Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

УДК 004.942

На правах рукопису

ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “ Засіб централізованого керування комп’ютерами через чат-бота ”

Спеціальність – 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

БР.ІПЗ - 31.00.00.000

Студент
ІПЗ-43_____/Максим МАТВЄЄВ/

Науковий керівник

д.т.н._____/Генадій ПОРЄВ /

Допускається до захисту

Консультант
з питань нормконтролю

фахівець_____/Тамара ЧАПОВСЬКА/

Завідувач кафедри

д.т.н., проф._____/Олексій БИЧКОВ/

Київ – 2021

Рішенням Екзаменаційної комісії

випускна кваліфікаційна робота студента

_____

захищена з оцінкою

_____

Голова Екзаменаційної комісії

професор, доктор техн. наук Вишнівський В.В

РЕФЕРАТ

Записка: 42 стор., 31 рис., 2 табл., 1 додаток, 16 джерел.

Об’єкт дослідження — процес дистанційного керування ПК.

Мета роботи — розробити telegram-бот для дистанційного керування ПК зі смартфону.

Методи дослідження — метод емпіричного дослідження.

Результати — створено телеграм бот для дистанційного керування персональним комп’ютером, який має наступні функції: прості маніпуляції з файлами; вимкнення та перезапуск комп’ютера; введення пристрою у режим сну; налаштування гучності; налаштування яскравості монітора; відкривання посилань у браузері; використання гарячих клавіш.

Записка: 42 СТР., 31 рис., 2 табл., 1 приложение, 16 источников.

Объект исследования - процесс дистанционного управления ПК.

Цель работы - разработать telegram-бот для дистанционного управления ПК со смартфона.

Методы исследования - метод эмпирического исследования.

Результаты - создано телеграмм бот для дистанционного управления персональным компьютером, который имеет следующие функции: простые

манипуляции с файлами; выключения и перезапуск компьютера; введение устройства в спящий режим; регулировки громкости; настройка яркости монитора открытие ссылок в браузере; использование горячих клавиш.

Note: 42 pages, 31 figures, 2 tables, 1 appendix, 16 sources.

The object of research is the process of remote control of a PC.

The purpose of the work is to develop a telegram bot for remote control of a PC from a smartphone.

Research methods - a method of empirical research.

Results - created a telegram bot for remote control of a personal computer, which has the following functions: simple file manipulation; shutting down and restarting the computer; putting the device to sleep; volume adjustment; monitor brightness adjustment; opening links in the browser; use hotkeys.

ТЕЛЕГРАМ БОТ, ДИСТАНЦІЙНЕ КЕРУВАННЯ ПЕРСОНАЛЬНИМ КОМП’ЮТЕРОМ, NODE.JS, ПОРІВНЯЛЬНИЙ АНАЛІЗ

ЗМІСТ

ВСТУП

Ми живемо в період, коли важко уявити своє існування без технологій. Вони зустрічають нас всюди і допомагають на кожному кроці, особливо, месенджери, які значно полегшити процес листування та передачі інформації. Тепер людина може сповістити адресата за лічені хвилини за допомогою всього декількох кліків. Іншими словами, месенджери стали своєрідним символом розвитку сучасного світу. Одним із найпопулярніших месенджерів є Телеграм, який створив Павло Дуров. Він посідає перші позиції серед всіх рейтингів, оскільки є зручним, анонімним та одним із найбезпечніших. Як підтвердження, можна навести велику кількість опитування, де

користувачі погоджувались із вищезгаданими властивостями Телеграму. Проте, є ще одна перевага в Телеграмі, яка виділяє його серед інших – це боти, які можна легко створювати та вільно користуватись.

Боти – це керовані програмами акаунти, які допоможе у дуже багатьох процесах, наприклад, покупки квитків на літак, пошук літератури чи фільму, який хочеться переглянути. Однак, у мене є бажання створити унікальний бот, за допомогою якого буде можливість керувати ПК.

Багато людей користуються комп’ютером з ціллю перегляну фільмів, серіалів чи відео, проте часто появляється якась реклама чи просто потрібно перемкнути далі. На мою думку, значно зручніше збільшити звук, регулювати яскравість екрану чи шукати щось у пошуку на відстані. Отже, в мене появилось бажання створити бота, що допоможе у цьому.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Аналіз існуючих реалізацій RC telegrambot

Перед початком роботи було проаналізовано наявність схожих ботів, які допомогаються дистанційного керувати комп'ютером. Один із найпопулярніших - RC (Remote Control) telegram bot, який можна завантажити на сайті (Рис. 1.1)[11].

Даний бот має декілька особливостей:

• Здійснювати керування можуть лише ті користувачі, який обрали в програмі;

• У програми є можливість роботи у фоновому режимі;

• Простий, зручний та зрозумілий інтерфейс;

• Велика кількість функцій;

• У разі втрати зв'язку, є можливість повторного автоматичного підключення.

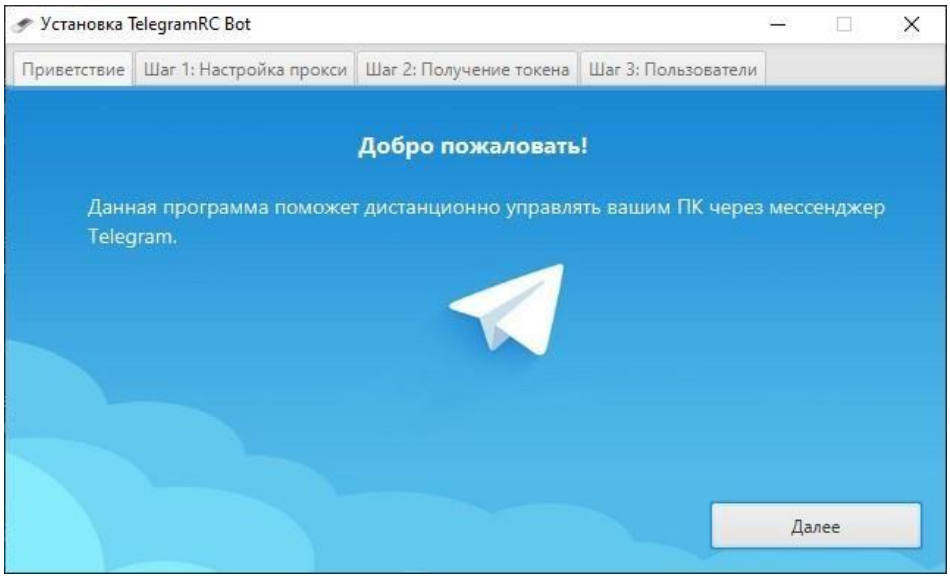

Рисунок 1.1 – Remote control telegram-bot

Для того, щоб була можливість користування програмою, необхідно створити бота.

Алгоритм створення бота у месенджері telegram:

1. звертання до боту @BotFather

2. після цього необхідно написати команду / newbot;

3. далі потрібно ввести ім’я, а потім нік бота, який мусить закінчуватись на bot;

4. після виконаних дій, бот надсилає token, який необхідно ввести у вікні установки програми (Рисунок 1.2);

5. для того, щоб була можливість керування програмами, необхідно ввести свій нік у списку користувачів, яким надається дозвіл.

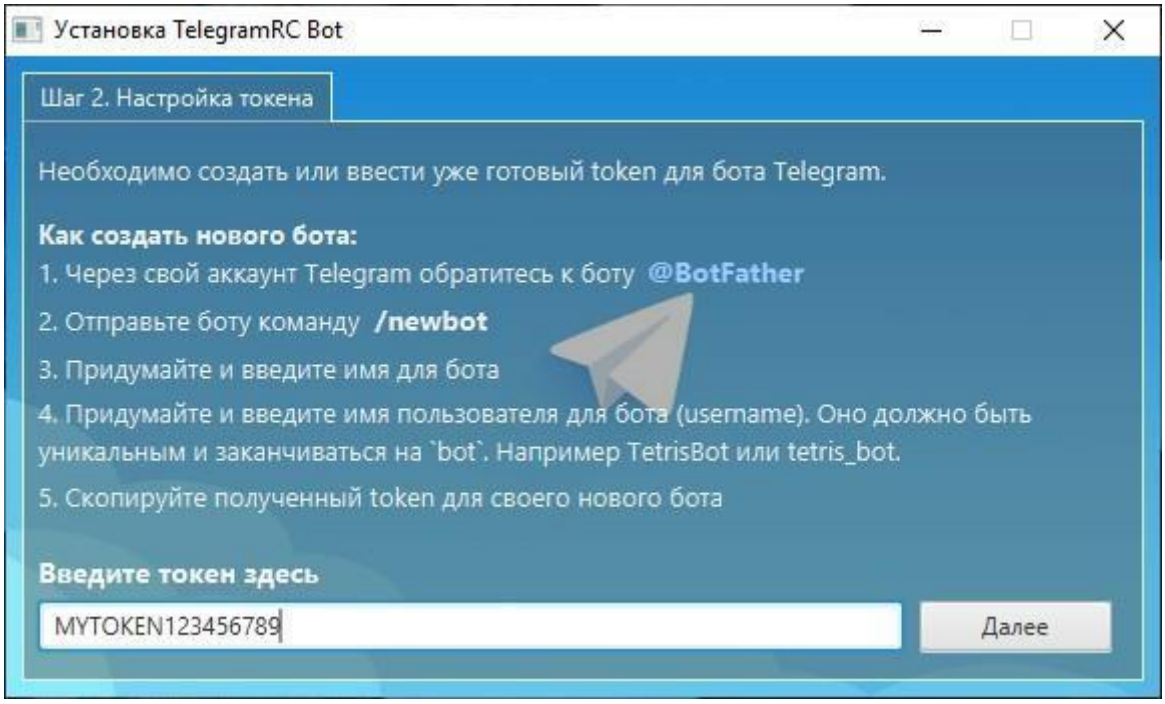

Рисунок 1.2 – Вікно встановлення програми

Керування ботом

У бота є можливість приймати команди зі стелешем на початку, а також без нього. Наприклад, / help і help – це однакова команда. Для аргументів теж є декілька способів –через знак пробіл, але у разі наявності пробілу в аргументі до того, його

варто обрамити в подвійні лапки; Для того, щоб зробити програму клікабельною замість пробілу потрібно написати "__", тобто два підряд нижніх підкреслювання.

Опис команд

Загальні команди:

/ start - Привітання бота

/ help – Довідка, описання команд та показ клавіатури для керування.

/ whoami - Демонстрація імені користувача, який на разі використовує програму і його id

/ ip - демонстрація зовнішнього IP, Geo-IP інформації та MAC адреси

/ browse [url] – можливість відкривання посилання, що починається з http і https в браузері за замовчуванням

/ alert [message] - показ повідомлення на ПК

/ uptime – інформує про час, який витратився на роботу програми і для системи windows

ПК система

/ systeminfo - показує інформацію про систему

/ exec [cmd] - здійснює системну команду, наприклад, / exec "cmd.exe / c ping 8.8.8.8" файлова система

/ cd – інформує про шлях до поточної директорії / Cd [path] - змінює директорію. Окрім того, в [path] може бути абсолютний, а також відносний шляхи.

/ ls - показ вмісту поточної директорії.

Окремо виділяють короткі команди, які необхідні для переходу до іншої директорії або файлу.

/ ls [path] [selectBy = name] - Показує вміст зазначеної в [path] директорії.

/ download [path] [selectBy = name] - завантаження файлу, тобто автоматично бот відправляє файл у діагол

/ delete [path] [selectBy = name] - видалення файлу з ПК.

/ print - відправлення файлу на друкування, проте для цього необхідно підключити принтер за замовчуванням, а також треба, щоб був встановлений Microsoft Word

Таймери

Таймери необхідні для того, щоб можна було контролювати час та ставити для виконання певного завдання виділений проміжок часу.

/ timers - пропонує список таймерів, які зараз працюють

/ timer [after] [command] – можливість створення нового таймеру.

[after] – проміжок часу, який виділяється на виконання команди; наприклад, 1s, 1m 10s, 1h 10m 20s.

[command] – назва або умова команди. Можна писати тексти команди використовуючи пробіли. Наприклад, / timer 30s / alert Hello World!

Media

/ screens - демонструє список екранів, а також їхнє розташування. Окрім цього, показує кнопки для створення знімків екрану.

/ screenshot [n = 0] – зробити знімок екрану, проте необхідно вказати номер екрану. Якщо цього не буде зроблено, тоді для створення скріншоту обереться екран за замовчуванням.

/ cameras - показує списоки web-камер, які можна використовувати, а також кнопки для створення знімків.

/ photo [n = 0] - зробити знімок з камери, однак тут також треба вказати номер камери з якого бажають зробити фото. У разі відсутності даної вказівки, знімок буде здійсненний з камери, що обрана за замовчуванням

/ volume - показує рівень гучності від 0 до 100. Проте, на жаль, дана функція працює лише на деяких пристроях.

/ volume [level] - встановлює рівень гучності. Значення [level] може бути від 0 до 100 або "up", "down", "+", "-".

/ brightness – контролює рівень яскравості від 0 до 100, однак також працює лише на деяких пристроях

/ brightness [level] - встановлює рівень яскравості. Значення [level] може бути від 0 до 100.

Клавіатура

/ keyboard - показ клавіатури з деякими клавішами і поєднаннями.

/ key [code] - натиснути клавішу (можна передати кілька - поєднання клавіш). Наприклад, / key back_space, / key windows a, / key control shift escape

/ key [media_code] – натискання на деякі медіа клавіші, які працюються на Windows, наприклад, play, stop, next, prev

/ media - відображає клавіатуру з медіа-клавішами.

Hardware

/ ram - показує оперативну пам'ять, яка зайнята ботом. У разі використання ОС Windows, тоді показує кількість всієї вільної та RAM на ПК.

/ battery – показує інформацію про акумулятор

/ temperature – показує інформацію з датчиків температури (якщо такі є і якщо є права доступу до них) живлення.

/ reboot – здійснює перезавантаження ПК.

/ shutdown - вимикає ПК.

Перед вимиканням або перезавантаженням ПК висвітлиться повідомлення з таймером (Рисунок 1.3).

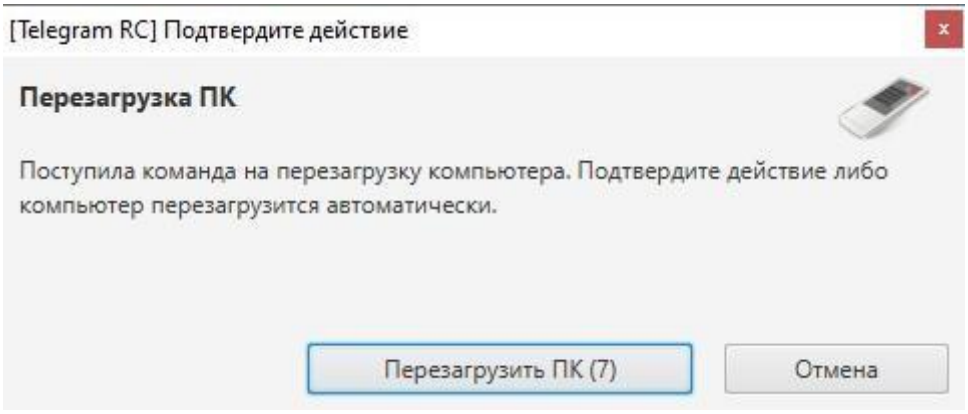

Рисунок 1.3 – Повідомлення з таймером

TeamViewer

TeamViewer – це програма, яка розрахована для віддаленого керування і проведення конференцій. Окрім того даний додаток має ще велику кількість функцій, наприклад:

• підтримка колег, друзів або клієнтів, які перебувають на відстані;

• з’єднання між комп’ютером з різноманітними ОС, адже додаток може працювати з системами Windows, macOS, Linux або Google Chrome OS;

• адміністрування серверів Windows і робочих станцій;

• Системна служба Windows, яка надає доступ до комп’ютера, щоб можна було здійснити вхід в Windows;

• Дозволяє підключитись з мобільних пристроїв Android, iOS,

Windows 10 Mobile або BlackBerry до пристроїв Windows, Mac або Linux.

• Під час конференцій дає можливість отримати доступ до свого робочого столу з метою демонстрації чи спільної роботи;

• Дає можливість входу до домашнього ПК, щоб можна було попрацювати з документами, зайти на електронну пошту чи завантажити, а згодом редагувати фото;

• Дозволяє підключення до ПК з ціллю отримання інформації;

• Окрім отримання доступу до пристроїв Android і iOS, може надати підтримку;

• Дає можливість віддаленого спостереження за активами, використовуючи перевірки стану інтегрованої системи і ITbrain;

• Не вимагає ніяких налаштувань, а також може працювати через брандмауери, маршрутизатори NAT і проксі;

Загальний опис

При з’єднанні даної програми з телефонним дзвінком буде використовуватись номер телефону, тобто він буде відповідати ID TeamViewer. За номер ми можемо окремо шукати користувачів. Однак при використанні програми через комп’ютери та мобільні пристрої одразу генерується індивідуальний ID, який створюється автоматично і згодом не змінюється. Важливо те, що він є доволі сильно захищений і не дає доступ третім особам.

Опис головного вікна

Нам одразу видно, що головне вікно має дві частини: вкладка віддалене керування і вкладка конференція (Рисунок 1.4).

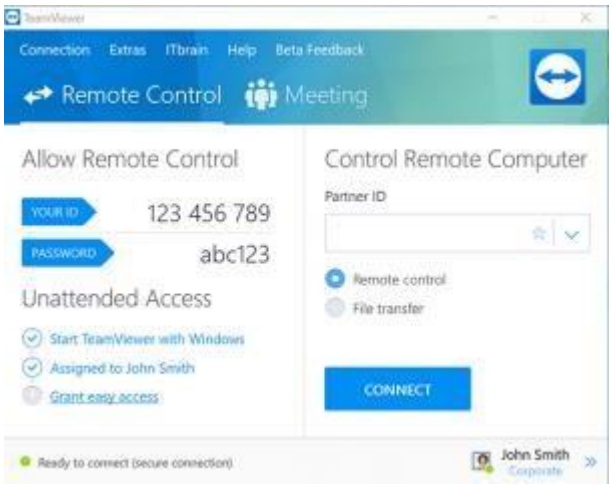

Рисунок 1.4 – Головне вікно

Вкладка «Віддалене керування»

У свою чергу, «Віддалене керування» також має декілька складових:

Дозволити керування

У області «дозволити керування» можна помітити ID, а також тимчасовий пароль, які можна надати партнеру з метою спільного підключення до одного комп’ютера. (Рисунок 1.5).

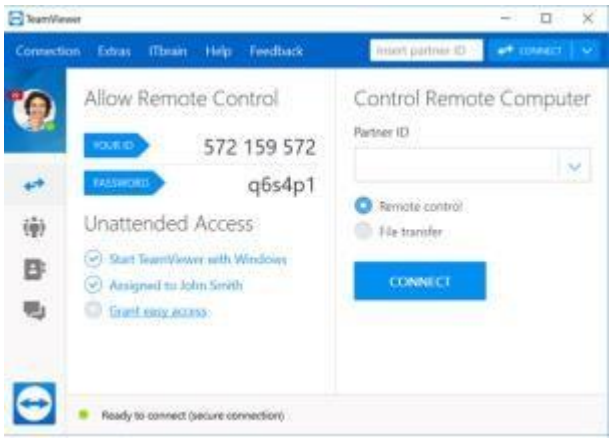

Рисунок 1.5 – Вкладка дозволити керування

Після того, як натиснути на піктограму в полі Пароль, ми одразу бачимо, що відкривається контекстне меню, яке дозволяє нам змінити пароль на випадковий або з ціллю копіювання паролю в буфер обміну, іншими словами «копіювати і вставити». Звичайно, є можливість сформувати пароль, який бажає користувач.

Якщо перейти до діалогового вікна Автоматичний доступ, тоді ми зможемо включити функцію, щоб програма автоматично запускалась при запуску Windows. Окрім цього, є можливість прив'язання пристрою до свого облікового запису і також обрати свій власний пароль. Після виконаних дій тепер користувач може отримати доступ до власного комп’ютера із різноманітних пристроїв за допомогою TeamViewer.

Керування комп'ютером

Для того, щоб користувач зміг керувати комп’ютером на відстані, потрібно ввести ID користувача в поле ID партнера.

Після підключення можна побачити, які режими з ціллю з’єднання доступні:

• Керування на відстані: контроль чи керування комп’ютером

• віддалене керування: керування комп'ютером партнера або спільна робота на одному комп'ютері;

• передача файлів з одного комп’ютера на інший;

• VPN: створення приватної мережі з іншим користувачем.

Після проведення детального аналізу, можна сказати, що TeamViewer справді є однією із найкращих програм, що дозволяє дистанційне керування комп’ютером. Однак, мова йде саме про користування додатком з комп’ютера з метою керування іншого комп’ютера. Якщо використовувати TeamViewer з мобільного додатку, то ситуація дещо гірша:

• Незручний інтерфейс, що ускладнює керування;

• Внаслідок того, що додаток не підлаштовується під телефон, то програма відкривається не на повний екран, що затрудняє відкриття інших додатків, які необхідні;

• Спостерігається постійне зависання картинки на телефоні.

1.2 Постановка задачі

Після того, як провели дослідження інших програм, які використовуються задля дистанційного керування комп’ютером, можна виокремити певні деталі, якими мала би володіти майбутня програма, що розробляється:

• Маніпуляції з файлами повинні бути простими та зрозумілими;

• Має бути можливість вимикати та перезавантажувати комп’ютер, а також вводити його в стан сну;

• Наявність функцій, що змогли би регулювати гучність та яскравість монітора;

• Програма має дозволяти відкривати різноманітні посилання у браузері;

• Можливість використання гарячих клавіш.

Отже, підсумовуючи вищезгадане, можна виділити декілька основних завдань:

1. Відшукати нові способи для того, щоб реалізувати поставлені задачі, тобто вибрати необхідні інструменти;

2. Створити телеграм бота, щоб реалізувати завдання на практиці;

3. Здійснити тестування телеграм бота.

2 ВИБІР ІНСТРУМЕНТІВ

2.1 Пошук можливих способів реалізації.

Існує декілька шляхів, які можна використати з метою реалізування функціональності телеграм бота. Наприклад, можна використати Microsoft Visual Studio.

Як мову програмування, можна використати C # або Node.js., адже це буде найраціональніше, оскільки не потрібно буде писати з нуля велику кількість кодів, а можна буде скористатись готовими шаблонами (Bot Application, Bot Framework Emulator, Bot Dialog і Bot Controller). Інформацію, яка нам необхідна для встановлення та те, як користуватись, можна знайти у документах Microsoft Bot Framework. Отже, необхідно тільки здійснити налаштування шаблони під власні вимоги. Варіант, що описаний вище ідеальний, однак він підходить лише для користувачів Windows.

Існує інший спосіб, щоб реалізувати Telegram бот і це – PHP, який теж характеризується наявністю готових бібліотек для роботи з Telegram. Яскравим прикладом є Telegram Bot SDK, що зводить всі зусилля до мінімального рівня.

Однак, не варто забувати і про ще одну мову програмування, яка надзвичайно швидко та без зайвих проблем дозволить створити бота в Telegram, а саме – Python. Дана мова доволі сильно користується попитом, оскільки дозволяє користуватись, як стандартними бібліотеками, так і спеціалізованими для роботи саме в Telegram, наприклад, PyTelagramBotAPI.

2.2 Python

Для роботи в телеграмі створена спеціальна бібліотека, а саме python-telegram-bot, яка гарантує чистий інтерфейс Python для API Telegram Bot. Також, не варто забувати, що він сумісний з версіями Python 3.5+ та PyPy.

Існує ще велика кількість переваг, однак одна із найсуттєвіших це те, що дана бібліотека включає в себе ряд класів високого рівня, які дозволяють зробити процес

створення телеграм боту легким, невимушеним та зрозумілим. Всі ці класи знаходяться на підмодулі telegram.ext

Встановлення:

Для того, щоб встановити бібліотеку на власний комп'ютер, потрібно ввести код, що вказаний нижче: (Рис. 2.1).

```
$ pip install python-telegram-bot --upgrade
```

Рисунок 2.1 – Код завантаження бібліотеки

Окрім того, якщо користувач бажає, то він може здійснити встановлення з репозиторію на сайті (Рисунок 2.2).

```
$ git clone https://github.com/python-telegram-bot/python-telegram-bot --recursive
$ cd python-telegram-bot
$ python setup.py install
```

Рисунок 2.2 – Альтернативний варіант завантаження

Відомо, що бібліотека включає в себе велику кількість необхідних та корисних модулів:

Наприклад, telegram.ext.CommandHandler, який використовується, щоб обробити всі команди, які були надіслані боту.

Варто сказати, що команди - це певні повідомлення в телеграмі, котрі починаються з /, вводяться після @ та імені бота або додатковим текстом. Після цього, обробник добавить список до CallbackContext з назвою CallbackContext.args. Тоді, він буде мати списки, що будуть текстом, що слідує за командою, яка розділена на окремі або послідовні символи пробілу.

Обробник має властивість за замовчування прослуховувати всі повідомлення, а також повідомлення, що були відредаговані. У разі, якщо користувачу не підходить дана поведінка, він може її змінити, використавши Filters.update.edited_message у аргументі filter.

Параметри:

• command (str | List[str]) - команда або список команд, які даний обробник необхідно виконувати;

• callback (callable) –буде викликано, у разі якщо check_update визначить, що модулю необхідне оновлення;

• pass_args (bool, optional) – визначає чи варто передавати аргумент команді. Аргумент – це ключове слово, яке позначається «args»;

• allow_edited (bool, optional) – визначання чи зможе обробник прийняти виправлені повідомлення.

telegram.InlineKeyboardButton – модуль для створення інлайн клавіатури

(клікабельних кнопок).

Параметри:

• text (str) – маркування на кнопці;

• url (str) – посилання, що відкривання у разі натискання на кнопку;

• login_url (telegram.LoginUrl, optional) – необхідно для авторизації користувача;

• callback_data (str, optional) – інформація, яка надсилається боту при натисканні на кнопку;

• switch_inline_query (str, optional) – дозволяє відкрити будь – який чат, щоб вставити в нього поле і ввести повідомлення юзернейм бота і певний запит для вбудованого режиму, у разі, якщо до цього був включений даний параметр;

Проте, якщо нічого не написати та відправити звичайне пусте поле, тоді буде вставлений тільки юзернейм бота**.** Дана функція необхідна для того, щоб була можливість швидкого перемикання між діалогами з ботом та режимом, що вбудований у цьому ж боті;

• switch_inline_query_current_chat (str, optional) – така ж функція, проте без вибору чату.

2.3 Node.js

Node або Node.js – це спеціалізована платформа, що базується на V8, що здійснює трансляцію JavaScript в машинний код, який перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення.

Дана платформа дозволяє здійснити взаємодію з пристроями введення-виведення через свій API написаний на C ++, а також допомагає підключити інші бібліотеки, котрі написані різними мовами, забезпечити виклики до них з JavaScript-коду.

Відомо, що Node.js використовується зазвичай на сервері, оскільки виконує роль веб-сервера, проте є можливість створити на Node.js і десктопні віконні додатки.

В основі Node.js можна помітити подієво-орієнтоване програмування, яке характеризується тим, що виконання програми визначається подіями, тобто власне діями користувача (клавіатура, миша), повідомленнями інших програм і потоків, подіями операційної системи, наприклад надходженням мережевого пакету.

Асинхронне, або реактивне програмування, тобто властивість виконувати дії паралельно, щоб не чекати завершення попередньої .

Для здійснення даного проекту мені необхідні також ще бібліотеки, такі як node-telegram-bot-api, brightness та robotjs.

Node-telegram-bot-api – це основна бібліотека, на якій буде базуватись весь проект. Дана бібліотека використовує декілька підходів, що дозволяють створити доволі складних ботів, однак із простим кодом.

Нижче пропоную розглянути декілька найцікавіших механізмів:

Middlewares – під час того, як коли користувач щось відправляє боту, ми помічаємо, що повідомлення не одразу проникає в контролер, а спочатку проходить через всі зареєстровані middlewares. Middlewares мають властивість можуть модифікації контексту, а також припинення виконання запиту. Яскравим прикладом є ситуація, коли користувач намагається потрапити до області, що призначена адміністраторам.

• Sessions - сесії, які можуть зберігати в собі інформацію, не прив'язану

до контролера. Можна сказати, що це схоже із глобальними змінними, які доступні з будь-якого місця в боті. Надзвичайно зручна річ, якою можна користуватись для локалізації. Також, спостерігається можливість зберігати сесії в різних режимах - БД, redis, локальні файли і т.д.

• Webooks – цей бот має властивості, що дозволяють працювати в двох режимах - long polling або Webhooks. Обидва вони характеризуються швидкою роботою, проте краще використовувати другий варіант. Отже, бот сам буде отримувати всі оновлення. Варто сказати, що обов'язковою умовою для Webhook є підтримка сервером SSL / TLS. Також треба переконатися, що порт, на якому запущений Webhook, відкритий і доступний ззовні.

• Markup – даний клас може навчити бота відповідати в markup / markdown розмітки.

• Stage – Даний бот може визначати тип повідомлення та одразу перенаправляти його в той чи інший контролер, який виконує свою функцію і відправляє відповідь користувачеві.

Brightness – бібліотека, яка дозволяє керувати налаштуваннями яскравості екрану.

Robotjs – бібліотека, що відповідає за керування мишкою та клавіатурою. Для клавіш нам знадобиться наступна функція (Таблиця 2.1)

Таблиця 2.1 – keyTap(key,[modefier]).

Аргумент	Описання
Key	Кнопка (Таблиця 2.2)
Modified	Модифікація кнопки (shift, ctrl, enter)

Таблиця 2.2 – Список можливих для натискання кнопок

Кнопка	Описання	Замітки

Backspace	
Delete	
Enter	
Tab	
Escape	
Up	Стрілка вгору
Down	Стрілка вниз
Right	Стрілка вправо
Left	Стрілка вліво
Home	
End	
PageUp	
PageDown	
F1	
F2	
F3	
F4	
F5	
F6	
F7	
F8	
F9	
F10	
F11	
F12	
Command	

Продовження таблиці 2.2

Alt	

Control		
Shift		
Right_shift	Правий шифт	
Space		
Printscreen	Зробити скріншот	Не підтримується на Мак ОС
Insert		Не підтримується на Мак ОС
Audio_mute	Вимкнути звук	
Audio_vol_down	Зменшити звук	
Audio_vol_up	Збільшити звук	
Audio_play	Грати	
Audio_stop	Вимкнути	
Audio_pause	Пауза	
Audio_prev	Попередній трек	
Audio_next	Наступний трек	
Audio_rewind	Відмотати трек	Тільки на Лінукс
Audio_forward	Перемотати трек	Тільки на Лінукс
Audio_repeat	Повторити трек	Тільки на Лінукс
Audio_random	Випадковий трек	Тільки на Лінукс
Numpad_0		Не підтримується на Лінукс
Numpad_1		Не підтримується на Лінукс
Numpad_2		Не підтримується на Лінукс

Продовження таблиці 2.2.

Numpad_3		Не підтримується на Лінукс
Numpad_4		Не підтримується на Лінукс
Numpad_5		Не підтримується на Лінукс
Numpad_6		Не підтримується на Лінукс
Numpad_7		Не підтримується на Лінукс
Numpad_8		Не підтримується на Лінукс
Numpad_9		Не підтримується на Лінукс
Lights_mon_up	Збільшити яскравість монітора	Ні підтримується на Віндоувс

Отже, на мою думку, найкраще для реалізації даного проекту використати програмну платформу Node.js

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Підготовка комп’ютера

Для початку, необхідно встановити Node.js. Для здійснення цього, потрібно зайти на офіційний сайт [16]. Після цього вибрати необхідну операційну систему та її розрядність і опісля почати завантаження програми (Рисунок. 3.1).

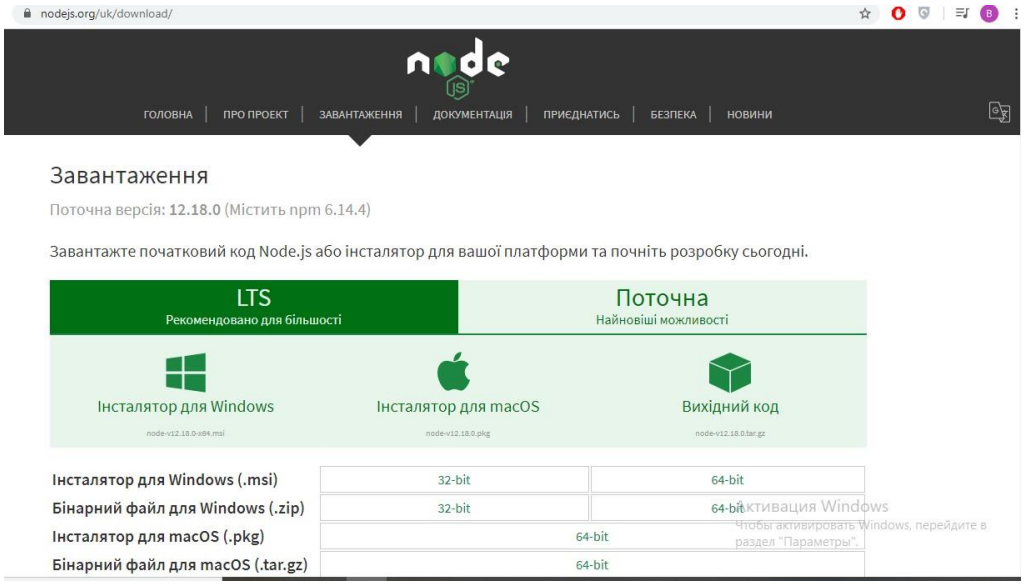

Рисунок 3.1 – Завантаження програми

Після цього встановлюємо Node.js (Рисунок 3.2-3.3).

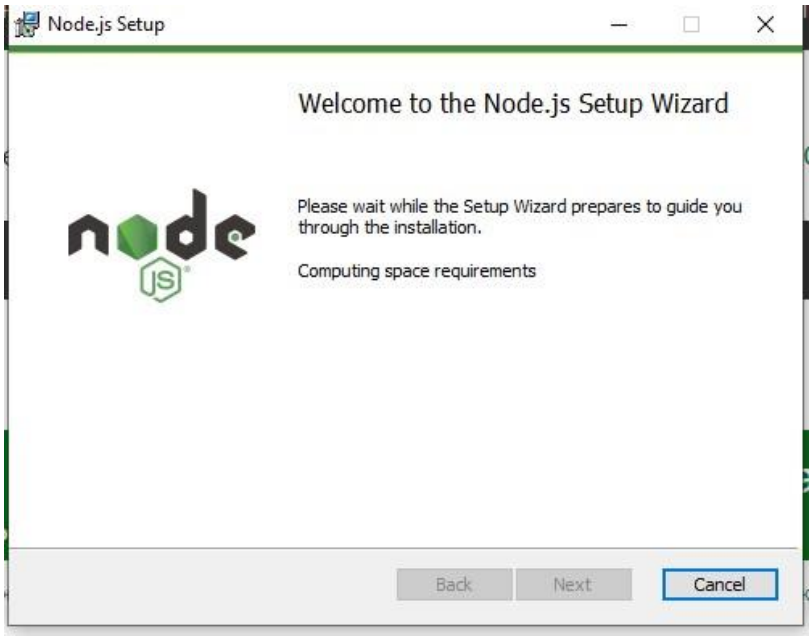

Рисунок 3.2 – Встановлення програми

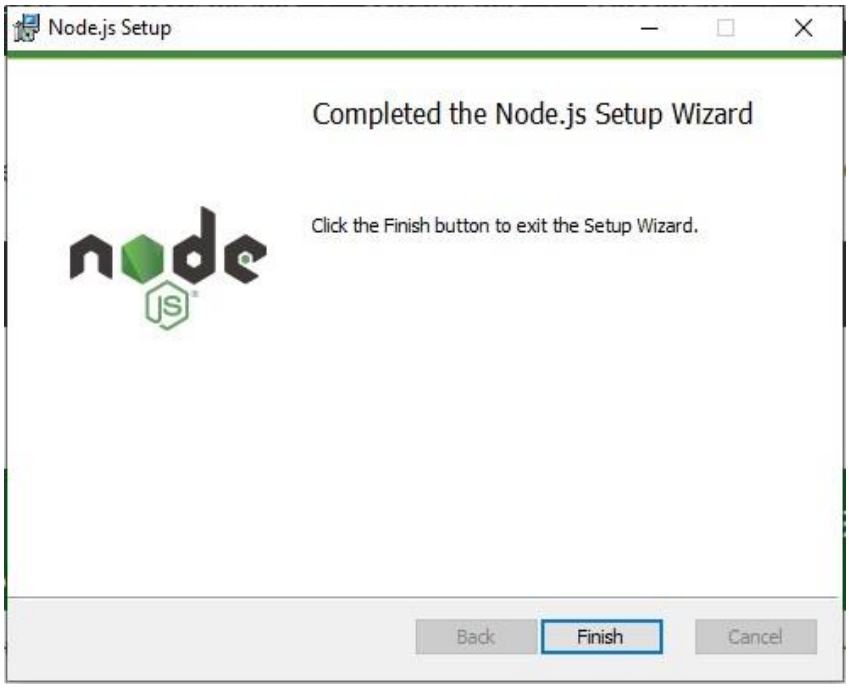

Рисунок 3.3 – завершення встановлення

Для подальшої роботи, необхідно також встановити бібліотеку. Для цього необхідно відкрити консоль і зайти у папку, в якій знаходиться проект. Після цього, потрібно ввести команду (Рисунок 3.4).

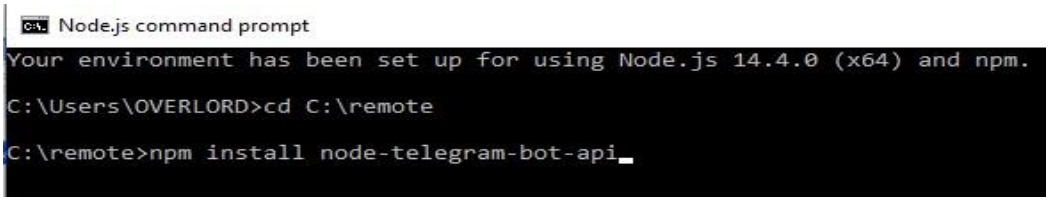

Рисунок 3.4 – Встановлення бібліотеки

У разі, коли все пройшло так, як зображено на картинці, нам пропонується наступне (Рис. 3.5).

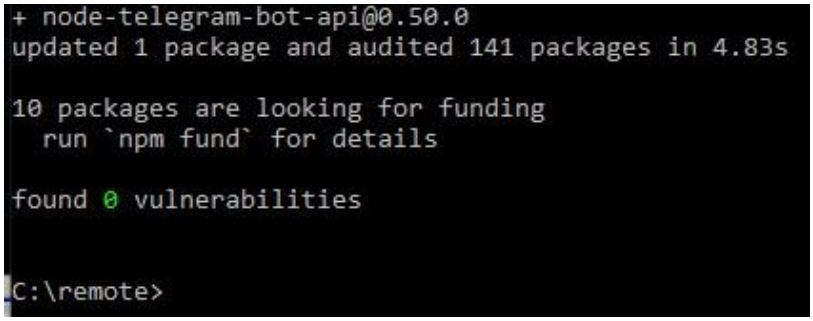

Рисунок 3.5 – Завершення встановлення

3.2 Створення бота

Для того, щоб бот міг функціонувати, його необхідно створити. Як уже говорилось вище, для цього в Телеграмі є спеціальний мета-бот BotFather (@BotFather), який легко знайти в пошуку. Його потрібно додати в клієнти Телеграма.

Щоб ознайомитись із списком всіх команд, що він пропонує потрібно написати в чаті з ним команду /help. Однак, у нас завдання створити нового бота. Дану дію можна здійснити, написавши команду /newbot і тоді у наступному повідомленні передати назву бота та після написати його нікнейм, який будуть користуватись люди, що будуть його шукати. Після цього, як відповідь прийде повідомлення з API ключем (Рисунок 3.6).

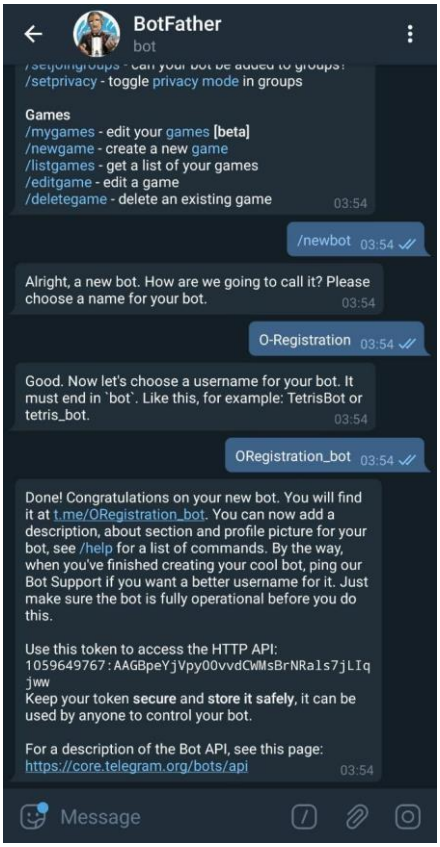

Рисунок 3.6 – Створення бота

API ключ, який отримали, необхідно ввести у програму (Рисунок 3.7).

```
var TelegramBot = require ('node-telegram-bot-api');
var token = '[redacted]';
var bot = new TelegramBot (token, {polling: true});
```

Рисунок 3.7 – Введення API ключа

Для того, щоб дізнатись чи працює бот можна створити тестовий метод (Рисунок 3.8)

```
bot.onText (/test/, function (msg, match) {
    var fromId = msg.from.id;
bot.sendMessage(fromId, 'Тест пройдено!');
});
```

Рисунок 3.8 – Створення тестового методу

Після цього ми використовуємо функцію onText, щоб створити метод, яким буде користуватись бот, щоб одразу при отриманні повідомлення test відправляти

користувачу повідомлення: «Тест пройдено!» . Для цього необхідно відкрити дужки та записати код. Для початку потрібно ввести регулярний вираз /test/, а після того, як бот отримає дане повідомлення, тоді функція повинна вже працювати.

Після цього потрібно створити змінну fromId, а у дану змінну зберегти id користувача, який перед цим надіслав повідомлення. Для того, щоб надіслати саме повідомлення, потрібно використати функцію sendMessage. Щоб зробити дану дію, необхідно записати спочатку id користувача, у нашому випадку воно знаходиться у змінній fromId, а потім після коми у лапках повідомлення. Тепер можна зберегти і запустити програму через консоль (Рисунок 3.9).

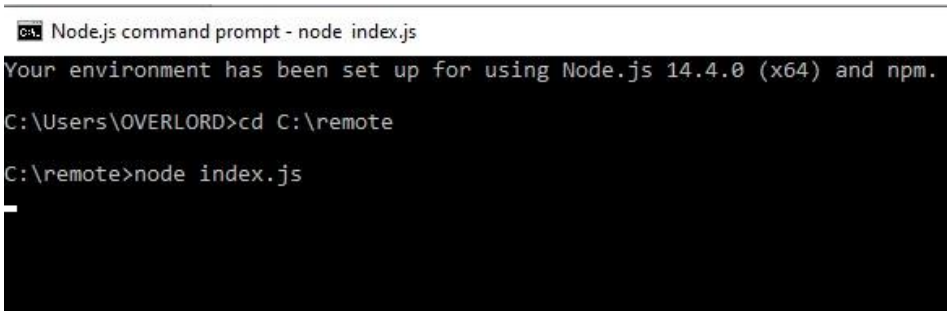

Рисунок 3.9 – Запуск програми через консоль

Перевірка програми (Рисунок 3.10).

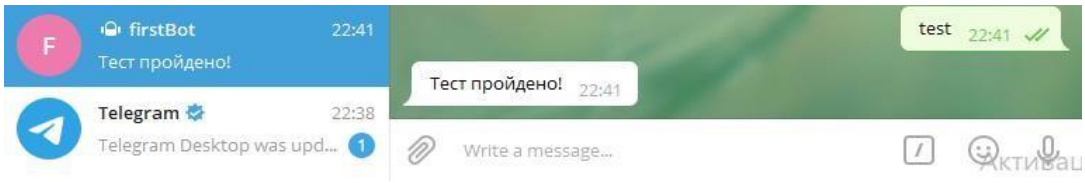

Рисунок 3.10 – Перевірка команди

3.3 Додавання функцій

Маніпуляції з файлами

Для того, щоб створити функцію, яка буде видаляти файли, необхідно навчити бота працювати з системною консоллю. З цим допоможе модуль childProcess.

Підключення відповідного модуля до програми (Рисунок 3.11).

```
var exec = require('child_process').exec;
```

Рисунок 3.11 – Підключення модуля childProcess

Після цього вводимо наступний код (Рисунок 3.12).

```
bot.onText (/del (.+)/, function (msg, match) {
    var fromId = msg.from.id;
    exec('del '+match[1]);
bot.sendMessage(fromId, 'Файл "'+match[1]+'"видалено!');
});
```

Рисунок 3.12 – Функція видалення файлів

У регулярному виразі необхідно написати del (.+). Після того, як бот отримає повідомлення, що починається з “del “, то зайве він збереже у масив match.

Після цього потрібно використати функцію exec**.** В дужках формується запит, який бот буде відправляти консолі, тоді як вкінці буде повідомлятись про те, що даний файл потрапив під видалення.

Тепер напишемо схожу команду для копіювання файлів (Рисунок 3.13).

```
bot.onText (/copy (.+) (.+)/, function (msg, match) {
    var fromId = msg.from.id;
    exec('copy '+match[1]+' '+match[2]);
bot.sendMessage(fromId, 'Файл "'+match[1]+'"скопійовано в "'+match[2]+'".');
});
```

Рисунок 3.13 – Функція копіювання файлів

Також можна написати код для перегляду файлів, які знаходяться у папці. Для цього знадобиться модуль fileStrim (fs).

Спочатку додається модуль до програми (Рисунок 3.14).

```
var bot = new TelegramBot (token, {polling: true});
var exec = require('child_process').exec;
var fs = require('fs');
```

Рисунок 3.14 – Підключення модулю fileStrim

Після цього прописується наступний код (Рисунок 3.15).

Рисунок 3.15 – Функція перегляду файлів

Спочатку бот читає повідомлення у якому вказано місце розташування папки: «bot.onText (/list (.+)/, function (msg, match)».

Після цього, він передає отриману інформацію у функцію:

«fs.readdir(match[1], function(err, files)».

Якщо папка була відсутня чи мала закритий доступ, тоді нам висвітлюється повідомлення про помилку. Для того, щоб такого не було, варто добавити ще такий варіант:

«if (err) bot.sendMessage(fromId, 'Ця папка не знайдена, або до неї нема доступу');
».

Після всіх дій, нам необхідно отримати список файлів та перетворити його у строку. Для цього пишемо:

var list = ''; files.forEach(function(file)
{ list += file + "\n";
});

Вимкнення, перезапуск та режим сну

За тим же принципом пишеться код (Рисунок 3.16).

Рисунок 3.16 – Функція перезапуску

Для того, щоб вимкнути консоль, варто написати: «shutdown /s /t 0».

Для введення у режиму сну: «Rundll32.exe powrprof.dll,SetSuspendState Sleep».

Налаштування гучності

У нас була мета також регулювати гучність і для того, щоб це можна було здійснити, необхідно завантажити програму з сайту [9], котра яка дає можливість робити це через консоль. Після завантаження програми, її необхідно розархівувати, щоб нею одразу можна було користуватись.

Прописавши у консолі: «setvol help», можна побачити усі можливості програми (Рисунок 3.17).

```
SetVol v2.2 Help

Options:
 ?                help

 n                set the master volume to n where n = 0 to 100

 +n               increase the master volume level by n where n = 1 to 100; maximum result = 100
 -n               decrease the master volume level by n where n = 1 to 100; minimum result = 0

 x                set the master volume level to x where x = "zero" to "one hundred" (without the quotes)

 align            make all channel volume levels equal the master volume level
                  most devices will have at least two channel levels
                  with the first channel representing the left speaker and the second channel representing the right speaker
```

Рисунок 3.17 – SetVol help меню

Після вищезгаданих дій, можна почати користування даними функціями:

• setvol n – встановити гучність на рівень n;

• setvol +-n – змінити рівень гучності на n;

• setvol align – вирівняти рівень гучності всіх пристроїв;

• setvol mute – вимкнути звук;

• setvol unmute – увімкнути звук.

За допомоги цього коду реалізуються усі ті команди (Рисунок 3.18).

```
bot.onText (/setvol(.+)/, function (msg, match) {
    var fromId = msg.from.id;
    exec('setvol'+match[1]);
bot.sendMessage(fromId, 'setvol'+match[1]);
});
```

Рисунок 3.18 – Функція регулювання звуку

Налаштування яскравості

Під час того, як здійснювалось дослідження, була знайдена бібліотека, щоб налаштувати яскравості екрану. Щоб розпочалось її завантаження, необхідно для початку написати в консолі: «npm install --save brightness».

Тепер з'являється можливість регулювати яскравість монітору цифрами від 0 до 1 (Рисунок 3.19).

```
bot.onText (/setbr (.+)/, function (msg, match) {
        brightness.set(match[1]).then(() => {
        console.log('Changed brightness to '+match[1]);
        });
    var fromId = msg.from.id;
bot.sendMessage(fromId, 'setbr'+match[1]);
});
```

Рисунок 3.19 – Функція налаштування яскравості

Робота з браузером

Одним із основних завдань проекту є створення бота, який міг би відправляти відкривати посилання у браузері. Для цього існує стандартний відомий модуль «childProcess». Якщо прописати у командній строці: «start http…», тоді ОС windows зреагує та відкриє посилання у браузері, який вибраний, як основний або за замовчуванням. Для того, щоб посилання відкривались, потрібно запрограмувати бота формувати запит «start» для усіх повідомлень, що починаються з «http» (Рисунок 3.20).

```
bot.onText (/http(.+)/, function (msg, match) {
    var fromId = msg.from.id;
    exec('start http'+match[1]);
});
```

Рисунок 3.20 – Функція відкривання посилань

Керування клавіатурою

Керування клавіатурою можна реалізувати за допомогою бібліотеки «robotjs», для встановлення якої необхідно прописати в консолі «npm install robotjs». Ця бібліотека корисна для нас однією функцією: robot.keyTap(key, [modifier])

Key – це кнопка, яку натискатиме бот. Усі вони відповідають своїм назвам.

Modifier – використовується з метою натискання комбінації клавіш.

Як вже говорилось, під час проекту планувалось реалізувати не лише використання клавіатури, але і гарячих клавіш, комбінацій, що успішно можна здійснити за допомогою Modifier .

Спочатку потрібно здійснити налаштування клавішей, котрі будуть необхідні для зручного користування сайту «you tube»:

• f – відкрити відео на повний екран;

• escape – згорнути відео;

• space – пауза та відтворення відео;

• shift + n – наступне відео;

• shift + p – попереднє відео та увімкнути спочатку.

У програмі це має наступний вигляд (Рисунок 3.21).

```
bot.onText (/p/, function (msg, match) {
    robot.keyTap("space");
});

bot.onText (/f/, function (msg, match) {
    robot.keyTap("f");
});

bot.onText (/esc/, function (msg, match) {
    robot.keyTap("escape");
});

bot.onText (/next/, function (msg, match) {
    //robot.keyToggle("shift");
    robot.keyTap("n",["shift"]);
});

bot.onText (/pr/, function (msg, match) {
    robot.keyTap("p",["shift"]);
});
```

Рисунок 3.21 – Функція керування клавіатурою

3.4 Тестування програми

Маніпуляції з файлами

Створюється декілька файлів (Рисунок 3.22).

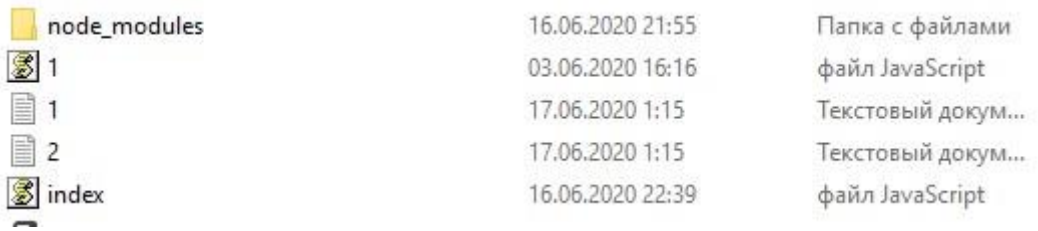

Рисунок 3.22 – Папка remote

Перевірка команди для видалення файлів (Рисунок 3.23)

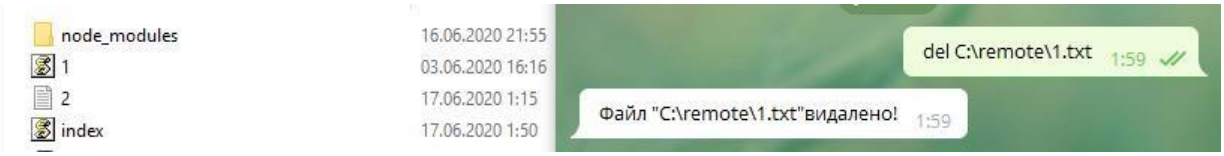

Рисунок 3.23 – Тестування функції видалення файлів

Ми можемо спостерігати те, що команда запрацювала.

Копіювання файлів. Створюється папка «est1» (Рисунок 3.24).

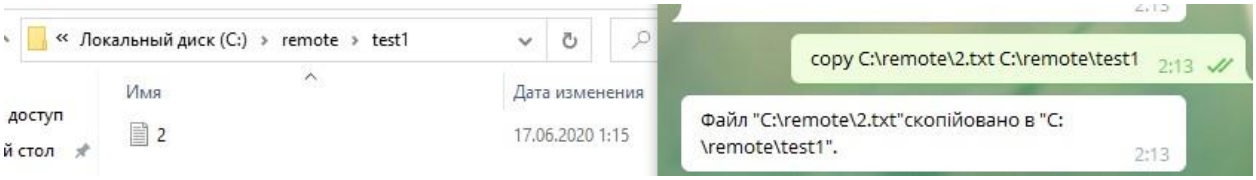

Рисунок 3.24 – Тестування функції копіювання файлів

Файл скопіювався у папку «test1».

Отримання списку файлів (Рисунок 3.25).

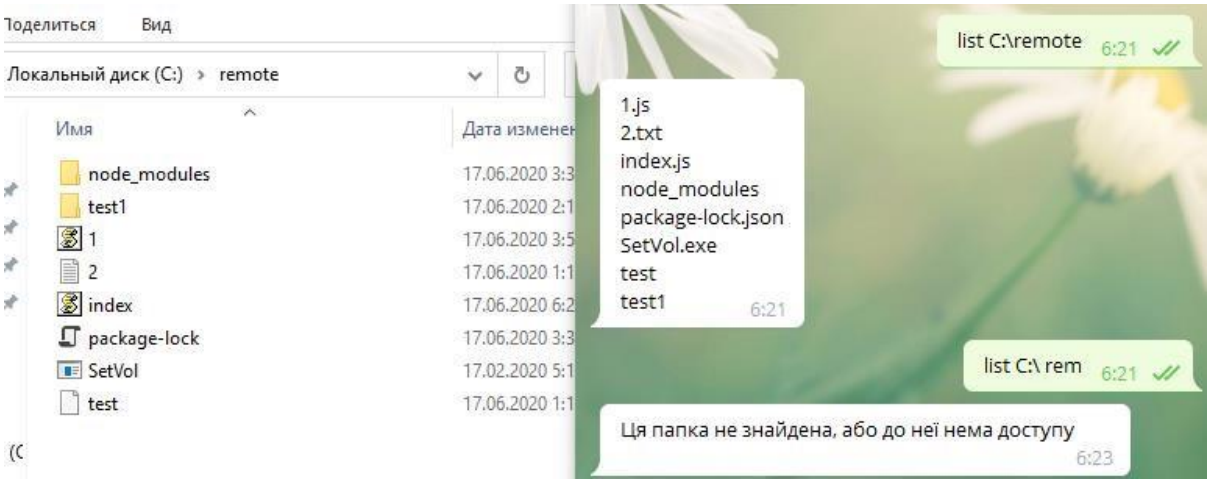

Рисунок 3.25 – Тестування функції перегляду файлів

Налаштування гучності

Боту успішно відправляються команди, які необхідні, щоб налаштовувати гучність (Рисунок 3.26).

Рисунок 3.26 – Тестування функції налаштування гучності

Налаштування яскравості

Перевірка програми (Рисунок 3.27).

Рисунок 3.27 – Тестування функції налаштування яскравості

Спостерігаючи за тестуванням, можна сказати, що дана програма працює, а отже, працює коректно.

ВИСНОВКИ

1. Аналіз, який проводився під час дослідження, показав, що технології розробки ботів в даний час надзвичайно сильно удосконалились та осучаснились. Відомо, що тепер є велика кількість можливостей, щоб створити програму, і що важливо, багато мов програмування підтримують telegram API. З кожним днем, стає все більше і більше бібліотек, які створюють розробники, щоб була можливість ще більше розширити функціонал ботів в телеграмі.

2. Незважаючи на те, що більшість програмістів при їх написанні більше користуються Python, на мою думку, інші мови такі, як Node.js та PHP не поступаються у своїх можливостях та мають певні переваги. Якщо знаєш декілька мов програмування, тоді легко підлаштуватись і виконати все згідно задоволення потреб. Після всіх досліджень, було обрано оптимальний інструмент реалізації, а саме Node.js.

3. Впродовж даного дослідження було створено бота в месенджері Телеграм, який включає в себе такі функції та відповідає вимогам, що ставились з самого початку:

Маніпуляції з файлами повинні бути простими та зрозумілими;

Має бути можливість вимикати та перезавантажувати комп’ютер, а також вводити його в стан сну;

Наявність функцій, що змогли би регулювати гучність та яскравість монітора;

Програма має дозволяти відкривати різноманітні посилання у браузері;

Можливість використання гарячих клавіш.

4. Під час тестування було виявлено, що створена програма є повноцінним пультом дистанційного керування для комп’ютера, яким можна користуватись

підчас прослуховування музики, перегляду фільмів та відео на телевізорі, використовуючи його як екран.

СПИСОК ЛІТЕРАТУРИ

1. Офіційний сайт Stack overflow: Інтернет-стаття. Посилання на джерело: https://ru.stackoverflow.com

2. Офіційний сайт Wibe.team: Інтернет-стаття. Посилання на джерело: https://wibe.team/sozdanie-bota-v-telegram/

3. MDN web docs: Інтернет-стаття. Посилання на джерело: https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Regular_Expr essions

4. Влад Скурішин стаття: «Telegram bot API і NodeJs». Посилання на джерело: https://medium.com/@vladthelittleone/telegram-api-%D0%B2-nodejsa82b0f7ec575

5. Telegram, документація створення телеграм бота. Посилання на джерело: https://tlgrm.ru/docs/bots/samples

6. Npm, репозиторій, документація бібліотеки robot.js. Посилання на джерело: https://www.npmjs.com/package/robotjs#examples

7. Github, репозиторій, бібліотека node-telegram-bot-api. Посилання на джерело: https://github.com/yagop/node-telegram-bot-api

8. Smmx стаття: «Створення ботів для телеграм на Python». Посилання на джерело: https://smmx.ru/telegram/how-to-use/bot-na-python.html

9. Rlatour.com, завантаження та опис програми SetVol. Посилання на джерело: https://rlatour.com/setvol/

10. Документація програми Team viewer. Посилання на джерело: https://dl.teamviewer.com/docs/ru/v13/TeamViewer13-Manual-RemoteControl-ru.pdf

11. Ts.Saltan tech blog завантаження та опис програми TelegramRC bot. Посилання на джерело: https://tssaltan.top/1928.telegramrc-bot/

12. Npm репозиторій, опис бібліотеки brightness. Посилання на джерело: https://www.npmjs.com/package/brightness

13. Github репозиторій з бібліотекою python-telegram-bot. Посилання на джерело: https://github.com/python-telegram-bot/python-telegram-bot

14. Документація бібліотеки python-telegram-bot. Посилання на джерело: https://python-telegrambot.readthedocs.io/en/stable/telegram.html

15. Документація бібліотеки python-telegram-bot, модуль inlinekeyboard. Посилання на джерело: https://pythontelegram.com

16. Офіційний сайт NodeJs. Посилання на джерело: https://nodejs.org/uk/

ДОДАТОК

Код програми

```
    const Promise =
require('bluebird');
Promise.config({   cancellation:
true
    }); process.env.NTBA_FIX_319 = 1; var
TelegramBot = require ('node-telegram-bot-api');
const Telegraf = require('telegraf'); var token
='******************************'; var bot = new
TelegramBot (token, {polling: true}); var exec =
require('child_process').exec; const brightness =
require('brightness'); const { Markup } =
require('telegraf'); var fs = require('fs'); var
robot = require("robotjs"); var options = {
reply_markup: JSON.stringify({     inline_keyboard:
[
          [{ text: 'Кнопка 1', callback_data: 'setvol 10'}],
          [{ text: 'Кнопка 2', callback_data: 'data 2' }],
          [{ text: 'Кнопка 3', callback_data: 'text 3' }]
        ]
      }) }; bot.onText(/\/start_test/, function (msg, match) {
bot.sendMessage(msg.chat.id, 'Выберите любую кнопку:', options);
    }); bot.onText (/p/, function (msg,
match) {
     robot.keyTap("space");
    }); bot.onText (/f/, function (msg,
match) {       robot.keyTap("f");
    }); bot.onText (/esc/, function (msg,
match) {
```

```
     robot.keyTap("escape");
    }); bot.onText (/next/, function (msg,
match) {       //robot.keyToggle("shift");
     robot.keyTap("n",["shift"]);
    }); bot.onText (/pr/, function (msg,
match) {
     robot.keyTap("p",["shift"]);
    }); bot.onText (/http(.+)/, function (msg,
match) {
     var fromId = msg.from.id;
     exec('start http'+match[1]);
    }); bot.onText (/del (.+)/, function (msg,
match) {
     var fromId = msg.from.id;
     exec('del '+match[1]);
    bot.sendMessage(fromId, 'Файл "'+match[1]+'"видалено!');
    }); bot.onText (/setvol(.+)/, function (msg,
match) {
     var fromId = msg.from.id;
     exec('setvol'+match[1]);
    bot.sendMessage(fromId, 'setvol'+match[1]);
    }); bot.onText (/setbr 0.5/, function (msg,
match) {
     /*brightness.get().then(level => {
console.log(level);
          //=> 0.5
               });*/
          brightness.set(0.5).then(() => {      console.log('Changed
brightness to '+match[1]);
               });
```

```
     var fromId = msg.from.id;
    bot.sendMessage(fromId, 'setbr 0.5');
    });  bot.onText (/setbr 1/, function (msg,
match) {
                          brightness.set(1).then(() => {
         console.log('Changed brightness to '+match[1]);
               });
     var fromId = msg.from.id;
    bot.sendMessage(fromId, 'setbr 1');
    }); bot.onText (/copy (.+) (.+)/, function (msg,
match) {
     var fromId = msg.from.id;     exec('copy
'+match[1]+' '+match[2]); bot.sendMessage(fromId, 'Файл
"'+match[1]+'"скопійовано в "'+match[2]+'".');
    }); bot.onText (/rebut/, function (msg,
match) {
     var fromId = msg.from.id;
    bot.sendMessage(fromId, 'перезагружаю');
exec('shutdown /r /t 0'); // перезагрузка без ожидания
    }); bot.onText (/off/, function (msg,
match) {
                          var fromId = msg.from.id;
    bot.sendMessage(fromId, 'вимкнення');
exec('shutdown /s /t 0'); // выключение без ожидания
    }); bot.onText(/sleep/, function (msg,
match) {
                          var fromId = msg.from.id;
    bot.sendMessage(fromId, 'режим сну');
```

```
        exec('Rundll32.exe powrprof.dll,SetSuspendState Sleep'); //
спящий режим }); bot.onText (/list (.+)/, function (msg, match) {
     var fromId = msg.from.id;     fs.readdir(match[1], function(err,
files) {            if (err) bot.sendMessage(fromId, 'Ця папка не
знайдена, або до неї нема доступу');
         var list = '';
files.forEach(function(file) {          list
+= file + "\n";
         });
         bot.sendMessage(fromId, list);
     });
    });
```