

**MOVIE RECOMMENDATION USING VOICE ASSISTANT**

**PYTHON PROJECT REPORT**

***Submitted by***

|  |  |  |
| --- | --- | --- |
| **NITHAN ANTO J** | **:** | **722822104115** |
| **NIRMAL S** | **:** | **722822104114** |
| **KARMUKILAN BP** | **:** | **722822104074** |
| **KAVIN M** | **:** | **722822104079** |

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**SRI ESHWAR COLLEGE OF ENGINEERING**

**(AN AUTONOMOUS INSTITUTION)**

**COIMBATORE – 641 202**

**JULY-AUGUST 2023**

# BONAFIDE CERTIFICATE

Certified that this project report **Movie Recommendation using Voice Assistant** is the bonafide work of

|  |  |  |
| --- | --- | --- |
| **NITHAN ANTO J** | **:** | **722822104115** |
| **NIRMAL S** | **:** | **722822104114** |
| **KARMUKILAN BP** | **:** | **722822104074** |
| **KAVIN M** | **:** | **722822104079** |

who carried out the project work under my supervision

**SIGNATURE**

**Ms. NIRANJANI V, M.E,.(Ph. D)**

**SUPERVISOR**

Assistant Professor,

Dept. of Computer Science & Engineering

Sri Eshwar College of Engineering,

Coimbatore-641202

# ACKNOWLEDGEMENT

We are extremely grateful and remain indebted to my guide **Ms. NIRANJANI V** for being a source of inspiration and for her constant support in the Design, Implementation, and Evaluation of the project. We were thankful to her for her constant constructive criticism and invaluable suggestions, which benefited us a lot while developing the project on “**MOVIE RECOMMENDATION USING VOICE ASSISTANT**”. She has been a constant source of inspiration and motivation for hard work. She has been very cooperative throughout this project work. Through this column, it would be my utmost pleasure to express my warm thanks to her for her encouragement, cooperation, and consent without which I might not be able to accomplish this project.

We also express our gratitude to **Ms. NIRANJANI V** for providing us with the infrastructure to carry out the project and to all staff members of our college who were, directly and indirectly, medium in enabling me to stay committed to the project.

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **1** | **INTRODUCTION** | **1** |
| **2** | **SYSTEM ANALYSIS AND DESIGN** |  |
|  | 2.1 PURPOSE | **2** |
|  | 2.2 SCOPE OF THE PROJECT | **3** |
|  | 2.3 OVERVIEW | **3** |
|  | 2.4 PROBLEM STATEMENT | **3** |
|  | 2.5 LITERATURE REVIEW | **3** |
|  | 2.6 ANALYSIS | **3** |
|  | 2.6.1 FEASIBILITY ANALYSIS | **4** |
|  | 2.6.2 TECHNICAL ANALYSIS | **5** |
|  | 2.7 DATA FLOW DIAGRAM | **6** |
| **4** | **PROPOSED SOLUTION** |  |
|  | 4.1 WORKING | **7** |
|  | 4.2 PROPOSED SYSTEM | **9** |
| **5** | **PROJECT DESCRIPTION** |  |
|  | 5.1 IMPLEMENTATION | **9** |
|  | 5.2 RESULT AND SCREENSHOTS | **14** |
| **6** | **CONCLUSION & FUTURE ENHANCEMENT** | **16** |
| **7** | **REFERENCE** | **17** |

# ABSTRACT

The Movie Recommendation using Voice Assistant project presents a comprehensive study on the development and implementation of an intelligent voice assistant that recommends movies to users based on their preferences and interests. The project aims to enhance user engagement and simplify the movie selection process, considering the vast catalog of available movies. The project utilizes advanced technologies, including Natural Language Processing (NLP), speech recognition, and movie recommendation algorithms, to create a user-friendly and interactive voice assistant. The system collects a large dataset of movies from diverse sources, capturing crucial metadata such as title, genre, director, cast, and synopsis.

The integration of speech recognition technology allows the voice assistant to convert user voice commands into text, enabling seamless communication between the user and the system. Pre-trained models like Google's Speech-to-Text API or the CMU Sphinx library are used for efficient and accurate speech recognition.

The heart of the system lies in the Movie Recommendation Algorithm, which generates personalized movie suggestions. Various recommendation techniques, such as collaborative filtering or content-based filtering, are explored to provide accurate and relevant movie recommendations. The algorithm takes into account the user's history, ratings, and preferences, ensuring that the suggestions cater to the individual's tastes.

## CHAPTER 1

## INTRODUCTION

The Movie Recommendation using Voice Assistant project aims to revolutionize the movie-watching experience by developing an intelligent voice assistant that assists users in finding their ideal movies effortlessly. In today's digital age, the vast and diverse range of movies available across various platforms can be overwhelming for users to navigate. As a result, personalized movie recommendation systems have gained tremendous popularity, as they offer a tailored and user-centric approach to content selection.

The objective of this project is to combine cutting-edge technologies, such as Natural Language Processing , speech recognition, and movie recommendation algorithms, to create an interactive and user-friendly voice assistant. The voice assistant will provide seamless communication between the user and the system, enabling users to search for movies using their natural voice commands.

The proliferation of smart speakers, voice-enabled devices, and virtual assistants has significantly impacted how users interact with technology. Voice assistants have become an integral part of everyday life, streamlining tasks and enhancing user experiences. Incorporating voice commands for movie recommendations not only offers convenience but also adds a touch of novelty to the movie selection process.

Evaluation of the project's success will involve assessing the accuracy of movie recommendations and analyzing user feedback to gauge the system's usability and user satisfaction. Additionally, the report will propose future enhancements to further enrich the voice assistant's capabilities and overall movie-watching experience.

## CHAPTER 2

**SYSTEM ANALYSIS AND DESIGN**

* 1. **PURPOSE:**

The purpose of the report is to present a comprehensive and detailed account of the Movie Recommendation Using Voice Assistant project, showcasing the innovative integration of technologies and its potential to revolutionize the movie-watching experience for users.

**The key purposes of Movie Recommendation using Voice Assistant include:**

**Personalized Movie Suggestions**:

The primary purpose of the Movie Recommendation using Voice Assistant is to provide users with personalized movie suggestions. By leveraging advanced recommendation algorithms, the voice assistant tailors its recommendations based on the user's historical data, preferences, and interactions, ensuring a more relevant and enjoyable movie-watching experience.

**Ease of Movie Selection:**

The voice assistant aims to simplify the movie selection process for users. Instead of manually browsing through a vast catalog of movies, users can simply interact with the voice assistant using natural language commands, making the content discovery process more intuitive and efficient.

**Convenience through Voice Interaction**:

By integrating speech recognition technology, the project enables users to interact with the recommendation system using voice commands. This adds convenience and ease of use, as users can make movie requests hands-free, making the experience more seamless and natural.

**Revolutionizing Movie-Watching Experience:**

The purpose of the project is to revolutionize the movie-watching experience by incorporating voice-enabled technology. By offering personalized recommendations and a user-friendly interface, the voice assistant aims to set new standards for content discovery in the entertainment industry.

* 1. **SCOPE OF THE PROJECT:**

The scope of the Movie Recommendation using Voice Assistant project encompasses the development and implementation of an intelligent voice assistant capable of providing personalized movie recommendations to users. The project aims to cover a wide range of movie genres and cater to diverse user preferences. The voice assistant will be designed to understand and process user queries in natural language, allowing users to interact with the system effortlessly using voice commands.

* 1. **OVERVIEW:**

The Movie Recommendation using Voice Assistant project is a cutting-edge and user-centric system that combines speech recognition, Natural Language Processing , and advanced recommendation algorithms to provide personalized movie suggestions to users. The project's core objective is to simplify the movie selection process for users who face the challenge of navigating a vast catalog of movies

* 1. **PROBLEM STATEMENT:**

The problem statement of the Movie Recommendation using Voice Assistant project is to address the challenge faced by users in selecting movies from a vast and diverse range of available options. With the increasing number of movies across various genres and platforms, users often find it overwhelming and time-consuming to browse through extensive catalogs.

**2.5** **LITERATURE REVIEW:**

The literature review of the Movie Recommendation using Voice Assistant project explores existing research and developments in the fields of movie recommendation systems, voice-enabled technologies, and natural language processing. Various studies have focused on personalized recommendation algorithms, collaborative filtering, and content-based filtering techniques to enhance the accuracy of movie suggestions. Additionally, research on voice assistants, such as Amazon's Alexa and Google Assistant, demonstrates the growing popularity of voice-enabled devices and their integration into daily life.

The literature review also emphasizes the importance of user satisfaction and engagement in recommendation systems. Studies have explored user preferences and the impact of personalized recommendations on user experience.

* 1. **SYSTEM ANALYSIS:**
     1. **FEASIBILITY ANALYSIS:**

The project demonstrates strong feasibility in terms of technical, economic, and operational aspects. From a technical standpoint, the integration of speech recognition, and movie recommendation algorithms is well-established and proven in the industry, ensuring the project's technical viability. The availability of large movie databases and APIs further supports the technical feasibility of data collection.

**Technical Feasibility:**

The integration of speech recognition, NLP, and movie recommendation algorithms is well-established and proven in the industry, ensuring the project's technical viability. The availability of large movie databases and APIs further supports the technical feasibility of data collection.

**Economic Feasibility:**

The project offers significant potential for commercialization and revenue generation. Personalized movie recommendations through voice interactions can enhance user engagement and attract a broader user base for entertainment platforms and content providers. The cost of implementing and maintaining the voice assistant system is reasonable, with a range of pre-existing tools and open-source libraries available for speech recognition and NLP integration.

**Operational Feasibility:**

The project aligns with current trends in consumer behavior, where voice-enabled devices and virtual assistants are increasingly popular. Users are already accustomed to interacting with technology using their voices, making the project's adoption seamless and user-friendly. Moreover, the project's ability to improve user satisfaction and content discovery contributes positively to operational efficiency for content providers, leading to potential partnerships and collaboration.

* + 1. **TECHNICAL ANALYSIS:**

**Data Collection and Preprocessing:**

* Evaluation of data sources and their relevance to the movie recommendation system.
* Assessment of data preprocessing techniques applied to clean and organize the movie dataset.
* Analysis of data quality and its impact on the recommendation algorithm's performance.

**Speech Recognition Technology:**

* Evaluation of the selected speech recognition system, such as Google's Speech-to-Text API or CMU Sphinx.
* Testing the accuracy and robustness of the speech recognition module under different voice inputs.
* Comparison of alternative speech recognition solutions and their suitability for the project.

**Integration of Voice Assistant:**

Assessment of the integration between the speech recognition, NLP, and recommendation modules to create a cohesive voice assistant system.

**Contribution to the Field:**

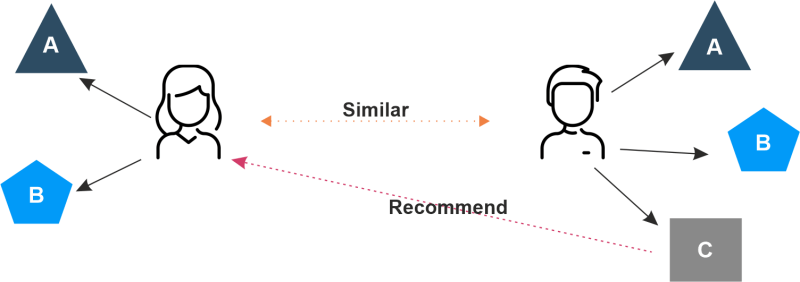
Assessment of the project's contribution to the field of voice-enabled recommendation systems and its potential impact on the entertainment industry.

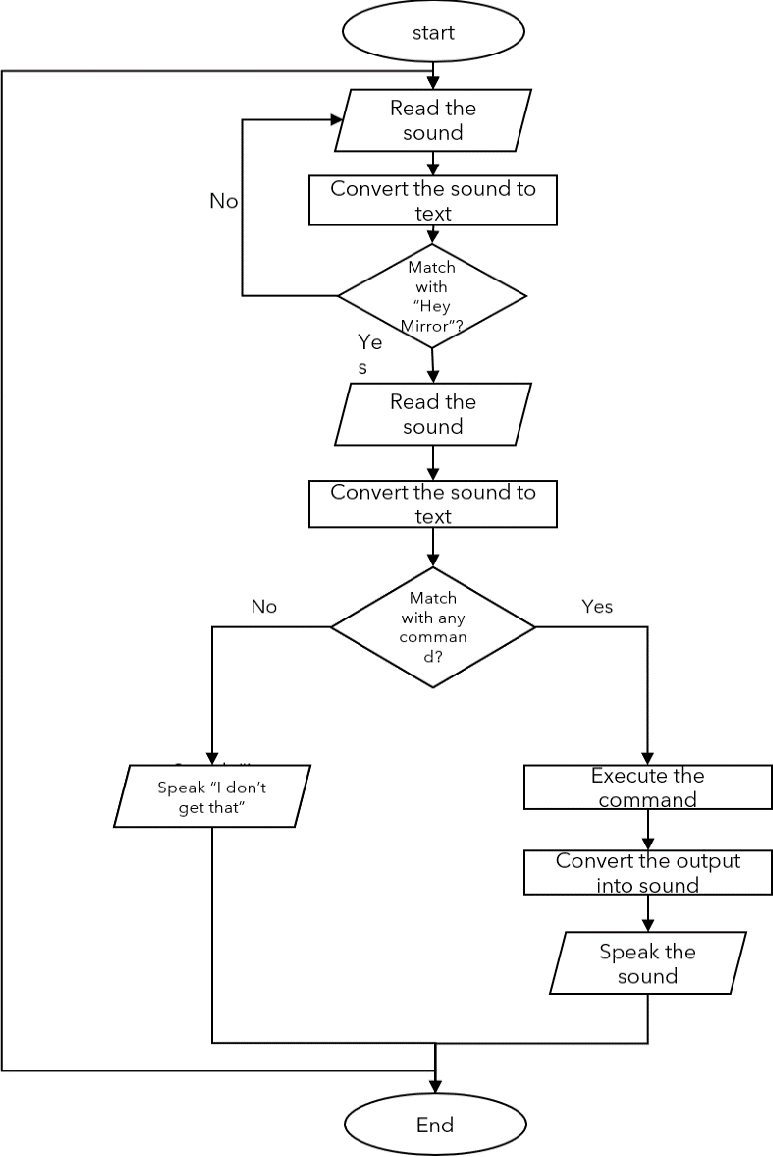
**User Interface and Interaction:**

Analysis of the user interface's design and intuitiveness in interacting with the voice assistant. Evaluation of the system's ability to engage users in natural and conversational interactions.

* 1. **DATA FLOW DIAGRAM:**

8





* 1. **Working:**

**CHAPTER 4**

**PROPOSED SOLUTION**

The proposed solution for the Movie Recommendation using Voice Assistant project is an intelligent and interactive system that seamlessly combines speech recognition, Natural Language Processing , and advanced movie recommendation algorithms. The system will enable users to effortlessly find personalized movie suggestions by interacting with the voice assistant through natural language commands.

The solution involves collecting a comprehensive movie dataset containing essential metadata. By integrating speech recognition technology, the voice assistant will convert user voice commands into text, enabling effective communication. The NLP module will process and interpret user queries, extracting relevant information like genres, actors, directors, and preferences. The core of the system lies in the movie recommendation algorithm, which will use collaborative filtering, content-based filtering, or hybrid approaches to generate tailored movie suggestions based on the user's historical data and interactions.

The system's architecture will facilitate seamless interactions between modules, providing users with an enjoyable and user-friendly movie selection process. The proposed solution aims to revolutionize the movie-watching experience, offering personalized and convenient movie recommendations through voice interactions, and showcasing the potential of advanced technologies in enhancing content discovery.

* 1. **Proposed System:**

The Movie Recommendation using Voice Assistant uses may features like,

**Voice Interaction:**

The system employs Automatic Speech Recognition (ASR) and Natural Language Processing techniques to understand user queries accurately.

**Genre and Mood-Based Recommendations:**

Users can ask for recommendations based on genres like action, comedy, romance, thriller, etc., or ask for suggestions tailored to their mood, such as "recommend a feel-good movie" or "suggest a suspenseful thriller."

**Voice Feedback and Rating:**

After watching a recommended movie, users can provide voice feedback or rate the movie using the Assistant created. This feedback helps the system refine its future recommendations.

**Real-time Updates:**

The system regularly updates its movie database to ensure it includes the latest releases and trending movies. Additionally, it stays up-to-date with user preferences to adjust its recommendations based on recent watching behavior.

## CHAPTER 5

**PROJECT DESCRIPTION**

* 1. **IMPLEMENTATION:**

import openai

import speech\_recognition as sr

import pyttsx3

import os

import json

import tkinter as tk

import pywhatkit as pw

import pyautogui

import threading

import pyaudio

import wave

import webbrowser

import imdb

# OpenAI configuration

openai.api\_key = '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'

personality = "p.txt"

with open(personality, "r") as file:

mode = file.read()

messages = [{"role": "system", "content": f"{mode}"}]

# pyttsx3 setup

engine = pyttsx3.init()

voices = engine.getProperty('voices')

engine.setProperty('voice', voices[1].id) # 0 for male, 1 for female

# Speech recognition setup

r = sr.Recognizer()

mic = sr.Microphone(device\_index=0)

r.dynamic\_energy\_threshold = False

r.energy\_threshold = 400

usewhisper = False

running = True

def whisper(audio):

with open('speech.wav', 'wb') as f:

f.write(audio.get\_wav\_data())

speech = open('speech.wav', 'rb')

wcompletion = openai.Audio.transcribe(

model="whisper-1",

file=speech

)

print(wcompletion)

user\_input = wcompletion['text']

print(user\_input)

return user\_input

def save\_conversation(save\_foldername):

os.makedirs(save\_foldername, exist\_ok=True)

base\_filename = 'conversation'

suffix = 0

filename = os.path.join(save\_foldername, f'{base\_filename}\_{suffix}.txt')

while os.path.exists(filename):

suffix += 1

filename = os.path.join(save\_foldername, f'{base\_filename}\_{suffix}.txt')

with open(filename, 'w') as file:

json.dump(messages, file, indent=4)

return suffix

def save\_inprogress(suffix, save\_foldername):

os.makedirs(save\_foldername, exist\_ok=True)

base\_filename = 'conversation'

filename = os.path.join(save\_foldername, f'{base\_filename}\_{suffix}.txt')

with open(filename, 'w') as file:

json.dump(messages, file, indent=4)

# Main execution

script\_dir = os.path.dirname(os.path.abspath(\_file\_))

foldername = "voice\_assistant"

save\_foldername = os.path.join(script\_dir, f"conversations/{foldername}")

suffix = save\_conversation(save\_foldername)

def is\_user\_speaking():

CHUNK = 1024

FORMAT = pyaudio.paInt16

CHANNELS = 1

RATE = 44100

RECORD\_SECONDS = 1 # Adjust the duration based on your needs

p = pyaudio.PyAudio()

stream = p.open(format=FORMAT,

channels=CHANNELS,

rate=RATE,

input=True,

frames\_per\_buffer=CHUNK)

frames = []

for i in range(0, int(RATE / CHUNK \* RECORD\_SECONDS)):

data = stream.read(CHUNK)

frames.append(data)

stream.stop\_stream()

stream.close()

p.terminate()

audio\_data = b''.join(frames)

audio\_file = 'user\_audio.wav'

with wave.open(audio\_file, 'wb') as wf:

wf.setnchannels(CHANNELS)

wf.setsampwidth(p.get\_sample\_size(FORMAT))

wf.setframerate(RATE)

wf.writeframes(audio\_data)

r = sr.Recognizer()

with sr.AudioFile(audio\_file) as source:

try:

audio = r.record(source)

text = r.recognize\_google(audio)

print("User spoke:", text)

return True

except sr.UnknownValueError:

return False

def loop(window=None, loop\_thread=None):

global user\_input\_var,running

def update\_label\_with\_user\_input(user\_input):

user\_input\_var.set(f"User: {user\_input}")

def stop\_song\_playback():

print("Stopping playback.")

messages.append({"role": "assistant", "content": "Stopping playback."})

pw.stop\_playback()

is\_playing = False

# Main while loop where the conversation occurs

while running:

with mic as source:

print("\nListening...")

r.adjust\_for\_ambient\_noise(source, duration=0.5)

audio = r.listen(source)

try:

if usewhisper:

user\_input = whisper(audio)

else:

user\_input = r.recognize\_google(audio)

print("Recognized:", user\_input)

except sr.UnknownValueError:

print("Could not understand audio")

continue

except sr.RequestError:

print("Could not request results from the speech recognition service. Check your internet connection.")

continue

messages.append({"role": "user", "content": user\_input})

if'play' in user\_input.lower() and 'review' in user\_input.lower():

song = user\_input.split('play')[1].strip()

print(f"Playing video: {song}")

messages.append({"role": "assistant", "content": f"Playing {song} on YouTube."})

if not is\_playing:

threading.Thread(target=pw.playonyt, args=(song,)).start()

is\_playing = True

elif 'stop' in user\_input.lower() and 'review' in user\_input.lower():

print("Stopping playback.")

messages.append({"role": "assistant", "content": "Stopping playback."})

# Stop the playback

pyautogui.press('space')

is\_playing = False

elif 'open' in user\_input.lower() and 'cine web page' in user\_input.lower():

url = "https://cineb.rs"

webbrowser.open(url)

print(f"Opening webpage: {url}")

messages.append({"role": "assistant", "content": f"Opening webpage: {url}"})

else:

completion = openai.ChatCompletion.create(

model="gpt-3.5-turbo",

messages=messages,

temperature=0.8,

max\_tokens=50

)

response = completion.choices[0].message.content

messages.append({"role": "assistant", "content": response})

print(f"\nAssistant: {response}\n")

save\_inprogress(suffix, save\_foldername)

engine.say(response)

engine.runAndWait()

if 'exit' in user\_input.lower():

break

user\_input\_var.set(f"User: {user\_input}")

# Get the latest user input and assistant's response

user\_input = user\_input\_var.get()

assistant\_response = messages[-1]['content']

# Update the Tkinter label with the latest user input and assistant's response

update\_label\_with\_user\_input(user\_input)

user\_input\_var.set(f"Assistant: {assistant\_response}")

# Update the Tkinter window every 100 milliseconds

window.update()

window.after(100)

def main():

global running

window = tk.Tk()

window.configure(bg="lightsalmon")

label = tk.Label(window,text="VOICE ASSISTANT",fg="violet red",bg="lightsalmon")

label.pack(pady=30)

label.config(font=("Arial",50,"bold"))

b = tk.Button(window,text='START',command=lambda:loop(window=window),fg='lemonchiffon2',bg='coral',height=2, width=20,font= ('Helvetica 25 bold italic'))

a=window.title("VOICE ASSISTANT")

b.pack(side="left")

b.place(x=50,y=50)

b.pack(anchor="center")

'''b.grid(row=100,column=200)'''

global user\_input\_var

user\_input\_var = tk.StringVar()

user\_input\_label = tk.Label(window, textvariable=user\_input\_var, font=("Arial", 18))

user\_input\_label.pack(pady=30)

window.mainloop()

running = False

loop\_thread.join()

if \_name\_ == "\_main\_":

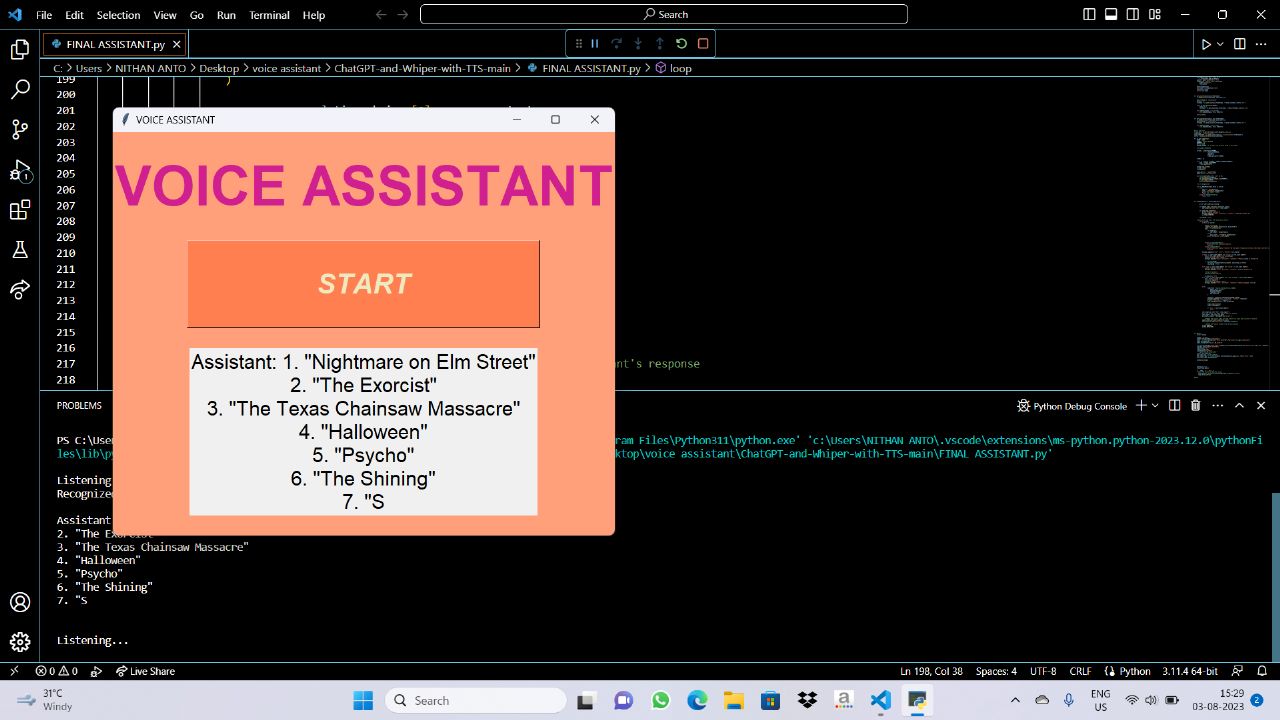
# Start the loop in a separate thread

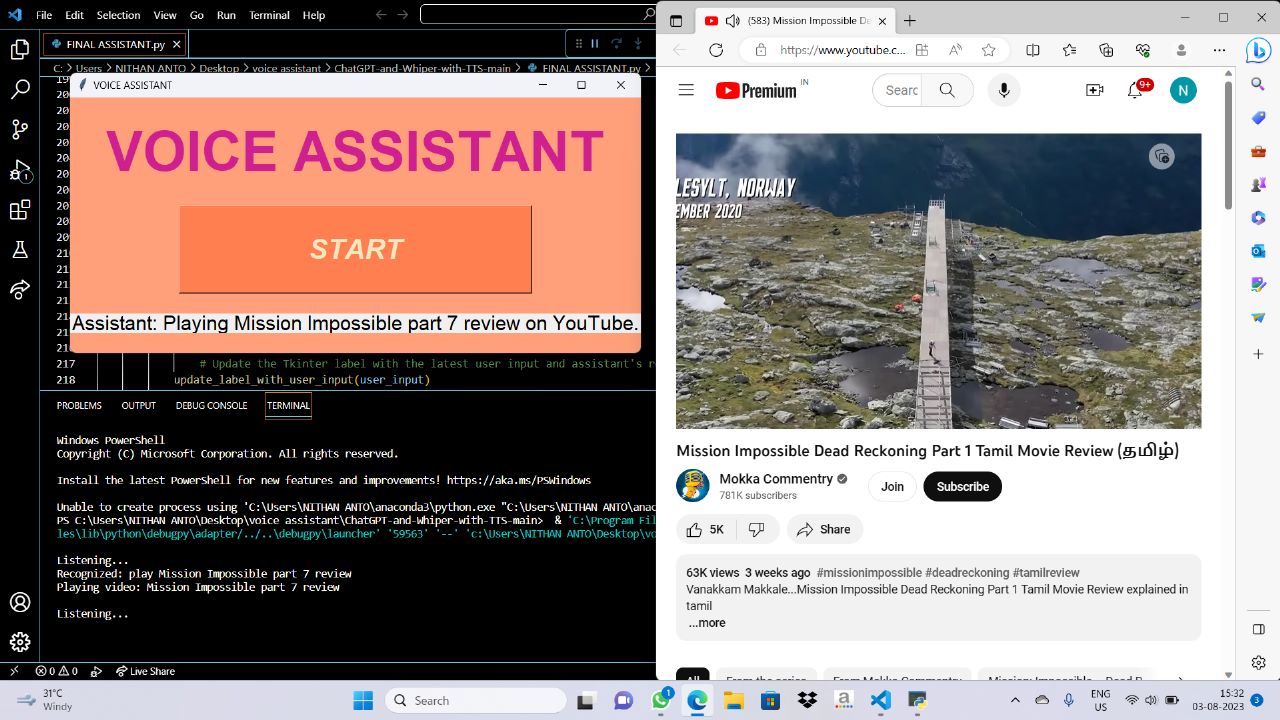
loop\_thread = threading.Thread(target=loop, args=(None, None))

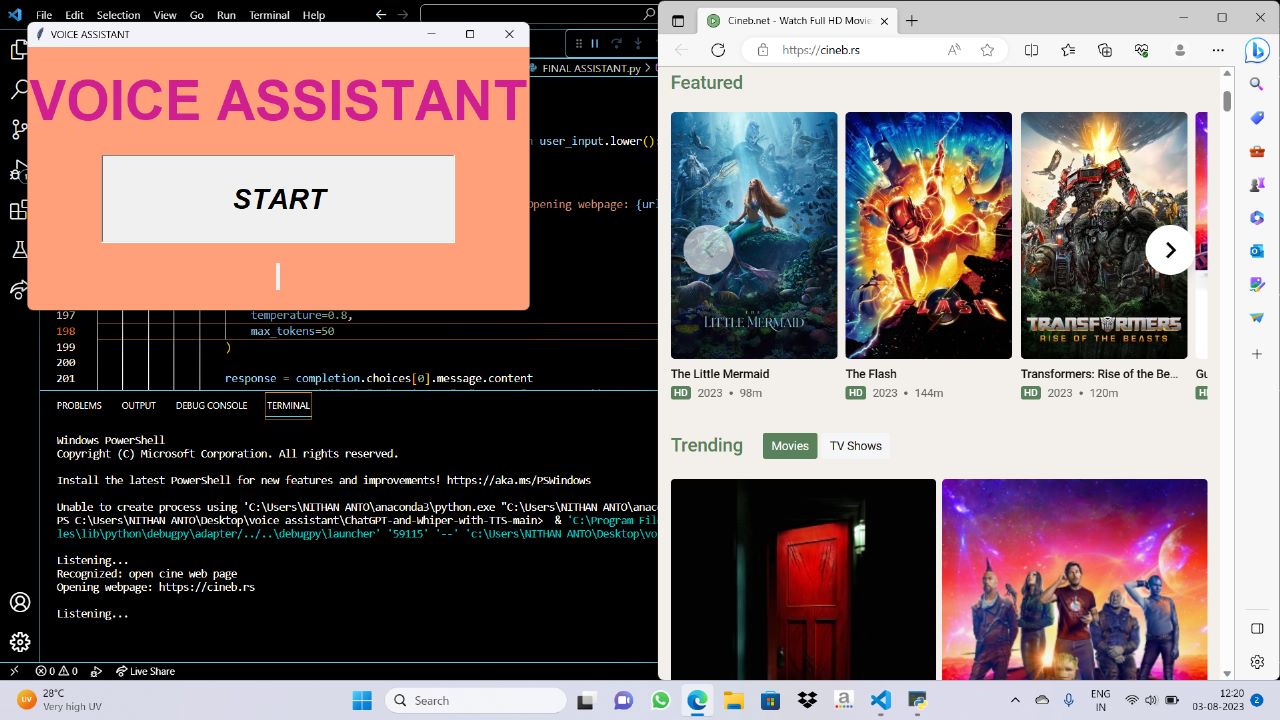
loop\_thread.start()

main()

**5.2** **RESULT AND SCREENSHOTS:**







**CHAPTER 7**

**CONCLUSION & FUTURE ENHANCEMENT**

**CONCLUSION:**

In conclusion, the Movie Recommendation using Voice Assistant project presents a successful and innovative application of advanced technologies to enhance the movie-watching experience for users. By integrating speech recognition, Natural Language Processing (NLP), and movie recommendation algorithms, the voice assistant offers a seamless and personalized approach to movie selection.

Many websites link cineb.com, M4FU.com, ABC Movies etc are some of the good and trustworthy websites where you can watch your desired language movies at any time with good Internet Connection.

**FUTURE ENHANCEMENT:**

For future enhancements, the Movie Recommendation using Voice Assistant project can explore several avenues to further improve its capabilities and user experience. Firstly, incorporating sentiment analysis into the recommendation algorithm can add a context-aware dimension to movie suggestions, considering users' emotions and moods while recommending movies.

This enhancement can lead to more relevant and personalized recommendations that align with users' current feelings. Secondly, the integration of advanced deep learning models for speech synthesis can enhance the naturalness of the voice assistant's responses, creating a more engaging and human-like interaction.

Some of the Future Enhancement in our project are:

* **Advanced Speech Synthesis**: Improves the voice control and develops a real person voice.
* **Multi-Lingual Support:** Extend the voice assistant's capabilities to support multiple languages.
* **Integration with Streaming Platforms:** Integrating the voice assistant with popular streaming platforms like Netflix, Amazon Prime Video, or Hulu can allow users to directly access and watch recommended movies.

**REFERENCE:**

1. <https://www.researchgate.net/publication/355486017_Voice_Assistant_based_Movie_Recommendation_System_utilizing_User_Filtered_Correlation_Factor/download>
2. <https://www.geeksforgeeks.org/voice-assistant-for-movies-using-python/>
3. <https://github.com/topics/movies-recommendation?o=desc&s=stars>
4. <https://www.digitaltrends.com/home/the-best-ai-assistants/>
5. <https://www.amazon.science/publications/two-stage-voice-application-recommender-system-for-unhandled-utterance-in-intelligent-personal-assistant>
6. <https://youtu.be/6xHg66qGjdQ?si=WDSpLjx5GoLtd4xJ>
7. <https://youtu.be/AWvsXxDtEkU?si=2tzepbrp2DwXnBO9>