PrivatBankBot

1.4.2 Перевезення - @railwaybot

Це бот буде вкрай корисний для тих, хто часто користується послугами «Укрзалізниці». Бот шукає квитки на поїзди України і показує графік руху поїзда по номеру. Його особливістю є моніторинг квитків по заданому напрямку з урахуванням обраних параметрів кожні 5 хвилин. @railwaybot також дозволяє перейти до процесу купівлі квитка через сервіс «Приватбанку» (рисунок 1.7).

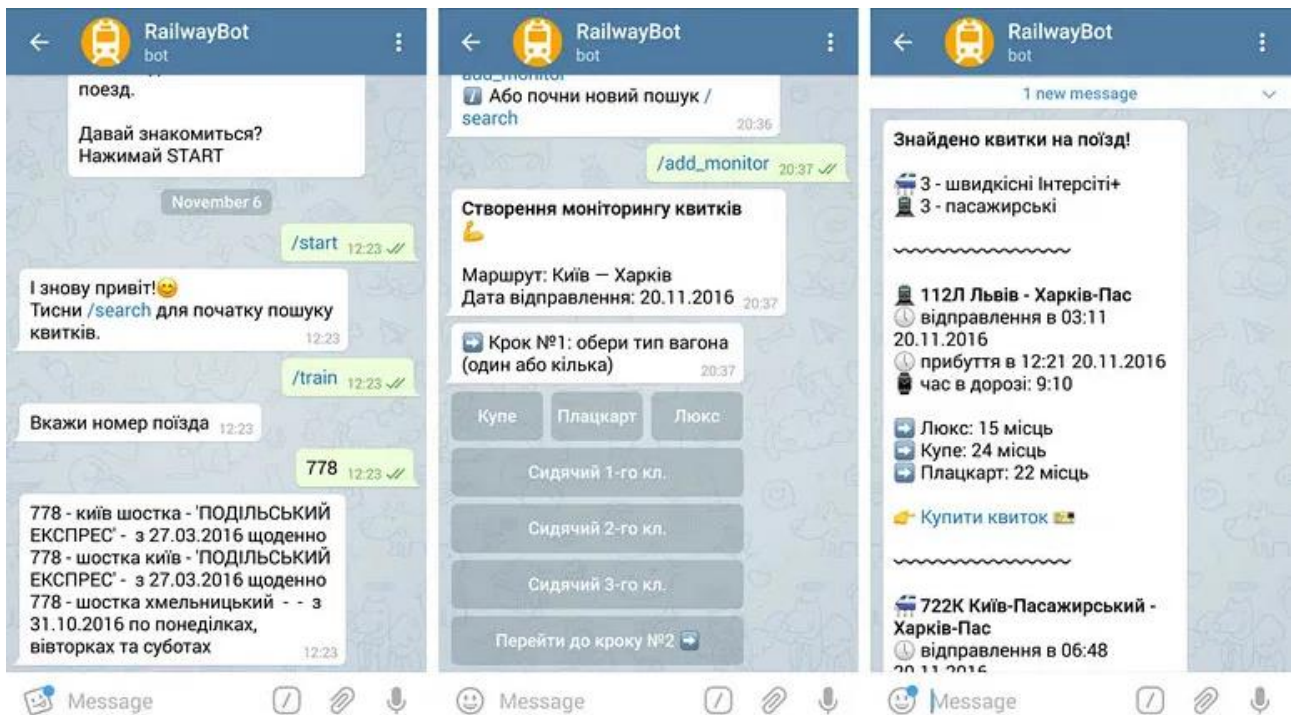


Рис. 1.7 Приклад роботи @railwaybot

1.4.3 Подорожі - @unexploredcity_bot

Робот українського travel-проекту про цікаві місця України. Ви відправляєте поточну геолокацію боту, і він підбере вам найближчі пішохідні маршрути, проведе по всіх найбільш визначних місцях, надасть відповідну інформацію і фотографії. The Unexplored City був запущений у вересні цього року, тому інформації в ньому зовсім мало і поки тільки по Києву. Творці проекту обіцяють додати маршрути по іншим українським містам в найближчому майбутньому (рисунок 1.8).

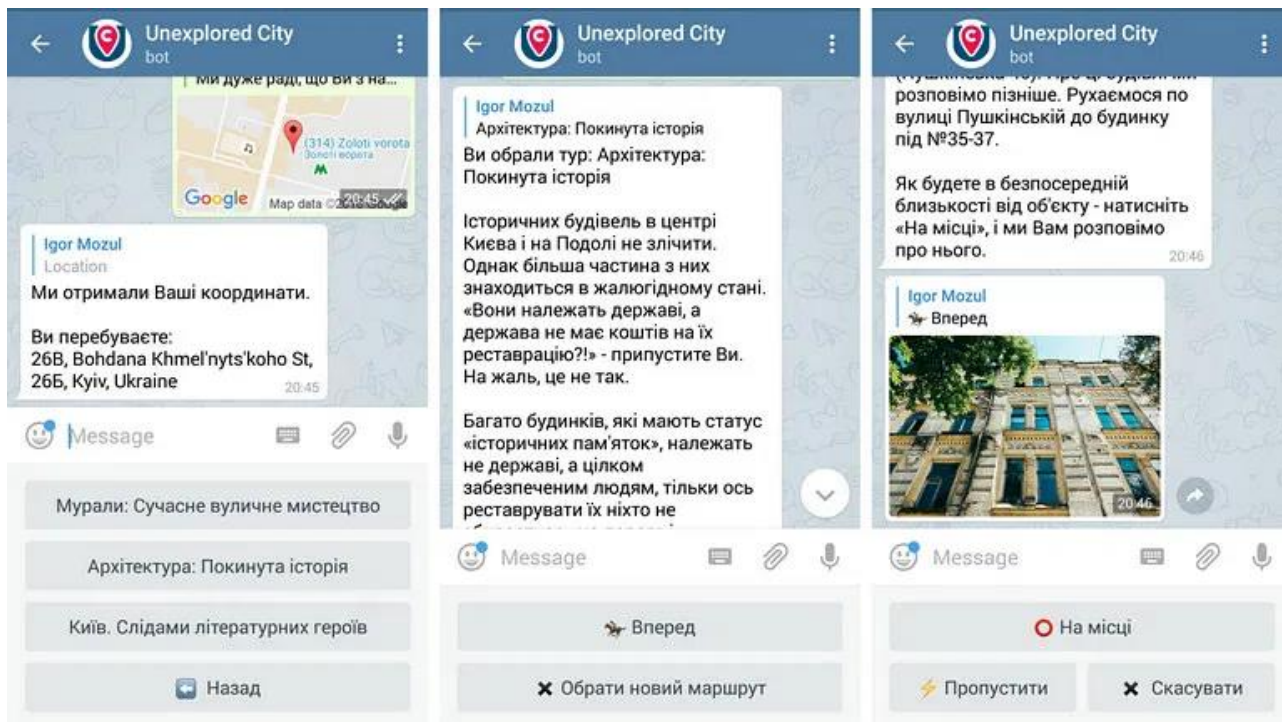


Рис. 1.8 Приклад роботи @unexploredcity_bot

1.4.4 Поштові послуги - @QTrackerBot

Бот, який вміє відслідковувати посилки кур'єрських служб доставок по Україні: «Нової Пошти», «Делівері», «Міст-Експрес», а також «Укрпошти».

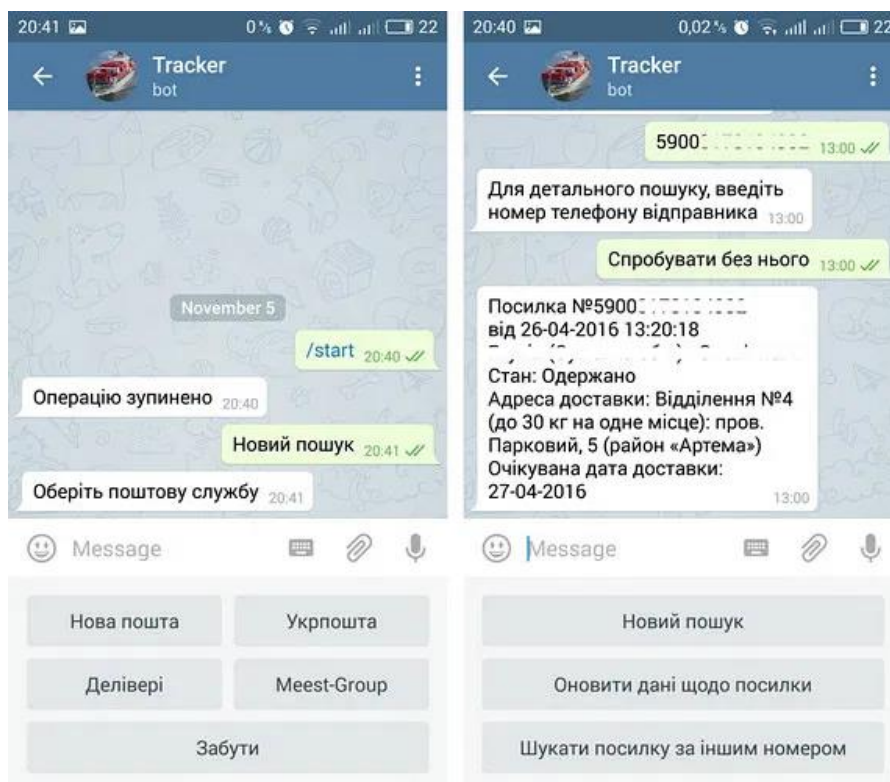


Рис. 1.9 Приклад роботи @QTrackerBot

1.5 Висновки до розділу

У цьому розділі було досліджено месенджери, їх популярність та кількість активних користувачів. Було визначено месенджер для подальшої розробки чат-бота. Також було проаналізовано всі варіанти ботів, та обрано той, що задовольнятиме потребам нашого користувача.

РОЗДІЛ 2

ОРГАНІЗАЦІЯ ПРОЦЕСУ РОБОТИ НАД TELEGRAM БОТОМ

2.1 Вибір мови програмування для розробки ЧБ

2.1.1 Статистика та аналіз мов програмування для ЧБ

У сучасному світі де все пов'язано з ІТ-технологіями, майже будь-яка компанія використовує додатки, що роблять процес роботи продуктивним та швидшим. Світ не стоїть на місці, а розвивається і тому тематика цієї роботи зумовлена тим, аби допомогти користувачу отримати нові знання та шлях для подальшого розвитку, навіть не закриваючи месенджер.

В свою чергу, переді мною стояла задача вибрати ту мову програмування, яка буде актуальною, буде мати всі необхідні можливості для розробки та допоможе реалізувати весь задуманий функціонал.

2.1.2 Основні поняття та види мов програмування

Мова програмування – це штучна мова, створена для передачі команд машинам, зокрема комп'ютерам.

Також зустрічається більш суворе визначення: мова програмування — це система позначень для опису алгоритмів і структур даних, певна штучна формальна система, засобами якої можна виражати алгоритми. Мову програмування визначає набір лексичних, синтаксичних і семантичних правил, що задають зовнішній вигляд програми та дії, які виконує виконавець (комп'ютер) під її управлінням.

Синтаксис комп'ютерної мови — це сукупність правил, що визначають комбінації символів, які вважаються правильно структурованим документом або фрагментом цієї мови. Це стосується як мов програмування, результатом якого є початковий код, так і мов розмітки, результатом якого є дані.

Семантика мови - це смислове значення слів. У програмуванні - значення операторів, основних конструкцій мови і т.п.

Мови програмування поділяються на два види – низькорівневі та високорівневі:

- Високорівнева мова програмування — мова програмування, розроблена для швидкого і зручного використання програмістом. Основна риса високорівневих мов — це абстракція, тобто введення смислових конструкцій, які коротко описують такі структури даних і операції над ними, опис яких на машинному коді або низькорівневій мові програмування був би дуже великим, складним для розуміння програмістів і зайняв би багато часу при написанні. Написані програми на мові високого рівня згодом перетворюються в машинний код для виконання безпосередньо операційною системою. В результаті безпосередній код, що виконується на комп'ютері і створений в процесі трансляції переважно містить певні зайві інструкції, тому виконання таких програм дещо повільніше.
- Низькорівнева мова програмування — мова програмування наближена до машинного коду. Перші комп'ютери доводилось програмувати двійковими машинними кодами. Проте програмувати таким чином — доволі трудомістке і важке завдання з тої причини, що людям важко запам'ятовувати цифрові позначення команд. Для спрощення цього завдання почали з'являтися мови програмування низького рівня, які дозволяли задавати машинні команди в зрозумілішому для людини вигляді.

Аналізуючи наші знання про мови програмування, а також враховуючи наші потреби ми будемо шукати мову програмування високого рівня для реалізації Telegram бота.

2.1.3 Вибір мови програмування для написання ЧБ

На сьогоднішній день існує безліч різних мов програмування і у кожній з них своя сфера застосування. Для того щоб обрати найкращу мову для написання чат-бота, необхідно вибрати декілька популярних мов, аби проводити між ними аналіз. Щоб це зробити необхідно звернутися до статистики по найпопулярнішим мовам програмування від сайту dou.ua. (рисунок 2.1.)

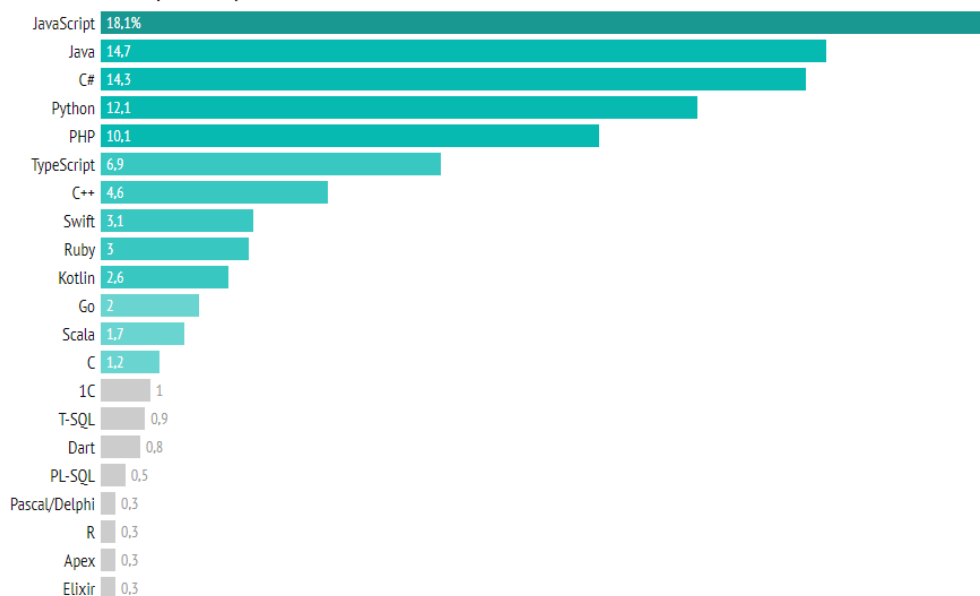


Рис. 2.1. Статистика популярності мов програмування у комерційній розробці, %

По даним, можна зробити висновок, що JavaScript значно випереджає Java і зараз є найпопулярнішою мовою програмування. В п'ятірку найпопулярніших мов також входять: C#, Python, PHP.

На рисунку 2.2. можна побачити як змінювались дані з 2012 по 2021 рік

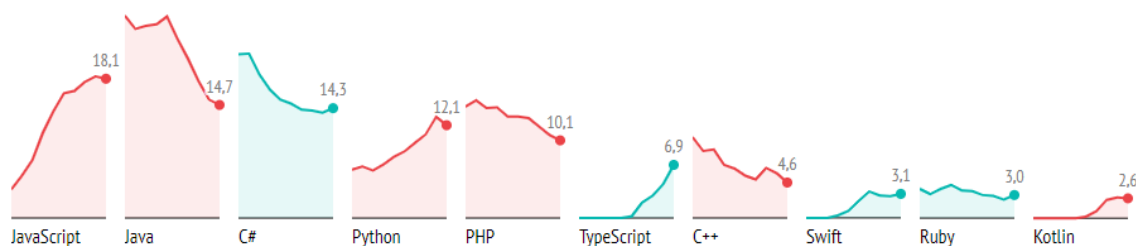


Рис. 2.2. Графік зміни популярності МП в комерційній розробці, %

По рисунку 2.2. можна зробити висновок, що популярність таких мов, як Java та C# сильно падає, в той час коли такі мови, як JavaScript, TypeScript та Python продовжує зростати.

Далі розглянемо рейтинг популярності по особистим вподобанням, але для комерційної розробки (рисунок 2.3.).

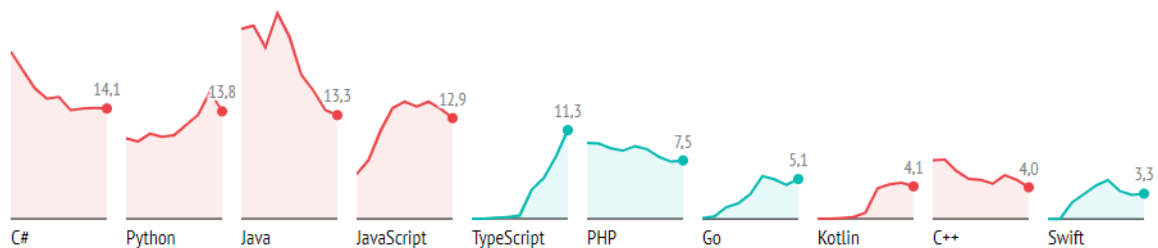


Рис. 2.3. Рейтинг популярності МП за особистими вподобаннями, %

По рисунку 2.3. можна зробити висновок, що по особистим вподобанням люди обирають C#, далі Python, а далі JavaScript.

З даних, що ми отримали можна зробити висновок, що найпопулярнішими мовами на даний момент є JavaScript, Python и C#. Тому проведемо порівняльний аналіз, аби визначити, яка з цих мов найбільше підходить для наших потреб.

2.1.4. Порівняльний аналіз мов програмування

Python — інтерпретована об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією. Структури даних високого рівня разом із динамічною семантикою та динамічним зв'язуванням роблять її привабливою для швидкої розробки програм, а також як засіб поєднання наявних компонентів.

Python підтримує модулі та пакети модулів, що сприяє модульності та повторному використанню коду. Інтерпретатор Python та стандартні бібліотеки доступні як у скомпільованій, так і у вихідній формі на всіх основних платформах. В мові програмування Python підтримується кілька парадигм програмування, зокрема: об'єктно-орієнтована, процедурна, функціональна та аспектно-орієнтована.

Переваги Python:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- переносність програм (що властиве більшості інтерпретованих мов);
- стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу);
- можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач);
- стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написано мовою Python;
- зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
- відкритий код (можливість редагувати його іншими користувачами).

Недоліки Python:

- не найкраща реалізація багатопоточності;
- на Python створено не так багато якісних програмних проектів, у порівнянні з більшістю універсальних мов програмування;
- відсутність підтримки засобів для комерційної розробки;
- початкова обмеженість засобів для роботи з базами даних;
- менша продуктивність у порівнянні з іншими мовами програмування, що створює цій мові репутацію повільної;



Рис.2.4. Логотип Python

JavaScript (JS) — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

- написання сценаріїв вебсторінок для надання їм інтерактивності;
- створення односторінкових та прогресивних вебзастосунків (React, AngularJS, Vue.js);
- програмування на боці сервера (Node.js(Express.js));
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладних програмах (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий «у спадок» від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme

Переваги JavaScript :

- Жоден сучасний браузер не обходиться без підтримки JavaScript;
- З використанням написаних на JavaScript плагінів і скриптів впорається навіть не фахівець;
- Корисні функціональні настройки;
- Постійно удосконалюється мова - зараз розробляється бета-варіація проекту, JavaScript2;
- Взаємодія з додатком може здійснюється навіть через текстові редактори - Microsoft Office і Open Office;
- Перспектива використання мови в процесі навчання програмуванню і інформації.

Недоліки JavaScript:

- Знижений рівень безпеки через повсюдного і вільного доступу до вихідного коду популярних скриптів;
- Безліч дрібних дратівливих помилок на кожному етапі роботи. Велика частина з них легко виправляється, але їх наявність дозволяє вважати цю мову менш професійним, порівняно з іншими;
- Повсюдне поширення. Своєрідним недоліком можна вважати той факт, що частина активно використовуваних програм (особливо додатків) перестануть існувати при відсутності мови, оскільки цілком базуються на ньому.



Рис. 2.5. Логотип JavaScript

C# (вимовляється Сі-шарп) — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Розроблена Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде під егідою Microsoft Research (належить Microsoft).

Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Переїнявши багато від своїх попередників — мов C++, Object Pascal, Модула і Smalltalk — C#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад, мова C#, на відміну від C++, не передбачає множинне успадкування класів.

Переваги C#:

- для малих підприємств і деяких окремих розробників безкоштовні інструменти включають Visual Studio, Azure Cloud, Windows Server, Parallels Desktop для Mac Pro і багато інших;
- велика кількість синтаксичних конструкцій, розроблених для кращого розуміння написання коду;
- дуже простий у вивченні;
- після покупки Xamarin на C # ви можете писати програми і додатки для операційних систем, таких як iOS, Android, MacOS і Linux;
- є ціле співтовариство з досвідчених програмістів.

Недоліки C#:

- пріоритетна орієнтованість на Windows платформу;
- мова безкоштовна тільки для невеликих фірм, індивідуальних програмістів, стартапів і учнів. Для великої компанії покупка ліцензійної версії цієї мови обійдеться в кругленьку суму;
- в мові залишилася можливість використання оператора безумовного переходу



Рис. 2.6. Логотип C#

Зараз хотілося б наглядно продемонструвати синтаксис мов за написанням рівнозначним за результатом коду на рисунку 2.7, рисунку 2.8, рисунку 2.9.

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!"); // Display the string.  
    }  
}
```

Рис. 2.7. Приклад коду на сімействі Java

```
#include<stdio.h>  
int main(int argc, char** argv)  
{  
    printf("Hello World");  
}
```

Рис. 2.8. Приклад коду на сімействі C

```
print "Hi there"
```

Рис. 2.9. Приклад коду на Python

По рисункам видно, що в той момент, коли Java та C потребують 4-5 рядків коду, Python використовує лише 1.

Тому, роблячи висновки, що базуються на представлених даних, моєму особистому аналізу та пошуку в інтернеті, я можу заявити, що для чат-ботів раціональніше за все використовувати Python, адже він найбільш простий в написанні, а також має достатню кількість документацій та бібліотек для створення максимально якісного сервісу.

2.2 Вибір IDE для розробки ЧБ

Інтегроване середовище розробки - комплексне програмне рішення для розробки програмного забезпечення. Зазвичай, складається з редактора початкового коду, інструментів для автоматизації складання та відлагодження програм. Більшість сучасних середовищ розробки мають можливість автодоповнення коду.

Як ми визначили раніше, мовою програмування, яку ми використовуватимемо для нашого ЧБ буде Python. Вибиремо найпопулярніші IDE з підтримкою Python та порівняємо їх.

2.2.1 Eclipse + PyDev

Якщо ви близькі з open-source спільнотою, то ви напевно чули про Eclipse (рисунок 2.10). Будучи доступним для Linux, Windows і OS X, Eclipse де-факто є open-source IDE для розробки на Java. Існує безліч розширень і аддонів, які роблять Eclipse корисним для різного роду завдань.

Одним з таких розширень є PyDev, що надає інтерактивну консоль Python і можливості для налагодження і автодоповнення коду. Встановити його просто: запусить Eclipse, виберіть Help → Eclipse Marketplace, потім знайдіть PyDev. Натисніть «Install» і при необхідності перезапустити Eclipse.

Переваги:

- якщо у вас вже був встановлений Eclipse, то установка PyDev пройде швидко і гладко. У досвідченого користувача Eclipse не виникне проблем з вивченням цього розширення.

Недоліки:

- якщо ви тільки починаєте вивчати Python або розробку в цілому, Eclipse може стати непосильним тягарем. Пам'ятайте, ми говорили, що IDE великі і вимагають більше досвіду для повноцінного використання? Все це можна сказати про Eclipse.

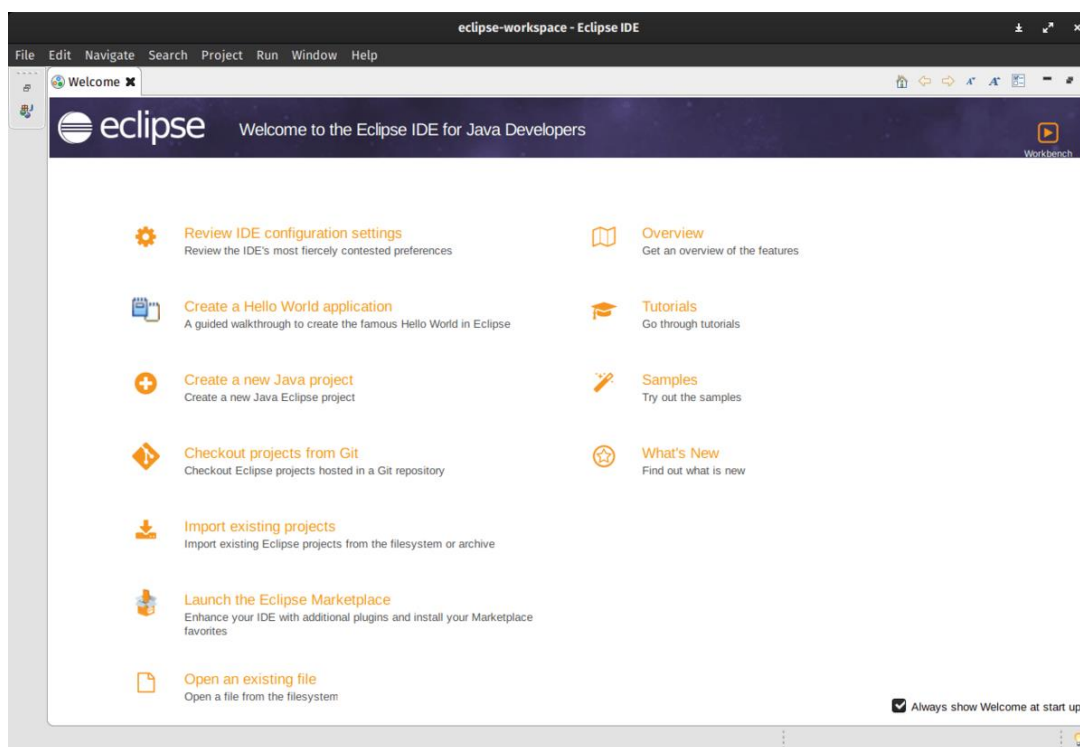


Рис.2.10. Інтерфейс Eclipse

2.2.2 Visual Studio + Python Tools for Visual Studio

Visual Studio - повнофункціональна IDE від Microsoft, яка багато в чому порівнянна з Eclipse. Доступна на Windows і Mac OS, Visual Studio представлена як в безкоштовному (Community), так і в платному (Professional і Enterprise) варіантах. Visual Studio дозволяє розробляти програми для різних платформ і надає свій власний набір розширень (рисунок 2.11).

Python Tools for Visual Studio (PTVS) дозволяє писати на Python в Visual Studio і включає в себе Intellisense для Python, налагодження та інші інструменти.

Переваги:

- як і у випадку з Eclipse, якщо у вас вже встановлена Visual Studio для інших завдань, установка PTVS пройде без проблем.

Недоліки:

- як і у випадку з Eclipse, Visual Studio буде забагато, якщо вам потрібен тільки Python. Крім того, якщо ви користуєтеся Linux, то Visual Studio для цієї платформи немає.

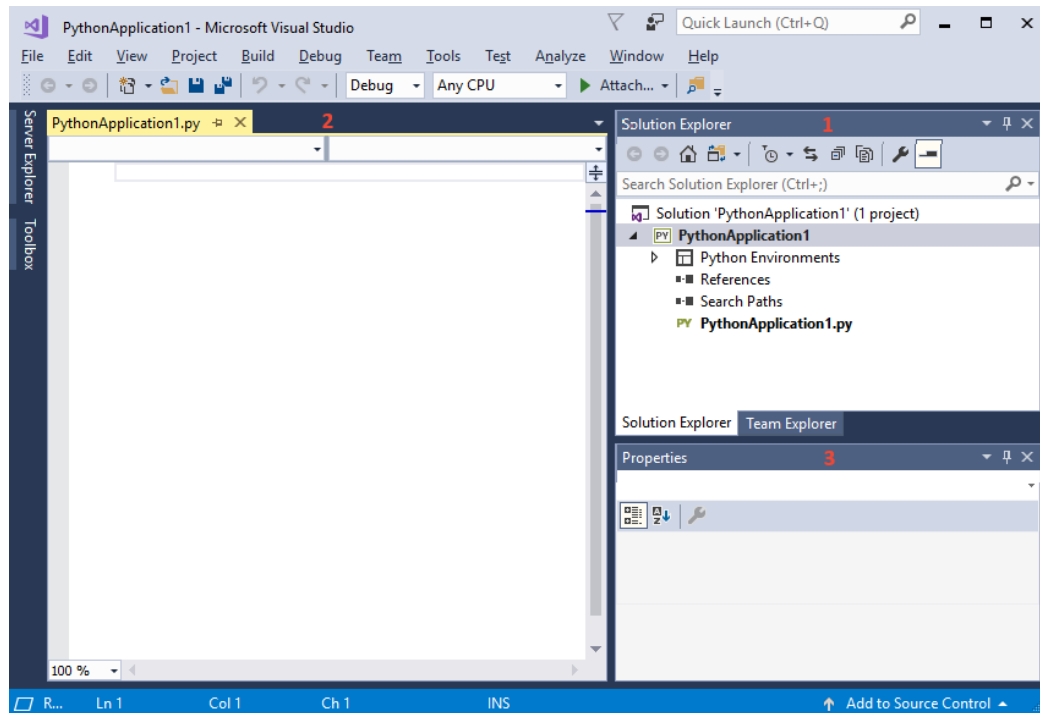


Рис. 2.11. Інтерфейс Visual Studio

2.2.3 PyCharm

Однією з кращих повнофункціональних IDE, призначених саме для Python, є PyCharm. Існує як безкоштовний open-source (Community), так і платний (Professional) варіанти IDE. PyCharm доступний на Windows, Mac OS X і Linux (рисунок 2.12).

PyCharm «з коробки» підтримує розробку на Python безпосередньо - відкрийте новий файл і починайте писати код. Ви можете запускати і налагоджувати код прямо з PyCharm. Крім того, в IDE є підтримка проектів і системи управління версіями.

Переваги:

- це середовище розробки для Python з підтримкою всього і вся і хорошим ком'юніті. У ній «з коробки» можна редагувати, запускати і налагоджувати Python-код.

Недоліки:

- PyCharm може повільно завантажуватися, а настройки за замовчуванням, можливо, доведеться підкоригувати для існуючих проектів.

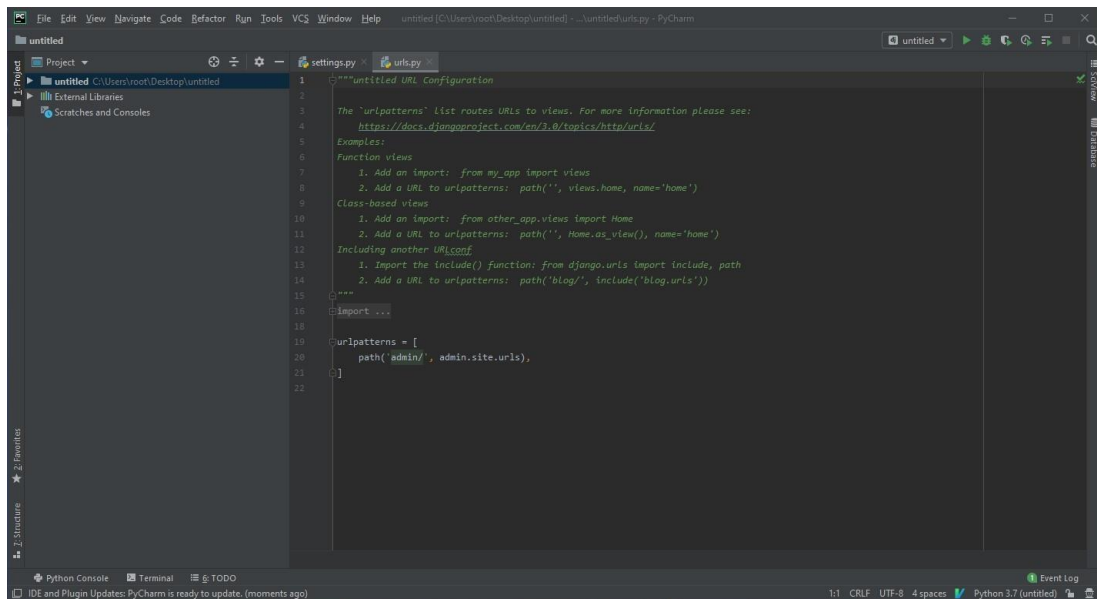


Рис. 2.12. Інтерфейс PyCharm

2.2.3 Thonny

Thonny називають IDE для новачків. Написаний і підтримуваний Інститутом інформатики Тартуського університету в Естонії, Thonny доступний на всіх основних платформах.

За замовчуванням Thonny встановлюється з версією Python, що йде в комплекті, тому вам не знадобиться встановлювати ще що-небудь. Просунутим користувачам, можливо, доведеться змінити цей параметр, щоб IDE знайшла і використовувала вже встановлені бібліотеки.

Переваги:

- ви початківець Python-програміст і вам потрібна IDE, з якої можна відразу йти в бій.

Недоліки:

- просунутим користувачам буде недостатньо функціональності, а вбудований інтерпретатор вони замінять. Крім того, з огляду на новизну IDE, можуть виникнути проблеми, вирішення яких на даний момент немає

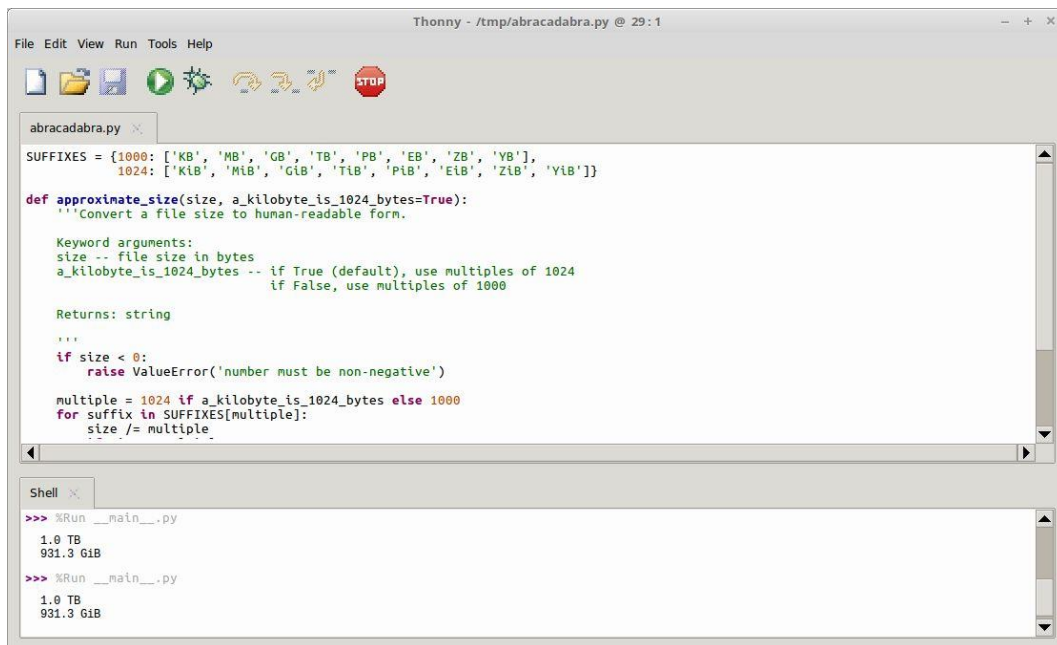


Рис.2.12 Інтерфейс Thonny

2.3 Python, як засіб розробки чат-боту

2.3.1 Необхідні модулі для розробки сервісу

В розділі 2.1 ми з'ясували, що мова програмування Python добре підходить для написання чат-боту для месенджера Telegram. Розглянемо певні бібліотеки, що можуть знадобитись нам в процесі написання ЧБ.

Модуль telebot необхідний для створення і чату в бота.

Модуль random надає функції для генерації випадкових чисел, букв, випадкового вибору елементів послідовності.

Модуль datetime надає класи для обробки часу і дати різними способами. Підтримується і стандартний спосіб представлення часу, проте більший акцент зроблено на простоту маніпулювання датою, часом і їх частинами.

Time - модуль для роботи з часом в Python.

Модуль `messagebox` надає базовий клас шаблону, а також безліч зручних методів для часто використовуваних конфігурацій. Поля повідомлень є модальними і повертатимуть підмножина (`True`, `False`, `OK`, `None`, `Yes`, `No`) на основі вибору користувача.

2.3.2 Функції в Python

Функція в Python - це об'єкт, який приймає аргументи і повертає значення. Як правило, функція визначається за допомогою оператора `def`. Приклад на рисунку 2.13.

```
def add(x, y):  
    return x + y
```

Рис. 2.13 Функція

Функція може приймати будь-яку кількість аргументів чи НЕ приймати їх взагалі. Часто зустрічаються функції з будь-якою кількістю аргументів, функції з позиційними і іменованими аргументами, обов'язкові і необов'язкові (рисунок 2.14).

```
>>> def func(a, b, c=2): # c - необязательный аргумент  
...     return a + b + c  
...
```

Рис.2.14. Аргументи функції

2.3.3 Умовні оператори та цикли Python

Умовний оператор `if-elif-else` (іноді званий оператором розгалуження) є основним інструментом вибору в Python. Простіше кажучи, вона вибирає, яку дію зробити, в залежності від значення змінних в момент перевірки умови.

Синтаксис: спочатку записується частина `if` з умовним виразом, потім одна або кілька необов'язкових частин `elif` і, нарешті, необов'язкова частина `else`. Загальна форма умовного листи виглядає наступним чином (рисунок 2.15):

```
if test1:
    state1
elif test2:
    state2
else:
    state3
```

Рис.2.15 Синтаксис умовного оператора If

2.4 Висновки до розділу

В цьому розділі було описано організаційні моменти підготовки до написання чат-бота, а саме було визначено оптимальну мову програмування та середовище розробки. Також були показані основні методи та функції, що будуть необхідні для написання якісного кінцевого продукту.

РОЗДІЛ 3

РОБОТА НАД СТВОРЕННЯМ TELEGRAM БОТА ДЛЯ САМООСВІТИ

3.1. Процес створення бота та огляд Bot API

В минулому розділі ми визначились, що для розробки ми використовуватимемо мову програмування Python та IDE PyCharm. А також виділили основні функції та модулі, що ми використаємо для розробки Telegram бота.

Далі буде продемонстровано процес створення бота та огляд інших додатків, що необхідні для розробки ЧБ.

Для створення ЧБ в месенджері Telegram існує бот – BotFather, його інтерфейс продемонстрований на рисунку 3.1. Він необхідний нам здебільшого для того, аби отримати токен авторизації бота, що дозволить ідентифікувати його серед інших, а також запрограмувати його.

Після старту роботи з BotFather по команді «/start», потрібно створити нового бота завдяки команді «/newbot». BotFather попросить вас дати боту назву та ім'я бота, перед створенням токена авторизації.

Назва та ім'я бота, це на перший погляд одне і те саме, але це не так. Назва бота – це те, що буде відображено у контактній інформації користувачів та в багатьох інших місцях, наприклад в профілі. В свою чергу ім'я буде використовуватись у згадуваннях та посиланнях. Воно має певні обмеження:

- нечутливе до регістру;
- може включати лише латинські символи, цифри та підкреслення;
- обов'язково має закінчуватися на «bot»;

Але не треба недооцінювати назву, адже вона завжди видима користувачу і дає змогу зрозуміти чий це бот, або що він робить.

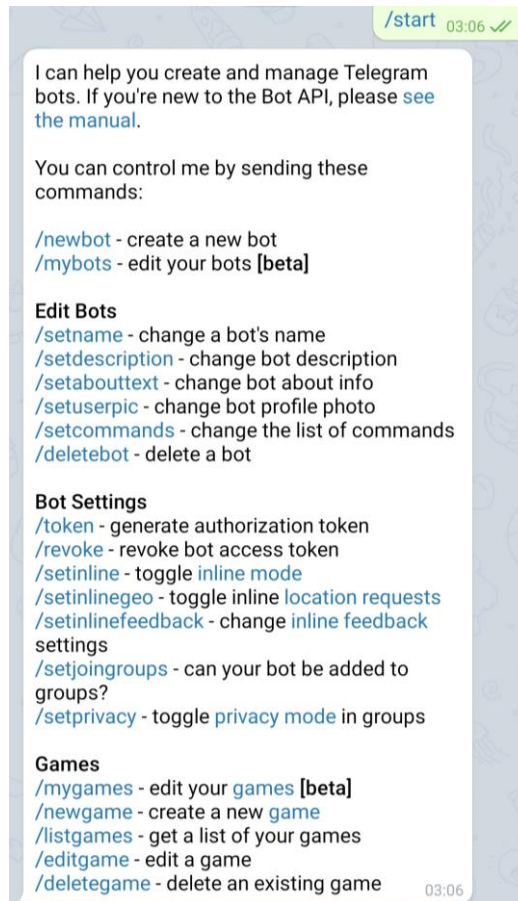


Рис. 3.1 Інтерфейс BotFather

BotFather – це ще й інструмент для керування уже створеними ботами. Список команд доступних для редагування ботів складається із:

- /mybots – видає список ботів та дозволяє їх редагування за допомогою наданих елементів керування;
- /setname – дає змогу змінити ім'я бота;
- /setdescription – дає можливість замінити опис бота – короткий текст до 512 символів, що описує його призначення. Користувач бачить цей текст перед початком розмови з ботом під заголовком;
- /setabouttext – дає змогу замінити інформацію про бота – короткий текст до 120 символів. Користувачі бачать цей текст на профілі бота. Коли вони діляться цим ботом з кимось, цей текст надсилається разом із посиланням;
- /setuserpic – дає змогу замінити аватар профілю бота. Зображення бота відображається під час пошуку, в діалозі та в списку контактів користувача;

- /setcommands – дає змогу замінити список команд, що підтримуються ботом.;
- /deletebot – дає змогу видалити бота. При цьому його ім'я стає доступним для використання іншими ботами;
- /setinline – дає змогу ввімкнути режим використання у будь-якому чаті для бота;
- /setinlinegeo – дає змогу використовувати ботом дані про місцезнаходження користувача, для забезпечення більш достовірних результатів на основі місцезнаходження;
- /setjoininggroups – дає змогу змінити можливість додавання бота до груп;
- /setprivacy – дає визначити які повідомлення бот отримуватиме при роботі в групах.

Після отримання токена можна приступати до подальшої розробки ЧБ, та слід пам'ятати які саме є відмінності ботів від користувачів в технічному плані:

- У ботів немає статусу що відображає коли бот був востаннє в мережі та чи онлайн він зараз. Натомість у всіх ботів є підпис «bot» під іменем;
- Боти мають обмежену кількість пам'яті у хмарному сховищі, саме тому, щойно повідомлення буде прочитане користувачем, воно буде видалятися з сервера. Натомість цю інформацію можуть зберігати програми на власних серверах.
- Боти зазвичай мають змоги почати діалог першими, отже користувач має написати особисте повідомлення боту першим або додати його в групу.
- Імена ботів завжди закінчуються на «bot»
- Після додання бота до групи, вони за замовчуванням не отримують усі повідомлення.

Після отримання токена ми можемо переходити до вибору необхідного прикладного програмного забезпечення (API). Прикладний програмний інтерфейс (англ. Application Programming Interface, API) — набір визначень

підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено — це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Telegram дає змогу використовувати два види API для розробників. Перше - це API Telegram, що існує для створення інших клієнтів платформи телеграм. Воно для нашої роботи не підходить. Друге - Bot API, дає змогу створювати без проблем нові програми, які використовують телеграм, як інтерфейс для комунікації з користувачем.

Саме Bot API дає змогу без зайвих складнощів створювати чат-ботів. Цей API дає змогу підключати своїх ботів до системи Телеграм. З точки зору системи боти – це спеціальні облікові записи, які не потребують встановлення додаткового номеру телефону. Ці записи є інтерфейсом для нашого коду.

Тепер ми можемо перейти до написання телеграм бота. Створивши бота завдяки BotFather, відеагувавши опис, аватар, та назву отримаємо подібний профіль (рисунок 3.2).

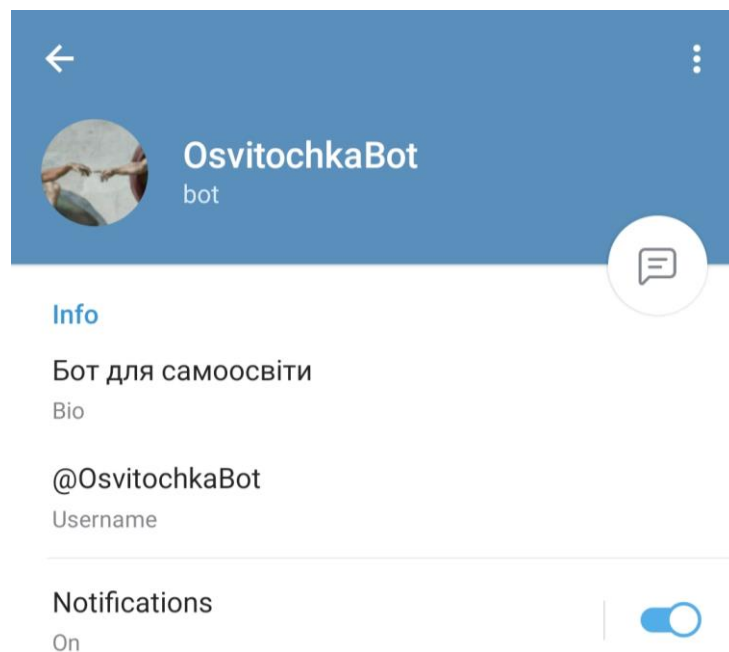


Рис. 3.2 Профіль нашого бота

Також на рисунку 3.3 ми бачимо користувацький інтерфейс бота, що готовий до взаємодії.

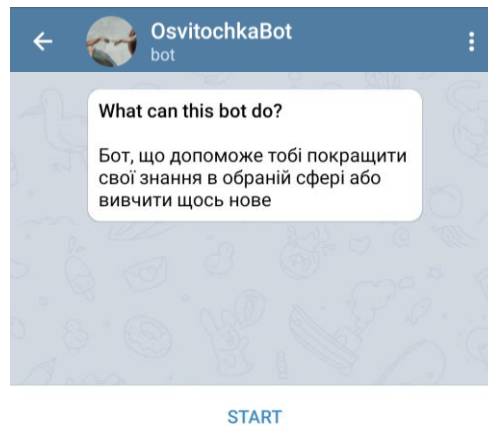


Рис. 3.3 Інтерфейс нашого чат бота

Далі нам необхідно, аби при взаємодії з користувача з чат-ботом, бот ідентифікував користувача, а також запропонував йому сфери знань, серед яких користувач зможе знайти для себе щось цікаве.(рисунок 3.4)

Аби отримати даний результат ми зробили метод, що показано нами на рисунку 3.5. Даний метод при натисканні на кнопку «start» створює кнопки для подальшої взаємодії з користувачем.

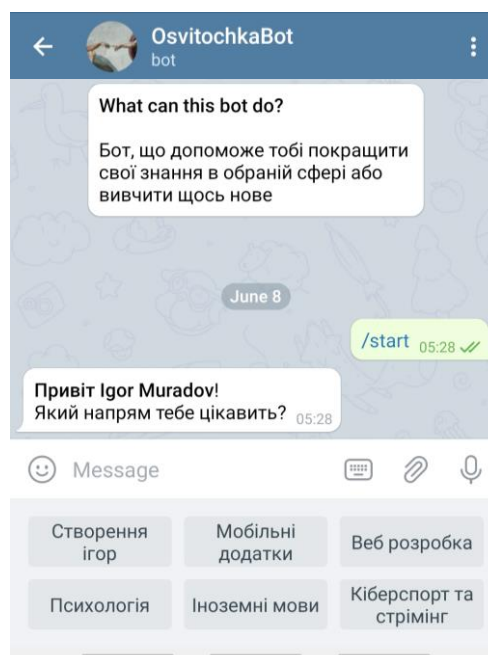


Рис. 3.4 Початок взаємодії користувача з ботом

```

@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True, row_width=3)
    btn1 = types.KeyboardButton('Створення ігор')
    btn2 = types.KeyboardButton('Мобільні додатки')
    btn3 = types.KeyboardButton('Веб розробка')
    btn4 = types.KeyboardButton('Психологія')
    btn5 = types.KeyboardButton('Іноземні мови')
    btn6 = types.KeyboardButton('Кіберспорт та стрімінг')
    markup.add(btn1, btn2, btn3, btn4, btn5, btn6)
    send_mess = f"<b>Привіт {message.from_user.first_name} {message.from_user.last_name}</b>!\nЯкий напрям тебе цікавить?"
    bot.send_message(message.chat.id, send_mess, parse_mode='html', reply_markup=markup)

```

Рис. 3.5 Фрагмент коду, що відповідає за початок взаємодії з користувачем

При виборі користувачем певної дисципліни для отримання знань в цій сфері, попередні кнопки замінюються іншими, які допомагають більш точно обрати необхідний для користувача курс. Також створюється кнопка «Назад», що повертає користувача в попереднє меню. (рисунк 3.6)

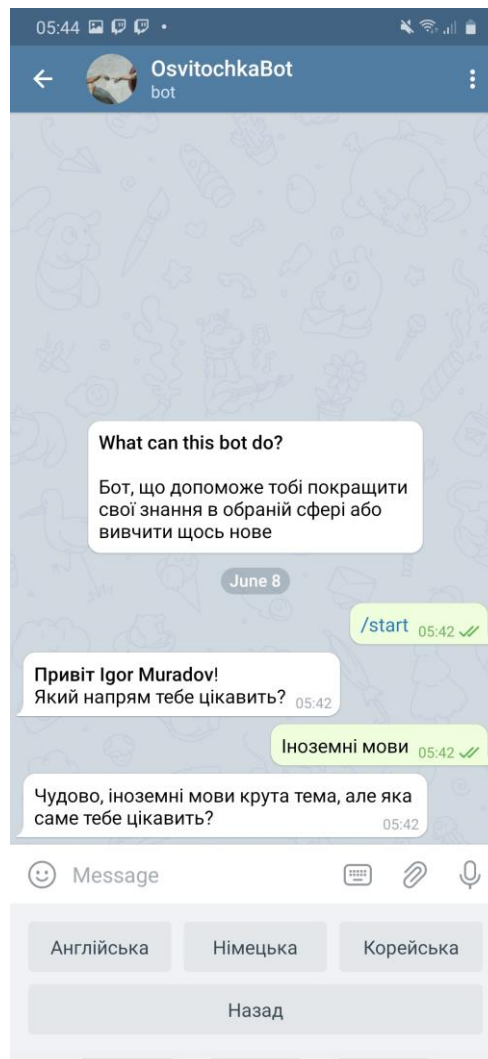


Рис. 3.6 Вибір напрямку в конкретній обраній дисципліні

Коли користувач обирає один з запропонованих варіантів, то він отримує посилання на певний курс, що допоможе йому у подальшому отриманні знань. (рисунок 3.7)

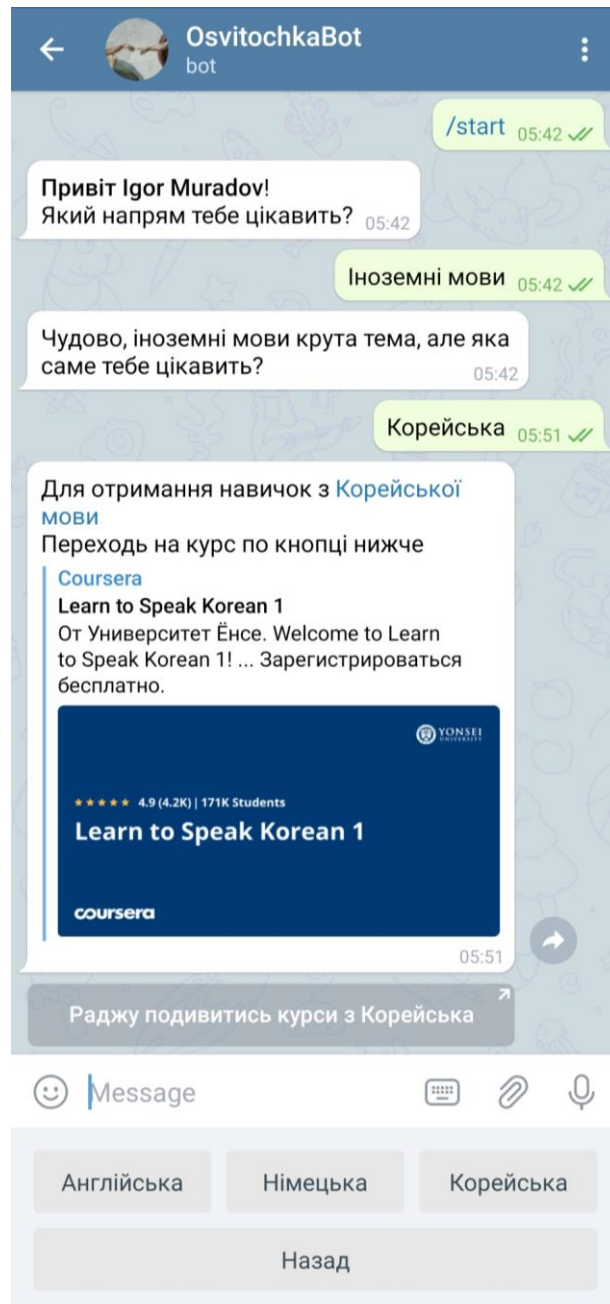


Рис. 3.7 Отримання користувачем посилання на курс

Також для зручності користувача додано посилання в вигляді кнопки-повідомлення під основним посиланням. Основне посилання інтегровано в текст, що виділено синім кольором.

Після переходу за посиланням користувач що взаємодіє з ботом за допомогою телефону потрапляє у інтернет-браузер Telegram в якому може пройти запропонований курс або перейти уже інстальований раніше браузер і продовжити роботу з ним там (рисунок 3.8). З ПК користувач одразу потрапляє за посиланням на сайт курсу у своєму інтернет-браузері.

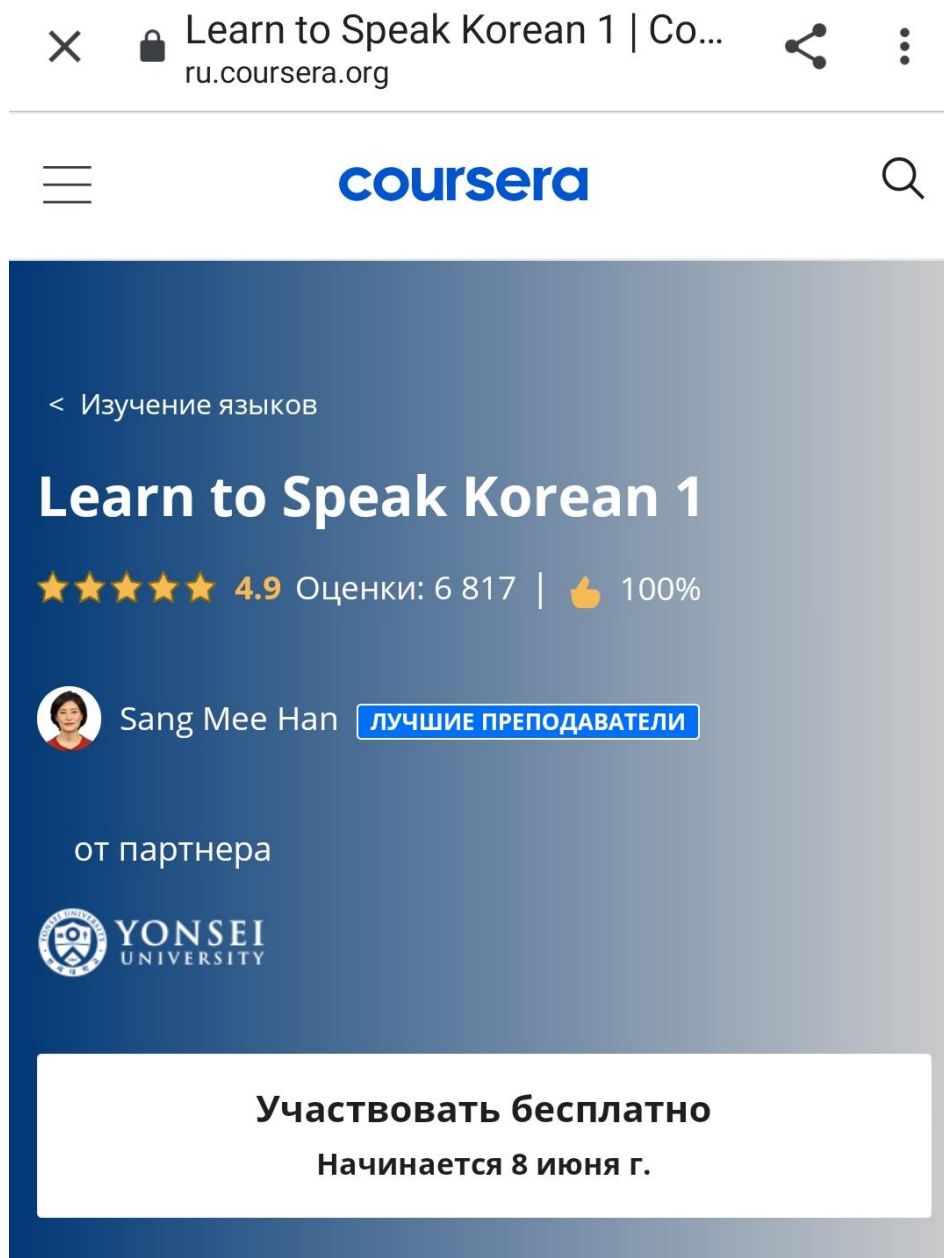


Рис. 3.8 Інтернет-браузер Telegram та відкрите в ньому посилання

Бот працює, але все ще лишилась одна проблема. Він працює лише тоді, коли код запущено на комп'ютері. Для вирішення цієї проблеми бота треба захостити.

3.2 Вибір хостинга та процес хостингу

Було обрано хостинг бота на хмарній платформі від Гугл (Google Cloud Platform). Google Cloud Platform — запропонований компанією Google набір хмарних служб, які виконуються на тій же самій інфраструктурі, яку Google використовує для своїх продуктів призначених для кінцевих споживачів, таких як Google Search та YouTube

Google Cloud Platform надає такі оточення як інфраструктура як послуга, платформа як послуга, та безсерверні обчислення. Інтерфейс цієї платформи зображений на рисунку 3.9.

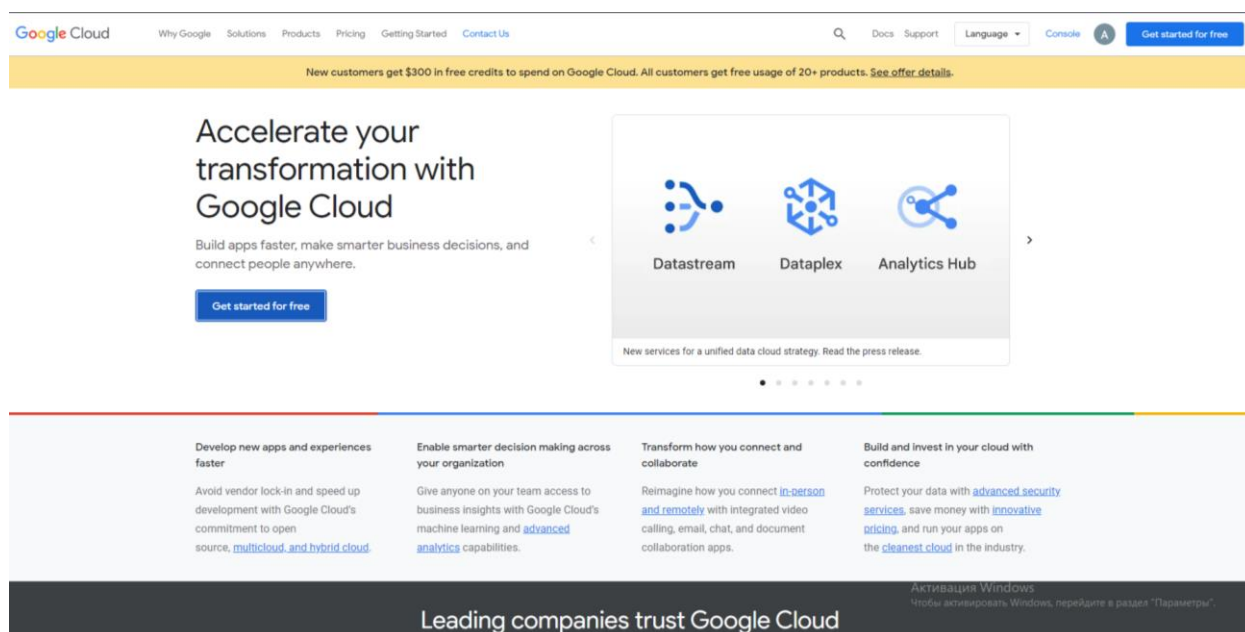


Рис. 3.9 Інтерфейс Google Cloud Platform

Роботу з даним сервісом починаємо з реєстрації. Після її проходження, верифікації акаунту ми можемо почати роботу з сервісом. Далі нам необхідно створити віртуальну машину на даному ресурсі. Меню створення майбутньої віртуальної машини показано на рисунку 3.10. Для її створення необхідно зазначити такі параметри: назва, розташування сервера, характеристики віртуальної машини, а також операційну систему.

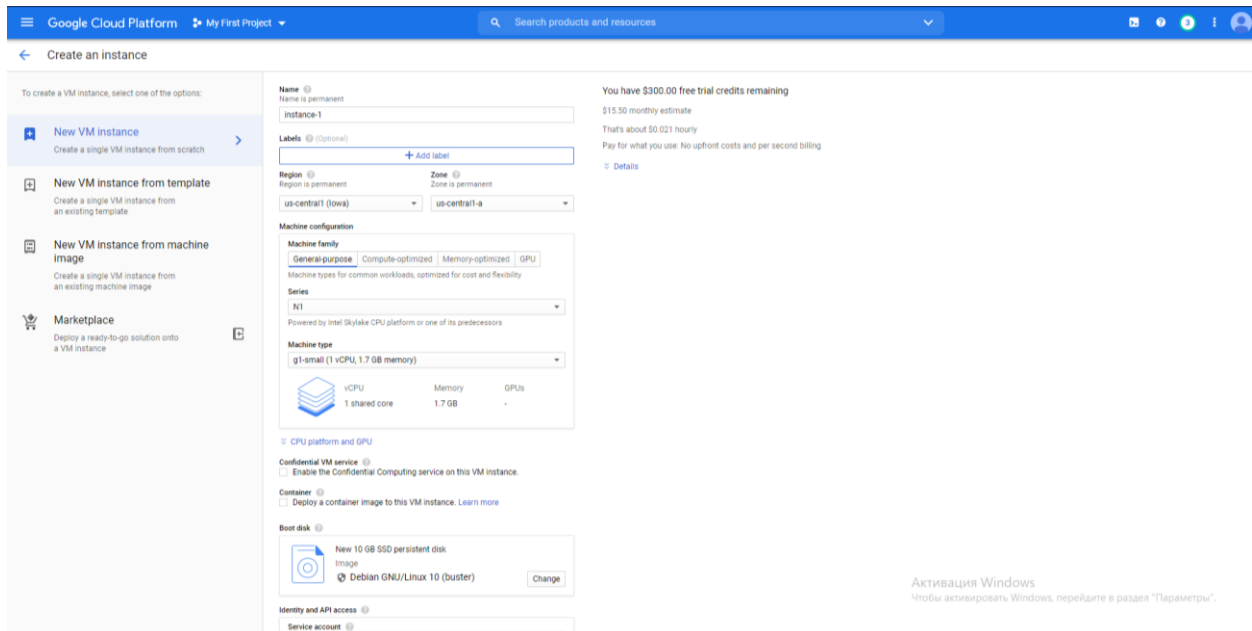


Рис. 3.10 Меню створення Віртуальної машини в Google Cloud Platform

Після створення віртуальної машини підключаємося до неї зі свого ПК через SSH (рисунок 3.11). А далі лишається лише налаштувати на віртуальній машині Python та за допомогою GitHub запусити бота на віртуальній машині.

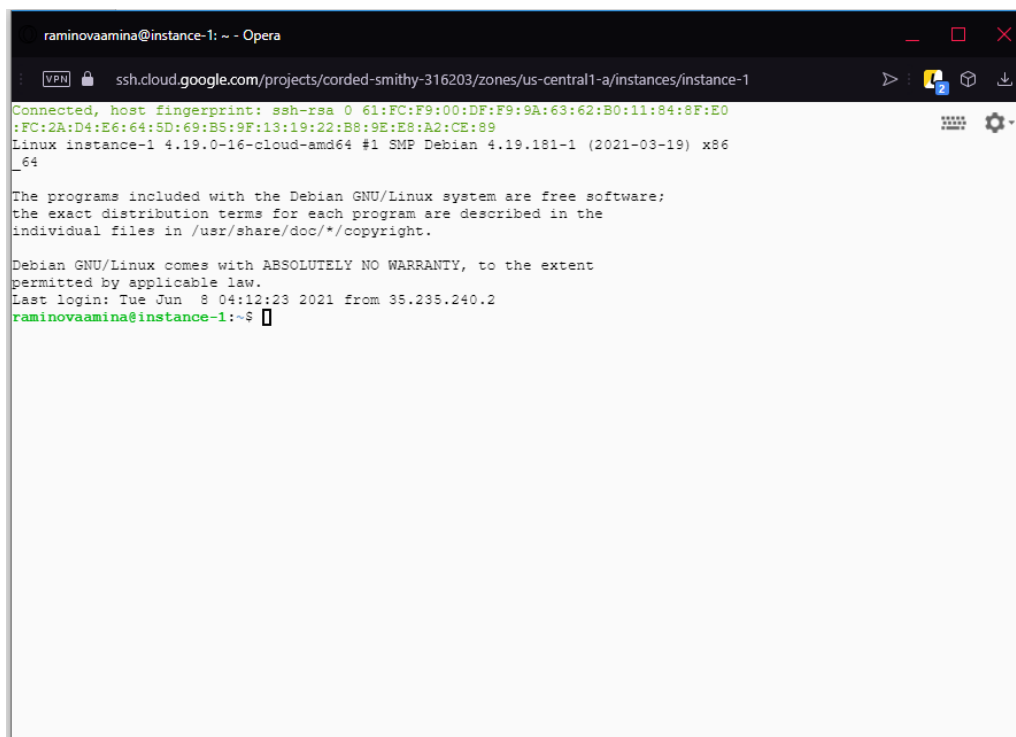


Рис. 3.11 Віртуальна машина з операційною системою Linux

3.3 Висновки до розділу

В цьому розділі було описано процес створення бота, а також було показано процес його роботи з точки зору користувача. Також було обрано систему хостингу для автономної роботи бота та застосування його на практиці. Бот працює повністю написано, за забезпечено повну його автономність.

ВИСНОВКИ

В сучасному суспільстві популярність месенджерів, як засобу спілкування безперечно зростає. Компанії, сім'ї, друзі щоденно використовують засоби обміну повідомленнями на відстані. У зв'язку з цим складно не помітити зростання зацікавленості людей до використання чат-ботів, які працюють на платформі месенджерів. Служба підтримки, конвертування документів, замовлення таксі, пошук необхідних даних та багато чого іншого завдяки ботів може бути реалізовано в рамках одного лише месенджера. Подібний підхід до користувачів дозволяє не завантажувати безліч вузькопрофільних додатків. Адже набагато легше мати лише месенджер та певний набір чат-ботів, які не займають місце в пам'яті телефона та будуть доступні в будь-який момент .

В рамках дипломного проекту було виконано поставлені задачі. По-перше, було проаналізовано найпопулярніші світові месенджери, вивчено Telegram та його популярність, також було визначено актуальність Telegram ботів. По-друге, було описано процес організації у роботі над Telegram ботом, а саме визначено найбільш актуальну мову програмування, зручне середовище розробки. Визначено основні функції мови програмування, що будуть необхідні в процесі розробки.

На основі цього було розроблено Telegram бот «OsvitochkaBot». Що задіяв мову програмування Python, середовище розробки PyCharm. Для автономної роботи бота, його було розміщено на віртуальній машині в хмарному сервісі від компанії Google.

Таким чином, результатом дипломного проекту є повністю реалізований та автономний Telegram бот «OsvitochkaBot», що може допомагати людям у пошуці курсів для отримання знань у певній сфері, або їх покращення.