**[AIoT]Smart Mood Light with ChatGPT + VoiceRecognition + TTS**

**TABLE OF CONTENTS:**

1. **INTRODUCTION**
2. **HARDWARE USED**
3. **SOFTWARE USED**
4. **PROGRAMMING LANGUAGES**
5. **APPS AND THE ONLINE SERVICES USED**
6. **WORKFLOW**
7. **SETTING UP WITH PYTHON**
8. **SAMPLE UTTERENCES**
9. **OUTPUT**
10. **CONCLUSION**
11. **INTRODUCTION:**

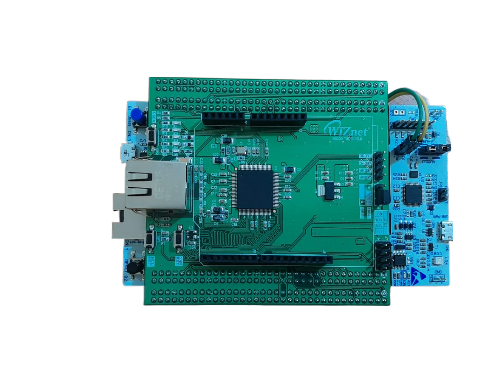
Welcome to the future of lighting innovation – the smart Lighting System! With its cutting-edge technology and seamless integration of voice commands, our system brings your dreams to life with a mere utterance. Imagine a world Where your words can paint your surrounding with vibrant hues and mesmerizing ambiance. With our Smart lighting System, you become the conductor, orchestrating a symphony of colours to suit your mood and preferences. Where Your voices becomes the pallete , and the lights are your canvas.

Introducing “Smart mood lighting system”. This cutting-edge system integrates voice commands and AI -driven text analysis to create a truly personalized and immersive experience. Basically with the Python programming we will give voice commands Which then communicates with the ChatGPT to accurately interpret then on basis of the interpretation it will extract the desired colour code and will take desired actions.

To implement this project, Wiznet-W5300 TOE SHIELD board which, is connected to the LED RGB, With the board we can give the commands to LED to give the particular ambiance of the room.

1. **HARDWARE USED:**

* WIZnet-W5300 TOE SHIELD + STM32-F429ZI board



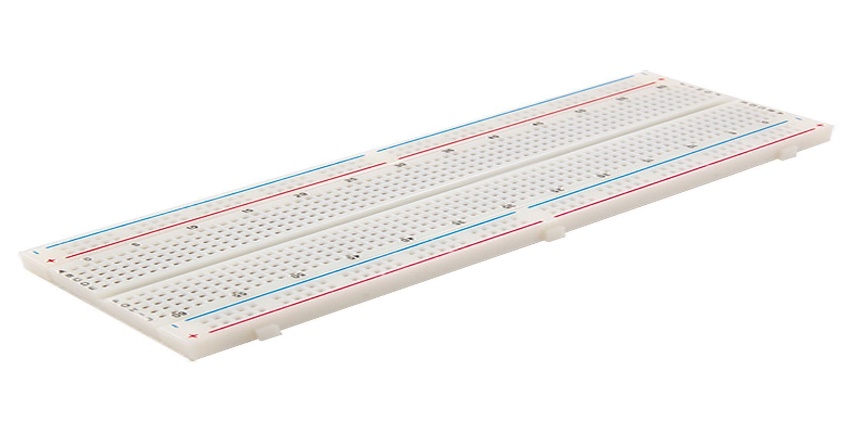
* LED RGB



* JUMPER WIRES



* BREAD BOARD



* MICROPHONE



1. **SOFTWARES AND SERVICES USED**

* MQTT SERVICE
* ARDUINO IDE
* PYCHARM

1. **PROGRAMMING LANGUAGES**

* C++
* PYTHON

1. **APPS AND THE ONLINE SERVICES USED:**
2. MQTT-BROKER
3. OPENAI

**MQTT-BROKER:**

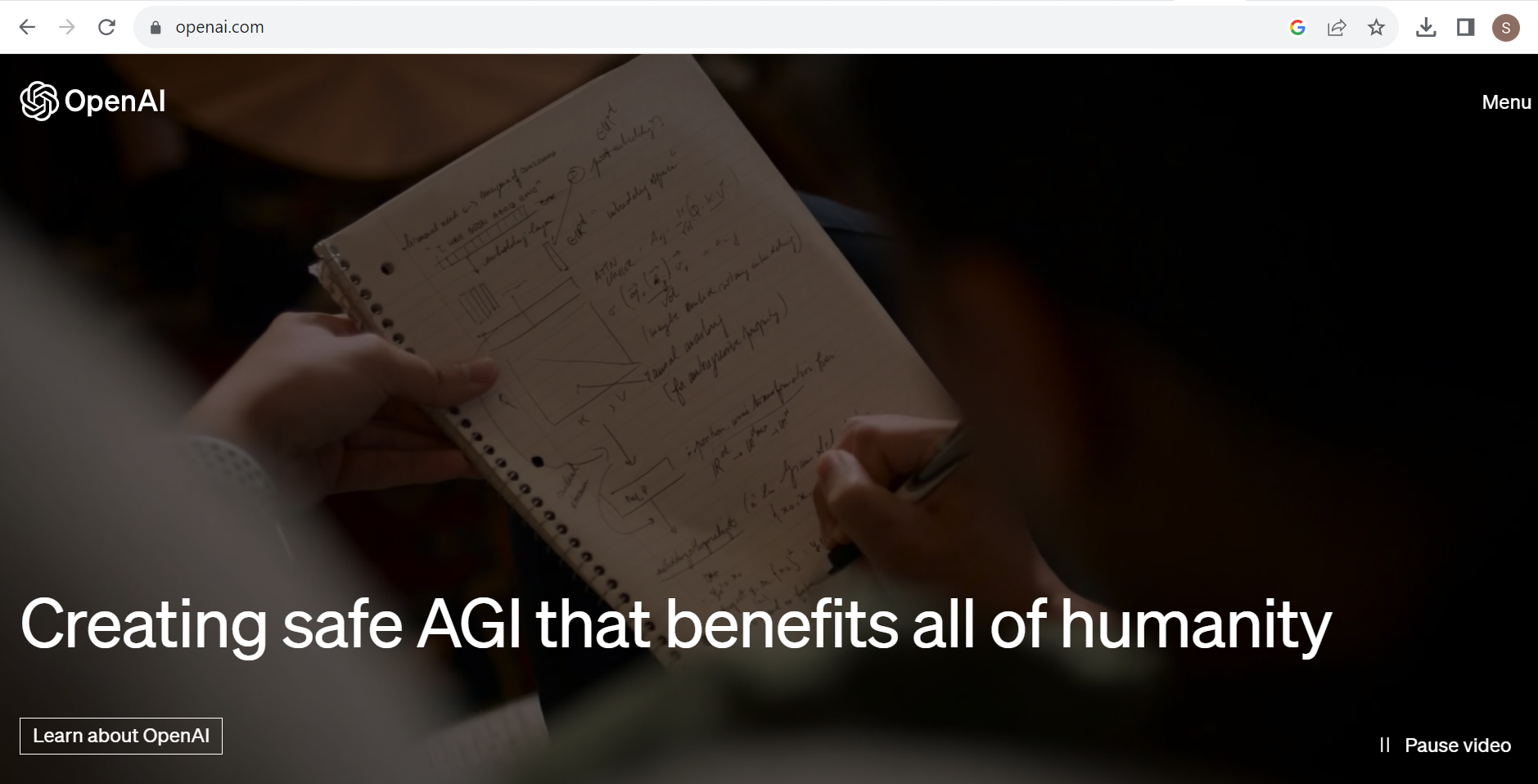
MQTT acts as a broker, facilitating communication between Python and Arduino. Python sends RGB values via MQTT to the Arduino. Arduino, configured to receive MQTT messages, interprets the RGB data and controls the connected RGB LED accordingly. This enables seamless remote control of the RGB LED’s colour using MQTT protocol, bridging Python and Arduino for effective IoT applications.

**OPEN AI :**

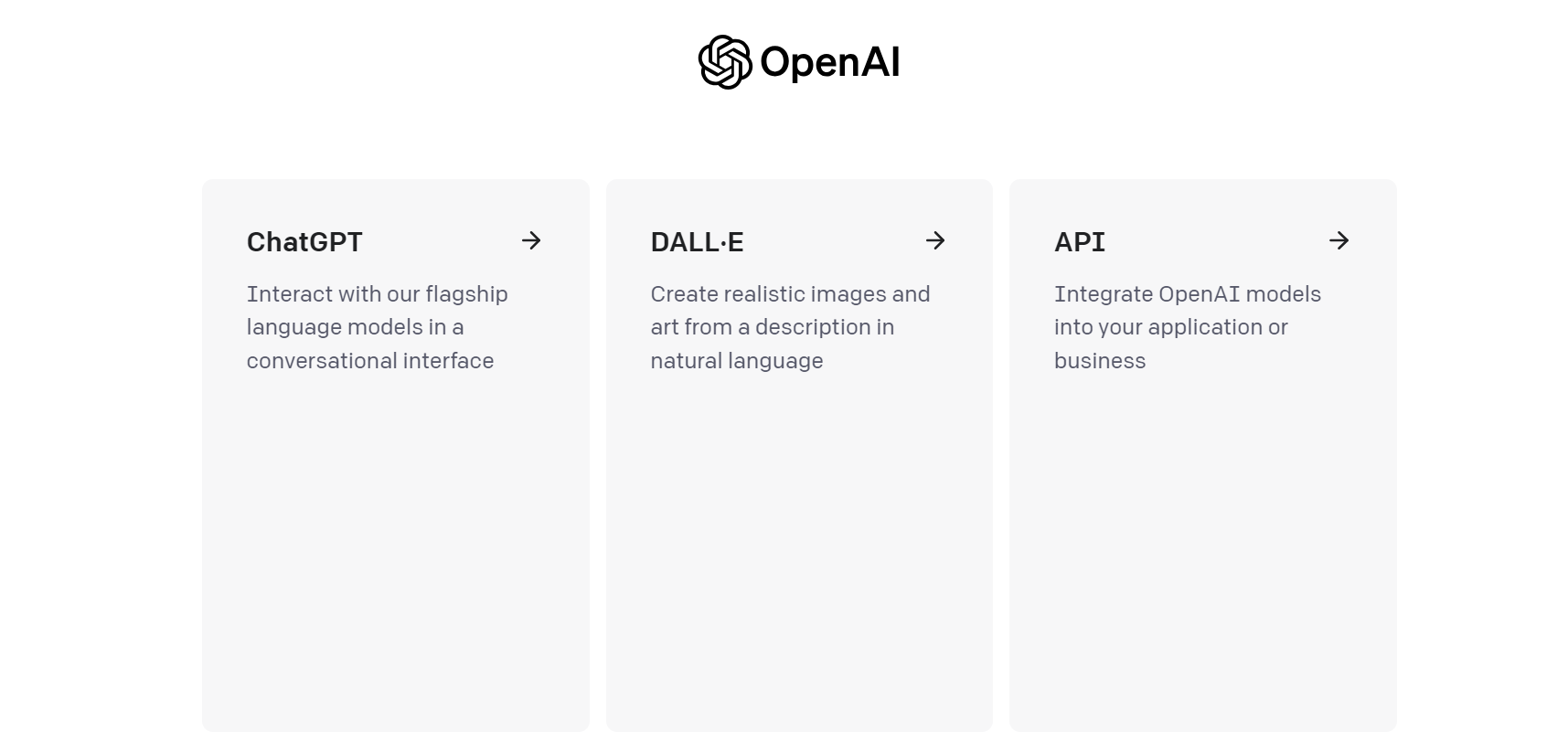
OpenAI provides web services that allow developers to access and utilize its advanced natural language processing models, like the GPT-3. These web services enable applications to integrate powerful language capabilities, such as text generation, language translation, sentiment analysis, and more. Developers can make API calls to interact with these models over the internet, enabling the integration of OpenAI's technology into a wide range of software applications and services.

Setting up with OpenAi :

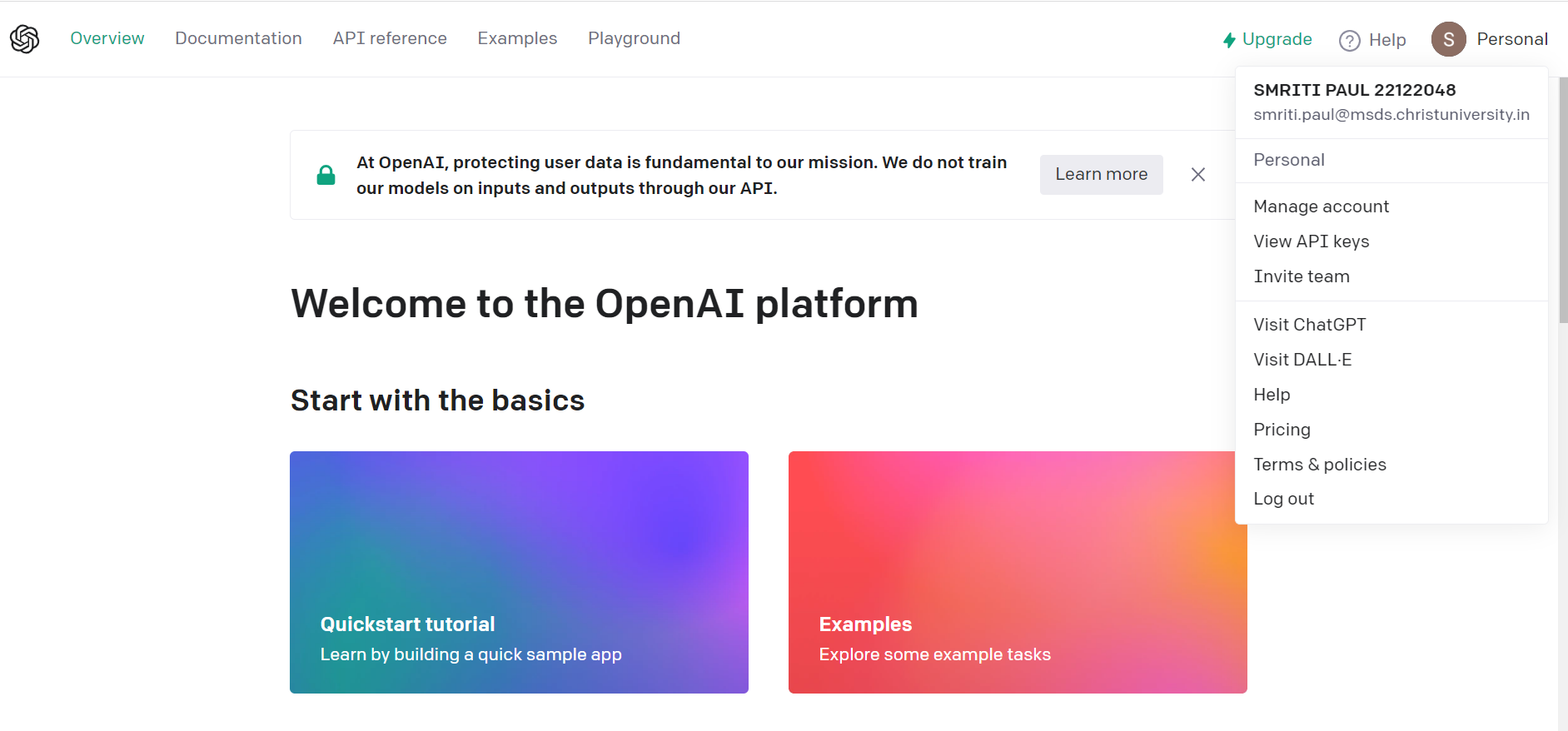
1. <https://openai.com/> Go to this link



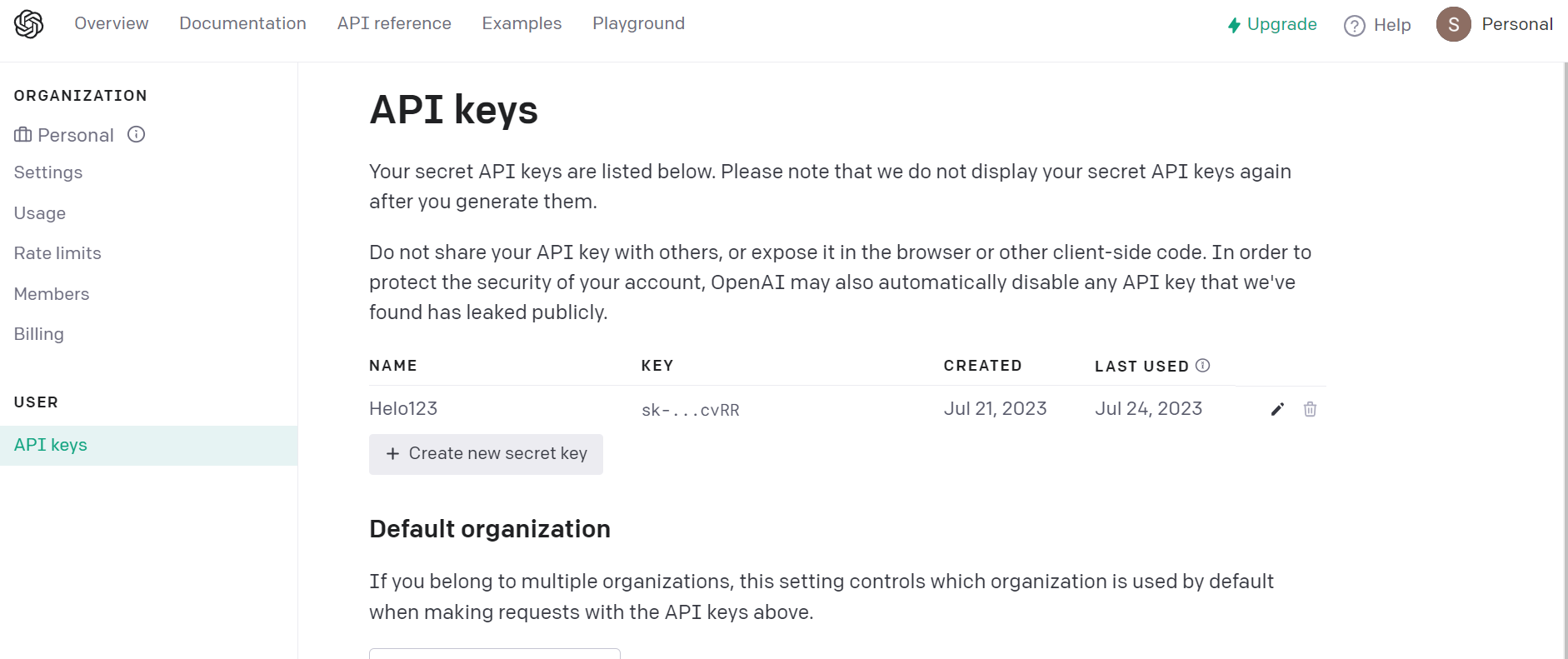
1. Go to menu and then Sign up (Create a new account) or otherwise just Log in

****

1. Then you can see above three options go to API for generating the API Key.



1. Then go to 🡪 Personal 🡪 view API Keys



From Create new secret key option you can create a new API , It will generate once and have take a copy of that.

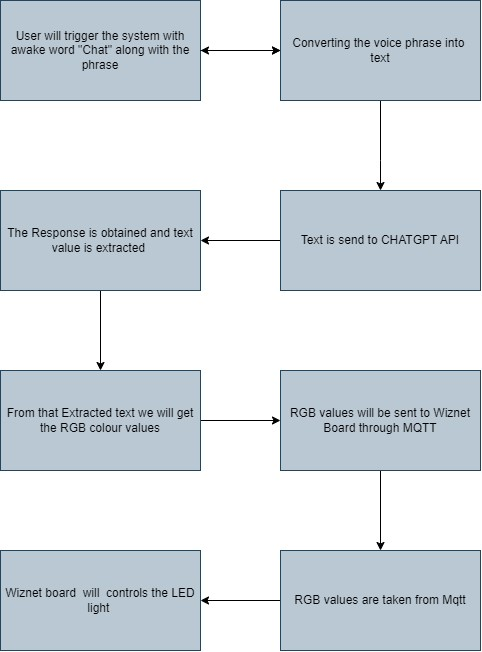
1. **WORKFLOW**

The workflow of this project “Smart Mood light” for turning on different colour on the basis of your desires command where you just gives the names of the shades you want on the basis of your mood and sometimes you just need explain the view and it will interpret the colour from that pharse and turn on the coloured LED, All the you don’t need say turn on for turning on the light.

1. Voice Input: The Systems Starts by receiving voice input from the user. User can provide commands like “I want cherry blossoms view” or simply describe a colour or view they desire. Also Need add the Special Word “Chat” to make the command recognise.
2. Speech -to -text Conversion: The voice input is converted into text using a Speech Recognition library in Python.So This process ensures that system can work with textual data for further analysis.
3. Text analysis: Once the voice command is converted into text, the System will go to ChatGPT inside this where the text analysis will happen through- GPT-3.5 model And give the desire command. This components identifies keywords, colours, and phrases related to views or moods mentioned by the user.
4. Colour interpretation: Based on the analyzed text, the system interprets the desired colour or view. For instance, if the user says “Cherry blossoms view”, the system recognizes the associated pink colour.
5. Colour Mapping: The interpreted colour is then mapped to specific RGB values or colour codes used by the LED lighting system. This mapping ensures that the system can accurately represent the desired colours using the available LEDs.
6. LED Control: The system sends the mapped colour information to the LED control module (WIZnet board), which then activates the corresponding LEDs to emit the desired colour. There is no need for user to explicitly say “turn on the light” since the system automatically interprets the commands.
7. Feedback: The system provides feedback to the user, confirming the successful interpretation and activation of the desired color. This feedback can be in the form of a voice response or visual indicators.
8. Continuous Listening: The Smart Lighting System continuously listens for new voice commands, allowing users to change colors or views on-the-fly by providing new input.

By combining voice recognition, text analysis, colour interpretation, and LED control, the smart Lighting System ensures a seamless and personalized experience, making it a truly innovative and intuitive way to interact with lighting in your living space.

**FLOW CHART:**



**SETTING UP USING PYTHON:**

**Setting up with MQTT:**

import paho.mqtt.client as mqtt

paho.mqtt.client: This library provides a client implementation for MQTT (Message Queuing Telemetry Transport), a lightweight messaging protocol widely used in the Internet of Things (IoT) domain for communication between devices. It allows you to connect to an MQTT broker and publish messages to topics or subscribe to topics to receive messages. It's commonly used for real-time data transmission in IoT applications.

Installation: To install the paho-mqtt library, you can use pip, the Python package manager. Open your terminal or command prompt and run the following command:

pip install paho-mqtt

**Setting up with OPENAI:**

import openai

openai: This library provides a Python interface for the OpenAI API, allowing you to interact with various natural language processing models and services provided by OpenAI. With this library, you can access powerful language models like GPT-3 to perform tasks such as text generation, language translation, question-answering, and more.

Installation: To install the openai library, you can use pip as well. Run the following command in your terminal or command prompt:

pip install openai

**Code:**

import openai

# Set up OpenAI API key  
openai.api\_key = "sk-b7\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*oPb4L60"

def process\_ai\_model(voice\_command):

# Prepare the prompt for the OpenAI API  
  
prompt = f"""Convert the given text to a command among the following options:  
- Turn off the light  
- Turn on the light  
- Change the light to a specific color  
  
Text: {voice\_command}  
  
Conditions:  
1. Find a specific color based on the text even if color is not directly mentioned.  
2. If the text contains a specific color, choose the command "Turn on [color] light".  
3. If the text contains the word "party" or "rainbow," choose the command "Turn on all hue lights."  
4. If the text mentions colors indirectly (e.g., sky-like, ocean-like, ocean blue , sea green ,midnight blue,navy blue ,navi blue ,etc .),  
 map the keywords like (e.g sky ,ocean, sea, midnight,navy, navi etc) to colors and choose the closest matching color. Then, use the command "  
 Turn on [keyword] [closest\_color] or [colour] light."  
5. If the text contains the word "dim" and a color, choose the command "Turn on light [color] with reduced intensity."  
6. If the text does not contain any color, choose either "Turn off the light" or "Turn on the light" command.  
"""  
  
# Call the OpenAI API to generate the AI response  
respond = openai.Completion.create(  
 model="text-davinci-003",  
 prompt=prompt,  
 temperature=0,  
 max\_tokens=100,  
 top\_p=1,  
 frequency\_penalty=0.2,  
 presence\_penalty=0  
)  
  
# Extract the generated text from the response  
generated\_text = respond.choices[0].text.strip()

This is the code for using Open Ai with python with the API key I will set up with python. Then

**Setting Up with Recogniser:**

import speech\_recognition as sr

This library provides an easy-to-use interface to work with various speech recognition APIs. It allows you to convert spoken language into text and supports multiple speech recognition engines, such as Google Web Speech API, Microsoft Bing Voice Recognition, and more.

Installation: To install the speech\_recognition library, you can use pip:

pip install SpeechRecognition

**Setting up with TTS:**

import pyttsx3

This library is a text-to-speech (TTS) engine that allows you to convert text into speech. It's useful for applications where you want your computer or device to speak out information to the user.

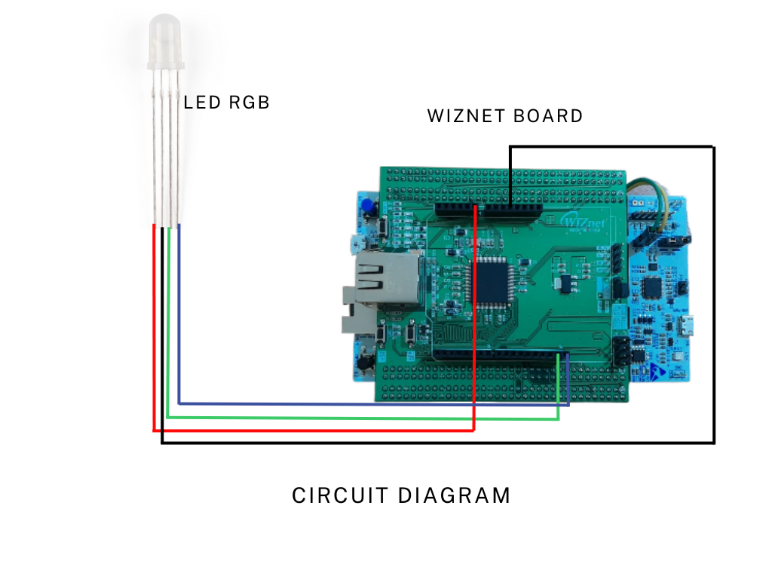
Installation: To install the pyttsx3 library, you can use pip:

pip install pyttsx3

Another two libraries are used here threading, re. “threading” -- Threads are particularly useful when you want to perform tasks simultaneously without blocking the main program's execution. This can improve the responsiveness of your application, especially for I/O-bound tasks.

“re” -- This library is part of Python's standard library and provides support for regular expressions. Regular expressions are a powerful tool for pattern matching and text manipulation. The re library allows you to search, find, and replace specific patterns within strings.

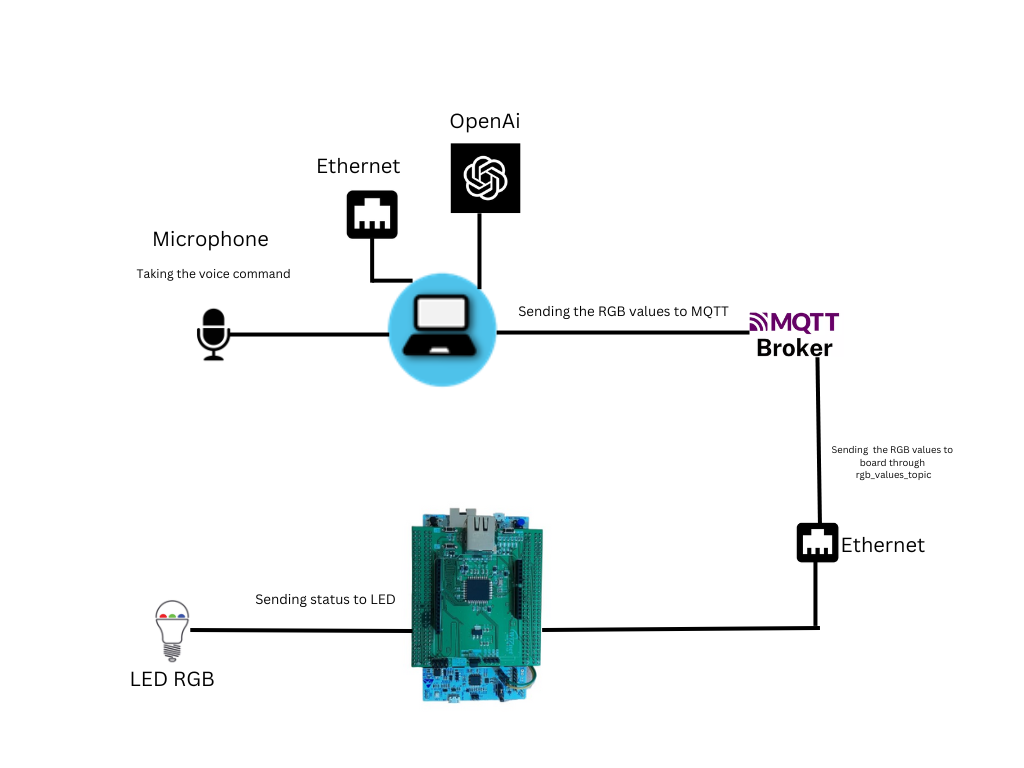
**CIRCUIT CONNECTION:**



LED RGB has 4 pins R (red pin), Common anode, G (green pin), B (blue pin) which is connected with W5100s-EVB-pico board

* Red pin is connected with A0
* VCC is connected with 5V
* Green pin is connected with D14
* Blue pin is connected with D15

**CONNECTION DIAGRAM:**



**SAMPLE UTTERANCES:**

I have tested the below given utterances for opening and closing and it was working fine. More phrases can be possible:

Turning on:

* Chat Turn on the light\4’
* Chat I can’t see anything
* Chat I want to look
* Make it bright
* Chat Make me see
* Chat Turn on

Turning on (colour):

* Chat I want to see cherry blossom’s view

(Turn on the pink light)

* Chat Make it greenery

(Turn on the green light)

* Chat I want to see forest view

(Turn on light green light)

* Chat Please make it sunset like

(Turn on the orange)

* Chat can I have purple sky like view

(Turn on the purple light)

* Chat Turn on magenta

(Turn on magenta light)

* Chat Let’s have a party

(Turn on all hue colours)

Turning off:

* Chat I want darkness
* Chat Turn off
* Chat I want to sleep
* Chat I want it to be dark
* Chat Let there be darkness

**EXAMPLES:**

|  |  |
| --- | --- |
| You said: Chat turn On Magenta  AI Response: Turn on magenta light |  |
|  |  |
| You said: Chat Can I have cherry blossom’s view  AI Response: Turn on pink light |  |
|  |  |
| You said: Chat can I see midnight view  AI Response: Turn on light midnight blue |  |
|  |  |
| You said: Chat Make it lime green  AI Response: Turn on light lime green |  |
|  |  |
| You said: Chat can I have a sunset view  AI Response: Turn on the orange light |  |

**OUTPUT:**

[](https://www.youtube.com/embed/zpPhmPdvTZE?feature=oembed)

Video Link : <https://youtu.be/zpPhmPdvTZE>

**CONCLUSIONS:**

A smart mood light with ChatGPT + voice recognition + TTS is a device that can change its color and brightness in response to a user's voice commands or the content of a conversation. The ChatGPT component allows the light to generate text-based responses to user queries, while the voice recognition and TTS components allow it to understand and respond to spoken commands. This type of light has the potential to be a valuable tool for people with disabilities, as it can provide them with a way to control their environment without having to use their hands. It could also be used to create a more immersive and interactive experience for users of smart home devices.Overall, the combination of ChatGPT, voice recognition, and TTS has the potential to make smart mood lights more versatile and user-friendly. This could lead to increased adoption of these devices by people with disabilities and other users who find traditional smart home controls difficult to use.