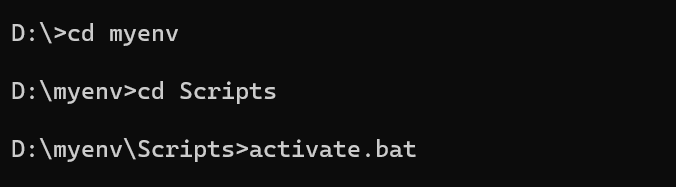
First check weather python is properly installed or not then follow the below steps

1. Set up a virtual environment and activate it using the following command.

**python -m venv myenv**

Then activate it by using **myenv\Scripts\activate.bat**

if it is properly created and activated then you should get like this

****

1. Install the required dependencies: Flask, Flask-RESTful, Flask-HTTPAuth and requests. Use the below command to install

**pip install flask flask-restful flask-httpauth requests**

if you get any errors then update your pip version using the below command

**D:\C++\myenv\Scripts\python.exe -m pip install --upgrade pip**

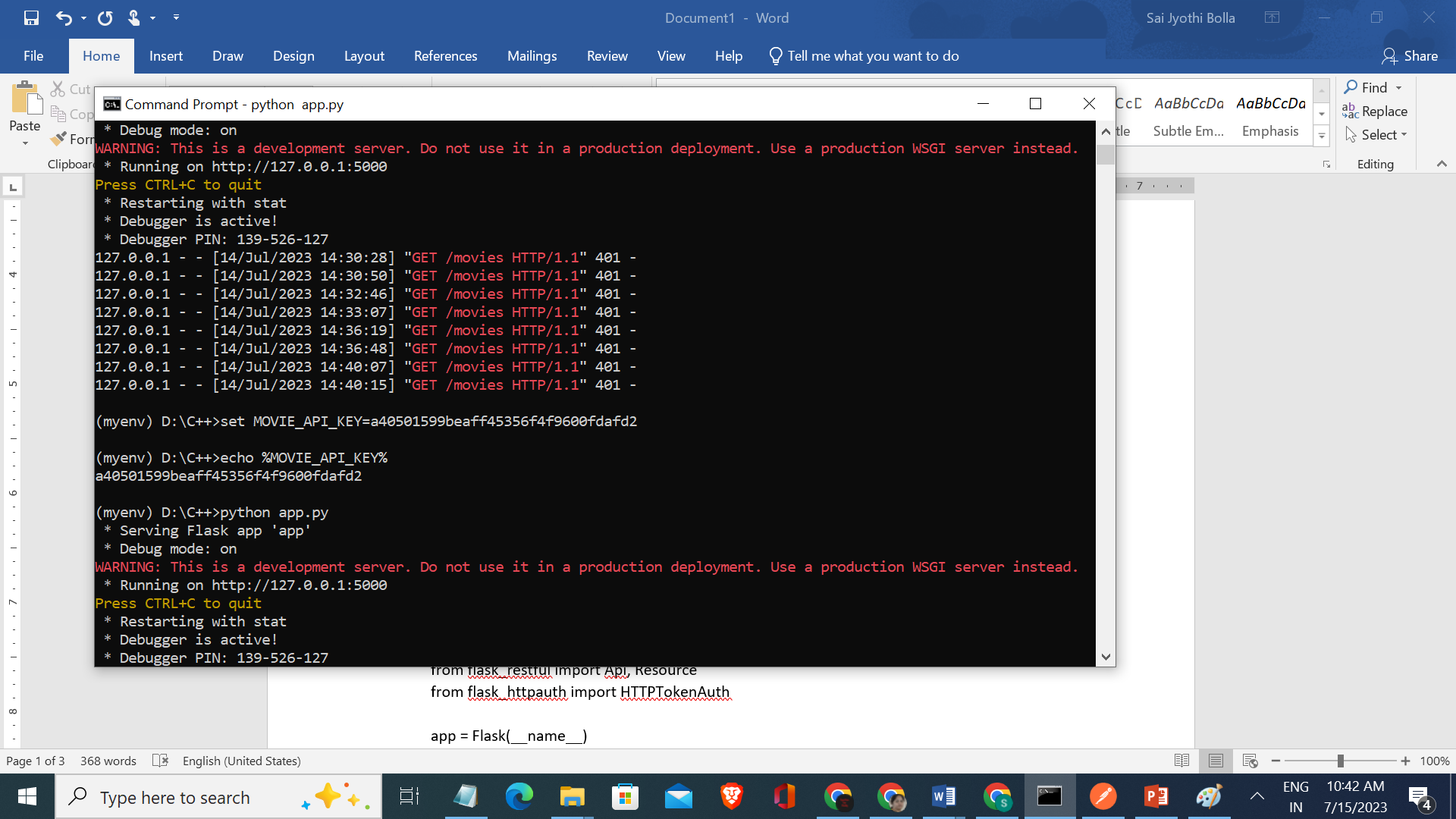
**D:\C++** is my current working directory, you need to use your current working directory

1. Then we need to set TMDb API key as the environment variable **MOVIE\_API\_ KEY.** Use the below command to set and use the same API key provided here.(I think you need not created new API key)

**set MOVIE\_API\_KEY=a40501599beaff45356f4f9600fdafd2**

you can check weather it is properly set or not with the command

**echo %MOVIE\_API\_KEY%**



1. **Then save the code as app.py**

import os

import requests

from flask import Flask

from flask\_restful import Api, Resource

from flask\_httpauth import HTTPTokenAuth

app = Flask(\_\_name\_\_)

api = Api(app)

auth = HTTPTokenAuth(scheme='Bearer')

# Retrieve the API key from environment variable

API\_KEY = os.environ.get('MOVIE\_API\_KEY')

if not API\_KEY:

raise ValueError('Movie API key is missing. Set the MOVIE\_API\_KEY environment variable.')

# Verify the API key

@auth.verify\_token

def verify\_token(token):

return token == API\_KEY

@auth.error\_handler

def unauthorized():

return {'message': 'Unauthorized'}, 401

# Endpoint for movie details

class MovieDetails(Resource):

@auth.login\_required

def get(self, movie\_name):

url = f'https://api.themoviedb.org/3/search/movie?api\_key={API\_KEY}&query={movie\_name}'

response = requests.get(url)

data = response.json()

if 'results' in data and len(data['results']) > 0:

movie = data['results'][0]

movie\_details = {

'title': movie['title'],

'release\_year': movie['release\_date'].split('-')[0],

'plot': movie['overview'],

'cast': self.get\_movie\_cast(movie['id']),

'rating': movie['vote\_average']

}

return movie\_details

else:

return {'message': 'Movie not found'}, 404

def get\_movie\_cast(self, movie\_id):

url = f'https://api.themoviedb.org/3/movie/{movie\_id}/credits?api\_key={API\_KEY}'

response = requests.get(url)

data = response.json()

cast = []

if 'cast' in data:

for actor in data['cast'][:5]:

cast.append(actor['name'])

return cast

# Endpoint for movie list

class MovieList(Resource):

@auth.login\_required

def get(self):

url = f'https://api.themoviedb.org/3/discover/movie?api\_key={API\_KEY}&sort\_by=popularity.desc'

response = requests.get(url)

data = response.json()

movie\_list = []

if 'results' in data:

for movie in data['results'][:10]:

movie\_list.append(movie['title'])

return {'movies': movie\_list}

# Add routes to the Flask application

api.add\_resource(MovieDetails, '/movies/<string:movie\_name>')

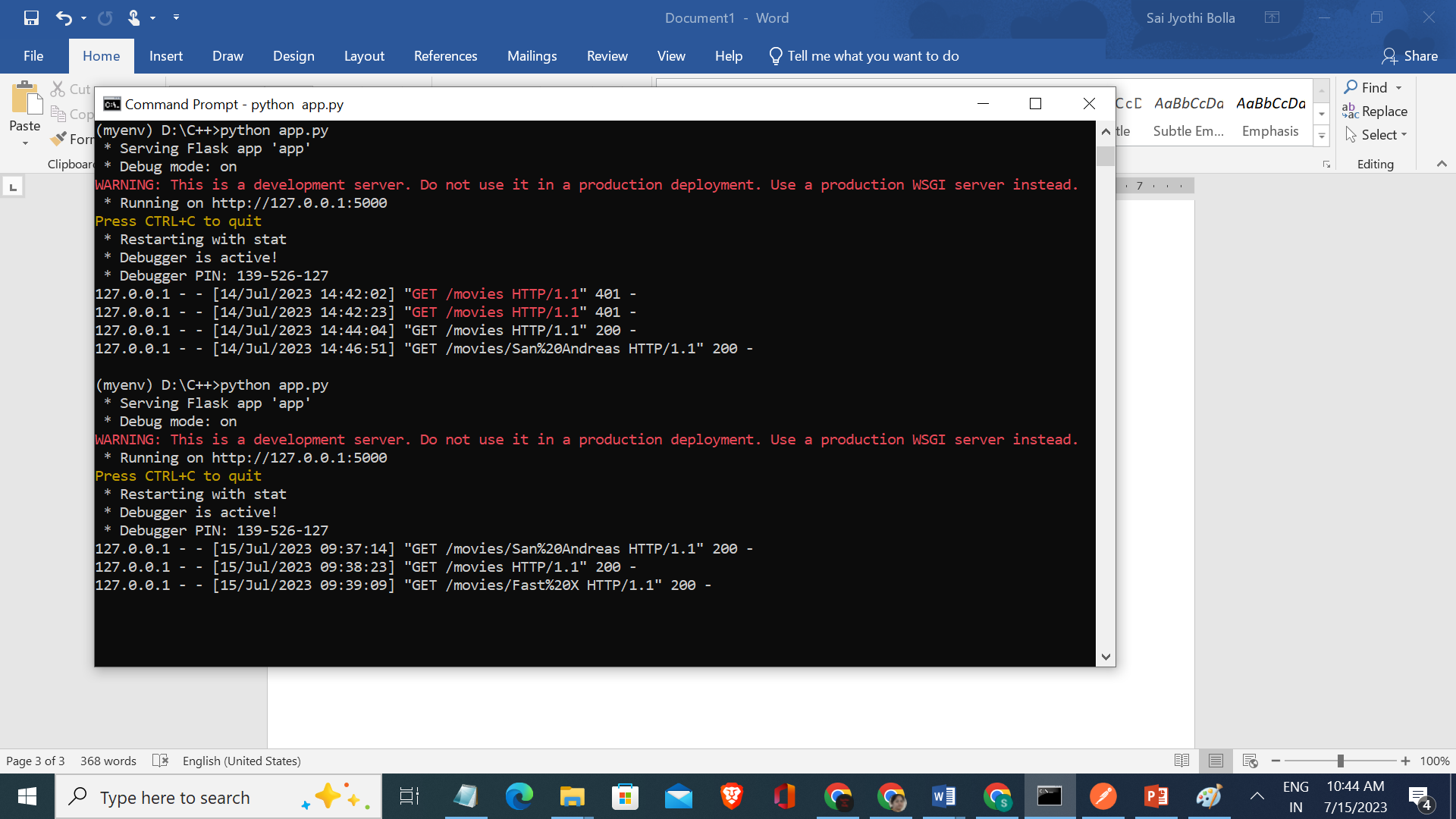
api.add\_resource(MovieList, '/movies')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

1. Then run the program using

**python app.py**



1. After this we need to test it with postman. Open the postman do the following.
2. Select new tab at top left corner
3. select GET request
4. select authorization tab, then from the dropdown select bearer token give the above API key
5. use [**http://127.0.0.1:5000/movies**](http://127.0.0.1:5000/movies) **in get request, now you will get all the moies**

if you want get a specific movie details, you can use http://127.0.0.1:5000/movies/John Wick: Chapter 4.

Here John Wick: Chapter 4 is the movie name.

