

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

## **ВИПУСКНА РОБОТА**

на тему:

**«Telegram-бот для дистанційного керування  
персональним комп'ютером»**

**Завідувач**

**Випускаючої кафедри**

**Довбиш А.С.**

**Керівник роботи**

**Ободяк В.К.**

**Студента групи ІІз-61с**

**Левченко В.І.**

**СУМИ 2020**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
Центр заочної, дистанційної і вечірньої форм навчання

**Кафедра комп'ютерних наук**

Затверджую \_\_\_\_\_

Зав. кафедрою Довбиш А.С.

“ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**до випускної роботи**

Студента четвертого курсу, групи ІНз-61С спеціальності “Інформатика”  
заочної форми навчання Левченка Владислава Ігоровича.

**Тема: “Telegram-бот для дистанційного керування персональним  
комп'ютером”**

Затверджена наказом по СумДУ

№ \_\_\_\_\_ от \_\_\_\_\_ 20\_\_ р.

**Зміст пояснювальної записки:** 1) аналітичний огляд методів реалізації програми для дистанційного керування персональним комп'ютером; 2) постановка завдання й формування завдань дослідження; 3) практична реалізація telegram – бота для дистанційного керування персональним комп'ютером; 4) тестування програмного продукту.

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

Керівник випускної роботи \_\_\_\_\_ Ободяк В.К.

Завдання прийняв до виконання \_\_\_\_\_ Левченко В.І.

## РЕФЕРАТ

**Записка:** 40 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 смартфона.

**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.

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

## ЗМІСТ

ВСТУП .....	5
1 АНАЛІТИЧНИЙ ОГЛЯД.....	6
1.1 Аналіз існуючих реалізацій.....	6
1.2 Постановка задачі.....	14
2 ВИБІР ІНСТРУМЕНТІВ.....	15
2.1 Пошук можливих способів реалізації .....	15
2.2 Python.....	15
2.3 Node.js.....	17
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	<b>Ошибка! Закладка не определена.</b>
3.1 Підготовка комп'ютера .....	<b>Ошибка! Закладка не определена.</b>
3.2 Створення бота .....	25
3.3 Додавання функцій .....	27
3.4 Тестування програмного продукту .....	32
ВИСНОВКИ.....	<b>Ошибка! Закладка не определена.</b>
СПИСОК ЛІТЕРАТУРИ.....	<b>Ошибка! Закладка не определена.</b>
ДОДАТОК .....	37

## ВСТУП

Ми живемо в ері інформаційних технологій. За допомогою сучасних девайсів та програм люди можуть обмінюватися інформацією зробивши всього декілька кліків. Тому месенджери стали невід’ємним елементом сучасного світу.

Одним із таких є Телеграм. Створений Павлом Дуровим, завдяки своїй зручності, безпеці та анонімності він став одним із най популярніших у світі. Я сам ним користуюся, тому можу сказати це напевно. Однією з причин популярності телеграма стали «боти», та можливість вільно їх створювати і використовувати.

Боти – це аккаунти якими керують програми. За допомоги простих текстових повідомлень ви можете поросити його зробити майже усе, від покупки квитків на потяг, до пошуку цікавої книги чи фільму для перегляду ввечері. Я хочу зробити дещо особливе. Бота, за допомоги якого я зможу керувати своїм ПК.

Багато людей використовують комп’ютер для перегляду фільмів чи відео. Сидячи на відстані від екрану було б зручно перемикаати відео, шукати щось у браузері чи на ютубі, регулювати гучність та яскравість. Тому було вирішено створити бота який зможе це робити.

## 1 АНАЛІТИЧНИЙ ОГЛЯД

### 1.1 Аналіз існуючих реалізацій

#### RC telegram bot

Під час пошуку існуючих реалізацій бота для дистанційного керування комп'ютером було знайдено декілька варіантів. Один із них RC (Remote Control) telegram bot (Рис. 1.1). Завантажити його можна на сайті [11].

Особливості:

- керувати можуть тільки обрані в програмі користувачі;
- програма може працювати у фоновому режимі;
- зручний інтерфейс;
- широкий функціонал;
- автоматичне підключення після втрати зв'язку.

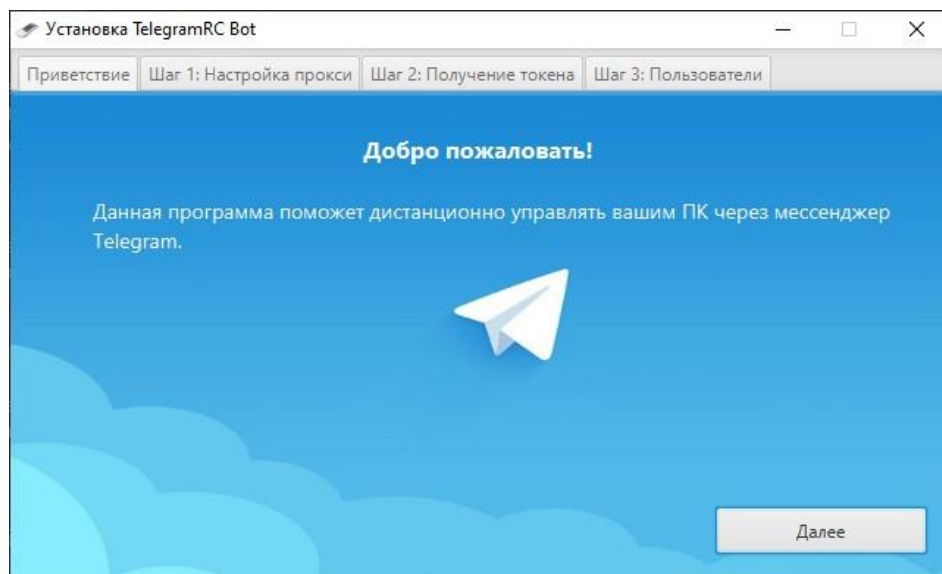


Рисунок 1.1 – Remote control telegram-bot

Для користування програмою потрібно створити бота.

Створення бота telegram:

- звертаємося до боту @BotFather;
- пишемо команду / newbot;

- вводимо спочатку ім'я бота потім його нік (повинен закінчуватися на bot);
- далі отримуємо від бота token і вводимо його в вікні установки програми (Рис. 1.2);
- щоб керувати ботом необхідно вказати свій нік в списку дозволених користувачів.

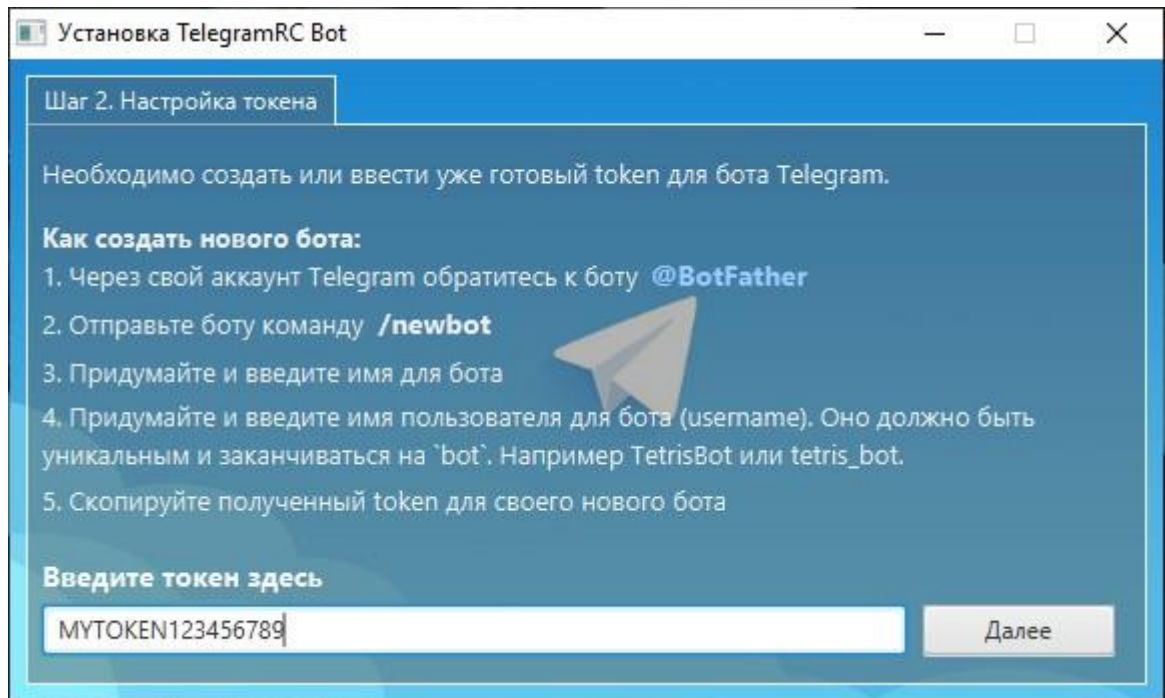


Рисунок 1.2 – Вікно встановлення програми

### Керування ботом

Бот приймає команди як зі слешем на початку, так і без нього, тобто / help і help це одна і та ж команда. Аргументи команди можна передати через знак пробіл, якщо аргумент містить його, потрібно обрармити його в подвійні лапки; щоб зробити команду клікабельно, можна замість пробілу використовувати " \_\_ " (два знака нижнє підкреслювання).

### Опис команд

#### Загальні:

/ start - Привітання бота

/ help - Довідка. Містить опис команд. Показує клавіатуру для керування.

/ whoami - показує ім'я поточного користувача і його id

/ ip - показує зовнішній IP, Geo-IP інформацію і MAC адреса

/ browse [url] - відкриває посилання (підтримуються http і https посилання) в браузері за замовчуванням

/ alert [message] - показує повідомлення на ПК

/ uptime - показує час роботи програми і для системи windows - час роботи ПК система

/ systeminfo - показує інформацію про систему

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

/ cd - показує шлях до поточної директорії / Cd [path] - змінити директорію. Тут і в інших командах в [path] може міститися як абсолютний, так і відносний шлях.

/ ls - показує вміст поточної директорії.

Додатково відображаються короткі команди для переходу до певної директорії або файлу.

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

/ download [path] [selectBy = name] - завантажити файл (бот відправляє файл в діалог)

/ delete [path] [selectBy = name] - видалити файл з ПК.

/ print - відправляє файл на друк (використовується принтер за замовчуванням, необхідна наявність встановленого Microsoft Word)

## Таймери

Використовуються для виконання команд через певний проміжок часу.

/ timers - показує список працюючих таймерів

/ timer [after] [command] - створює новий таймер. [after] - час, через який буде виконана команда, наприклад, 1s, 1m 10s, 1h 10m 20s. [command] -



текст команди, можна з пробілами. Наприклад, / timer 30s / alert Hello World!

## Media

/ screens - показує список екранів і їх розташування + кнопки для створення скріншотів.

/ screenshot [n = 0] - робить скріншот. Якщо не вказано номер екрану [n], то вибирається екран за замовчуванням.

/ cameras - відображає список web-камер + кнопки для знімків.

/ photo [n = 0] - робить знімок з камери. Якщо не вказано номер камери [n], то вибирається камера за замовчуванням.

/ volume - відображає рівень гучності від 0 до 100. (працює не на всіх пристроях).

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

/ brightness - відображає рівень яскравості від 0 до 100. (працює не на всіх пристроях)

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

## Клавіатура

/ keyboard - показує клавіатуру 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] - на системі windows працюють деякі медіа клавіші: play, stop, next, prev

/ media - відображає клавіатуру з медіа-клавішами.

## Hardware

/ ram - відображає оперативну пам'ять, займану ботом. На ОС Windows відображає кількість всієї вільної та RAM на ПК.

/ battery - відображає дані про орендованому акумуляторі (якщо такий є).

/ temperature - відображає дані з датчиків температури (якщо такі є і якщо є права доступу до них) живлення.

/ reboot - виконує перезавантаження ПК.

/ shutdown - вимикає ПК.

Перед вимиканням або перезавантаженням ПК з'явиться повідомлення з таймером (Рис. 1.3).

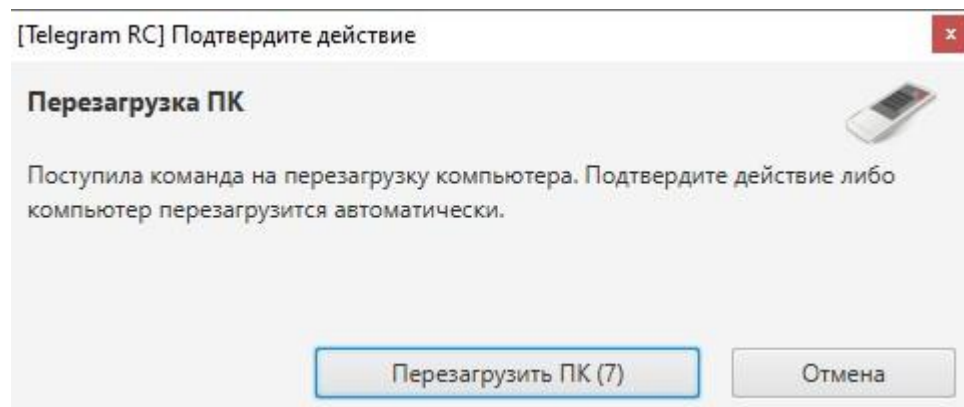


Рисунок 1.3 – Повідомлення з таймером

## TeamViewer

TeamViewer - додаток для віддаленого керування і проведення конференцій. Також може використовуватися:

- для надання підтримки колегам, друзям або клієнтам на відстані;
- для встановлення з'єднання між комп'ютерами з різними операційними системами. TeamViewer працює під операційними системами Windows, macOS, Linux або Google Chrome OS;
- для адміністрування серверів Windows і робочих станцій;

- цей додаток може бути запущений як системна служба Windows, що дозволить отримати доступ до комп'ютера ще до входу в Windows;
- є можливість підключення з мобільних пристроїв Android, iOS, Windows 10 Mobile або BlackBerry до пристроїв Windows, Mac або Linux.
- надавати доступ до свого робочого столу під час конференцій, демонстрацій або спільної роботи;
- увійти до домашнього ПК і працювати з документами, перевірити електронну пошту або завантажувати і редагувати зображення в дорозі;
- підключитися до робочого ПК, перебуваючи в дорозі для отримання інформації;
- отримати доступ до пристроїв на базі Android і iOS і надавати їм підтримку;
- здійснювати віддалений моніторинг системи і відстежувати активи за допомогою перевірок стану інтегрованої системи і ITbrain;

Team Viewer працює через брандмауери, маршрутизатори NAT і проксі, не вимагаючи налаштувань.

### **Загальний опис**

Якщо порівняти з'єднання цього додатку з телефонним дзвінком, то ID TeamViewer буде відповідати номеру телефону, за яким можна окремо знайти користувачів.

Комп'ютери та мобільні пристрої ідентифікуються за унікальним ID, який генерується автоматично при першому запуску, ґрунтується на характеристиках забезпечення і надалі не змінюється.

Всі з'єднання зашифровані і захищені від доступу третіх осіб.

## Опис головного вікна

Головне вікно складається з двох вкладок: віддалене керування і конференція (Рис. 1.4).

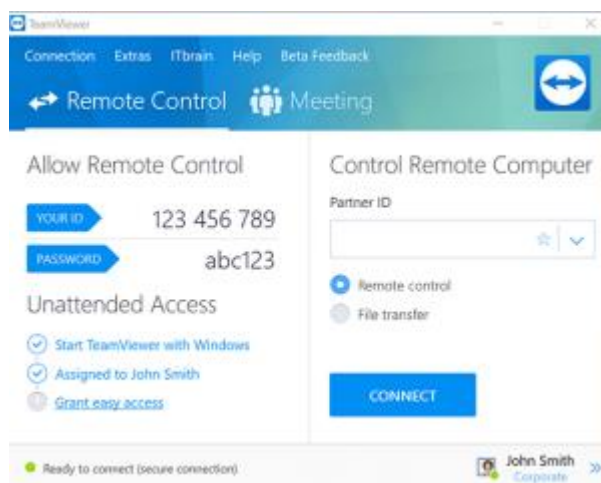


Рисунок 1.4 – Головне вікно

## Вкладка «Віддалене керування»

Вкладка Віддалене керування складається з наступних областей:

### Дозволити керування

У цій області знаходиться ID і тимчасовий пароль. Якщо користувач надасть ці дані своєму партнеру, то він зможе підключитися до комп'ютера користувача (Рис. 1.5).

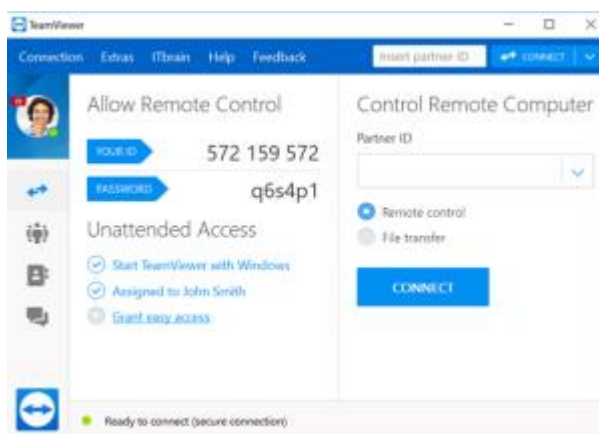


Рисунок 1.5 – Вкладка дозволити керування

Клацанням по піктограмі в полі Пароль відкривається контекстне меню для зміни випадкового пароля або для копіювання його в буфер обміну (функція

Windows «Копіювати і вставити»). Крім того, тут можна задати особистий пароль.

У діалогових вікнах Автоматичний доступ можна налаштувати програму на автоматичний запуск одночасно з Windows (системна служба), прив'язати пристрій до свого облікового запису і вибрати особистий пароль. Після цього можна у будь-який час отримати доступ до свого комп'ютера за допомогою TeamViewer.

### **Керування комп'ютером**

Для віддаленого керування комп'ютером необхідно ввести його ID в поле ID партнера. При цьому доступні різні режими з'єднання:

- віддалене керування: керування комп'ютером партнера або спільна робота на одному комп'ютері;
- передача файлів: відправлення файлів на комп'ютер партнера і назад.
- VPN: створення приватної віртуальної мережі з партнером.

Провівши дослідження можна стверджувати, що це одна з найкращих програм для дистанційного керування пристроями, але усе це стосується лише керування одного комп'ютера іншим. Коли справа доходить до мобільного додатку усе по іншому:

- нестабільне з'єднання;
- незручне керування;
- через недостатній розмір екрану смартфона неможливо відкрити певний додаток;
- зображення на телефоні постійно зависає.

## 1.2 Постановка задачі

Після аналізу існуючих програм дистанційного керування комп'ютером можна зробити висновок, що проект який розроблюється крім зручності використання зі смартфона повинен мати такі 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.

Таким чином потрібно виконати такі завдання:

1. Знайти можливі способи реалізації поставленої задачі, а саме вибрати інструменти;
2. виконати практичну реалізацію телеграм бота;
3. провести його тестування.

## 2 ВИБІР ІНСТРУМЕНТІВ

### 2.1 Пошук можливих способів реалізації.

Функціональність Telegram-бота можна реалізувати різними шляхами, кожен з яких має свої переваги.

З використанням Microsoft Visual Studio. В якості мови програмування в цьому випадку раціонально буде вибрати C # або Node.js. При цьому, не знадобиться з нуля писати великі масиви коду - набагато простіше скористатися готовим набором шаблонів, таких як Bot Application, Bot Framework Emulator, Bot Dialog і Bot Controller. Всю необхідну інформацію про їх встановлення та використання містить документація Microsoft Bot Framework, залишається тільки налаштувати ці шаблони під свої вимоги. Такий варіант підходить в повній мірі тільки для користувачів Windows.

Серед інших варіантів реалізації Telegram бота, PHP також дає чимало можливостей через наявність вже готових бібліотек для роботи з Телеграм, таких, наприклад, як Telegram Bot SDK, що дозволяє звести зусилля до мінімуму.

Нарешті, серед мов програмування, що дозволяють швидко і безпроблемно створити бота в телеграмі, Python є одним з найбільш популярних рішень. В основному це пов'язано з широкими можливостями, доступними як при використанні стандартних бібліотек, так і з застосуванням вже готових варіантів, таких як PyTelegramBotAPI, розрахованих на роботу безпосередньо з Telegram.

### 2.2 Python

Основна бібліотека для роботи з телеграм це python-telegram-bot.

Ця бібліотека забезпечує чистий інтерфейс Python для API Telegram Bot. Він сумісний з версіями Python 3.5+ та PyPy.

Окрім реалізації API, ця бібліотека має ряд класів високого рівня, щоб зробити розробку ботів легкою та зрозумілою. Вони містяться в підмодулі telegram.ext.

Встановлення:

Для встановлення бібліотеки на комп'ютер в консолі необхідно прописати наступний код (Рис. 2.1).

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

Рисунок 2.1 – Код завантаження бібліотеки

Також є можливість встановити з репозиторію на сайті (Рис. 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

Ця бібліотека містить багато корисних модулів:

`telegram.ext.CommandHandler` – використовується для обробки команд надісланих боту. Команди - це повідомлення Telegram, які починаються з /, можуть вводитись після @ та ім'ям бота або додатковим текстом. Обробник додасть список до `CallbackContext` з назвою `CallbackContext.args`. Він буде містити список рядків, який є текстом, що слідує за командою, розділеною на окремі або послідовні символи пробілу.

За замовчуванням обробник прослуховує повідомлення, а також відредаговані повідомлення. Щоб змінити цю поведінку, можна використати `Filters.update.edited_message` у аргументі `filter`.

Параметри:

- `command (str | List[str])` - команда або список команд, які цей обробник повинен виконувати;
- `callback (callable)` –буде викликано, коли `check_update` визначить, що модуль потребує оновлення;
- `pass_args (bool, optional)` – визначає, чи слід передавати аргумент команді.

Аргумент - ключове слово, що позначається «args»;

- `allow_edited (bool, optional)` – визначає чи буде обробник приймати виправлені повідомлення.



`telegram.InlineKeyboardButton` – модуль для створення інлайн клавіатури (клікабельних кнопок).

Параметри:

- `text (str)` – напис на кнопці;
- `url (str)` – посилання яке відкривається при натисканні на кнопку;
- `login_url (telegram.LoginUrl, optional)` – використовується для авторизації користувача;
- `callback_data (str, optional)` – інформація яку отримує бот коли натискається кнопка;
- `switch_inline_query (str, optional)` – Якщо цей параметр заданий, то при натисканні на кнопку додаток запропонує користувачеві вибрати будь-який чат, відкриє його і вставить в поле вводу повідомлення юзернейм бота і певний запит для вбудованого режиму. Якщо відправляти пусте поле, то буде вставлений тільки юзернейм бота. Це потрібно для того, щоб швидко перемикатися між діалогом з ботом і вбудованим режимом з цим же ботом;
- `switch_inline_query_current_chat (str, optional)` – така ж функція, але без вибору чату.

### 2.3 Node.js

Node або Node.js - програмна платформа основана на V8, що здійснює трансляцію JavaScript в машинний код, який перетворює JavaScript з вузькоспеціалізованої мови в мову загального призначення. Node.js додає можливість JavaScript взаємодіяти з пристроями введення-виведення через свій API написаний на C ++, підключати інші зовнішні бібліотеки, написані на різних мовах, забезпечуючи виклики до них з JavaScript-коду.

Node.js застосовується переважно на сервері, виконуючи роль веб-сервера, але є можливість розробляти на Node.js і десктопні віконні додатки.

В основі Node.js лежить подієво-орієнтоване програмування, в якому виконання програми визначається подіями - діями користувача (клавіатура,

миша), повідомленнями інших програм і потоків, подіями операційної системи, наприклад надходженням мережевого пакету. І асинхронне, або реактивне програмування, тобто можливість паралельно виконувати дії не чекаючи на завершення попередньої. Для цього проекту мені також знадобляться наступні бібліотеки: `node-telegram-bot-api`, `brightness` та `robotjs`.

`Node-telegram-bot-api` – основна бібліотека, на якій буде написано проект. Вона використовує кілька цікавих підходів, які дозволяють створювати складних ботів зі збереженням простоти коду. Нижче представлені найбільш цікаві механізми:

- **Middlewares** - коли користувач щось відправляє боту, повідомлення спочатку проходить через всі зареєстровані `middlewares`, а вже після потрапляє в контролер. `Middlewares` можуть модифікувати контекст, а також припиняти виконання запиту, наприклад, якщо звичайний користувач спробує потрапити в область для адміністраторів.
- **Sessions** - сесії, які можуть зберігати в собі інформацію, не прив'язану до контролера. Можна навести аналогію з глобальними змінними, які доступні з будь-якого місця в боті. Дуже зручна річ, яка використовується, наприклад, для локалізації. Є можливість зберігати сесії в різних режимах - БД, `redis`, локальні файли і т.д.
- **Webhooks** - бот може працювати в двох режимах - `long polling` або `Webhooks`. І хоча працюють вони однаково швидко, все ж в `production` середовищі краще використовувати другий варіант. Таким чином, бот сам буде отримувати всі оновлення. Обов'язковою умовою для `Webhook` є підтримка сервером `SSL / TLS`. Також треба переконатися, що порт, на якому запущений `Webhook`, відкритий і доступний ззовні.
- **Markup** - за допомогою цього класу можна навчити бота відповідати в `markup / markdown` розмітки.

- Stage - Бот в залежності від типу повідомлення перенаправляє користувача в той чи інший контролер, який виконає свою функцію і відправляє відповідь користувачеві.

Brightness – бібліотека завдяки якій стає можливою реалізація функції налаштування яскравості екрану.

Robotjs – бібліотека для керування мишкою та клавіатурою. Для натискання клавіш нам знадобиться наступна функція (Таблиця 2.1)

Таблиця 2.1 – keyTap(key,[modifier]).

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	

## Продовження таблиці 2.2

left	Стрілка вліво	
home		
end		
pageup		
pagedown		
f1		
f2		
f3		
f4		
f5		
f6		
f7		
f8		
f9		
f10		
f11		
f12		
command		
alt		
control		

Продовження таблиці 2.2

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

## Продовження таблиці 2.2

numpad_0		Не підтримується на лінукс
numpad_1		Не підтримується на лінукс
numpad_2		Не підтримується на лінукс
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	Не підтримується на віндоус

Таким чином для реалізації телеграм бота для вирішення поставленої задачі буде використовуватись програмна платформа Node.js

## 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

### 3.1 Підготовка комп'ютера

Спочатку треба встановити Node.js. Для цього треба перейти на офіційний сайт [16]. Обрати операційну систему та її розрядність і завантажити програму (Рис. 3.1).

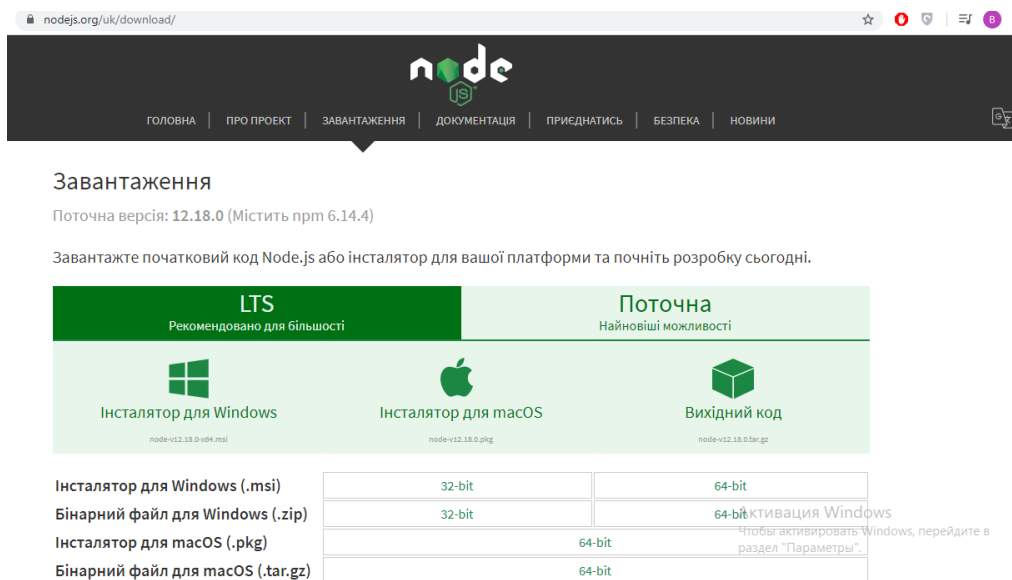


Рисунок 3.1 – Завантаження програми

Далі встановлюємо Node.js (Рис. 3.2), (Рис. 3.3).

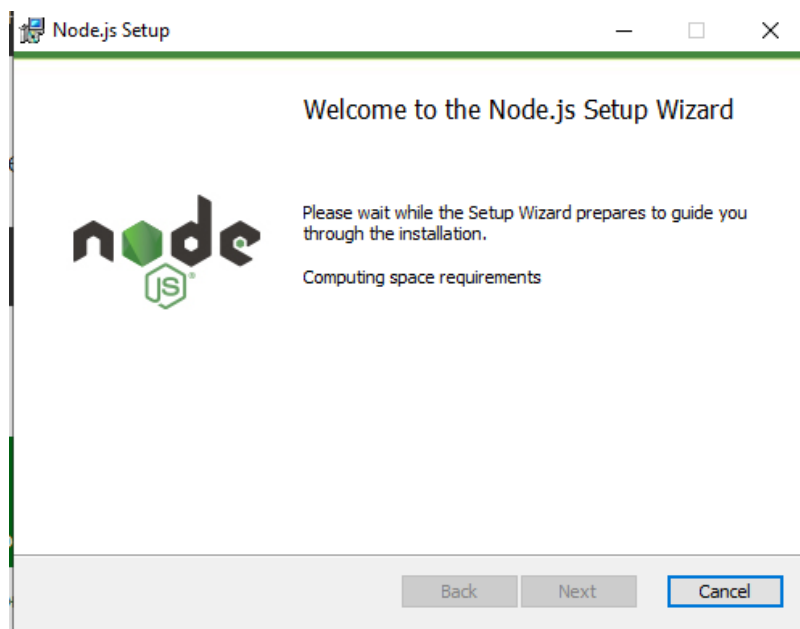


Рисунок 3.2 – Встановлення програми

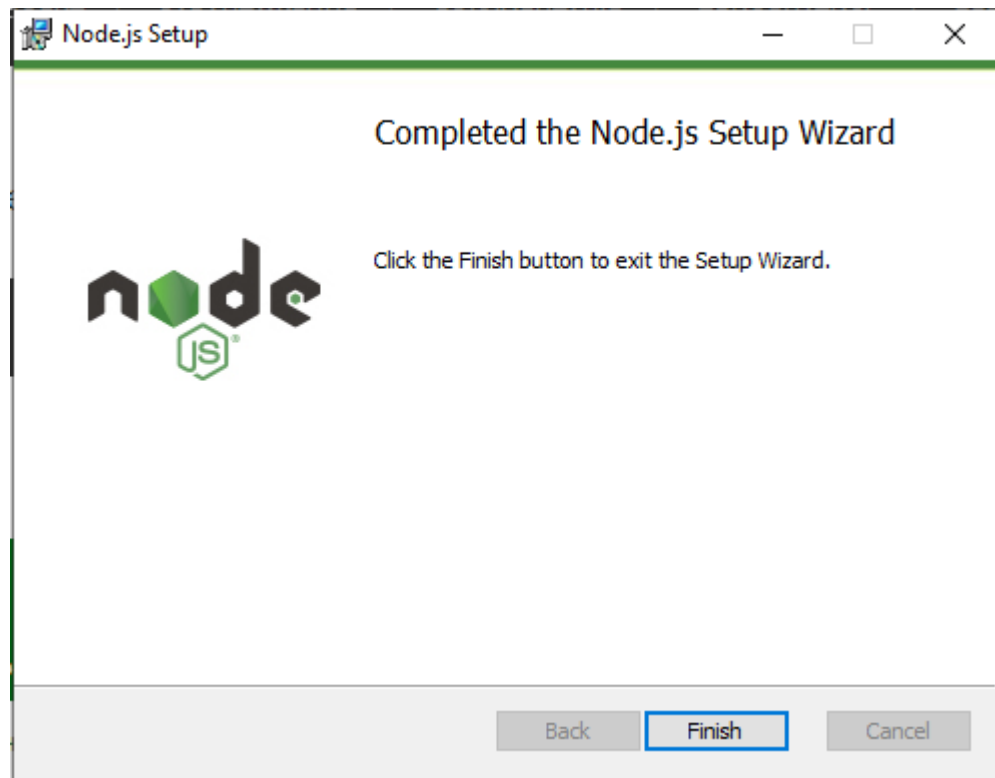


Рисунок 3.3 – завершення встановлення

Потім потрібно інстальювати бібліотеку. Для цього потрібно відкрити консоль і перейти у папку з проектом. Далі ввести команду (Рис. 3.4).

```
Node.js command prompt
Your environment has been set up for using Node.js 14.4.0 (x64) and npm.
C:\Users\OVERLORD>cd C:\remote
C:\remote>npm install node-telegram-bot-api_
```

Рисунок 3.4 – Встановлення бібліотеки

Якщо усе пройшло успішно побачимо наступне (Рис. 3.5).

```
+ node-telegram-bot-api@0.50.0
updated 1 package and audited 141 packages in 4.83s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

C:\remote>
```

Рисунок 3.5 – Завершення встановлення



### 3.2 Створення бота

Щоб бот функціонував, потрібно його створити. Для цього в Телеграмі є спеціальний мета-бот BotFather (@BotFather). Його потрібно додати через пошук, в клієнті Телеграма. Список його команд можна отримати, написавши в чаті з ним команду /help. Для створення нового бота потрібно написати команду /newbot і в наступному повідомленні передати назву бота та після написати його нікнейм (за яким користувачі будуть його знаходити) . У відповідь прийде повідомлення з API ключем (Рис. 3.6).

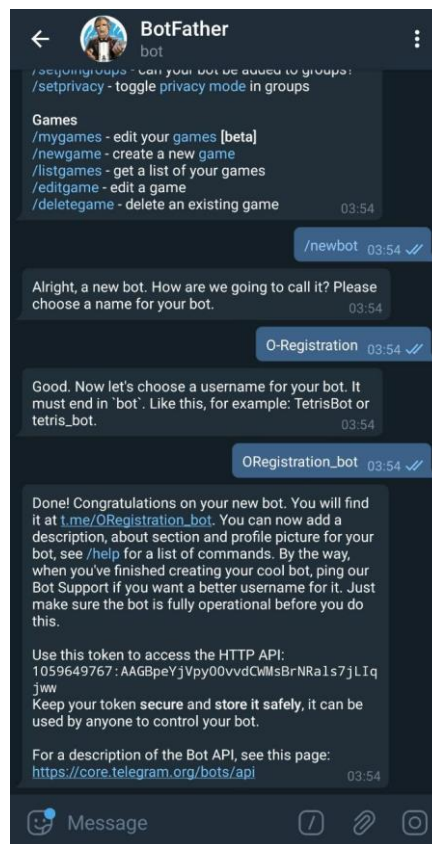


Рисунок 3.6 – Створення бота

Отриманий API ключ потрібно ввести у програму (Рис 3.7).

```
var TelegramBot = require('node-telegram-bot-api');
var token = '1059649767:AAGBpeYjVpy00vvdCWMSBrNRa1s7jLIqjww';
var bot = new TelegramBot(token, {polling: true});
```

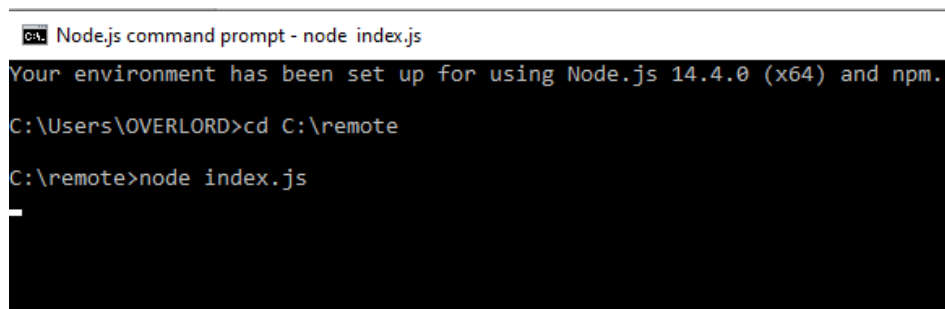
Рисунок 3.7 – Введення API ключа

Щоб перевірити чи працює бот можна створити тестовий метод (Рис. 3.8)

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

Рисунок 3.8 – Створення тестового методу

Використавши функцію onText створюємо метод завдяки якому бот при отриманні повідомлення test відправляє користувачу повідомлення: «Тест пройдено!» . Відкриваємо дужки і записуємо код. Спершу потрібно написати регулярний вираз /test/ коли бот отримає це повідомлення функція буде спрацьовувати. Потім створити змінну fromId у цю змінну зберегти id користувача який прислав повідомлення. Потім використовуючи функцію sendMessage відправити повідомлення. Для цього в дужках треба записати спершу id користувача, у нашому випадку воно знаходиться у змінній fromId, потім після коми у лапках повідомлення. Тепер можна зберегти і запустити програму через консоль (Рис. 3.9).



```
Node.js command prompt - node index.js
Your environment has been set up for using Node.js 14.4.0 (x64) and npm.
C:\Users\OVERLORD>cd C:\remote
C:\remote>node index.js
```

Рисунок 3.9 – Запуск програми через консоль

Перевірка програми (Рис. 3.10).

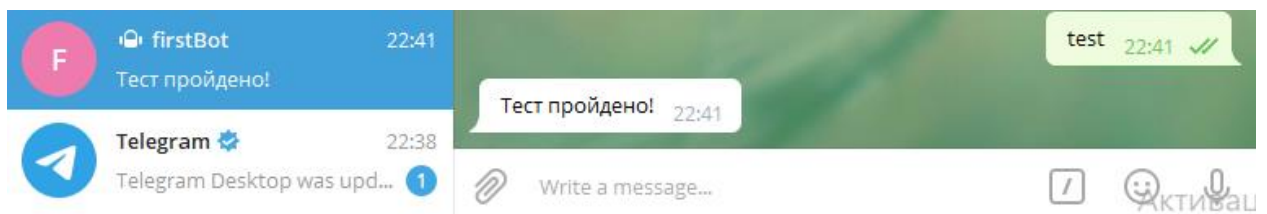


Рисунок 3.10 – Перевірка команди

### 3.3 Додавання функцій

#### Маніпуляції з файлами

Для створення функцію для видалення файлів треба навчити бота працювати з системною консоллю. З цим допоможе модуль `childProcess`. Підключення відповідного модуля до програми (Рис. 3.11).

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

Рисунок 3.11 – Підключення модуля `childProcess`

Далі вводимо наступний код (Рис. 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 потрібно записати `del (.+)`. Коли бот отримує повідомлення яке починається з “del “ решту він збереже у масив `match`.

Далі використовується функцію `exec`. В дужках формується запит який бот буде відправляти консолі. В кінці повідомляється про те, що файл був видалений.

Тепер напишемо схожу команду для копіювання файлів (Рис 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 (Рис. 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`

Далі прописується наступний код (Рис. 3.15).

```
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);
  });
});
```

Рисунок 3.15 – Функція перегляду файлів

Спочатку бот зчитує повідомлення у якому вказано місце знаходження папки:

«bot.onText (/list (.+)/, function (msg, match))».

Далі передає цю інформацію у функцію:

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

У разі відсутності папки, або закритого доступу з'являється помилка, тому потрібно опрацювати цей варіант:

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

Тепер потрібно 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 (Рис. 3.16).

```
bot.onText (/rebut/, function (msg, match) {
    var fromId = msg.from.id;
    bot.sendMessage(fromId, 'перезагружаю');
    exec('shutdown /r /t 0'); // перезагрузка без ожидания
});
```

Рисунок 3.16 – Функція перезапуску

Для вимкнення до консолі потрібно записати: «shutdown /s /t 0».

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

## Налаштування гучності

Для того, щоб мати змогу регулювати гучність була завантажена програма з сайту [9] яка дає змогу робити це через консоль. Програму треба завантажити та розархівувати, після чого вона готова для використання. Прописавши у консолі: «setvol help», можна побачити усі можливості програми (Рис. 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

Тепер можна користуватися наступними командами:

- 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 (Рис 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 дослідження була знайдена бібліотека для налаштування яскравості екрану. Для того, щоб її завантажити потрібно прописати в консолі: «npm install --save brightness».

Тепер з'являється можливість регулювати яскравість монітору цифрами від 0 до 1 (Рис 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

Однією с цілей проекту являється створення бота з можливістю відкривати посилання у браузері. Для цього можна скористатись уже знайомим модулем «childProcess». Якщо написат у командній строці: «start http...» то операційна система windows відкриє посилання у браузері який виставлений основним. Тому треба запрограмувати бота формувати запит «start» для усіх повідомлень, що починаються з «http» (Рис. 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

Данну функція реалізується за допомоги бібліотеки «robotjs». Для її встановлення потрібно прописати в консолі «npm install robotjs». Данна бібліотека має одну необхідну для нас функцію:

```
robot.keyTap(key, [modifier])
```

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

Modifier – використовується щоб натиснути комбінацію клавіш.

Використовуючи ці знання можна керувати комп'ютером за допомоги гарячих клавіш та їх комбінацій.

Для початку налаштування клавіші які знадобляться 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 (Рис. 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 (Рис. 3.22).

node_modules	16.06.2020 21:55	Папка с файлами
1	03.06.2020 16:16	файл JavaScript
1	17.06.2020 1:15	Текстовый докум...
2	17.06.2020 1:15	Текстовый докум...
index	16.06.2020 22:39	файл JavaScript

Рисунок 3.22 – Папка remote

Перевірка команди для видалення файлів (Рисунок 3.23)

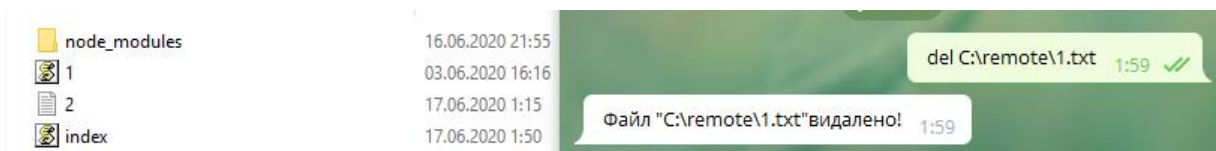


Рисунок 3.23 – Тестування функції видалення файлів

Як можна побачити команда спрацювала.

Копіювання файлів. Створюється папка “test1” (Рис. 3.24).

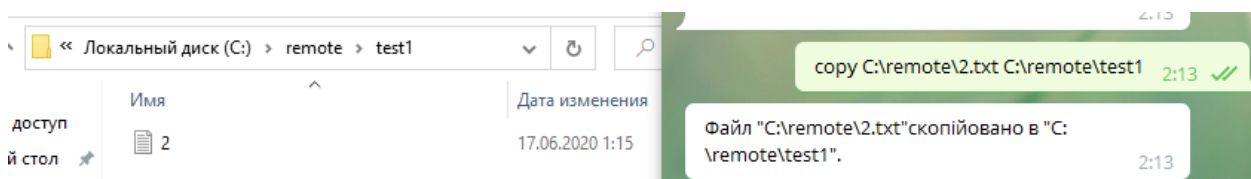


Рисунок 3.24 – Тестування функції копіювання файлів

Файл скопіювався у папку “test1”.

Отримання списку файлів (Рис. 3.25).

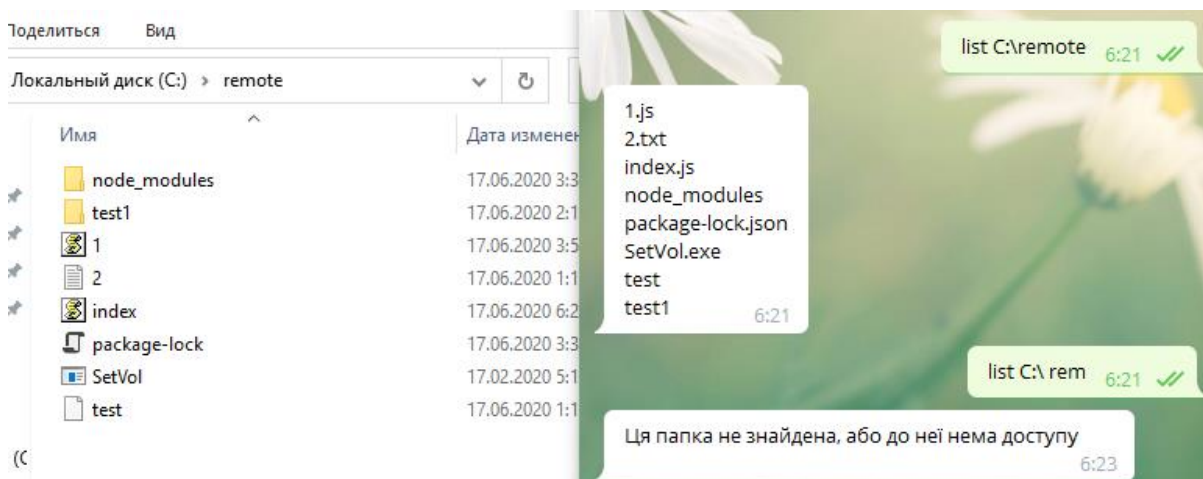


Рисунок 3.25 – Тестування функції перегляду файлів



## Налаштування гучності

Боту відправляються команди для налаштування гучності (Рис. 3.26).



Рисунок 3.26 – Тестування функції налаштування гучності

## Налаштування яскравості

Перевірка програми (Рис. 3.27).

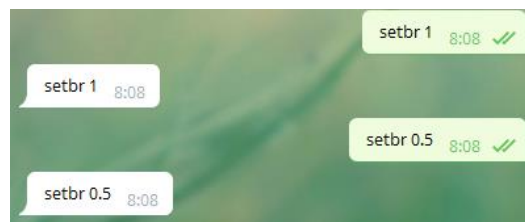


Рисунок 3.27 – Тестування функції налаштування яскравості

З результатів тестування можна зробити висновок, що розроблена програма працює коректно.

## ВИСНОВОК

1. Проведений під час дослідження аналіз показав, що технології розробки ботів в даний час розвинулися. Існує багато можливостей для створення програми. багато мов програмування підтримують telegram API. Розробники продовжують створювати і розвивати бібліотеки, які постійно розширюють функціонал телеграм ботів.
2. Хоча більшість програмістів при їх написанні віддають перевагу Python, інші мови такі, як Node.js та PHP не поступаються у своїх можливостях та мають певні переваги. Знаючи на достатньому рівні декілька мов програмування можна підлаштовувати програми під певні потреби. Дослідивши проблему було знайдено оптимальний інструмент реалізації, а саме Node.js.
3. Створено телеграм бота який відповідає вимогам і 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.
4. Тестування створеної програми показало, яка являється повноцінним пультом дистанційного керування для комп'ютера. Який можна використовувати підчас прослуховування музики, перегляду фільмів та відео на телевізорі використовуючи його як екран.

## СПИСОК ЛІТЕРАТУРИ

1. Stack overflow – <https://ru.stackoverflow.com/questions2>
2. Wibe.team – <https://wibe.team/sozдание-bota-v-telegram/>
3. MDN web docs –  
[https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Regular_Expressions)
4. Влад Скурішин стаття: «Telegram bot API і NodeJs» –  
<https://medium.com/@vladthelittleone/telegram-api-%D0%B2-nodejs-a82b0f7ec575>
5. Telegram, документація створення телеграм бота –  
<https://tlgrm.ru/docs/bots/samples>
6. Npm, репозиторій, документація бібліотеки robot.js –  
<https://www.npmjs.com/package/robotjs#examples>
7. Github, репозиторій, бібліотека node-telegram-bot-api –  
<https://github.com/yagor/node-telegram-bot-api>
8. Smmx стаття: «Створення ботів для телеграм на Python» –  
<https://smmx.ru/telegram/how-to-use/bot-na-python.html>
9. Rlatour.com, завантаження та опис програми SetVol –  
<https://rlatour.com/setvol/>
10. Документація програми Team viewer –  
<https://dl.teamviewer.com/docs/ru/v13/TeamViewer13-Manual-Remote-Control-ru.pdf>
11. Ts.Saltan tech blog завантаження та опис програми TelegramRC bot –  
<https://tssaltan.top/1928.telegramrc-bot/>
12. Npm репозиторій, опис бібліотеки brightness –  
<https://www.npmjs.com/package/brightness>
13. Github репозиторій з бібліотекою python-telegram-bot –  
<https://github.com/python-telegram-bot/python-telegram-bot>

14. Документація бібліотеки python-telegram-bot – <https://python-telegram-bot.readthedocs.io/en/stable/telegram.html>
15. Документація бібліотеки python-telegram-bot, модуль inlinekeyboard – <https://python-telegram-bot.readthedocs.io/en/stable/telegram.inlinekeyboardbutton.html>
16. Офіційний сайт NodeJs – <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);
    });
  });
});
```