FKLT/06-00

UNIVERSITI POLY-TECH MALAYSIA

Assignment Cover– **be sure to keep a copy of all work submitted.**

Name(s): **NUR HAIFAA BINTI HANAPI**		
ID Number(s): **AM2207011647**		
Lecturer Name: SIR MOHD AKMAL BIN MOHD AZMER		Section Number: **01**
Course and Course Code **EMERGING TECHNOLOGY (SWC2373)**		Submission Date: **10/11/2023**
Assignment No. / Title **API ASSIGNMENT**		Extension & Late submission: Allowed / **Disallowed**
Assignment type: **INDIVIDUAL**	% of Assignment Mark	Returning Date:

Penalties:

1. 10% of the original mark will be deducted for every one-week period after the submission date.
2. No work will be accepted after two weeks of the deadline.
3. If you were unable to submit the coursework on time due to extenuating circumstances, you may be eligible for an extension.
4. Extension will not exceed one week

Declaration: I/we the undersigned confirm that I/we have read and agree to abide by these regulations on plagiarism and cheating. I/we confirm that this piece of work is my/our own. I/we consent to appropriate storage of our work for checking to ensure that there is no plagiarism/ academic cheating.

Signature(s):

This section may be used for feedback or other information

TABLE OF CONTENT

GitHub link:

INTRODUCTION

What is Webex?

Webex is Cisco’s communication and collaboration platform for everything and everyone. It combines calling, messaging, and meeting into a single, unified application so you can streamline teamwork and produce results faster than ever. Webex provides the functionality, security, and quality required to migrate your mission-critical communications to the cloud. It’s an easy-to-use and powerful collaboration solution that keeps people and teams connected from anywhere, anytime, on any device.

What is Webex API?

In simple terms, the Webex API is a set of tools that lets software developers create custom apps that work with Cisco Webex, a platform for online meetings and messaging. These apps can send messages, set up video meetings, and integrate with other services. It's like a toolkit for building extra features and automation on Webex.

How can third-party apps be developed with Webex API? How can we develop the app from Webex API and other relevant?

1. Begin by signing up as a developer on the Webex for Developers website, which provides access to the API tools and resources.
2. Create an integration or app within the Webex Developer portal. This step involves defining your app's details and capabilities, resulting in an access token for authentication.
3. Select the specific features from the Webex API that your app will utilize, aligning with your app's purpose, such as messaging, meetings, or user data.
4. Use the access token obtained during app creation for authentication when requesting the Webex API.

5. Interact with the Webex API by sending HTTP requests (e.g., GET, POST) to access and manage data. Include the access token in the request headers for authentication.
6. If your app requires real-time functionality, consider setting up webhooks to receive instant notifications about events within Webex, like new messages.
7. Process the data received from the Webex API to display information within your app's user interface. This may involve parsing JSON data and presenting it to users.
8. If applicable, design and develop the user interface (UI/UX) of your app, ensuring it accommodates the Webex API's functionalities and user interactions.
9. Conduct thorough testing to ensure your app functions correctly and handles potential errors gracefully, covering functional and integration testing.
10. Deploy your app to a suitable hosting environment or distribute it through app stores or other platforms, depending on your target audience.
11. If your app is intended for a broader user base, consider marketing and promoting it to reach potential users or customers.
12. Continuously monitor your app's performance, manage updates, and address any issues or user feature requests to maintain its functionality.

OBJECTIVES

- **Objective:** Create a troubleshooting tool for the organization using the Webex API.
- **User Authentication:** Users start by entering their Webex token number for authentication.
- **Menu Options:** The application offers four options for users:
    a. **Test Connection:** To check the connection with the Webex server.
    b. **Display User Information:** View their details like Displayed Name, Nickname, and Emails.
    c. **List Rooms:** To display up to five rooms with more information like Room ID, Title, date created, and last activity.
    d. **Create a Room:** To make a new space by providing a title; an acknowledgment is displayed if successful.
    e. **Send Message to a Room:** To select a room, send a message and receive an exposure upon successful sending.
- The tool is designed to assist in troubleshooting and managing Webex-related tasks for the organization.

LITERATURE REVIEW

The code is a Python program that talks to the Cisco Webex collaboration platform through its Webex API. Here's what it does:

Software and Language:

The code is written in Python, a standard programming language. It uses two essential tools:

- **`requests`:** This helps it communicate with Webex by sending and receiving data over the internet.
- **`json`:** This helps it handle data in JSON format.

What it Needs:

- **Access Token:** To connect to Webex, the code asks for a unique access token, like a password.

How it Works:

The program works like a menu:

1. **Login:** You start by giving it your access token.
2. **Options:** It shows you a menu with different choices: check the connection, see your info, list chat rooms, create a new space, or send a message.
3. **Action:** When you pick an option, the program uses the access token and 'requests' to talk to Webex.
4. **Feedback:** It tells you if things went well.
5. **Interaction:** When needed, it asks you for more information, like the room's name or a message.
6. **Repeat or Exit:** After each action, it asks if you want to do more or quit.

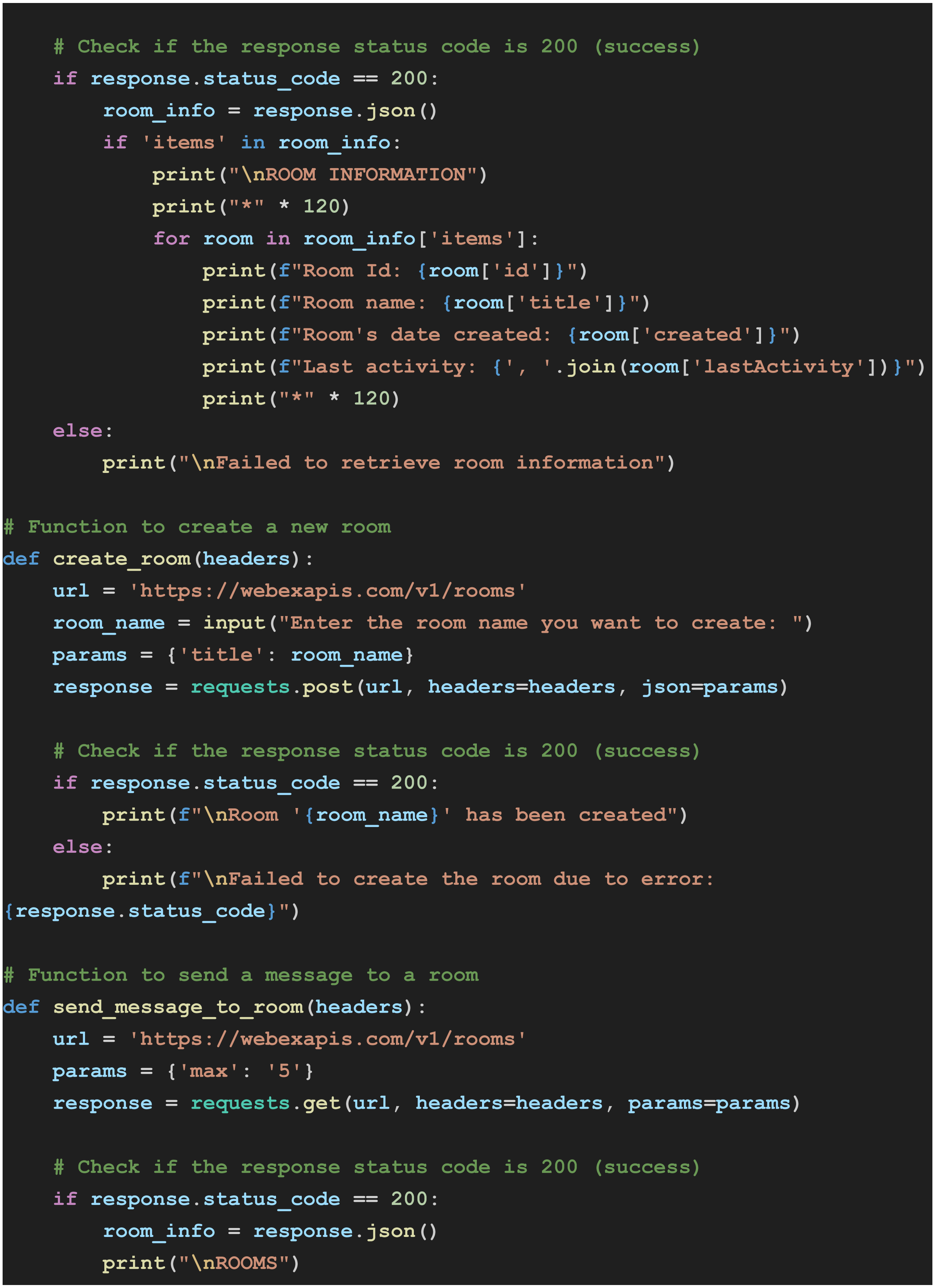
Connecting to Webex:

To talk to Webex, the program sends requests to Webex's web addresses. It uses your access token to prove it can speak with your Webex account. It gets data from Webex, like your user info or chat rooms, and shows it to you.

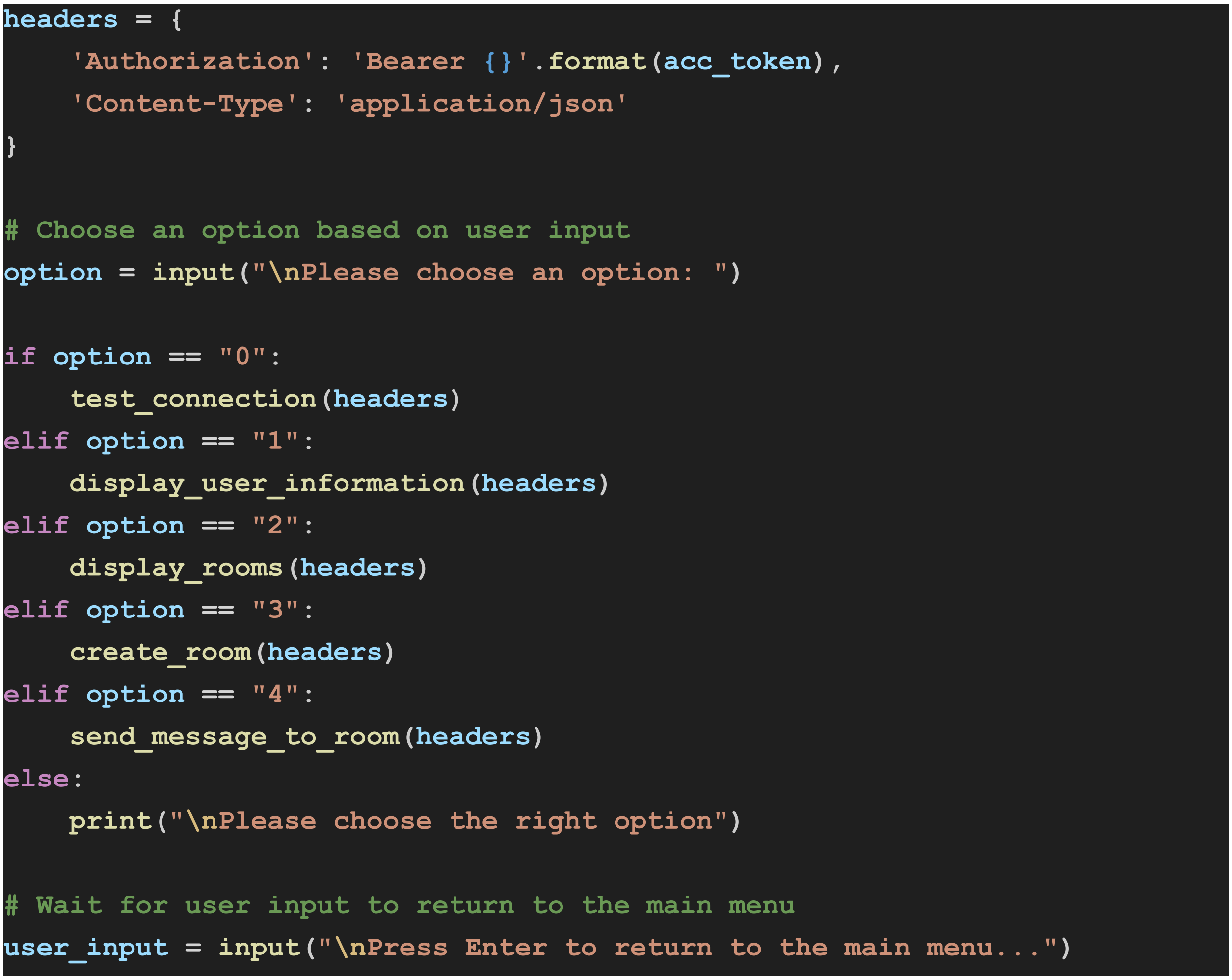
In short, this Python program is like a friendly helper that lets you do things in Webex without going to the Webex website. It's designed to be easy to use, and it talks to Webex behind the scenes to get the job done.

DEVELOPMENT OF THE APPLICATION

```
# Import necessary libraries
import requests
import json

# Function to test the connection to the Webex server
def test_connection(headers):
    url = 'https://webexapis.com/v1/people/me'
    response = requests.get(url, headers=headers)

    # Check if the response status code is 200 (success)
    if response.status_code == 200:
        print("\nConnection to Webex server successful")
    else:
        print("\nConnection to Webex server failed")

# Function to display user information
def display_user_information(headers):
    url = 'https://webexapis.com/v1/people/me'
    response = requests.get(url, headers=headers)

    # Check if the response status code is 200 (success)
    if response.status_code == 200:
        user_info = response.json()
        print("\nUSER INFORMATION")
        print("*" * 130)
        print(f"User Displayed name: {user_info['displayName']}")
        print(f"User Nickname: {user_info['nickName']}")
        print(f"User email: {', '.join(user_info['emails'])}")
        print("*" * 130)
    else:
        print("\nFailed to retrieve user information")

# Function to display a list of rooms
def display_rooms(headers):
    url = 'https://webexapis.com/v1/rooms'
    params = {'max': '5'}
    response = requests.get(url, headers=headers, params=params)
```



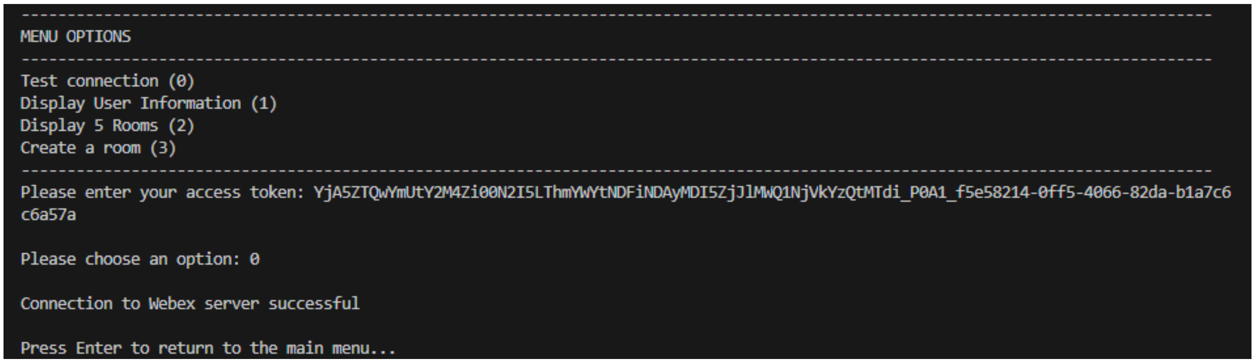
```
    # Check if the response status code is 200 (success)
    if response.status_code == 200:
        room_info = response.json()
        if 'items' in room_info:
            print("\nROOM INFORMATION")
            print("*" * 120)
            for room in room_info['items']:
                print(f"Room Id: {room['id']}")
                print(f"Room name: {room['title']}")
                print(f"Room's date created: {room['created']}")
                print(f"Last activity: {', '.join(room['lastActivity'])}")
                print("*" * 120)
    else:
        print("\nFailed to retrieve room information")

# Function to create a new room
def create_room(headers):
    url = 'https://webexapis.com/v1/rooms'
    room_name = input("Enter the room name you want to create: ")
    params = {'title': room_name}
    response = requests.post(url, headers=headers, json=params)

    # Check if the response status code is 200 (success)
    if response.status_code == 200:
        print(f"\nRoom '{room_name}' has been created")
    else:
        print(f"\nFailed to create the room due to error:
{response.status_code}")

# Function to send a message to a room
def send_message_to_room(headers):
    url = 'https://webexapis.com/v1/rooms'
    params = {'max': '5'}
    response = requests.get(url, headers=headers, params=params)

    # Check if the response status code is 200 (success)
    if response.status_code == 200:
        room_info = response.json()
        print("\nROOMS")
        print("*" * 120)
        for i, room in enumerate(room_info['items']):
            print(f"({i + 1}) {room['title']}")
            print("*" * 120)
        room_choice = int(input("\nPlease choose a room to send a message:
")) - 1

        if 0 <= room_choice < len(room_info['items']):
            room_id_selected = room_info['items'][room_choice]['id']
            room_name_selected = room_info['items'][room_choice]['title']
            message_to_room = input("Enter the message you want to send:
")
            params = {'roomId': room_id_selected, 'markdown':
message_to_room}
            url = 'https://webexapis.com/v1/messages'
            response = requests.post(url, headers=headers, json=params)

            # Check if the response status code is 200 (success)
            if response.status_code == 200:
                print(f"\nMessage: '{message_to_room}' is sent
successfully to room: '{room_name_selected}'")
            else:
                print(f"\nFailed to send the message due to error:
{response.status_code}")
    else:
        print("\nFailed to retrieve room information")

# Main menu with options
print("-" * 130)
print("MENU OPTIONS")
print("-" * 130)
print("Test connection (0)")
print("Display User Information (1)")
print("Display 5 Rooms (2)")
print("Create a room (3)")
print("Send a message to a room (4)")
print("-" * 130)

# Input access token for Webex API
acc_token = input("Please enter your access token: ")
```



```
headers = {
    'Authorization': 'Bearer {}'.format(acc_token),
    'Content-Type': 'application/json'
}

# Choose an option based on user input
option = input("\nPlease choose an option: ")

if option == "0":
    test_connection(headers)
elif option == "1":
    display_user_information(headers)
elif option == "2":
    display_rooms(headers)
elif option == "3":
    create_room(headers)
elif option == "4":
    send_message_to_room(headers)
else:
    print("\nPlease choose the right option")

# Wait for user input to return to the main menu
user_input = input("\nPress Enter to return to the main menu...")
```

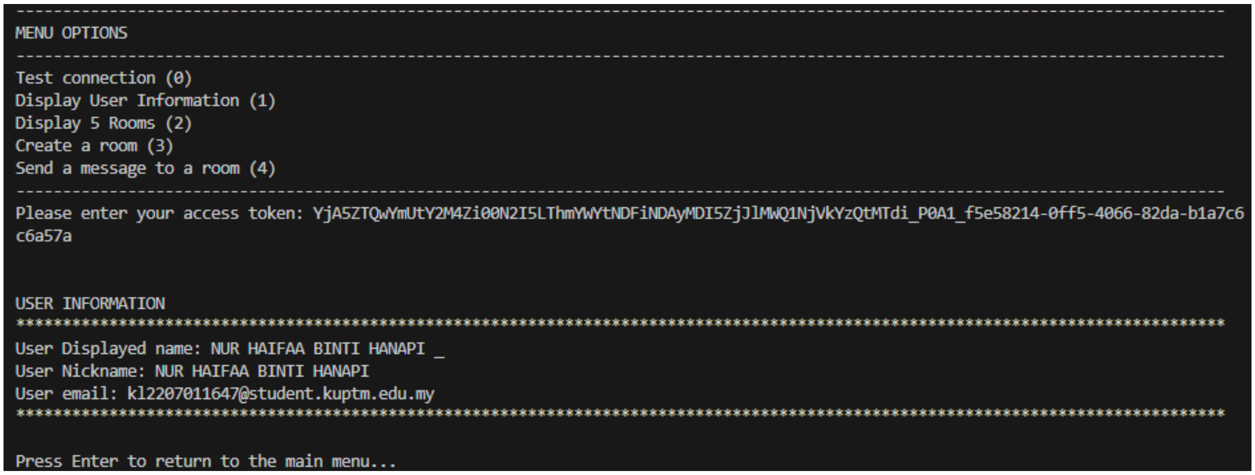
TESTING OF THE APPS

OPTION 0



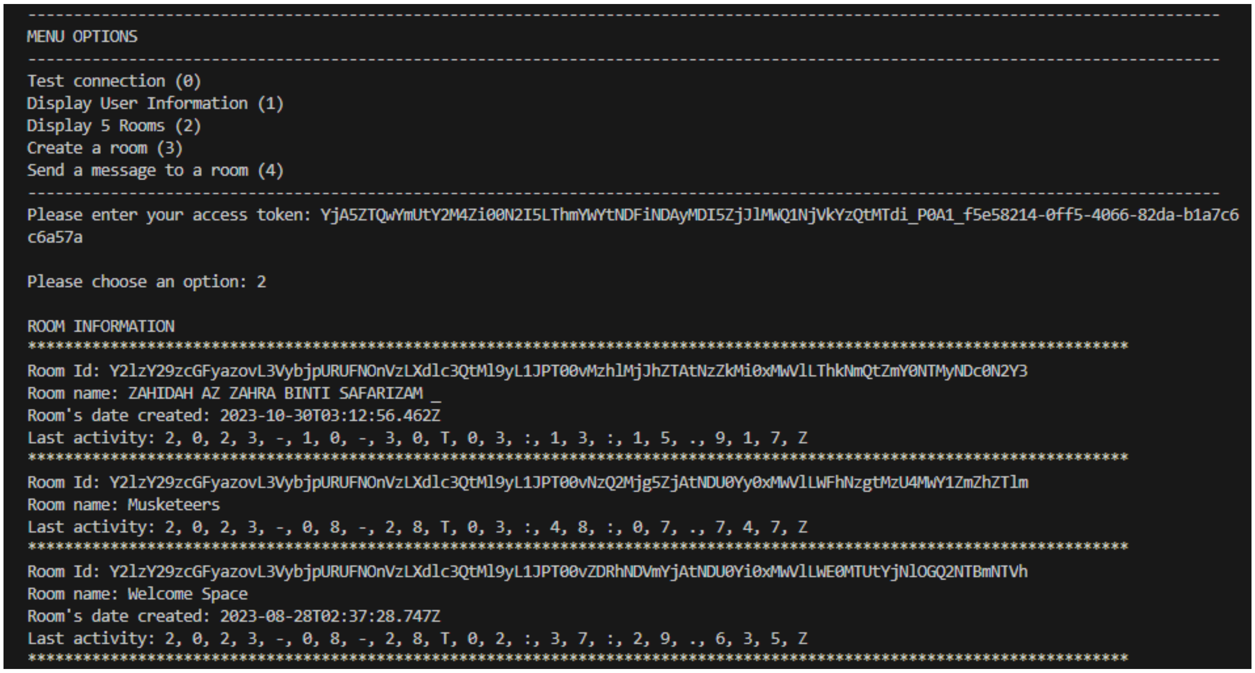
```
------------------------------------------------------------------------------------------------------------------------
MENU OPTIONS
------------------------------------------------------------------------------------------------------------------------
Test connection (0)
Display User Information (1)
Display 5 Rooms (2)
Create a room (3)
------------------------------------------------------------------------------------------------------------------------
Please enter your access token: YjA5ZTQwYmUtY2M4Zi00N2I5LThmYWYtNDFiNDAyMDI5ZjJlMWQ1NjVkYzQtMTdi_P0A1_f5e58214-0ff5-4066-82da-b1a7c6
c6a57a

Please choose an option: 0

Connection to Webex server successful

Press Enter to return to the main menu...
```

OPTION 1



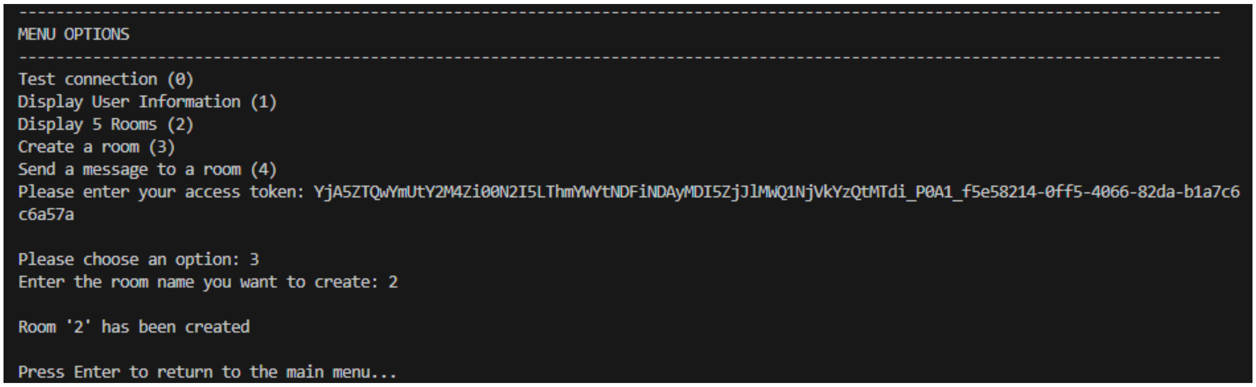
```
------------------------------------------------------------------------------------------------------------------------
MENU OPTIONS
------------------------------------------------------------------------------------------------------------------------
Test connection (0)
Display User Information (1)
Display 5 Rooms (2)
Create a room (3)
Send a message to a room (4)
------------------------------------------------------------------------------------------------------------------------
Please enter your access token: YjA5ZTQwYmUtY2M4Zi00N2I5LThmYWYtNDFiNDAyMDI5ZjJlMWQ1NjVkYzQtMTdi_P0A1_f5e58214-0ff5-4066-82da-b1a7c6
c6a57a


USER INFORMATION
************************************************************************************************************************
User Displayed name: NUR HAIFAA BINTI HANAPI _
User Nickname: NUR HAIFAA BINTI HANAPI
User email: kl2207011647@student.kuptm.edu.my
************************************************************************************************************************

Press Enter to return to the main menu...
```

OPTION 2



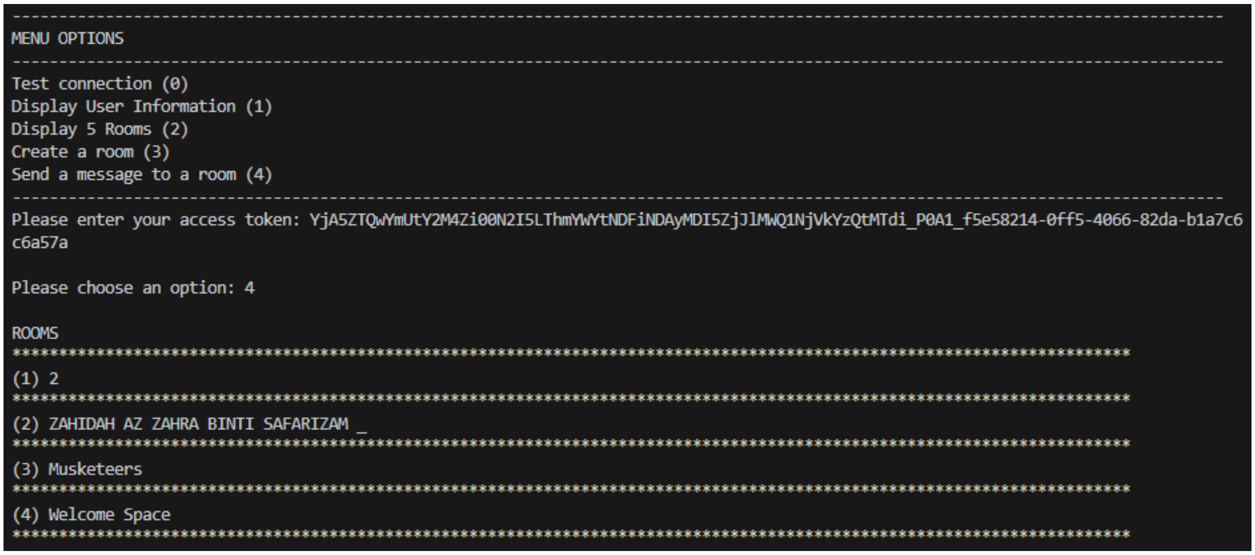
```
-----------------------------------------------------------------------------------------------------------------------------------
MENU OPTIONS
-----------------------------------------------------------------------------------------------------------------------------------
Test connection (0)
Display User Information (1)
Display 5 Rooms (2)
Create a room (3)
Send a message to a room (4)
-----------------------------------------------------------------------------------------------------------------------------------
Please enter your access token: YjA5ZTQwYmUtY2M4Zi00N2I5LThmYWYtNDFiNDAyMDI5ZjJlMWQ1NjVkYzQtMTdi_P0A1_f5e58214-0ff5-4066-82da-b1a7c6
c6a57a

Please choose an option: 2

ROOM INFORMATION
************************************************************************************************************************
Room Id: Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vMzhlMjJhZTAtNzZkMi0xMWVlLThkNmQtZmY0NTMyNDc0N2Y3
Room name: ZAHIDAH AZ ZAHRA BINTI SAFARIZAM _
Room's date created: 2023-10-30T03:12:56.462Z
Last activity: 2, 0, 2, 3, -, 1, 0, -, 3, 0, T, 0, 3, :, 1, 3, :, 1, 5, ., 9, 1, 7, Z
************************************************************************************************************************
Room Id: Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vNzQ2Mjg5ZjAtNDU0Yy0xMWVlLWFhNzgtMzU4MWY1ZmZhZTlm
Room name: Musketeers
Last activity: 2, 0, 2, 3, -, 0, 8, -, 2, 8, T, 0, 3, :, 4, 8, :, 0, 7, ., 7, 4, 7, Z
************************************************************************************************************************
Room Id: Y2lzY29zcGFyazovL3VybjpURUFNOnVzLXdlc3QtMl9yL1JPT00vZDRhNDVmYjAtNDU0Yi0xMWVlLWE0MTUtYjNlOGQ2NTBmNTVh
Room name: Welcome Space
Room's date created: 2023-08-28T02:37:28.747Z
Last activity: 2, 0, 2, 3, -, 0, 8, -, 2, 8, T, 0, 2, :, 3, 7, :, 2, 9, ., 6, 3, 5, Z
************************************************************************************************************************
```

OPTION 3

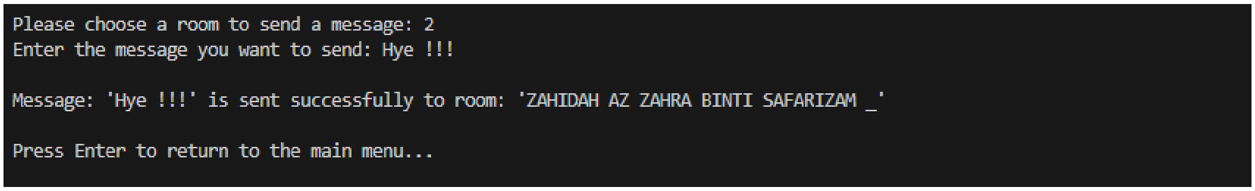


```
------------------------------------------------------------------------------------------------------------------------------------
MENU OPTIONS
------------------------------------------------------------------------------------------------------------------------------------
Test connection (0)
Display User Information (1)
Display 5 Rooms (2)
Create a room (3)
Send a message to a room (4)
Please enter your access token: YjA5ZTQwYmUtY2M4Zi00N2I5LThmYWYtNDFiNDAyMDI5ZjJlMWQ1NjVkYzQtMTdi_P0A1_f5e58214-0ff5-4066-82da-b1a7c6
c6a57a

Please choose an option: 3
Enter the room name you want to create: 2

Room '2' has been created

Press Enter to return to the main menu...
```

OPTION 4



```
---------------------------------------------------------------------------------------------------------------------------
MENU OPTIONS
---------------------------------------------------------------------------------------------------------------------------
Test connection (0)
Display User Information (1)
Display 5 Rooms (2)
Create a room (3)
Send a message to a room (4)
---------------------------------------------------------------------------------------------------------------------------
Please enter your access token: YjA5ZTQwYmUtY2M4Zi00N2I5LThmYWYtNDFiNDAyMDI5ZjJlMWQ1NjVkYzQtMTdi_P0A1_f5e58214-0ff5-4066-82da-b1a7c6
c6a57a

Please choose an option: 4

ROOMS
*****************************************************************************************************************
(1) 2
*****************************************************************************************************************
(2) ZAHIDAH AZ ZAHRA BINTI SAFARIZAM _
*****************************************************************************************************************
(3) Musketeers
*****************************************************************************************************************
(4) Welcome Space
*****************************************************************************************************************
```



```
Please choose a room to send a message: 2
Enter the message you want to send: Hye !!!

Message: 'Hye !!!' is sent successfully to room: 'ZAHIDAH AZ ZAHRA BINTI SAFARIZAM _'

Press Enter to return to the main menu...
```

CONCLUSION

To summarise, developing the troubleshooting tool using the Webex API has effectively achieved the organization's goals. The device not only ensures secure user access but also provides a user-friendly experience with a straightforward menu. Users can efficiently perform tasks such as checking their Webex connection, accessing their details, managing rooms, creating new spaces, and sending messages.

This tool significantly enhances the user's ability to troubleshoot and manage interactions within the Webex platform, making communication and collaboration more efficient. By simplifying tasks like connection verification, room management, and messaging, it becomes a valuable resource for the organization, ultimately boosting productivity and making Webex usage more effective.

REFERENCES

Dudlets, K. (2022). *What Is Webex and How Does It Work? | TelNet Worldwide*. [online] www.telnetww.com. Available at: https://www.telnetww.com/blog/unified-communications/what-is-webex-how-does-it-work/.

AWS Identity And Access Management User Guide. https://usermanual.wiki/Document/AWS20Identity20and20Access20Management2020user20guide.1201199466/help

Go client library for Atlassian Jira. https://reposhub.com/go/third-party-apis/andygrunwald-go-jira.html

Data Binding | DataGrid | WinForms | Syncfusion. https://help.syncfusion.com/windowsforms/datagrid/databinding

A complete list of 2013 - 2014 Microsoft Dynamics AX Monthly webinars with links to recordings. - Microsoft Dynamics 365 Blog. https://cloudblogs.microsoft.com/dynamics365/no-audience/2014/07/31/a-complete-list-of-year-2013-2014-microsoft-dynamics-ax-monthly-webinars-with-links-to-recordings/