**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**
**ІМЕНІ ТАРАСА ШЕВЧЕНКА**
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

**Кваліфікаційна робота**
**на здобуття ступеня бакалавра**
за спеціальністю 121 Інженерія програмного забезпечення
на тему:

# РОЗРОБКА ТЕЛЕГРАМ-БОТА ДЛЯ ПАРСИНГУ СОЦІАЛЬНИХ МЕРЕЖ

Виконав: студент 4-го курсу
Липинець Ярослав _____
(підпис)

Науковий керівник:
доцент, кандидат фізико-математичних наук
Іванов Євген _____
(підпис)

Засвідчую, що в цій кваліфікаційній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри інтелектуальних програмних систем
25 травня 2022 р.,
протокол № 10

Завідувач кафедри
Олександр Провотар _____
(підпис)

КИЇВ-2022

# РЕФЕРАТ

Обсяг роботи 29 сторінок, 10 ілюстрацій, 11 джерел посилань.

БОТ, ЧАТ-БОТ, МЕСЕНДЖЕР, TELEGRAM, ПАРСИНГ, СОЦІАЛЬНІ МЕРЕЖІ.

Об’єктом розроблення програмного засобу є процес парсингу, тобто збору і систематизації даних з інтернет-джерел з подальшою передачею цих даних чат-боту для взаємодії з користувачем.

Метою кваліфікаційної роботи є створення програмного засобу для збору і систематизації публікацій користувачів таких соціальних мереж як Instagram, Facebook та Twitter з подальшою їх публікацією у Telegram-канали та групи за допомогою чат-бота.

Інструменти розроблення: середовище розробки Microsoft Visual Studio Code, мова програмування JavaScript, платформа Node.js, технологія автоматизації веб-браузера Nightmare.

Результати роботи: досліджені способи створення ботів на платформі Telegram, досліджені методи збору і систематизації даних з інтернет-джерел, створений програмний засіб з інтерфейсом чат-бота для парсингу публікацій із соціальних мереж.

Розроблений програмний засіб може бути використаний публічними особами для полегшення адміністрування власних спільнот у Телеграм, а також іншими користувачами, які мають бажання слідкувати за новинами їх авторів із інших соціальних мереж.

# ЗМІСТ

# ВСТУП

**Оцінка сучасного стану об’єкта розробки.**

Сьогодні існує велика кількість засобів, способів і форм поглинання інформації і новин, і більшість з них пов’язана з сучасними інформаційно-комунікативними технологіями. Одним із засобів комунікації і сприйняття нової інформації є платформа Telegram. Платформа вже давно еволюціонувала з простого месенджера у потужний інструмент для безпечної і конфіденційної комунікації людей, надійного зберігання і обміну необмеженої кількості файлів, а також ведення свого мікроблогу. Завдяки спеціальному API сторонні розробники мають можливість створювати так званих «чат-ботів», які розширюють можливості Telegram ще більше: від пошуку і прослуховування музикальних композицій до допомоги у веденні бізнесу. Боти можуть виконувати найрізноманітніші фунції, саме тому боти набули широкої популярності як серед звичайних користувачів Telegram, так і серед публічних осіб і видань, які мають свої мікроблоги у вигляді Telegram-каналів.

**Актуальність роботи та підстави для її виконання.** Оскільки сьогодні Телеграм – це не тільки месенджер, а ще він має спільноти (групи, канали), існує потреба у постійному потоці нового контенту. Нові користувачі генерують новий для платформи контент, але при цьому все ще є необхідність у контенті з інших платформ, якого поки немає у Телеграмі. Тому з’явилась потреба у ретрансляції контенту з інших соціальних мереж, найпопулярнішими з яких є Instagram, Facebook і Twitter.

**Мета й завдання роботи.** Метою кваліфікаційної роботи є створення програмного засобу для збору і систематизації публікацій користувачів таких соціальних мереж як Instagram, Facebook та Twitter з подальшою їх публікацією у Telegram-канали та групи за допомогою чат-бота. Виходячи з цієї мети, були поставлені такі завдання:

- Дослідити існуючі методи збору та систематизації даних з інтернет-джерел.
- Дослідити способи створення чат-ботів для платформи Telegram та способи їх взаємодії зі стороннім програмним забезпеченням.
- Вибір і підготовка інструментів розробки для створення відповідного програмного засобу.
- Реалізація програмного засобу із інтерфейсом телеграм-бота для збору інформації з обраних соціальних мереж і подальшою їх передачею у Телеграм.

**Об'єкт, методи й засоби розроблення.** Об’єктом розробки програмного засобу є процес парсингу, тобто збору і систематизації даних з інтернет-джерел з подальшою передачею цих даних чат-боту для взаємодії з користувачем.

Під час розробки використана система емуляції дій користувача у веб-браузері для авторизації в обраних соціальних мережах, знаходження потрібного джерела (сторінки у соц. мережі) та знаходження потрібних даних, які необхідно зберегти для наступної передачі їх користувачу за допомогою чат-бота.

**Можливі сфери застосування.** Розроблене програмне забеспечення може бути використане як допоміжний інструмент публічним особам, які вже мають певну аудиторію у соціальних мережах та планують розширити її, або перевести на платформу Telegram за допомогою ретрансляції дописів зі своїх сторінок у соціальних мережах у власному Telegram-каналі. Зокрема, кінцевий продукт може бути використаний спільнотою фанатів деякої публічної особи, створюючи групи і канали у Telegram, посвячені цій особі, публікуючи туди її дописи з інших соціальних мереж.

# РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ

## 1.1 Месенджери і чат-боти

Інтернет-сервіси для спілкування почали свій розвиток з форумів, їм на зміну прийшли соціальні мережі, де користувачі могли спілкуватися один з іншим в особистих чатах, тепер популярність серед засобів комунікації через інтернет набирають месенджери, які повністю повторили концепцію чатів у соціальних мережах, але переросли у повноцінні додатки. Згідно з опитанням, проведених порталом datareportal.com з третього кварталу 2018 року [1] по перший квартал 2022 року [2] загальна кількість щомісячних користувачів таких месенджерів як WhatsApp, FB Messenger та Telegram виросла в середньому на 73%. За той же проміжок часу кількість користувачів таких соціальних мереж як Facebook, Instagram і Twitter в середньому зросла на 41%. Це показує, що на даний момент месенджери набирають популярність і однією з найшвидше зростаючих платформ є Telegram.

Телеграм – це кросплатформенний месенджер для обміну текстовими, голосовими та відеоповідомленнями, а також медіа-файлами і документами різних форматів [3]. Телеграм доступний для більшості популярних операційних систем, серед яких Android, iOS, Windows, macOS і Linux.

Телеграм має декілька основних переваг, завдяки яким, власне, і набув такої популярності:

- Конфіденційність – повідомлення надійно шифруються;
- Безкоштовність – ніяких абонплат за користування месенджером і дзвінками;
- Швидкодія – велика шкидкість як доставки повідомлень, так і інтерфейсу;
- Відсутність будь-яких обмежень на розмір медіа-файлів і документів для передачі;

- Відкритість для сторонніх розробників.

Саме завдяки останньому пункту у Телеграмі набула великого поширення платформа для створення чат-ботів і роботи з ними.

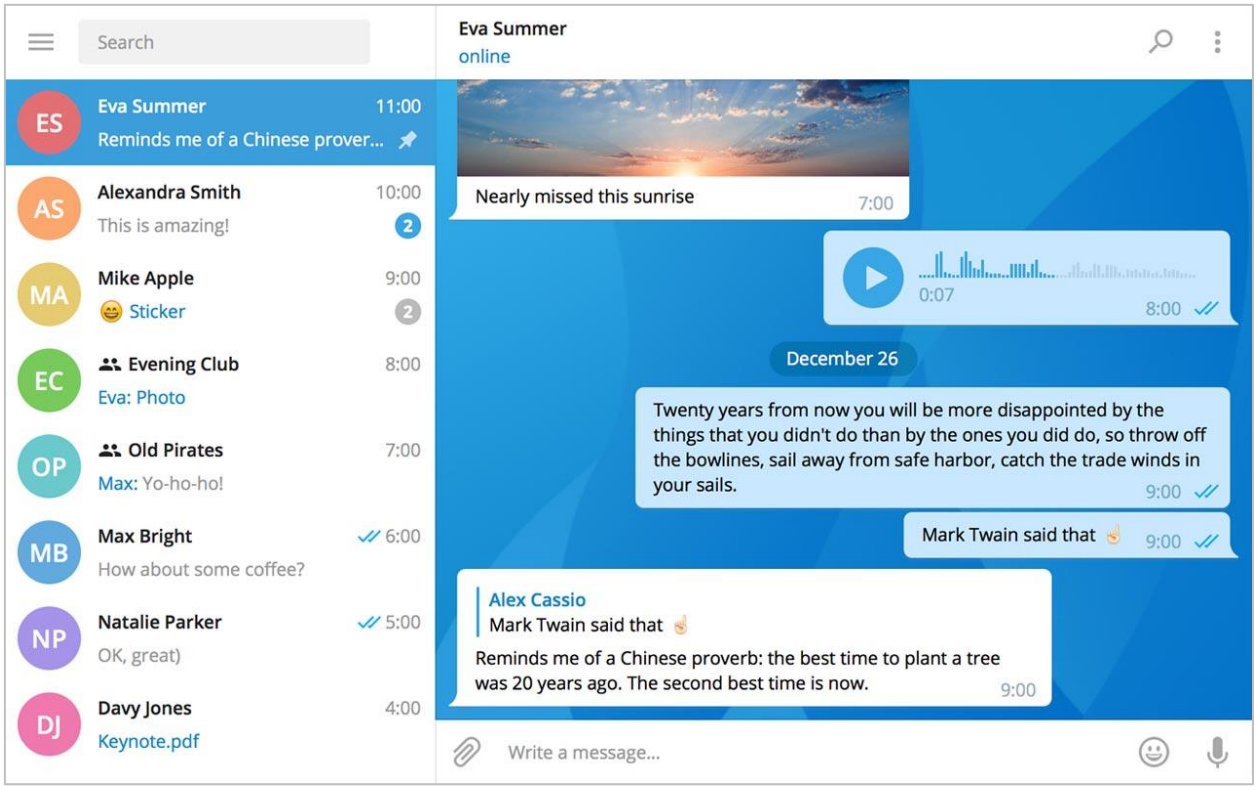



Рисунок 1 – користувацький інтерфейс клієнта Telegram для Windows

Визначимо що таке чат-бот (або просто бот). Бот – це спеціальна программа, яка може виконувати різноманітні функції, які вкладає в неї розробник, для взаємодії з користувачем за допомогою інтерфейсу чату (в нашому випадку чат в Telegram) [5]. Найчастіше боти використовуються для автоматизації дій реальної людини. Тому основною задачею будь-якого бота є автоматична відповідь на команду, яку вводить у діалог користувач. Оскільки програма може виконувати деякі дії людини у багато разів швидше, боти також допомагають зменшити часові витрати при виконанні тих же дій, на які людині потрібно набагато більше часу.

Бот, як віртуальний асистент, має деякий набір можливих питань, який може задати користувач і відповідний їм набір відповідей або сценарій дій,

який має відбутися при відповідній команді користувача. Отже, можна сказати, що у бота є деяка база знань для взаємодії з користувачем. Інколи, за допомогою штучного інтелекту, бот може аналізувати будь-які повідомлення від користувача, але найчастіше розробник заздалегідь розробляє можливі команди, які може обрати користувач і на які буде реагувати бот. Це може бути реалізовано у вигляді спеціальної клавіатури з необхідними клавішами, кожна з яких відповідна за окрему команду, а може мати вигляд звичайного набору команд, які користувач буде вводити у діалог з ботом вручну.

Боти мають декілька значних переваг, як для користувача, так і для розробника програмного засобу:

- Бота, на відміну від звичайного додатку, не потрібно завантажувати і встановлювати, оскільки весь програмний код бота знаходиться на стороні сервера. Для початку роботи потрібно тільки почати діалог з ним.
- Кросплатформенність та синхронізація. Неважливо на якому пристрої користувач використовує бота, він буде працювати однаково на всіх пристроях, які підтримує сам месенджер.
- Простота розробки користувацького інтерфейсу. Оскільки бот спілкується з користувачем за допомогою чату у месенджері, то розробнику потрібно лише спроектувати можливі сценарії взаємодії з ботом за допомогою заздалегідь продуманих команд.

Таким чином, боти є зручним засобом для реалізації багатьох цілей. Від виклику таксі по заданій адресі до отримання новин з різних джерел.

### 1.2 Поняття парсингу

Оскільки основною задачею бота, що розробляється, є отримання публікацій з інших соціальних мереж за допомогою парсингу веб-сторінок потрібно розглянути поняття парсингу.

Парсинг – це синтаксичний аналіз веб-документа і вилучення необхідної інформації з подальшою систематизацією у бази даних або електронні таблиці. Під парсером мається на увазі скрипт, який використовується для автоматичного збирання інформації з веб-сторінок і надання її користувачу у систематизованому вигляді. Парсинг сайту включає у себе доступ до потрібної веб-сторінки напряму через HTTP-запит, або через веб-браузер.

Робота парсера відбувається наступним чином:

1) Отримання URL-адреси потрібної сторінки;
2) Завантаження вмісту HTML-дерева сторінки;
3) Пошук та вилучення всієї необхідної інформації.

# РОЗДІЛ 2. ІНСТРУМЕНТИ ТА ТЕХНОЛОГІЇ РОЗРОБКИ

## 2.1 Технології розробки бота

### 2.1.1 Платформа Node.js

Як платформа для написання програмного засобу була обрана Node.js. Node.js - це кросплатформенне середовище виконання JavaScript з відкритим вихідним кодом, яке може запускати код JavaScript поза межами веб-браузера [4]. Це може бути використано, наприклад, для роботи веб-сервера.

Node.js працює в середовищі виконання JavaScript V8, яка початково була розроблена для Google Chrome, і яка відрізняється ресурсоефективною архітектурою, що може підтримувати велику кількість одночасних мережевих підключень. V8 компілює вихідний код JavaScript у свій машинний код під час виконання. Станом на 2016 рік він також включає Ignition – інтерпретатор байт-коду.

Мова сценаріїв JavaScript визначає архітектуру, керовану подіями. При роботі сервера це має ту перевагу, що для кожного існуючого з'єднання потрібно менше основної пам'яті, ніж для аналогічних програм, які запускають окремий потік для кожного відкритого з'єднання. Після кожного підключення запускається зворотний виклик, але якщо немає ніякої роботи, Node.js перейде в режим сну. Це відрізняється від більш поширеної сьогодні моделі паралелізму, в якій використовуються потоки ОС. Мережа на основі потоків відносно неефективна та дуже складна у використанні. Крім того, користувачі Node.js не переймаються блокуванням процесу, оскільки блокувань немає. Майже жодна функція в Node.js безпосередньо не виконує введення-виведення, тому процес ніколи не блокується, за винятком випадків, коли введення-виведення виконується за допомогою синхронних методів стандартної бібліотеки Node.js.

Для серверної платформи Node.js за замовчуванням встановлений менеджер пакетів – npm. Він встановлює модулі Node.js з реєстру npm, організуючи встановлення та керування сторонніми пакетами Node.js. Пакети в реєстрі npm можуть варіюватися від простих допоміжних бібліотек, таких як Lodash, до засобів запуску завдань, таких як Grunt.

Для реалізації методів парсингу будуть використані такі модулі:

- Модуль FS – дозволяє взаємодіяти з файловою системою методом, змодельованим на основі стандартних функцій POSIX;
- Модуль HTTP – дозволяє Node.js передавати дані по протоколу передачі гіпертексту (HyperText Transfer Protocol). Модуль HTTP може створити HTTP-сервер, який буде відстежувати порти сервера і повертати відповідь клієнту.
- Модуль npm mysql2 – дозволяє взаємодіяти з системою управління базами даних MySQL.
- Модуль npm download – дозволяє завантажувати медіа та інші файли.

Інші модулі, які потрібні для парсингу і взаємодії з користувачем, більш докладно розглянуті нижче.

### 2.1.2 Telegram Bot API

Bot API – це інтерфейс на основі HTTP, створений для розробників, які займаються створенням ботів для Телеграм. Цей API дозволяє підключати ботів до нашого програмного засобу. Телеграм-боти виступають як спеціальні аккаунти, для створення і налаштування яких не потрібен номер телефону [6]. Ці облікові записи слугують інтерфейсом для коду, який буде працювати на ПК або веб-сервері.

Існує два способи отримувати оновлення від Телеграм-бота: long pulling та webhook. При використанні long pulling наш додаток періодично перевіряє

наявність оновлень на серверах Телеграм. При використанні webhook сервери самостійно повідомляють додаток при появі оновлень.

Сервер Телеграм виступає мостом між клієнтом і ботом, логіка якого розміщена на сервері. Коли користувач, який використовує додаток Телеграму взаємодіє з ботом, спочатку відправляється запит на сервери Телеграм, де запускається процес запиту с протоколом HTTPS до бота, який працює десь на сервері. Бот-сервер надає оновлення відповідно запиту сервера Телеграм, тобто відповідь надсилається у зворотньому порядку. Відповідно, дизайн виконується с використанням Telegram Bot API у якості проміжної ланки для бота-сервера. Telegram Bot API - це API (інтерфейс прикладного програмування), який використовується для віддаленого моніторингу використання ботів як програмного забезпечення, що виконується на віддаленому сервері. Telegram Bot API використовує протокол шифрування MTPproto, який був створений для розробників як засіб безпеки.

Для написання ботів на Node.js існує відповідна бібліотека npm node-telegram-bot-api. Для підключення бота потрібно підключити до проекту даний модуль, а також спеціальний токен, мова про який піде в описі реалізації бота. Нижче наведені приклади методів [11], які надає node-telegram-bot-api:

- sendMessage – відправляє повідомлення від імені бота;
- sendPhoto - відправляє фото;
- openWebHook – відкриває вебхук для прослуховування оновлень від сервера Телеграм;
- getUpdates – надсилає запит на сервери Телеграм для отримання оновлень (long pulling).

### 2.1.3 Бібліотека Nightmare

Для парсингу веб-сторінок методом вилучення необхідної інформації з HTML-розмітки сторінки потрібна технологія автоматизації роботи з веб-

браузером. Такою технологією у даній роботі буде виступати бібліотека Nightmare з менеджера пакетів npm.

Nightmare – це високорівнева бібліотека автоматизації браузера, яку можна використовувати для тестування веб-сторінок, автоматизації введення даних на веб-сайтах і парсингу даних з веб-сайтів [7]. Nightmare використовує середовище браузера Chromium, яке надає фреймворк Electron. Electron - це фреймворк для розробки десктопних програм з використанням HTML, CSS і JavaScript, у двійковий код якого вже вбудовані Chromium та Node.js [8].

Принцип роботи Nightmare:

1. Nightmare ініціалізує новий додаток Electron зі стартовою сторінкою, яку необхідно піддати подальшій обробці.
2. Перед завантаженням сторінки, що досліджується, завантажуються скрипти, які дозволяють підтримувати двосторонню взаємодію розробника і сторінки.
3. Nightmare надає розробнику набір API, що дозволяють зробити будь-які маніпуляції з сайтом та отримати необхідні дані.

Серед плюсів цієї технології варто виділити невелике навантаження відносно інших браузерних парсерів, можливість відключати відображення вікна браузера для демонстрації його роботи і таким чином переводити його у фоновий режим роботи, підтримка проксі.

### 2.1.4 СУБД MySQL

Оскільки боту для нормальної роботи необхідно зберігати інформацію про id користувача, id груп та каналів, номер публікації і т.д., потрібна система управління даними. У даній роботі у якості системи управління базами даних була обрана MySQL.

MySQL – це система управління реляційною базою даних з відкритим кодом. MySQL є безкоштовним програмним забезпеченням з відкритим

вихідним кодом згідно з умовами Загальної публічної ліцензії GNU, а також доступний за різними власними ліцензіями. MySQL має окремі клієнти, які дозволяють користувачам безпосередньо взаємодіяти з базою даних MySQL за допомогою SQL, але часто MySQL використовується з іншими програмами для реалізації програмного забезпечення, яке потребує можливостей реляційної бази даних. MySQL використовується багатьма веб-додатками, що керуються базами даних, включаючи Drupal, phpBB, Joomla і WordPress. MySQL також використовується багатьма популярними веб-сайтами, включаючи Facebook, Flickr, Twitter і YouTube.

По суті, MySQL складається з сервера MySQL і клієнтів MySQL. Сервер MySQL – це система управління базами даних, через яку здійснюється доступ до даних, які зберігаються у базах даних. Потім ці дані можуть бути запитані одним або декількома клієнтами MySQL. На сервері MySQL можуть бути встановлені декілька баз даних, в яких дані зберігаються у двовимірних таблицях. Кожна така таблиця може містити стовбці, в яких дані мають вказаний тип даних (наприклад integer, тобто ціле число). Ці типи даних можуть складатися із числових типів, текстових рядків або типів дати та часу. Рядки реляцій називаються кортежами і являють собою конкретну комбінацію значень стовбців (значень атрибутів). Всі кортежі можуть бути запитані через механізм бази даних. Мова бази даних SQL використовується для створення і редагування структур даних у базі даних і запитів до них. Як правило, MySQL підходить для великої кількості різноманітних баз даних, які можут отримувати запити від декількох клієнтів. Бази даних MySQL можливо зробити загальнодоступними через Інтернет або ж застосувати для локального використання. Клієнт, який ініціює запити, може бути простою програмою командного рядка, складним програмним забезпеченням або веб-додатком. Доступ до баз даних можливо контролювати за допомогою прав доступу, тому можливо реалізувати модель, при якій конкретний додаток буде мати доступ лише до певних баз даних, а зміни вноситись тільки адміністраторами.

Існує декілька способів роботи з MySQL. Наприклад, це може відбуватися за допомогою командного рядку, або ж графічного інтерфейсу, що є набагато зручнішим, оскільки графічний інтерфейс може представляти базу даних у вигляді ієрархічних списків. Одним з засобів представлення баз даних MySQL за допомогою графічного інтерфейсу є PhpMyAdmin.

PhpMyAdmin – веб-додаток з відкритим кодом, написаний мовою PHP і який є веб-інтерфейсом для адміністрування СУБД MySQL. PhpMyAdmin надає можливість за допомогою браузера виконувати адміністрування сервера MySQL, запускати команди SQL та переглядати вміст таблиць баз даних.

Оскільки веб-додаток надає зручний графічний інтерфейс, він дозволяє керувати даними і таблицями без безпосереднього введеня SQL команд.

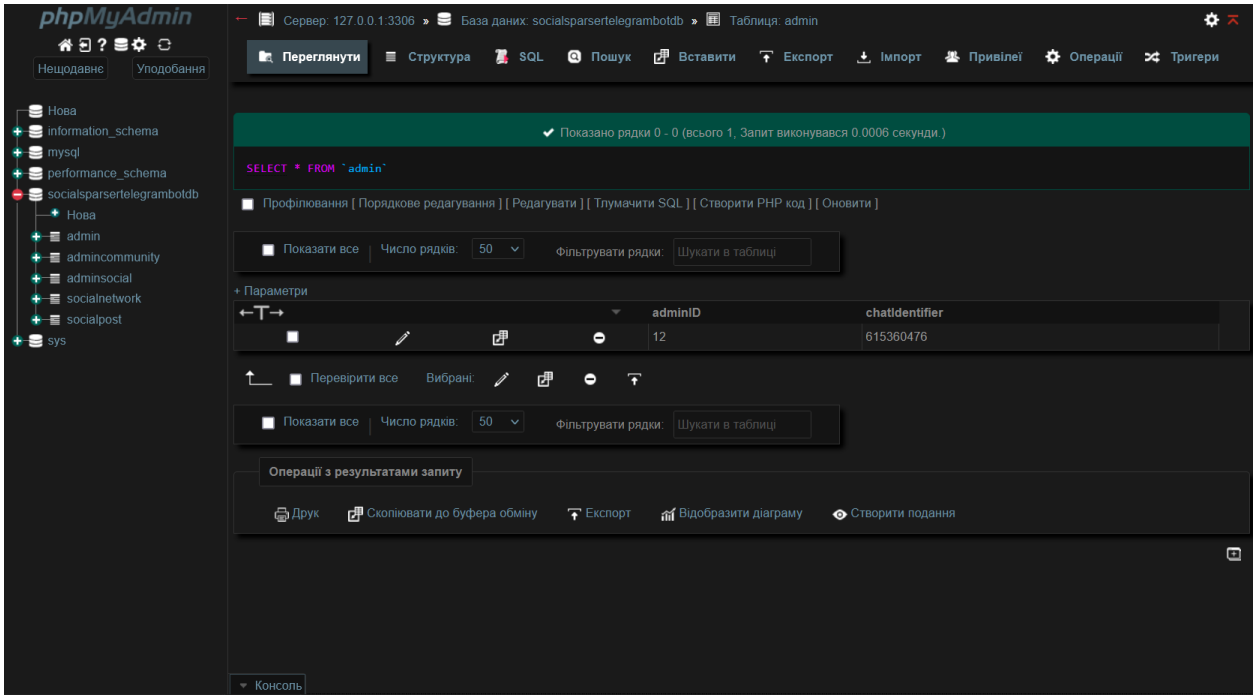



Рисунок 2 – Інтерфейс PhpMyAdmin

## 2.2 Середовище розробки

### 2.2.1 Microsoft Visual Studio Code

Visual Studio Code (або VS Code) – це безкоштовний текстовий редактор з відкритим вихідним кодом, розроблений компанією Microsoft. VS Code доступний для таких платформ як Windows, Linux і macOS [9]. Хоча редактор відносно легкий, він включає в себе декілька потужних функцій, які зробили VS Code одним з найпопулярніших середовищ розробки:

- Контроль версій. Однією з основних вбудованих функцій VS Code є контроль версій. Спеціальна вкладка у рядку меню надає користувачам доступ до налаштувань контролю версій. Крім того, користувачі мають можливість переглядати зміни, які були внесені до проекту. Однак використання цього способу управління вихідним кодом потребує прив’язки VS Code до необхідних систем контролю версій, наприклад Git. Користувачі можуть створювати репозиторії, відправляти і отримувати запити безпосередньо з самого редактора [10].
- Підтримка мов програмування. VS Code підтримує більшість популярних мов програмування, включаючи Python, C++, Go, JavaScript та інші. Надзвичайно зручними для розробника є функції згортання частин коду, підсвічування синтаксису і автопідстановки дужок. Інші функції можуть відрізнятися в залежності від мови програмування. VS Code також пропонує IntelliSense для TypeScript, CSS, HTML і т.д., а також засоби відладки для Node.js. Користувачі можуть використовувати безкоштовні розширення для підтримки додаткових мов, тем та відладчиків [10].
- Автодоповнення IntelliSense. IntelliSense – це технологія автодоповнення Microsoft. Вона дописує назву функції під час введення початкових літер. Крім прямого призначення, IntelliSense використовується для доступу до документації та усунення неоднозначності в іменах змінних, функцій та методів [10].

Розробники можуть швидко переміщатися між декількома інструментами, щоб внести необхідні зміни у свій код. VS Code також пропонує зручне налаштування гарячих клавіш та їх комбінацій.

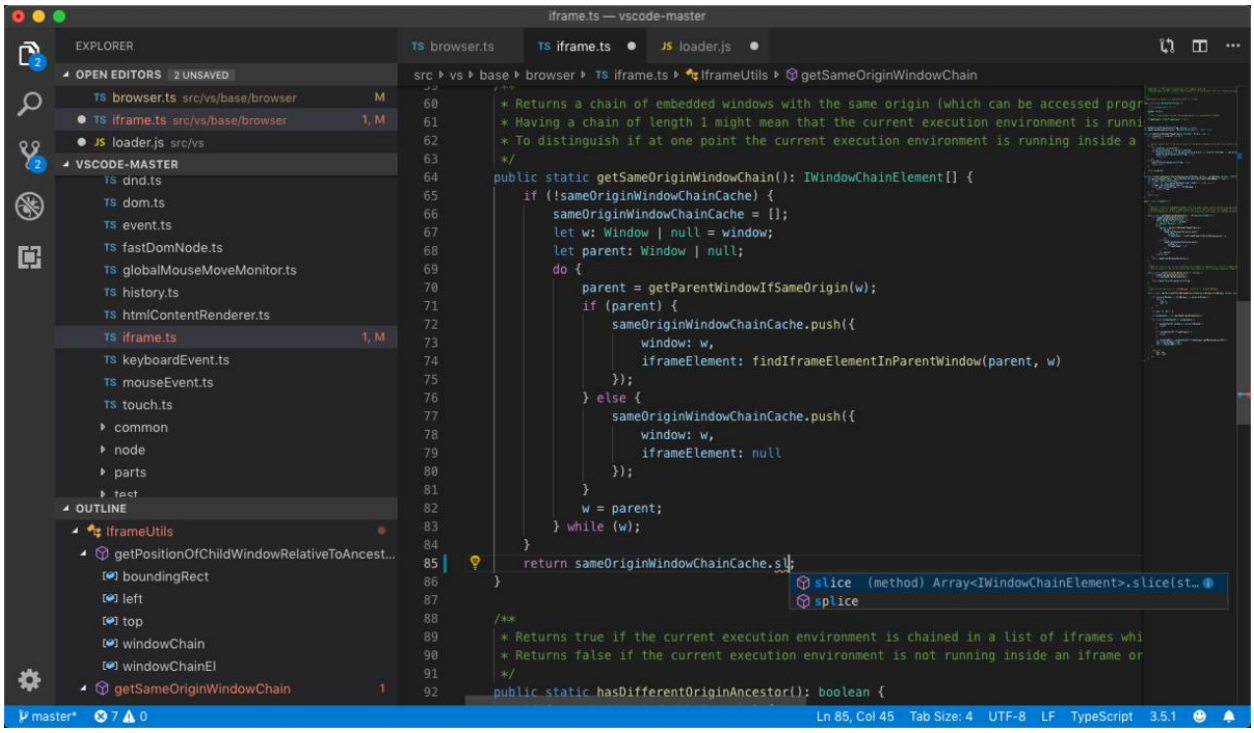



Рисунок 3 – зовнішній вигляд редактора VS Code

# РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ

## 3.1 Реєстрація бота у системі управління ботами Telegram

Щоб користувачі Телеграм могли знайти нашого бота через пошук аккаунтів потрібно зареєструвати бота у системі Телеграм. Для цього розробниками Телеграм був створений окремий бот (рисунок 4), за допомогою якого можна створювати і управляти власними ботами. Знайти цього бота можна ввівши у пошуковий рядок @BotFather [5].

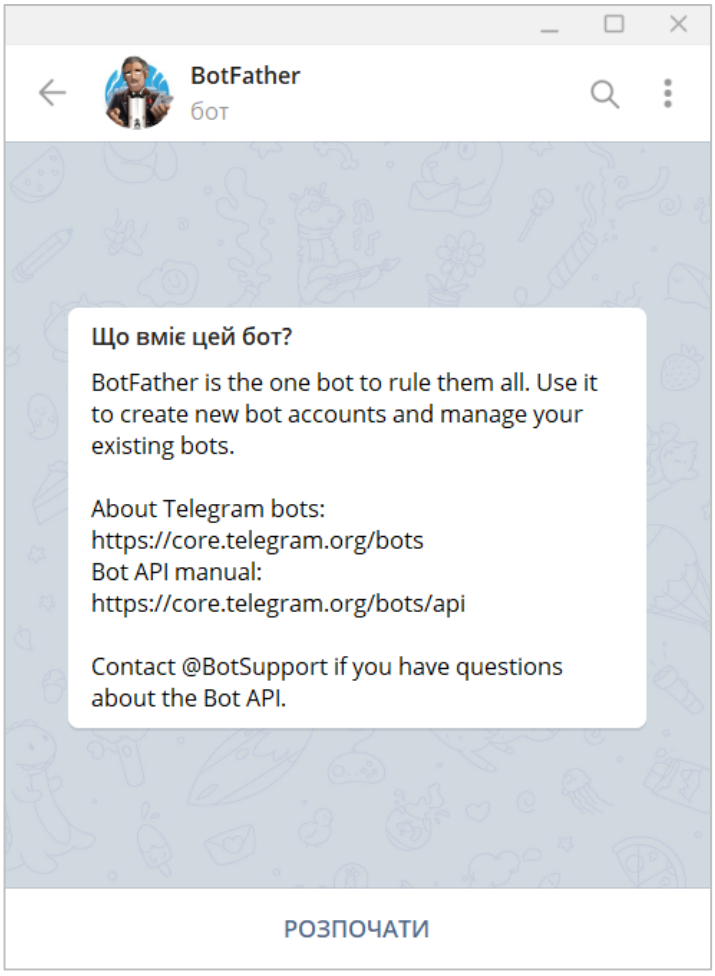



Рисунок 4 – Офіційний бот для створення і управління користувацькими ботами у Telegram

Зареєструвати свого бота можна ввівши у чат команду /newbot. Після цього запропонується ввести назву нашого бота, яку інші користувачі будуть

бачити у чаті з ботом. Після цього потрібно ввести тег нашого бота. Щодо тегу є дві умови:

1) Тег має закінчуватися на «Bot» або «_bot»;
2) Тег має бути унікальним.

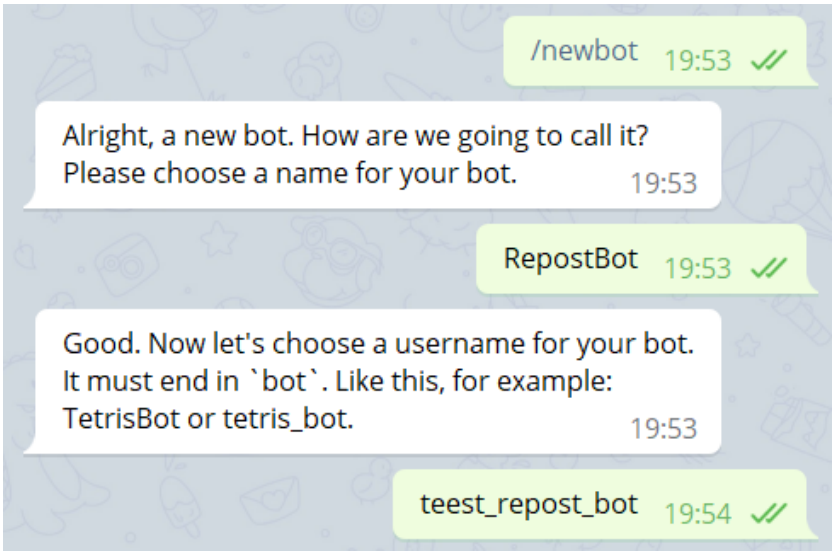



Рисунок 5 – Приклад реєстрації нового бота за допомогою BotFather

Якщо всі умови задоволені, BotFather повідомить нас, що бота створено, видасть посилання, за яким можна перейти у діалог зі створеним ботом, а також видасть спеціальний токен для доступу до HTTP API для підключення бота з вихідним кодом, який буде виконуватися при роботі бота.

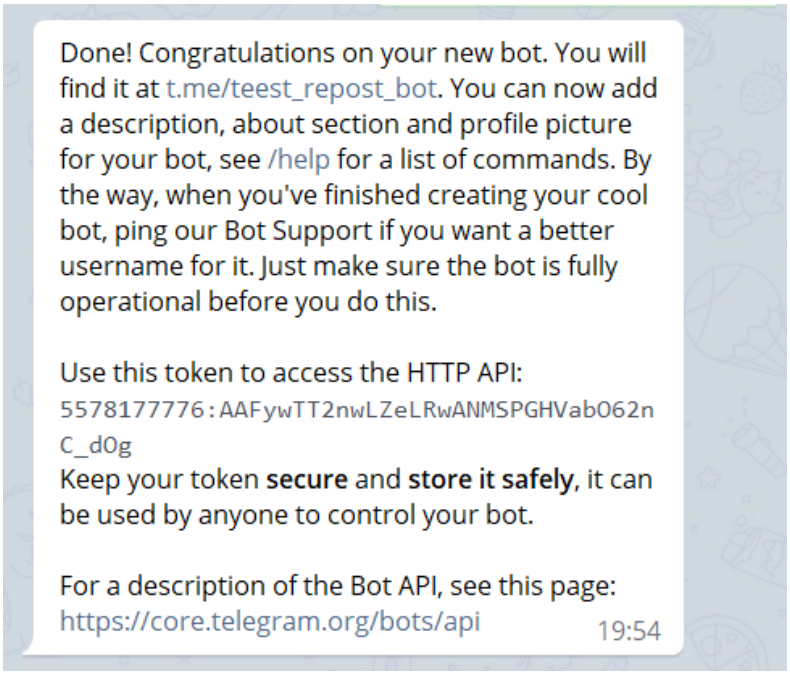



Рисунок 6 – повідомлення про успішну реєстрацію нового бота

BotFather дозволяє налаштувати бота у системі управління ботами Телеграм. Ввівши команду /mybots і обравши потрібного бота, користувач потрапляє у меню управління ботом, де він може:

- Отримати унікальний токен бота;
- Змінити зовнішній вигляд бота: назву, опис, інформацію про можливості бота, яку той виводить при початку діалогу з ним, змінити фотографію, додати і редагувати команди;
- Змінити налаштування бота: дозволити додавання у групи, налаштування приватності у групах, права адміністратора груп та каналів за замовчуванням, що може унеможливити додавання бота у канал без надання йому прав адміністратора для відправки публікацій;
- Налаштувати систему оплати за допомогою бота;
- Передати управління ботом іншому користувачу;
- Видалити бота.

Стандартний зовнішній вигляд бота було змінено: додано опис та інформацію про можливості, також було створено зображення акаунта бота.

Для нашого бота буде корисним налаштувати права адміністратора у каналах та групах за замовчуванням, щоб користувач випадково не забув це зробити і щоб нам не довелося робити перевірку на статус бота і чи може він відправляти публікації у канал.

### 3.2 Розробка користувацького інтерфейсу бота

Тепер бота можна знайти через пошук у Телеграм і навіть розпочати чат. Але поки що не можуть взаємодіяти з ним, оскільки спочатку бота потрібно підключити до проекту і описати користувацький інтерфейс. Користувачі можуть взаємодіяти з ботами за допомогою команд (це можуть бути екранні

кнопки, які з’являються в області клавіатури і які будуть відповідати за певні команди) або ж за допомогою наекранних кнопок, які буде виводити сам бот під кожним своїм повідомленням.

Сам процес спілкування з користувачем має мати наступний вигляд:

1) При стандартній команді /start бот має привітати користувача та надіслати інструкцію по його налаштуванню. У користувача у той час виводиться кнопка «Почати» для початку роботи.
2) Після команди «Почати» бот має запропонувати обрати соціальну мережу автора, публікації якого користувач хоче відстежувати. У користувача виводиться три кнопки з вибором потрібної соціальної мережі: Instagram, Facebook та Twitter.
3) Коли користувач обрав соціальну мережу, бот просить надіслати посилання на автора.
4) При отриманні посилання, бот має перевірити його на валідність, а користувач має отримати повідомлення, що посилання перевіряться.
5) При успішній перевірці посилання бот пропонує додати його у канал або групу і пояснює як правильно це зробити.

Також необхідно реалізувати можливості зміни автора та видалення групи. Для цього будуть реалізовані кнопки «Змінити соціальну мережу» та «Відв’язати групу» відповідно.

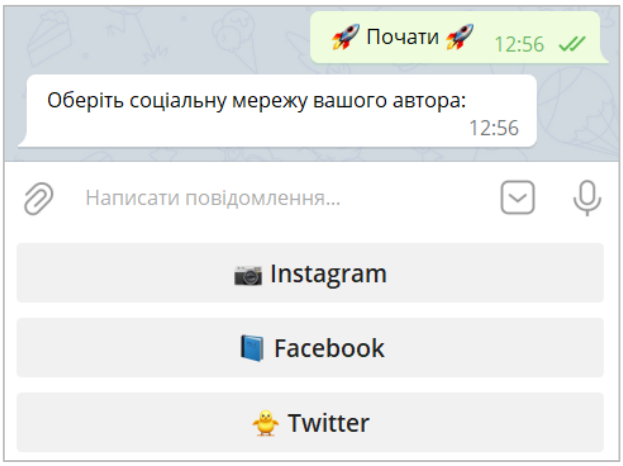
Розберемо на прикладі реалізацію другого пункту, тобто вибору соціальної мережі. Якщо бот отримує повідомлення від користувача, у якому той відправив у чат з ботом команду або «Почати», або «Назад», або «Змінити соціальну мережу», то бот відправляє повідомлення «Оберіть соціальну мережу автора» і викликає метод вибору соціальної мережі, який виводить користувачу відповідні кнопки. Так це виглядає у вигляді вихідного коду (рисунок 7 (а), (б)) та у вигляді користувацького інтерфейсу (рисунок 7 (в)):

```
if ((🚀 Почати 🚀)|(⬅ Назад)|(🔃 Змінити соціальну мережу)/.test(msg.text)) {
    users.filter(user => user.userId == chatId)[0].addingURL = false;
    return bot.sendMessage(chatId, "Оберіть соціальну мережу вашого автора:", selectSocialKeyboard);
}
```

а)

```
let selectSocialKeyboard = {
    "reply_markup": {
        hide_keyboard: false,
        resize_keyboard: true,
        "keyboard": [
            ["📷 Instagram"],
            ["📘 Facebook"],
            ["🐥 Twitter"]
        ]
    }
};
```

б)




в)

Рисунок 7 – Реалізація кнопки вибору соціальної мережі:
а) реалізація повідомлення від бота; б) реалізація клавіатури користувача;
в) вигляд кнопок та повідомлення у користувацькому інтерфейсі

Таким чином була реалізована вся взаємодія з користувачем.

### 3.3 Структура бази даних

Оскільки бот повинен працювати з багатьма користувачами, які в свою чергу зможуть додавати бота до декількох груп або каналів, необхідно створити базу даних, яка буде задовільняти потреби у систематизації унікального номера користувача, груп, опублікованих записів та іншого. Для цих задач була спроектована і створена база даних за допомогою системи MySQL і веб-інтерфейсу PhpMyAdmin, діаграму якої наведено на рисунку 8.

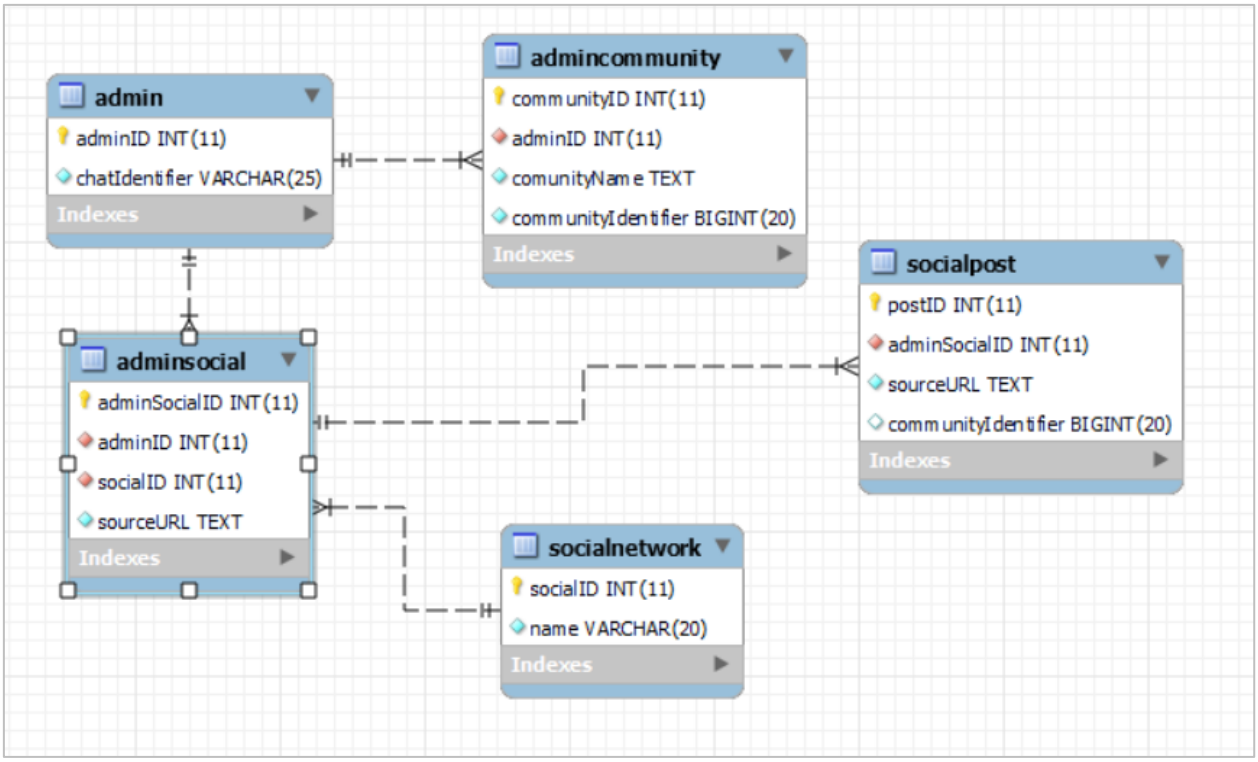



Рисунок 8 – Діаграма таблиць бази даних бота

Нижче наведений більш детальний опис кожної з п’яти таблиць бази даних бота:

- admin – зберігає унікальний id користувача та ідентифікатор діалогу з користувачем;
- admincommunity – групи або канали, куди користувач додав бота, зберігає id користувача, назву групи або каналу та його ідентифікатор;
- adminsocial – містить інформацію про обрану соціальну мережу та посилання на автора (джерело) відповідно до кожного користувача. Для цього в таблиці містяться id користувача, id соціальної мережі та посилання на автора у вигляді тексту;
- socialpost – містить інформацію про опубліковані записи, зберігаючи id публікації, id набору даних про прив’язку джерела відповідно до конкретного користувача, посилання на джерело та ідентифікатор групи або каналу, де був опублікований запис;

- socialnetwork – містить відповідність назв соціальних мереж до їх ідентифікаторів для більш зручної взаємодії з базою даних і користувачем.

### 3.4 Реалізація серверної сторони бота

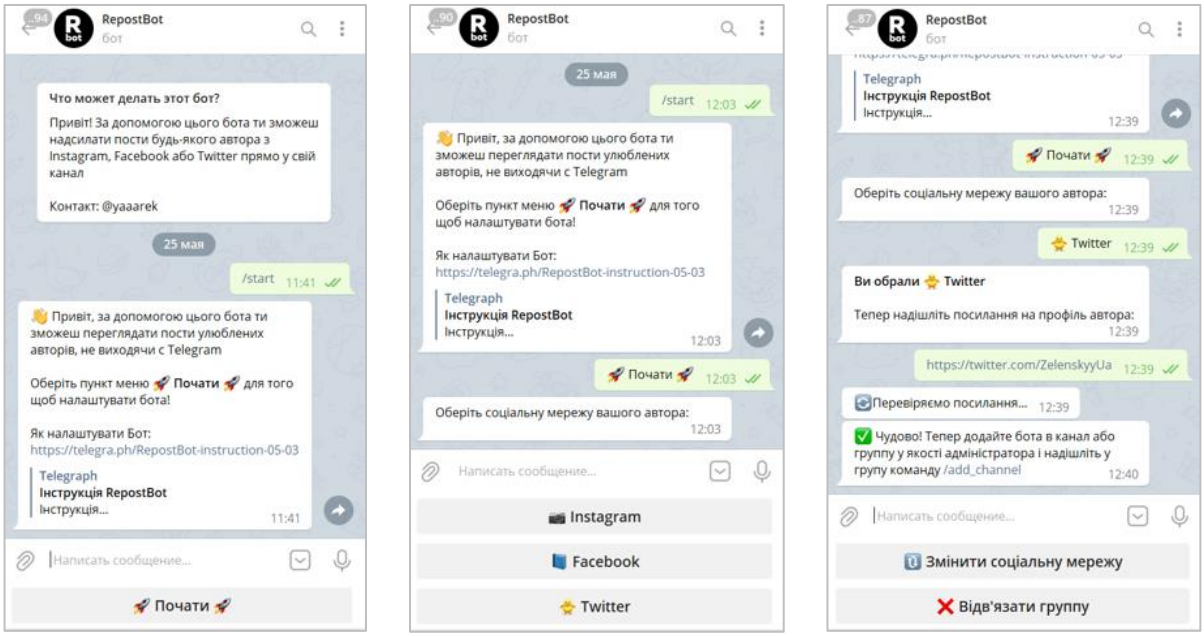
Для початку потрібно сформулювати основні задачі, які повинна вирішувати програмна частина бота:

- Отримувати, розпізнавати і реагувати на команди користувача;
- Виводити екранні кнопки для користувача;
- Додавати і редагувати інформацію в базі даних;
- Парсити задані джерела, зберігати отриманий контент і відправляти його у групу, куди був доданий бот;
- Перевіряти валідність посилань на джерела, які надає користувач;
- Взаємодіяти з групами і каналами, куди був доданий бот.

Структуру програмної частини бота було вирішено реалізувати у вигляді трьох основних частин:

1) index.js – основа бота, де програмний код з’єднується з користувацькою стороною бота за допомогою унікального токена і створюється сам клієнт Телеграм-бота. Ця частина також буде розпізнавати і відповідати на команди бота, виводити відповідні кнопки, запускати методи парсингу соціальних мереж, видаляти додані групи.
2) pageParser.js – частина, в якій реалізовані три методи парсингу соціальних мереж, а також метод перевірки сторінки на валідність.
3) db.js – описані методи взаємодії з базою даних на мові SQL та відповідні дії бота, які повинні виконуватися при зверненні до бази даних.

Розглянемо процес роботи користувача з ботом і що у цей час відбувається на програмній стороні бота. У першу чергу користувач має розпочати діалог з ботом, обрати соціальну мережу і надіслати посилання на автора, якого бажає відслідковувати. Бот у цей час відправляє відповідні повідомлення і виводить потрібні екранні кнопки.




а) б) в)

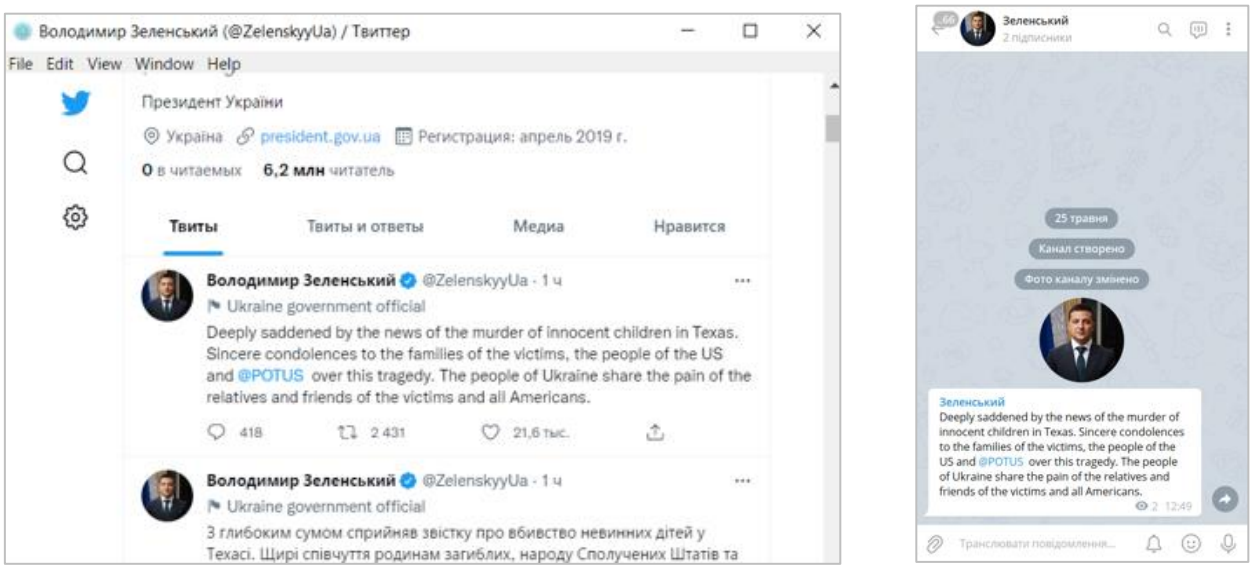
Рисунок 9 – Процес початку роботи користувача з ботом:
а) початок діалогу; б) вибір соціальної мережі;
в) отримання посилання на джерело і відправка відповідних повідомлень ботом.

Оскільки програмна частина процесу отримання, відправки і виводу кнопок вже розглядалася при описі користувацького інтерфейсу, розглянемо те, що відбувається після отримання ботом посилання на автора. Після отримання посилання, бот виводить повідомлення, що здійснюється перевірка посилання. У цей час запускається метод перевірки посилання: метод перевіряє чи користувач надіслав посилання на ту соціальну мережу, яку обрав, чи не має посилання помилок, оскільки ми маємо представлення про те, який вигляд повинно мати посилання. Якщо посилання правильне, то бот запускає метод валідації сторінки evaluatePage з файлу pageParser.js. Метод ініціалізує новий об’єкт класу nightmare для запуску автоматизованого веб-

браузера. Бот вже перевірив, що посилання правильне, тож при відкритті веб-браузера і переході за посиланням відкриється відповідна сторінка у соціальній мережі. Метод виконує пошук по HTML-дереву сторінки, аналізуючи її на наявність таких елементів як «Сторінки не існує», «Сторінка приватна» та подібних, в залежності від кожної соціальної мережі. Якщо таких елементів не було знайдено, робимо висновок, що сторінка загальнодоступна. Тому відправляємо користувачу повідомлення, що проблем не виникло, надсилаючи інструкцію по додаванню бота у спільноти і додаємо відповідну інформацію про користувача та джерело до бази даних. У іншому випадку, якщо сторінка не загальнодоступна, бот надсилає користувачу відповідне повідомлення і пропонує надіслати посилання ще раз.

Розглянемо процес додавання у спільноту на прикладі додавання бота у канал, але процес додавання у групу аналогічний. Під час налаштування нашого бота у BotFather були вказані права адміністратора за замовчуванням при додаванні у канал, тому користувач зможе додати бота лише у якості адміністратора. Це потрібно для того, щоб бот мав можливість надсилати публікації у канал. Дотримуючись інструкції, користувач додає бота у канал як адміністратора з відповідними правами за замовчуванням і надсилає у канал спеціальну команду /add_channel. Це потрібно для того, щоб бот міг розпізнати, що його додали до нової спільноти, і отримати ідентифікатор спільноти. У цей момент бот додає канал до бази даних і повідомляє про це користувача. Потім починається процес парсингу заданого джерела: в залежності від соціальної мережі, запускається відповідний метод парсингу соціальної мережі. Розглянемо детальніше цей метод на прикладі парсингу сторінки у Twitter. Метод parsingInterval у db.js буде використовувати метод parseTwitter для роботи саме з Twitter. Метод parseTwitter, як і метод evaluatePage, ініціалізує об’єкт веб-браузера і виконує відповідні скрипти. За допомогою скрипта «article[data-testid='tweet']» виконуємо пошук останньої (найновішої ) публікації. Скрипт «div[data-testid='tweetText']» виконує пошук

тексту автора у публікації. Якщо він є, то він зберігається. Аналогічно виконується пошук фото за допомогою скрипта «"div[data-testid='tweetPhoto'] img» і пошук відео – «div[data-testid="videoPlayer"]». Відповідний контент зберігається. Після цього веб-браузер автоматично зупиняє роботу, а збережений контент публікується в обраному каналі. Також автоматично задається інтервал парсингу і через деякий час відбувається новий процес парсингу, під час якого отримується остання публікація і перевіряється чи вона вже існує у каналі. Якщо з'явилася нова публікація, то бот публікує її у канал. У іншому випадку бот нічого не публікує.




а) б)

Рисунок 10 – Процес парсингу сторінки у соціальній мережі:
а) автоматизований пошук потрібного контенту у веб-браузері;
б) публікація запису в оригінальному вигляді у каналі користувача Телеграм.

Таким чином був реалізований процес парсингу та публікації записів бажаного джерела у телеграм-канал користувача бота. Схожим чином відбувається процес парсингу Facebook та Instagram з поправками на відповідні скрипти.

# ВИСНОВКИ

У наш час все більшу популярність набирають месенджери, де користувачі обмінюються не тільки текстовими повідомленнями, а й медіаконтентом, використовуючи месенджери як джерело новин. Одним з найпопулярніших сьогодні месенджерів є Telegram, що, як і будь-яка соціальна платформа, потребує постійного потоку нового контенту, у тому числі і з інших соціальних мереж. Для вирішення цієї проблеми було прийнято рішення створити програмний продукт для ретрансляції контенту з інших соціальних мереж у Telegram, використовуючи популярну нині модель чат-ботів.

У процесі виконання кваліфікаційної роботи було проаналізовано предметну область та визначено основні задачі, які повинен вирішувати програмний засіб.

Були досліджені методи парсингу інформації із соціальних мереж, способи реалізації ботів для платформи Telegram, були проаналізовані інструменти і технології для вирішення поставлених задач.

Результатом кваліфікаційної роботи є створений програмний продукт з інтерфейсом чат-бота для парсингу інформації заданих джерел із трьох соціальних мереж і подальшої її передачі у Telegram-спільноти.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ


1. Dateportal [Електронний ресурс] : Digital 2018: Q3 Global digital statshot. – Режим доступу: https://datareportal.com/reports/digital-2018-q3-global-digital-statshot?rq=DIGITAL%202018%20statshot
2. MessengerPeople [Електронний ресурс] : WhatsApp, WeChat and Meta Messenger Apps – Global usage of Messaging Apps, Penetration and Statistics. – Режим доступу: https://www.messengerpeople.com/global-messenger-usage-statistics/
3. Wikipedia [Електронний ресурс] : Телеграм. – Режим доступу: https://uk.wikipedia.org/wiki/Telegram
4. Wikipedia [Електронний ресурс] : Node.js. – Режим доступу: https://uk.wikipedia.org/wiki/Node.js
5. Офіційний сайт Telegram [Електронний ресурс] : Bots: An introduction for developers. – Режим доступу: https://core.telegram.org/bots
6. Офіційний сайт Telegram [Електронний ресурс] : Telegram API. – Режим доступу: https://core.telegram.org/api
7. Офіційний сайт NPM [Електронний ресурс] : Nightmare. – Режим доступу: https://www.npmjs.com/package/nightmare
8. Офіційний сайт Electron [Електронний ресурс] : Документація Electron. – Режим доступу: https://www.electronjs.org/ru/docs/latest
9. Educative [Електронний ресурс] : What is Visual Studio Code? – Режим доступу: https://www.educative.io/edpresso/what-is-visual-studio-code
10. Webopedia [Електронний ресурс] : Visual Studio Code. – Режим доступу: https://www.webopedia.com/definitions/visual-studio-code/
11. GitHub [Електронний ресурс] : Документація node-telegram-bot-api. – Режим доступу: https://github.com/yagop/node-telegram-bot-api/blob/release/doc/api.md