

INDEX

Sr No	Date	Name of the Practical	Signature
1		Loading Raspbian on Raspberry Pi and executing applications on it using Python and Node.js	
2		Create a home automation system and control the devices remotely.	
3		Implement Microservices on IoT device	
4		Face Detection using IoT Device	
5		Create the programs using the Microsoft Cognitive APIs for IoT.	
6		Build your own IoT platform	

jabtak tere pair challenge
tumko journal k pages milenge

Table of Contents

PRACTICAL No. 1 : Loading Raspbian and Windows IoT Core on Raspberry Pi and executing applications on it using Python and Node.js.....	3
--	---

jabtak tere pair chalenge
tumko journal k pages milenge

PRACTICAL No. 1 : Loading Raspbian and Windows IoT Core on Raspberry Pi and executing applications on it using Python and Node.js.

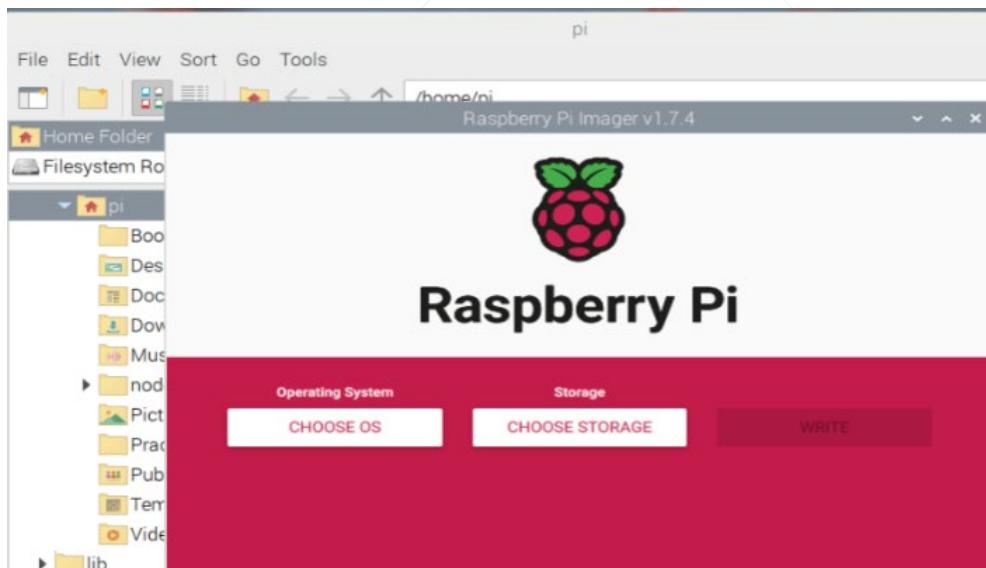
Components used: Raspberry Pi 3B+ Kit, cables.

Procedure:

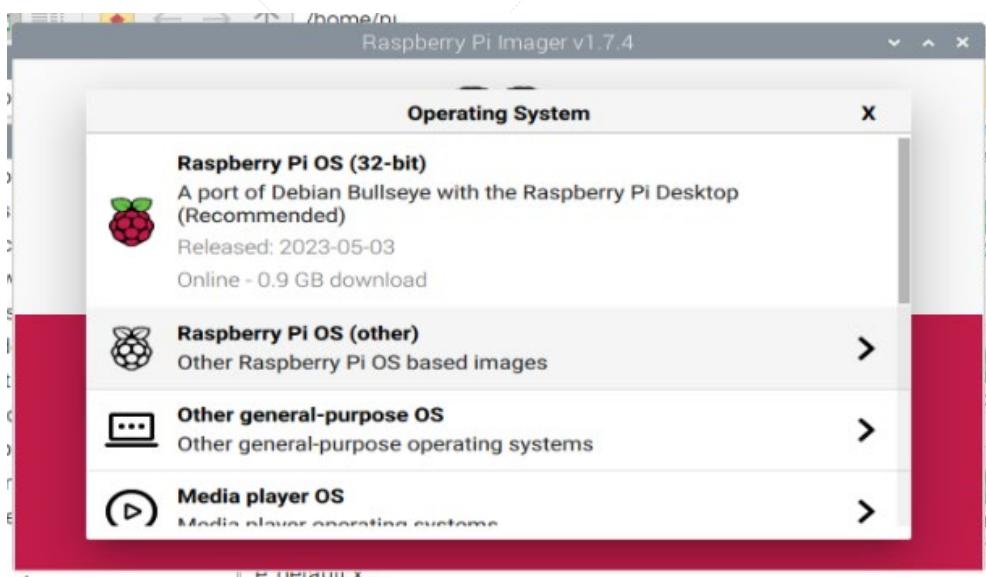
Raspbian is a free and open-source operating system (OS) based on the Debian Linux distribution, specifically designed for the Raspberry Pi single-board computers. The Raspberry Pi is a series of low-cost, credit card-sized computers that can be used for various projects and applications.

Raspbian provides a user-friendly interface and a collection of software packages optimized for the Raspberry Pi's ARM processor architecture. It offers a complete Linux environment with tools and libraries to support programming languages such as Python, C/C++, and Java.

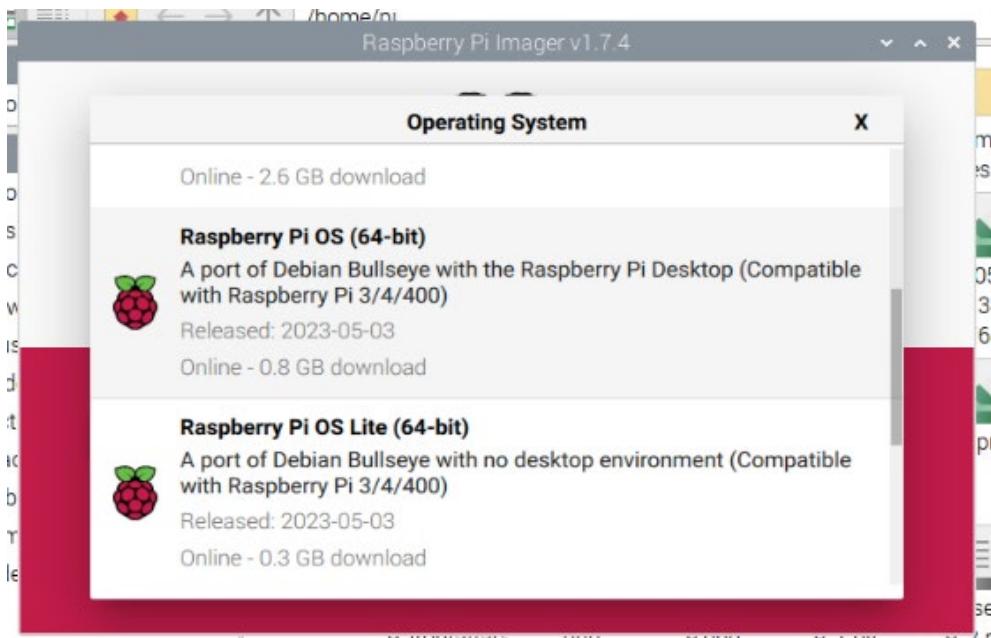
1. To Install Raspberry Pi OS in SD Card, use Raspberry Pi Imager



2. Select 'Choose OS'



3. Now we are only installing Raspberry PI OS only select Raspberry PI OS (OTHER)



4. Choose This Version 64bit (don't select the LITE VERSION)

5. After Connecting all the required connection to the Raspberry Pi now boot it up.

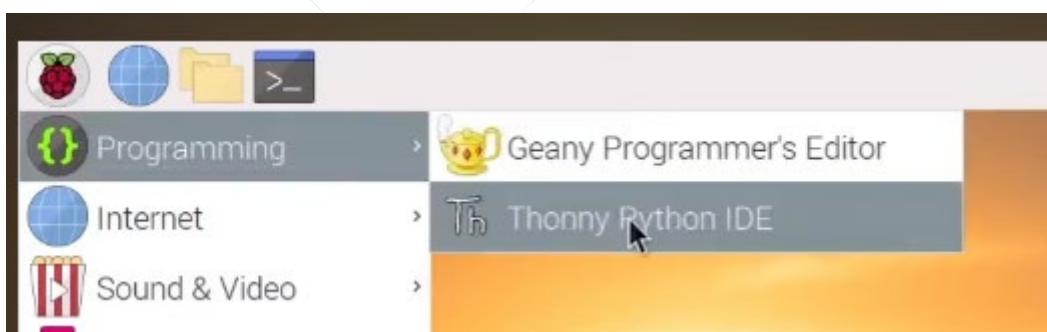
Step 1-->Install Python and Node.js

- For Python Open terminal and type
 - sudo apt update
 - sudo apt install python3 idle3
- Then set the variable path:
 - echo 'export PATH="\$PATH:/home/admin/.local/bin"'>> ~/bashrc
 - In the same terminal run sudo apt-get install nodejs

```
Microsoft Windows [Version 10.0.22621.1778]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sharo>echo 'export PATH="$PATH:/home/admin/.local/bin"'>> ~/bashrc|
```

Step 2-->Open Thonny Python IDE



For Python Code

```
num1 = input("Enter first digit")
num2 = input("Enter second digit")
sum0 = int(num1) + int(num2)
sum1 = int(num1) - int(num2)
sum2 = int(num1) * int(num2)
print("Add ", sum0)
print ("Sub ", sum1)
print ("Multiply ", sum2)
```

For Node.js

```
const readline = require('readline');
const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout
});

rl.question('Enter the first number: ', (num1) => {
    num1 = parseInt(num1);
    rl.question('Enter the second number: ', (num2) => {
        num2 = parseInt(num2);
        const sum0 = num1 + num2;
        const sum1 = num1 - num2;
        const sum2 = num1 * num2;
        console.log('Add:', sum0);
        console.log('Sub:', sum1);
        console.log('Multiply:', sum2);
        rl.close();
    });
});
```

Step 3-->Run both the files in terminal

```
File Edit Tabs Help  
pi@raspberrypi:~ $ python addpy.py  
Enter first digit3  
Enter second digit4  
Add 7  
Sub -1  
Multiply 12  
pi@raspberrypi:~ $ node addnode1.js  
Enter the first number: 4  
Enter the second number: 5  
Add: 9  
Sub: -1  
Multiply: 20  
pi@raspberrypi:~ $
```

jabtak tere pair chalenge
tumko journal k pages milenge

PRACTICAL - 2 : To create a home automation system and control the devices remotely.

Components & Software used: Raspberry Pi kit,cables,breadboard,Telegram Messenger, Telepot

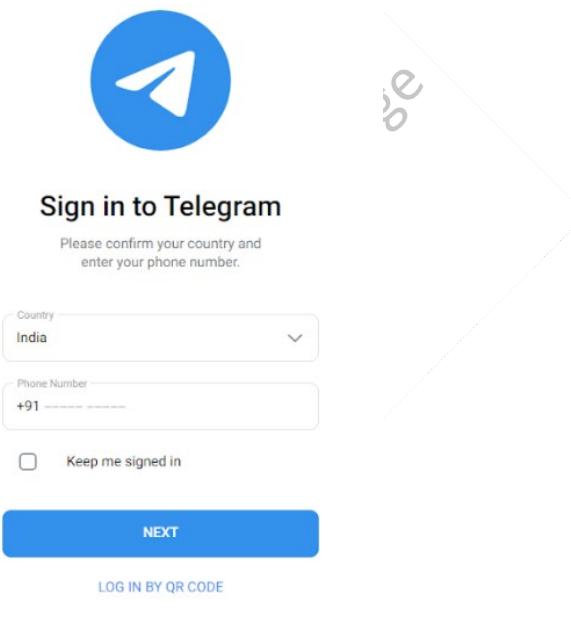
Procedure:

Telepot is a Python library for Telegram Bot API. It provides an easy-to-use interface for developing Telegram bots using the Python programming language. With Telepot, you can send and receive messages, images, documents, and other types of media through the Telegram Bot API.

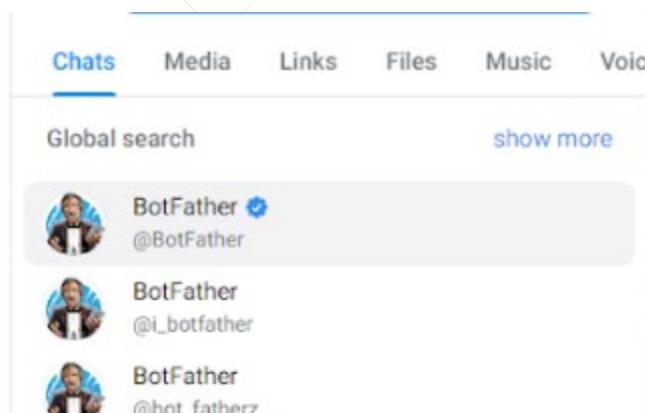
To use Telepot, you need to create a bot on the Telegram platform and obtain an API token.

Steps:

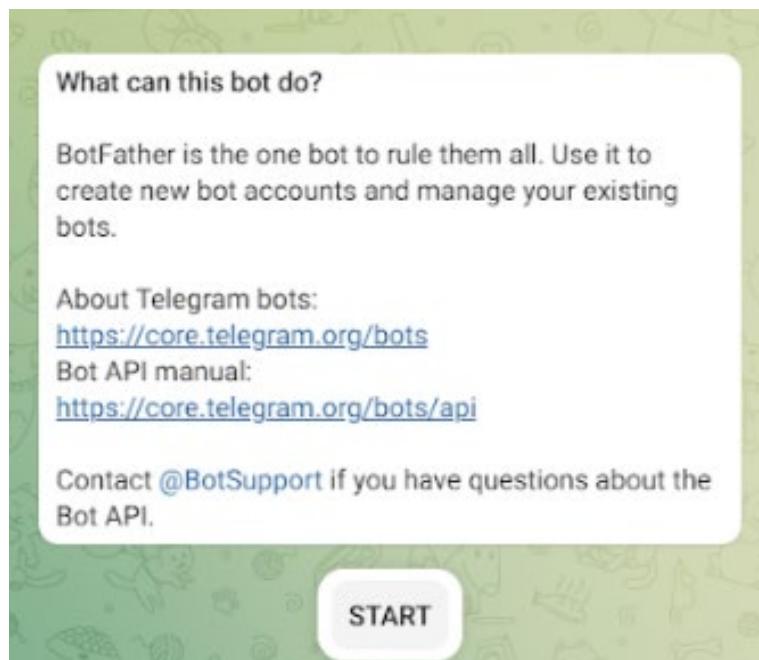
Step 1-->Sign in to Telegram



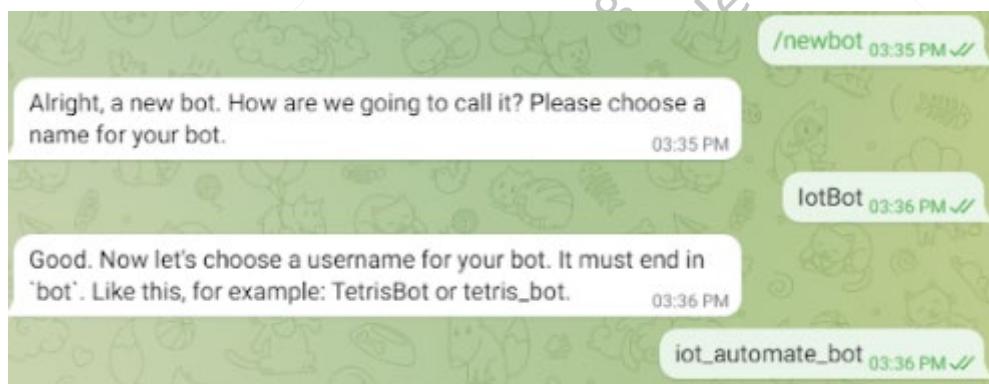
Step 2-->Search for BotFather in Telegram



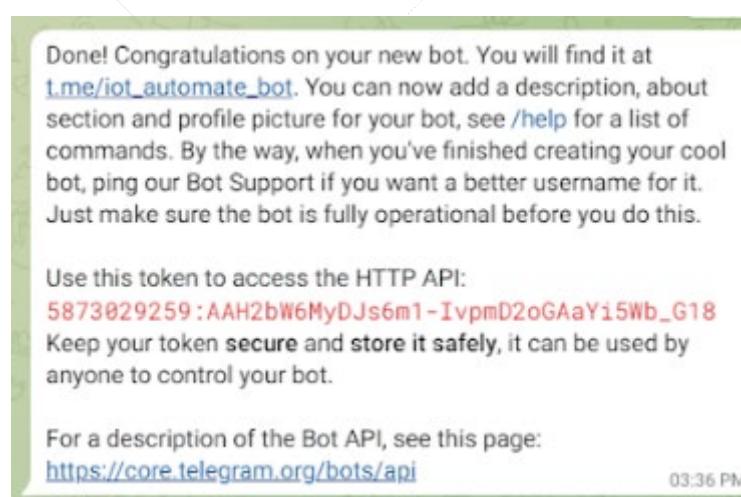
Step 3-->Start the chat with BotChat



Step 4-->Send required Message to create a bot, give name and username to it.



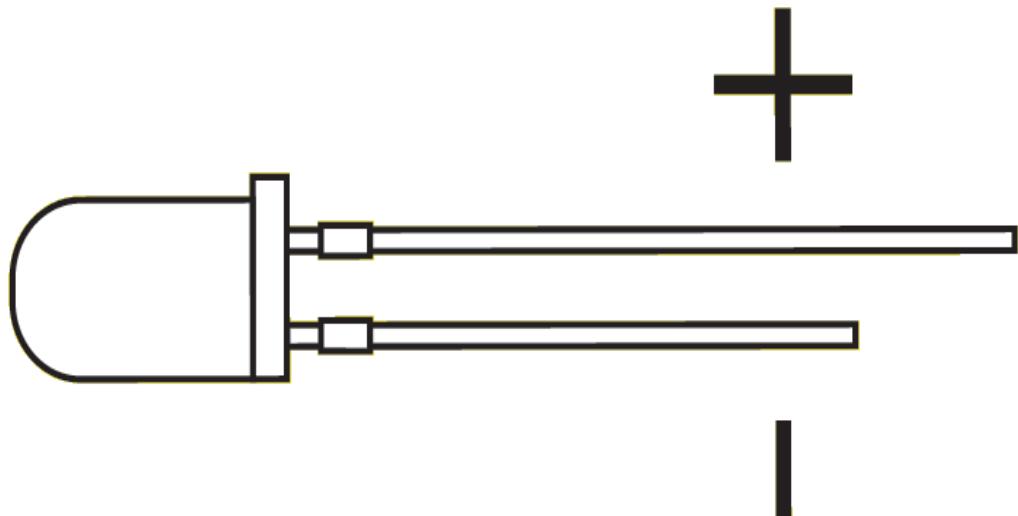
Step 5-->After this BotFather will give a Token to access the HTTP API (BOT)



Now before accessing the bot we created through telegram we need to create a python program in Raspberry Pi with LED BULB Connected to it.



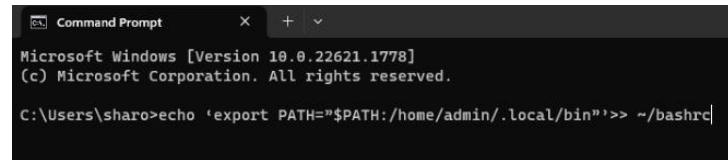
(-)Ground Wire will go to Pin #3
(+) Other Pin will go to Pin #20



Step 6-->After setting up the led bulb to Raspberry Pi boot it up and type this command first to install python first in Raspberry Pi Terminal

- a. sudo apt update
- b. sudo apt install python3 idle3

Step 7-->Then set the variable path



```
Command Prompt
Microsoft Windows [Version 10.0.22621.1778]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sharo>echo 'export PATH=$PATH:/home/admin/.local/bin' >> ~/.bashrc
```

Step 8-->Then install the required library in Raspberry Pi Terminal

1. pip install telepot

Step 9-->Use this python code in Raspberry Pi to access the bot

```
import time

import RPi.GPIO as GPIO
import telepot
from telepot.loop import MessageLoop

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
led_pin = 20
GPIO.setup(led_pin, GPIO.OUT)

def handle(msg):
    chat_id = msg["chat"]["id"]
    command = msg["text"]
    print("Got command: %s" % command)
    bot.sendMessage(chat_id, "Command received")
    try:
        if command == "/on":
            print("on")
            GPIO.output(led_pin, GPIO.HIGH)
        elif command == "/off":
            print("off")
            GPIO.output(led_pin, GPIO.LOW)
    except Exception as e:
        print(e)

bot = telepot.Bot("HTTP-API")
MessageLoop(bot, handle).run_as_thread()
print("I am listening ...")
while True:
    time.sleep(10)
```

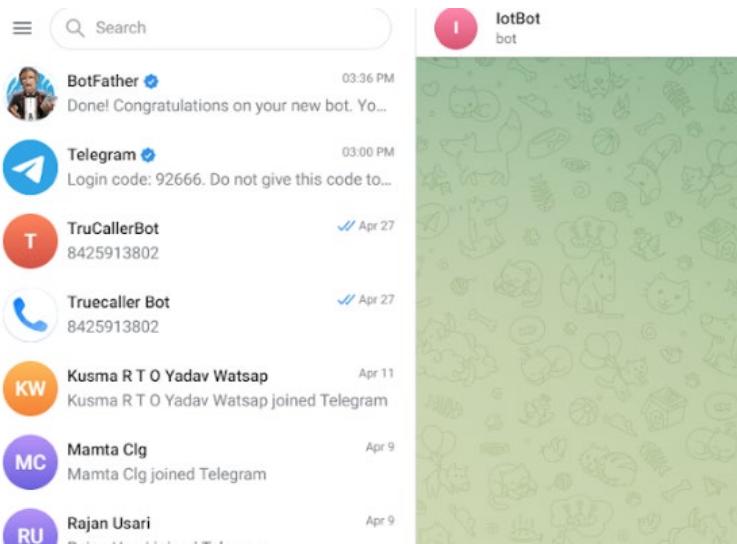
(MAKE SURE TO REPLACE **HTTP-API** with the token that BotFather gave us after giving(created a bot) the username to it.

Use this token to access the HTTP API:

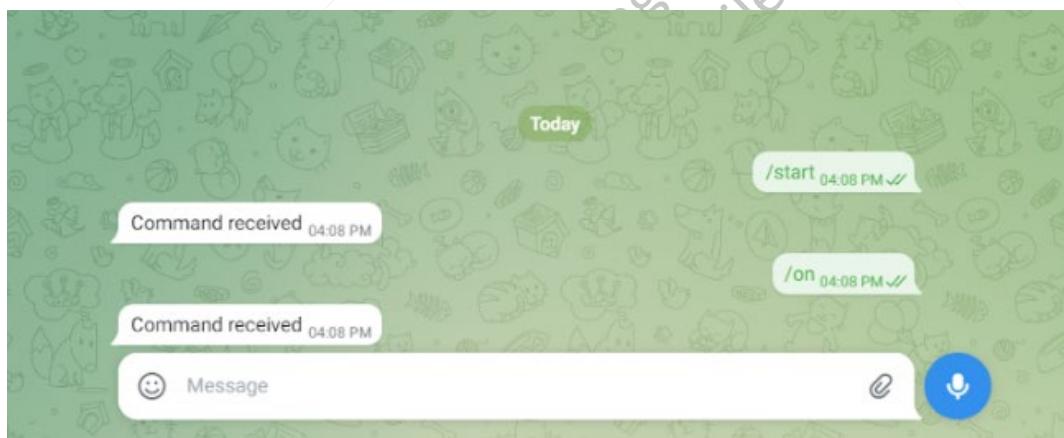
5873029259:AAH2bW6MyDJs6m1-IvpmD2oGAaYi5Wb_G18

Step 10-->After running this code head to the telegram and search for the Name in telegram your bot name you provided

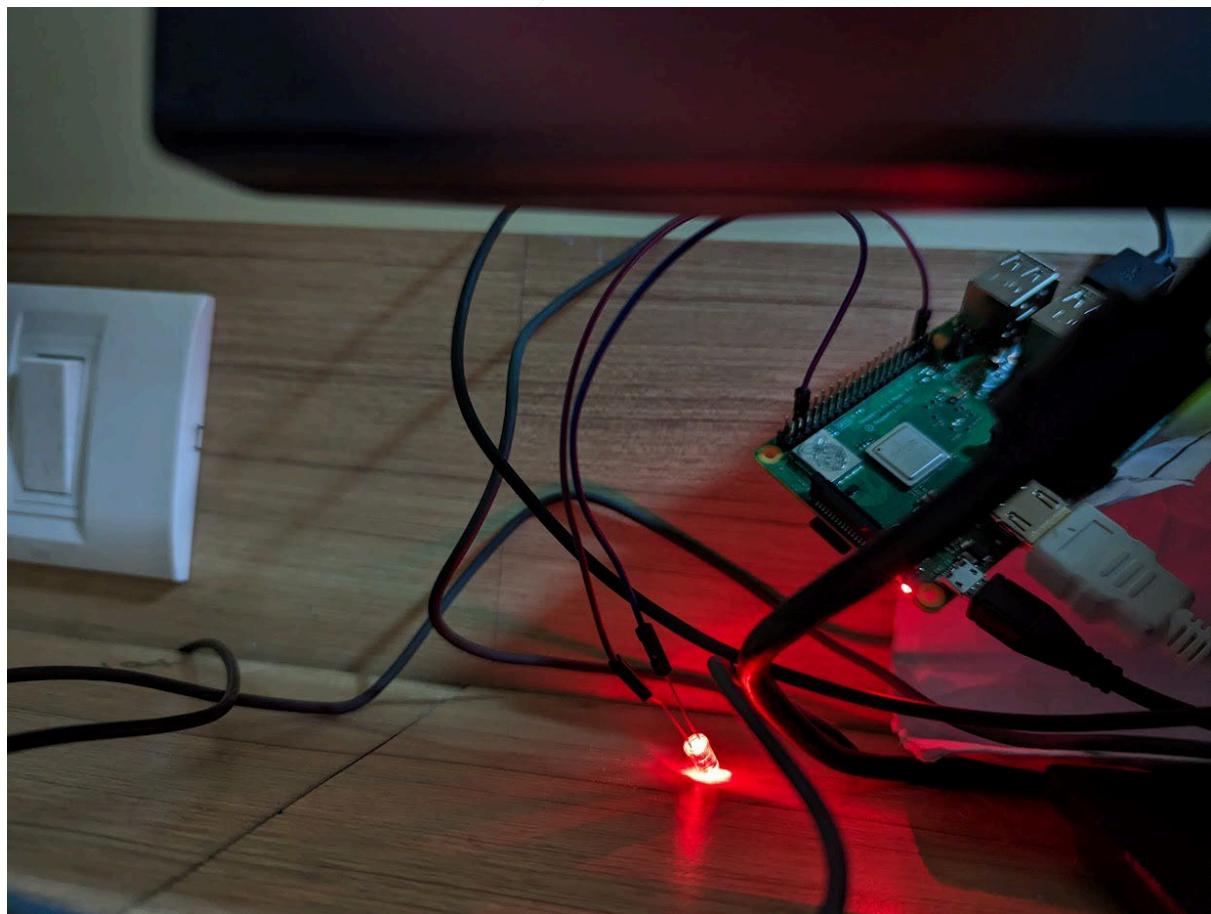
NOTE: NOT THE USERNAME



Start the chat with this bot



```
admin@raspberrypi:~ $ python tele_auto.py
admin@raspberrypi:~ $ python tele_auto.py
I am listening ...
Got command: /start
Got command: /on
on
Got command: /off
off
Got command: /on
on
Got command: /off
off
```



PRACTICAL - 3 : To implement Microservices on IoT device.

Packages used: Flask, jinja2

Description:

1. Flask

Flask is a popular web framework for building web applications using the Python programming language. It is lightweight, flexible, and easy to use, making it a popular choice for both small and large-scale web development projects.

2. jinja2

Jinja2 is a powerful and popular templating engine for Python. It is used in web development frameworks such as Flask and Django to separate the presentation layer (HTML) from the application logic (Python code).

Steps:

Step 1-->Install Flask



```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ pip install --upgrade flask jinja2
pi@raspberrypi:~ $ pip install --upgrade flask jinja2
```

pip install --upgrade flask jinja2

Step 2-->Create 2 Python files

Python files name

- microservice_1.py
- microservice_2 .py

In microservice_1.py code this

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello from Microservice 1!"

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```

In microservice_2.py code this

```
import requests

response = requests.get("ipcode")
print(response.text)
```

Step 3-->Run microservice_1.py first

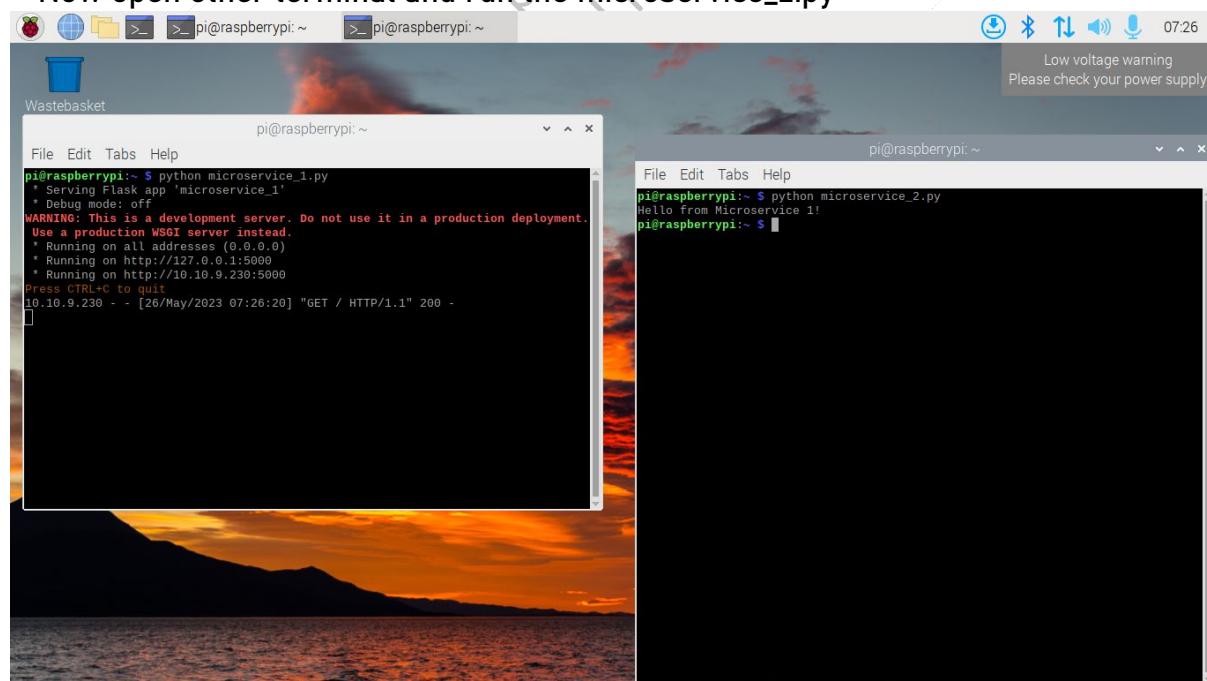
```
pi@raspberrypi:~ $ python microservice_1.py
 * Serving Flask app 'microservice_1'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.10.9.230:5000
Press CTRL+C to quit
```

Step 4-->Copy that `http://<your_ip_address>:5000` and paste it to microservice_2.py where 'ipcode' is written.

```
import requests

response = requests.get("http://<your_ip_address>:5000")
print(response.text)
```

Step 5-->Now open other terminal and run the microservice_2.py



PRACTICAL - 4 : To perform Face Detection using IoT device.

Components used: Raspberry Pi kit, cables, Web cam.

Description:

OpenCV

OpenCV (Open Source Computer Vision Library) is a popular open-source computer vision and image processing library. It provides a wide range of functions and algorithms for tasks such as image and video manipulation, object detection and recognition, feature extraction, camera calibration, and more.

Haarcascade_frontalface_default.xml

This file is a pre-trained Haar cascade classifier XML file that is often used for face detection in computer vision applications. It is a part of the OpenCV library and provides a ready-to-use model for detecting frontal faces in images or video frames.

Haar cascade classifiers are machine learning-based algorithms that can be trained to identify specific objects or features in images. The "haarcascade_frontalface_default.xml" file specifically focuses on detecting frontal faces. It has been trained on a large dataset of positive (images with faces) and negative (images without faces) samples to learn the characteristics of frontal faces.

To use the "haarcascade_frontalface_default.xml" file for face detection, you need to have OpenCV installed and configured in your development environment. Once you have OpenCV set up, you can load the XML file and apply it to your images or video frames to detect faces.

Steps:

Step 1--> Connect a Webcam to the Raspberry Pi with a USB Cable

Step 2--> Install opencv using pip install opencv-python

```
pi@raspberrypi:~ $ pip install opencv-python
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting opencv-python
  Downloading opencv_python-4.7.0.72-cp37-abi3-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (40.4 MB)
    |██████████| 40.4 MB 14 kB/s
Requirement already satisfied: numpy>=1.17.3 in /usr/lib/python3/dist-packages (from opencv-python) (1.19.5)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.7.0.72
```

Step 3--> Type this following python code

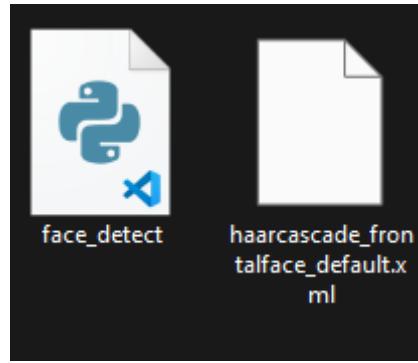
```
import cv2
import numpy as np

# Initialize the camera capture object
cap = cv2.VideoCapture(0) # 0 represents the default camera
# Load the pre-trained face detection model
face_cascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
# Start the main loop to capture frames from the camera
while True:
    ret, frame = cap.read() # Read a frame from the camera
    # Convert the frame to grayscale for face detection
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # Perform face detection
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.3,
minNeighbors=5)
    # Draw rectangles around the detected faces
    for x, y, w, h in faces:
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
    # Display the frame with detected faces
    cv2.imshow("Face Detection", frame)
    # Break the loop if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break
# Release the camera and close the windows
cap.release()
cv2.destroyAllWindows()
```

save it as face_detect.py

Do note we need a haarcascade_frontalface_default.xml along with this python file in the same directory. Download the file from

https://github.com/kipr/opencv/blob/master/data/haarcascades/haarcascade_frontalface_default.xml



Step 4--> Run The Python code

A screenshot of a terminal window on a Raspberry Pi. The terminal shows the command `pi@raspberrypi:~ $ pip install opencv-python` being run, followed by the output of the package installation. Below the terminal is a window titled "Face Detection" showing a video feed of a person's face, which is highlighted with a green bounding box. The background of the desktop shows a sunset over water.

```
pi@raspberrypi:~ $ pip install opencv-python
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting opencv-python
  Downloading opencv_python-4.7.0.72-cp37abi3-manylinux_2_17_aarch64_manylinux2014_aarch64.whl (40.4 MB)
Requirement already satisfied: numpy>=1.17.3 in /usr/lib/python3/dist-packages (from opencv-python) (1.19.5)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.7.0.72
pi@raspberrypi:~ $ python face-detect.py
python: can't open file '/home/pi/face-detect.py': [Errno 2]
pi@raspberrypi:~ $ python face_detect.py
```

PRACTICAL - 5 : Create a program using the Microsoft Cognitive APIs for IOT

Components used: Raspberry Pi kit, cables.

Description:

Creating a program using Microsoft Cognitive APIs for text translation involves harnessing the power of AI to build a language translation solution. By integrating Microsoft's Text Translator API into your program, you can enable automatic translation of text from one language to another with high accuracy and efficiency. The program utilizes advanced natural language processing techniques to analyze and understand the input text, and then generates translated output in the desired target language. This enables seamless communication and understanding across language barriers. With Microsoft's Text Translator API, you can create applications, chatbots, or services that facilitate multilingual communication, content localization, and international collaboration. The program empowers users to easily translate and understand text in different languages, making it a valuable tool for global communication and information exchange.

Step 1--> Go to portal.azure.com

The screenshot shows the Microsoft Azure portal homepage. At the top, there is a navigation bar with various icons and a search bar. Below the navigation bar is the Microsoft Azure logo and a search bar labeled "Search resources, services, and docs (G+ /)". On the right side of the header, there is a user profile icon and the text "jitenra.1234_rg@gmail... DEFAULT DIRECTORY [JITENDRA...]".

The main content area is divided into sections:

- Azure services:** A row of icons for creating a resource, App Service Environments, App Services, Cost Management, Help + support, Key vaults, API Connections, Azure Active Directory, Function App, and More services.
- Resources:** A table showing recent resources. The columns are Name, Type, and Last Viewed. The data is as follows:

Name	Type	Last Viewed
advance-iot	IoT Hub	49 minutes ago
DefaultResourceGroup-southindia	Resource group	53 minutes ago
potatodiseasebeba	Storage account	7 days ago
potato-disease	Function App	7 days ago
Azure for Students	Subscription	a week ago

[See all](#)

- Navigate:** A taskbar at the bottom of the screen showing various pinned application icons.

Step 2--> Click on Resource Groups and on next screen click on create Click on Review + create and press create. This is the resource group for the practical

Microsoft Azure

Search resources, services, and docs (G+)

Home > Resource groups >

Create a resource group

Basics Tags Review + create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

Project details

Subscription * ▼

Resource group * ✓

Resource details

Region * ▼

Step 3--> Now go back to home screen and click on create a resource

Microsoft Azure

Search resources, services, and docs (G+)

portal.azure.com/#home

Microsoft Azure

Search resources, services, and docs (G+)

jtendra.1234.rg@gmail...
DEFAULT DIRECTORY (JTENDRA...)

Azure services

Create a resource + App Service Environments App Services Cost Management ... Help + support Key vaults API Connections Azure Active Directory Function App More services

Resources

Name	Type	Last Viewed
advance-iot	IoT Hub	49 minutes ago
DefaultResourceGroup-southindia	Resource group	53 minutes ago
potatodiseasebeba	Storage account	7 days ago
potato-disease	Function App	7 days ago
Azure for Students	Subscription	a week ago

Recent Favorite

See all

Navigate

https://portal.azure.com/#create/hub

Step 4--> It will direct you to the page where you can access the resources

The screenshot shows the Microsoft Azure portal's 'Create a resource' interface. On the left, there's a sidebar with categories like AI + Machine Learning, Analytics, Blockchain, Compute, Containers, Databases, Developer Tools, DevOps, Identity, Integration, Internet of Things, IT & Management Tools, and Media. The main area has sections for 'Popular Azure services' and 'Popular Marketplace products'. The search bar at the top contains the URL 'portal.azure.com/#create/hub'. The status bar at the bottom shows '13:57 26-05-2023'.

Step 5--> In the search bar search for "Cognitive Services"

The screenshot shows the Microsoft Azure portal's 'Create a resource' page with the search bar containing 'cognitive services'. The search results list items such as 'cognitive', 'cognosys', 'cognizant', 'Speech', and 'Language service'. The rest of the interface is similar to the previous screenshot, including the sidebar, marketplace products section, and status bar.

Step 6--> It will direct you to the Marketplace page where all the resources related to "Cognitive services" under the "Cognitive Service" from Microsoft click on "Create"

The screenshot shows the Microsoft Azure Marketplace interface. A search bar at the top contains the text 'cognitive services'. Below the search bar, there are several filter options: 'Pricing : All', 'Operating System : All', 'Publisher Type : All', and 'Product Type : All'. There is also a checked checkbox for 'Azure services only' and an unchecked checkbox for 'Publisher name : All'. The main area displays a grid of service cards. The first card is 'Cognitive Services' by Microsoft, described as 'Connect powerful AI to your apps'. The second card is 'Bosch Cognitive Services' by Bosch, described as 'SaaS Service to identify your replacement part based on individual training data'. The third card is 'ignio Cognitive Procurement' by ignio, described as 'A Virtual Procurement Analyst to monitor, detect and predict corrective actions'. The fourth card is 'Personalizer' by Microsoft, described as 'An AI service that delivers a personalized user experience'. The fifth card is 'Azure Health Insights' by Microsoft, described as 'Analyzes multimodality patient data provided as input, and outputs insight inferences including evidence intended to support a human decision'. Each service card has a 'Create' button at the bottom.

Step 7--> Fill the Project details "Subscription - Azure for students". Instance details "Region - Central India" and "Name - cognitive-iot".

The screenshot shows the 'Create Cognitive Services' wizard on the Microsoft Azure portal. The 'Project Details' step is active. Under 'Subscription *', 'Azure for Students' is selected. Under 'Resource group *', 'DefaultResourceGroup-southindia' is selected. A note below the subscription dropdown states: 'Cognitive Services resource creation requires subscription registration, we detected that your selected subscription did not register Cognitive Services resource type before, we will help you to register Cognitive Services resource type when you select a subscription in subscription dropdown. Click to learn more how to check registration state for your selected subscription.' The 'Instance Details' step is the next one. Under 'Region', 'Central India' is selected. Under 'Name *', 'cognitive-iot' is entered. A note below the region dropdown states: 'Location specifies the region only for included regional services. This does not specify a region for included non-regional services.' At the bottom, there are buttons for 'Review + create', '< Previous', 'Next : Network >', and 'Give feedback'.

Step 8--> "Pricing tier - Standard"

Pricing tier * Standard S0

By checking this box I acknowledge that I have read and understood all the terms below *

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Cognitive Service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will use this service in accordance with it. This Cognitive Services is intended to process Customer Data that includes Biometric Data (as may be further described in product documentation) that Customer may incorporate into its own systems used for personal identification or other purposes. Customer acknowledges and agrees that it is responsible for complying with the Biometric Data obligations contained in the Online Services DPA.

Review + create < Previous Next : Network > Give feedback

Step 9--> Click on checkbox for acknowledgement.

View full pricing details

By checking this box I acknowledge that I have read and understood all the terms below *

Responsible AI Notice

Microsoft provides technical documentation regarding the appropriate operation applicable to this Cognitive Service that is made available by Microsoft. Customer acknowledges and agrees that they have reviewed this documentation and will use this service in accordance with it. This Cognitive Services is intended to process Customer Data that includes Biometric Data (as may be further described in product documentation) that Customer may incorporate into its own systems used for personal identification or other purposes. Customer acknowledges and agrees that it is responsible for complying with the Biometric Data obligations contained in the Online Services DPA.

Online Services DPA

Responsible Use of AI documentation for Spatial Analysis

Responsible Use of AI documentation for Text Analytics for Health

Responsible Use of AI documentation for Text Analytics PII

Review + create < Previous Next : Network > Give feedback

Step 10--> Select "All network, including the internet, can access this resource".

Configure network security for your Cognitive Services resource.

Type *

All networks, including the internet, can access this resource.
 Selected networks, configure network security for your Cognitive Service resource.
 Disabled, no networks can access this resource. You could configure private endpoint connections that will be the exclusive way to access this resource.

Review + create < Previous Next : Identity > Give feedback

Step 11--> Then click on "Identity" and make the status "off"

Search resources, services, and docs (G+)

Microsoft Azure

Home > Create a resource > Marketplace > Create Cognitive Services ...

Basics Network Identity Tags Review + create

System assigned managed identity

Enable system assigned identity to grant the resource access to other existing resources.

Status Off On

User assigned managed identity

Add user assigned identities to grant the resource access to other existing resources.

+ Add Remove

Name	resource group	subscription
No user assigned managed identities assigned to this resource. Select 'Add' to add more.		

Review + create < Previous Next : Tags > Give feedback

Microsoft Azure Search > Create a resource > Marketplace > Create Cognitive Services

Basics Network Identity Tags Review + create

Tags are name/value pairs that enable you to categorize resources and view consolidated billing by applying the same tag to multiple resources and resource groups. [Learn more about tags](#)

Note that if you create tags and then change resource settings on other tabs, your tags will be automatically updated.

Name	Value	Resource
cognitive-iot	Cognitive Service	

Review + create < Previous Next : Review + create > Give feedback

Step 12--> Click on "Review and create"

Microsoft Azure Search > Create a resource > Marketplace > Create Cognitive Services

Running final validation...

Basics Network Identity Tags Review + create

Basics

Subscription	Azure for Students
Resource group	DefaultResourceGroup-southindia
Region	Central India
Name	cognitive-iot
Pricing tier	Standard S0

Network

Type: All networks, including the internet, can access this resource.

Identity

Create < Previous Next Give feedback Download a template for automation

Step 13--> It will show "Validation Passed".

Validation Passed

Basics Network Identity Tags Review + create

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Azure for Students
Resource group	DefaultResourceGroup-southindia
Region	Central India
Name	cognitive-iot
Pricing tier	Standard S0

Create < Previous Next Give feedback Download a template for automation

Step 14--> Click on "Create" it will show "Submitting Deployment".

Submitting deployment...

Submitting the deployment template for resource group 'DefaultResourceGroup-southindia'.

Basics Network Identity Tags Review + create

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Azure for Students
Resource group	DefaultResourceGroup-southindia
Region	Central India
Name	cognitive-iot
Pricing tier	Standard S0

Create < Previous Next Give feedback Download a template for automation

Step 15--> A message will pop saying "Deployment Succeeded"

The screenshot shows the Microsoft Azure portal with a deployment overview page. The main content area displays a green checkmark icon and the message "Your deployment is complete". Below this, it shows deployment details: Deployment name: Microsoft.CognitiveServicesAllInOne-20230526135738, Start time: 5/26/2023, 1:59:51 PM, Subscription: Azure for Students, Correlation ID: ce3f7894-a47a-47ae-afe6-f71ead1f22d3, and Resource group: DefaultResourceGroup-southindia. To the right of the main content, a sidebar displays a "Deployment succeeded" message: "Deployment 'Microsoft.CognitiveServicesAllInOne-20230526135738' to resource group 'DefaultResourceGroup-southindia' was successful." Below this message are links to "Pin to dashboard" and "Go to resource group". The sidebar also includes sections for "Cost management", "Microsoft Defender for Cloud", "Free Microsoft tutorials", and "Work with an expert". The bottom right corner of the screen shows the Windows taskbar with the date and time (26-05-2023, 14:00).

Step 16--> Click on "Go to resources"

This screenshot is identical to the previous one, showing the Microsoft Azure portal deployment overview page. The main content area shows the deployment is complete with the same details as before. The "Go to resource" button is visible at the bottom of the main content area. The sidebar and taskbar are also present, including the "Cost management", "Microsoft Defender for Cloud", "Free Microsoft tutorials", and "Work with an expert" sections, along with the system date and time (26-05-2023, 14:00).

Step 17--> Now it will direct you to the homepage of the resource made "cognitive-iot".

The screenshot shows the Microsoft Azure portal interface. The URL in the address bar is <https://portal.azure.com/#@jitendra123rg@gmail.onmicrosoft.com/resource/subscriptions/1f441321...>. The page title is "cognitive-iot" under "Microsoft Azure". The main content area displays the "Essentials" section for the "cognitive-iot" resource. Key details include:

- Resource group: DefaultResourceGroup-southindia
- Status: Active
- Location: Central India
- Subscription: Azure for Students
- Subscription ID: 1f441321-673b-4c6b-9aa7-7dfd36e921e3
- Tags: Click here to add tags
- API type: All Cognitive Services
- Pricing tier: Standard
- Endpoint: <https://cognitive-iot.cognitiveservices.azure.com/>
- Manage keys: Click here to manage keys
- Autoscale: Disabled

The "Get Started" tab is selected. Below it, there's a section titled "Build intelligent apps using a comprehensive family of AI services and cognitive APIs" with links to various Azure Cognitive Services like Decision, Language, Speech, Vision, Form Recognizer, Metrics Advisor, and Containers.

Step 18--> Now in the search bar type "Translators" and enter.

The screenshot shows the Microsoft Azure portal search results for "Translators". The search bar at the top contains the text "Trans". The results are categorized into several sections:

- Services**:
 - Translators
 - Azure Maps Creator Resources
 - Azure Maps Accounts
 - Cognitive Services
- Marketplace**:
 - Speech
 - Hugging Face AzureML
 - Translator
 - TicketingBot - IT Support Ticketing System
 - Speech to Text (STT)
 - Power Translate - Power BI .PBIX Translator
 - Scriptix speech to text for automatic transcription and subtitling
 - Matillion ETL
- Documentation**:
 - azURE.ai.translation.text.models.SentenceLength class
 - Use Windows DFS-N to support flexible SAPMINT share creation f...
 - Azure VMs high availability for SAP NetWeaver on SLES for SAP A...
 - Create a Custom Endpoint
 - SAP workload data platform - Microsoft Azure Well-Architected F...
 - Continue searching in Azure Active Directory

The "Language" and "Speech" tabs are visible at the bottom right of the search results page.

Step 19--> Select "Create" or "Create Translator"

The screenshot shows the Microsoft Azure Cognitive Services | Translator interface. On the left, there's a sidebar with categories like Overview, All Cognitive Services, Azure OpenAI, Speech, Language, and Vision. Under the Language category, the 'Translator' option is selected. The main area displays a message: 'No translators to display' with a small icon of two arrows. Below it is a 'Create translator' button. At the top, there are various filters and search options.

Step 20--> Under Project Details "Subscription - Azure for student"

The screenshot shows the 'Create Translator' wizard. The 'Basics' tab is selected. In the 'Project Details' section, 'Subscription' is set to 'Azure for Students' and 'Resource group' is set to 'DefaultResourceGroup-southindia'. A note says: 'Please choose the Global region unless your business or application requires a specific region. Applications that do not offer a region selection use the Global region.' In the 'Instance Details' section, the 'Region' dropdown is empty. At the bottom, there are buttons for 'Review + create', '< Previous', 'Next : Network >', and 'Give feedback'.

Step 21--> Under Instance Details "Region - Central India" and "Name - cognitive-aiot" and "pricing tier - Free F0

communication, and more.

Project Details

Subscription * Resource group * [Create new](#)

Instance Details

Please choose the Global region unless your business or application requires a specific region. Applications that do not offer a region selection use the Global region.

Region *

Name *

Pricing tier *

[Review + create](#) [< Previous](#) [Next : Network >](#) [Give feedback](#)

Step 22--> Click on Review + create. Click on "Create"

Validation Passed

TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Azure for Students
Resource group	DefaultResourceGroup-southindia
Region	Central India
Name	cognitive-aiot

[Create](#) [< Previous](#) [Next](#) [Give feedback](#) [Download a template for automation](#)

Step 23--> It will show a message for successfull deployment and then click on "Go to resource"

The screenshot shows the Microsoft Azure portal interface. The main title bar reads "Microsoft.CognitiveServicesTextTranslation-20230526140147 | Overview". Below the title, there's a message: "Your deployment is complete". A "Deployment name" is listed as "Microsoft.CognitiveServicesTextTransla...". Deployment details include "Subscription: Azure for Students", "Start time: 5/26/2023, 2:03:12 PM", and "Correlation ID: 05b52173-ba9c-49a4-ac5e-8e73df2de...". There are sections for "Inputs", "Outputs", and "Template". A "Deployment details" section is expanded, showing "Next steps". A prominent blue "Go to resource" button is located at the bottom of this section. Below the main content, there are links for "Give feedback" and "Tell us about your experience with deployment". On the right side, there are promotional cards for "Cost management", "Microsoft Defender for Cloud", and "Free Microsoft tutorials". The bottom status bar shows the date and time as "26-05-2023 14:03".

Step 24--> It will direct you to the HomePage (translator).

The screenshot shows the Microsoft Azure portal interface for the "cognitive-aiot" service. The title bar reads "Microsoft.CognitiveServicesTextTranslation-20230526140147 | Overview". The left sidebar lists "Overview", "Activity log", "Access control (IAM)", "Tags", and "Diagnose and solve problems". Under "Resource Management", there are sections for "Keys and Endpoint", "Encryption", "Pricing tier", "Networking", "Identity", "Cost analysis", "Properties", "Locks", and "Monitoring". The main content area is titled "Essentials" and displays resource details: "Resource group (move) : DefaultResourceGroup-southindia", "Status : Active", "Location : Central India", "Subscription (move) : Azure for Students", "Subscription ID : 1f441321-673b-4c6b-9aa7-7dfd36e921e3", and "Tags (edit) : Click here to add tags". It also shows API type "Translator", Pricing tier "Free", and Endpoints "Click here to view endpoints". A "Manage keys" link is provided. Below this, there's a "Try it" section with a "Translate text to another language" button and a note: "This demo runs against your resource and will incur usage against your account". A "From" dropdown set to "Auto detect" and a "To" dropdown set to "Spanish" are shown. The bottom status bar shows the date and time as "26-05-2023 14:03".

Step 25--> Click on "Sample Code" and click on "Python".

The screenshot shows the Microsoft Azure portal interface for a Cognitive Services Text Translation resource named 'cognitive-aiot'. The 'Sample code' tab is selected, and the 'Python' section is active. The code snippet is as follows:

```
1 import requests, uuid, json
```

Step 26--> Copy the code

The screenshot shows the Microsoft Azure portal interface for the same Cognitive Services Text Translation resource. The 'Sample code' tab is selected, and the 'Python' section is active. The code snippet is as follows:

```
1 import requests, uuid, json
2
3 # Add your key and endpoint
4 key = "<your-translator-key>"
5 endpoint = "https://api.cognitive.microsofttranslator.com"
6
7 # location, also known as region.
8 # required if you're using a multi-service or regional (not global) resource. It can be found in the Azure portal on the Keys
9 location = "<YOUR-RESOURCE-LOCATION>"
10
11 path = '/translate'
12 constructed_url = endpoint + path
```

Step 27--> Click on the "Manage Keys"

The screenshot shows the Microsoft Azure portal interface. On the left, there's a sidebar with various service icons like Activity log, Access control (IAM), Tags, and Diagnose and solve problems. Under Resource Management, 'Keys and Endpoint' is selected. In the main content area, the 'Essentials' tab is active, displaying resource group (DefaultResourceGroup-southindia), status (Active), location (Central India), subscription (Azure for Students), and a 'Manage keys' link. Below this, there's a sample code section for Python, showing how to import requests, uuid, and json. The status bar at the bottom right shows the date and time as 26-05-2023 14:05.

Step 28--> Copy a key from "Key1 or Key 2" and "Location"

The screenshot shows the 'Keys and Endpoint' page for the 'cognitive-aiot' resource. The sidebar on the left is identical to the previous screenshot. In the main area, there's a note about the importance of securely storing keys. Below it, there are two sections: 'Show Keys' (KEY 1 and KEY 2) and 'Location/Region' (centralindia). A 'Copy to clipboard' button is available next to each key. At the bottom, there's a 'Web API' section with a note about using specific endpoints. The status bar at the bottom right shows the date and time as 26-05-2023 14:05.

Note: In the code you have to update the "key" and the "location"

Code :

```
import json
import uuid

import requests

# Add your key and endpoint
key = "<your_key>"
endpoint = "https://api.cognitive.microsofttranslator.com"

# location, also known as region.
# required if you're using a multi-service or regional (not global)
# resource. It can be found in the Azure portal on the Keys and Endpoint
# page.
location = "<your_resource_region>

path = "/translate"
constructed_url = endpoint + path

params = {"api-version": "3.0", "from": "en", "to": ["fr", "hi", "zh-
Hans"]}

headers = {
    "Ocp-Apim-Subscription-Key": key,
    # location required if you're using a multi-service or regional (not
    # global) resource.
    "Ocp-Apim-Subscription-Region": location,
    "Content-type": "application/json",
    "X-ClientTraceId": str(uuid.uuid4()),
}

# You can pass more than one object in body.
body = [{"text": "I would really like to drive your car around the block a
few times!"}]

request = requests.post(constructed_url, params=params, headers=headers,
json=body)
response = request.json()

print(
    json.dumps(
        response, sort_keys=True, ensure_ascii=False, indent=4,
        separators=(", ", ": "))
)
```

NOTE: Update the "KEY" and "Location" in this code.

The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - /home/ad...". The menu bar includes "File", "Edit", "Run", "Tools", "Help", and "About". The toolbar features icons for New (green plus), Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. A status bar at the bottom right shows "Local Python 3 • /usr/bin/python3" and the time "14:41".

cognitive.py

```
1 import requests, uuid, json
2
3 # Add your key and endpoint
4 key = "c3f09bc5d8604c8692c489134070ffd3"
5 endpoint = "https://api.cognitive.microsoft.com"
6 |
7 # location, also known as region.
8 # required if you're using a multi-service or regional (not global) resource. It can be found in the Azure portal on the Keys and
9 location = "centralindia"
10
11 path = '/translate'
12 constructed_url = endpoint + path
13
```

Shell

```
Python 3.9.2 (/usr/bin/python3)
>>>
```

Output:

The screenshot shows the Thonny IDE interface. The top bar displays the title "Thonny - /home/ad..." and the status "Thonny - /home/admin/iot/cognitive.py @ 11:20". The menu bar includes "File", "Edit", "Run", "Tools", "Help", and "About". The toolbar contains icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, Quit, and Support. A status bar on the right says "Switch to regular mode". The main window shows a code editor with "cognitive.py" open, containing the following Python code:

```
1 import requests, uuid, json
2
3 # Add your key and endpoint
4 key = "c3f09bc5d8604c8692c489134070fdd3"
5 endpoint = "https://api.cognitive.microsofttranslator.com"
```

Below the code editor is a "Shell" tab with the command:

```
>>> %Run cognitive.py
```

The output in the shell shows the JSON response from the Microsoft Translator API:

```
[{"translations": [{"text": "J'aimerais vraiment conduire votre voiture autour du p\u00e2t\u00e9 de maisons plusieurs fois!", "to": "fr"}, {"text": "मैं वास्तव में कुछ बार दलोक के चारों ओर अपनी कार चलाना चाहता हूं!", "to": "hi"}, {"text": "我真的很想开你的车绕街区转几次！", "to": "zh-Hans"}]}
```

PRACTICAL - 6 : Build your own IoT Platform.

Components used: Raspberry Pi kit, cables, led and jumper wires.

Step 1--> To create the file use "nano"

```
File Edit Tabs Help
admin@raspberrypi:~/iot $ nano iot_platform.py
```

Step 2--> The Python code

Code:

```
import RPi.GPIO as GPIO
from flask import Flask, request

# Define GPIO pin
led_pin = 18

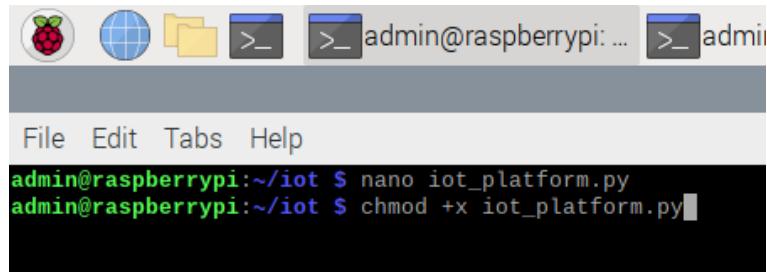
# Set GPIO mode
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(led_pin, GPIO.OUT)

# Create Flask app
app = Flask(__name__)

# Define route to handle HTTP POST request
@app.route('/', methods=['POST'])
def handle_post():
    message = request.get_data(as_text=True)
    if message == "ON":
        GPIO.output(led_pin, GPIO.HIGH)
    elif message == "OFF":
        GPIO.output(led_pin, GPIO.LOW)
    return 'OK'
```

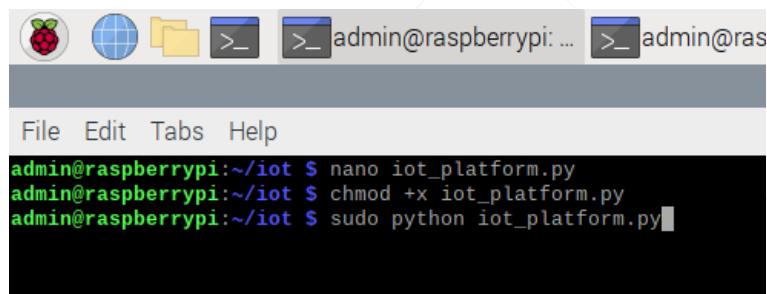
```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

Step 3--> To make the file executable use this command.



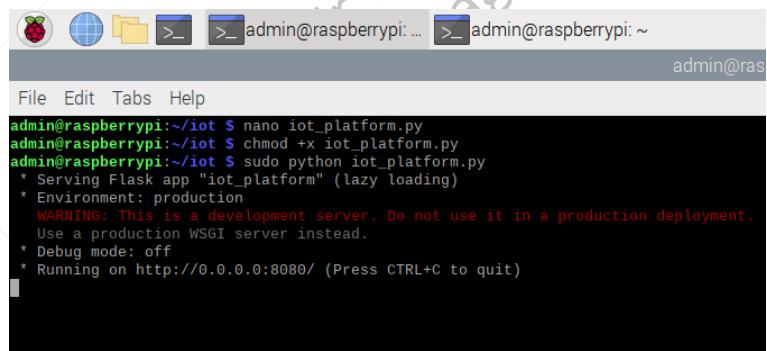
```
File Edit Tabs Help
admin@raspberrypi:~/iot $ nano iot_platform.py
admin@raspberrypi:~/iot $ chmod +x iot_platform.py
```

Step 4--> Execute the file.



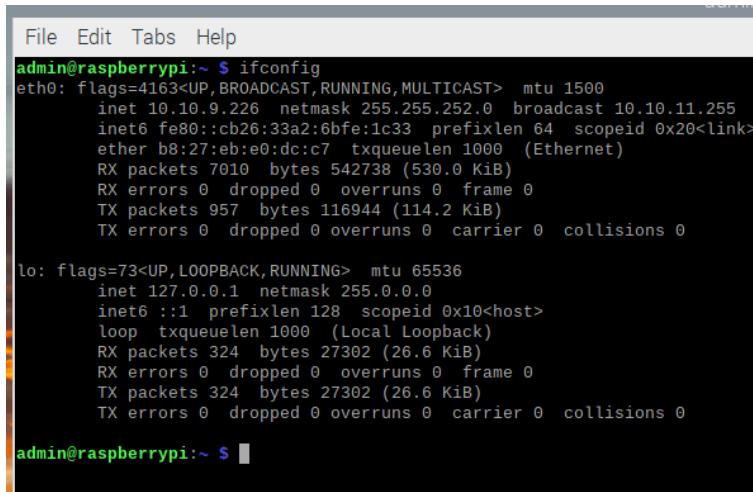
```
File Edit Tabs Help
admin@raspberrypi:~/iot $ nano iot_platform.py
admin@raspberrypi:~/iot $ chmod +x iot_platform.py
admin@raspberrypi:~/iot $ sudo python iot_platform.py
```

Step 5--> With this the server will open



```
File Edit Tabs Help
admin@raspberrypi:~/iot $ nano iot_platform.py
admin@raspberrypi:~/iot $ chmod +x iot_platform.py
admin@raspberrypi:~/iot $ sudo python iot_platform.py
 * Serving Flask app "iot_platform" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

Step 6--> "ifconfig" to get the ip address



```
File Edit Tabs Help
admin@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.10.9.226 netmask 255.255.252.0 broadcast 10.10.11.255
        inet6 fe80::cb26:33a2:6bfe:1c33 prefixlen 64 scopeid 0x20<link>
            ether b8:27:eb:e0:dc:c7 txqueuelen 1000 (Ethernet)
            RX packets 7010 bytes 542738 (530.0 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 957 bytes 116944 (114.2 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

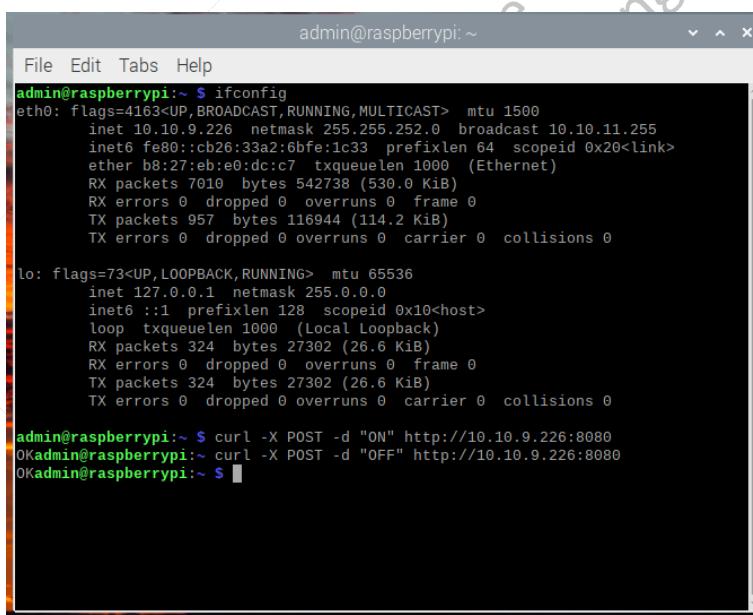
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 324 bytes 27302 (26.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 324 bytes 27302 (26.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

admin@raspberrypi:~ $
```

Output:

NOTE: Make sure you need to open 2 terminals one for running the "iot_platform.py"(server) and second for sending the request to the server.

Here we have sent POST request "ON" to the server

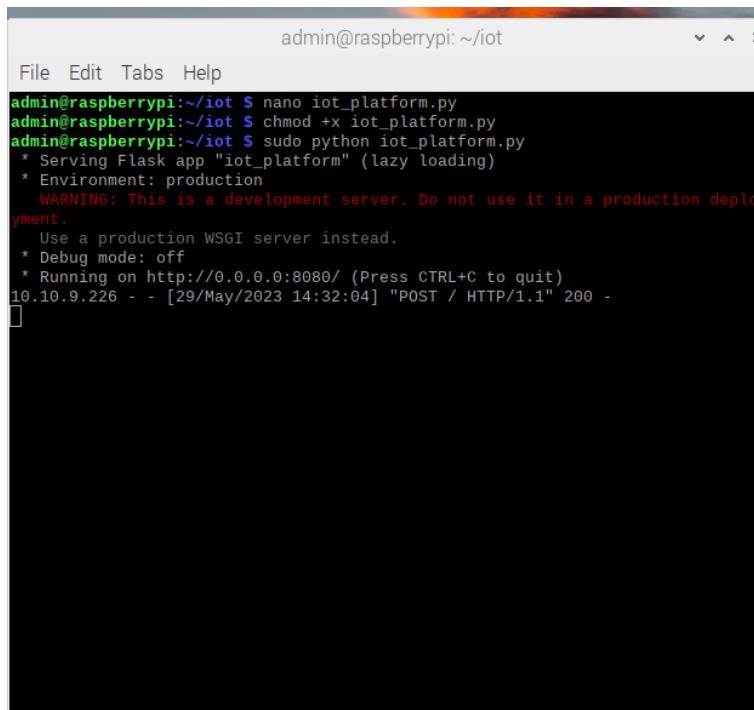


```
File Edit Tabs Help
admin@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 10.10.9.226 netmask 255.255.252.0 broadcast 10.10.11.255
        inet6 fe80::cb26:33a2:6bfe:1c33 prefixlen 64 scopeid 0x20<link>
            ether b8:27:eb:e0:dc:c7 txqueuelen 1000 (Ethernet)
            RX packets 7010 bytes 542738 (530.0 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 957 bytes 116944 (114.2 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
            loop txqueuelen 1000 (Local Loopback)
            RX packets 324 bytes 27302 (26.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 324 bytes 27302 (26.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

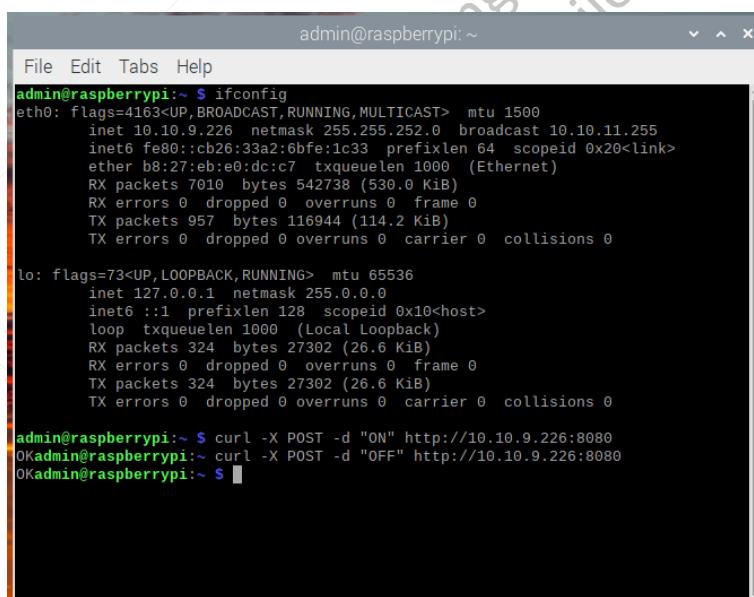
admin@raspberrypi:~ $ curl -X POST -d "ON" http://10.10.9.226:8080
OKadmin@raspberrypi:~ curl -X POST -d "OFF" http://10.10.9.226:8080
OKadmin@raspberrypi:~ $
```

Step 7--> Response for the POST method "ON"



```
admin@raspberrypi:~/iot
File Edit Tabs Help
admin@raspberrypi:~/iot $ nano iot_platform.py
admin@raspberrypi:~/iot $ chmod +x iot_platform.py
admin@raspberrypi:~/iot $ sudo python iot_platform.py
* Serving Flask app "iot_platform" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
    Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
10.10.9.226 - - [29/May/2023 14:32:04] "POST / HTTP/1.1" 200 -
```

Step 8--> Here we have sent POST request "OFF" to the server



```
admin@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.10.9.226 netmask 255.255.252.0 broadcast 10.10.11.255
        inet6 fe80::cb26:33a2:66fe:1c33 prefixlen 64 scopeid 0x20<link>
          ether b8:27:eb:e0:dc:c7 txqueuelen 1000 (Ethernet)
            RX packets 7010 bytes 542738 (530.0 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 957 bytes 116944 (114.2 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 324 bytes 27302 (26.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 324 bytes 27302 (26.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

admin@raspberrypi:~ $ curl -X POST -d "ON" http://10.10.9.226:8080
OKadmin@raspberrypi:~ curl -X POST -d "OFF" http://10.10.9.226:8080
OKadmin@raspberrypi:~ $
```

Step 9--> Response POST method for "OFF"

```
admin@raspberrypi:~/iot
File Edit Tabs Help
admin@raspberrypi:~/iot $ nano iot_platform.py
admin@raspberrypi:~/iot $ chmod +x iot_platform.py
admin@raspberrypi:~/iot $ sudo python iot_platform.py
* Serving Flask app "iot_platform" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
10.10.9.226 - - [29/May/2023 14:32:04] "POST / HTTP/1.1" 200 -
10.10.9.226 - - [29/May/2023 14:33:37] "POST / HTTP/1.1" 200 -
[]
```

jabtak tere pair chalenge
tumko journal k pages milenge