ChatBot development using OpenAI API and Streamlit

By: Vaishnavi Patil

1. **Set Up Virtual Environment**

   First, ensure you have virtualenv installed. If not, you can install it by running:

```
vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$ pip install virtual
env
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: virtualenv in /usr/lib/python3/dist-packages (20.13.0+ds)
Requirement already satisfied: distlib<1,>=0.3.1 in /usr/lib/python3/dist-packages (from virtualenv) (0.
3.4)
Requirement already satisfied: six<2,>=1.9.0 in /usr/lib/python3/dist-packages (from virtualenv) (1.16.0
)
Requirement already satisfied: platformdirs<3,>=2 in /usr/lib/python3/dist-packages (from virtualenv) (2
.5.1)
Requirement already satisfied: filelock<4,>=3.2 in /usr/lib/python3/dist-packages (from virtualenv) (3.6
.0)
vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$
```

   a. Create a Virtual Environment

      Navigate to your project directory and create a virtual environment:

```
vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$ pwd
/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit
vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$ virtualenv venv
created virtual environment CPython3.10.12.final.0-64 in 3186ms
  creator CPython3Posix(dest=/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit/venv, clear=False
, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir
=/home/vaishnavi/.local/share/virtualenv)
    added seed packages: pip==22.0.2, setuptools==59.6.0, wheel==0.37.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActi
vator
vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$
```

      This will create a venv folder inside your project directory.

   b. Activate the Virtual Environment

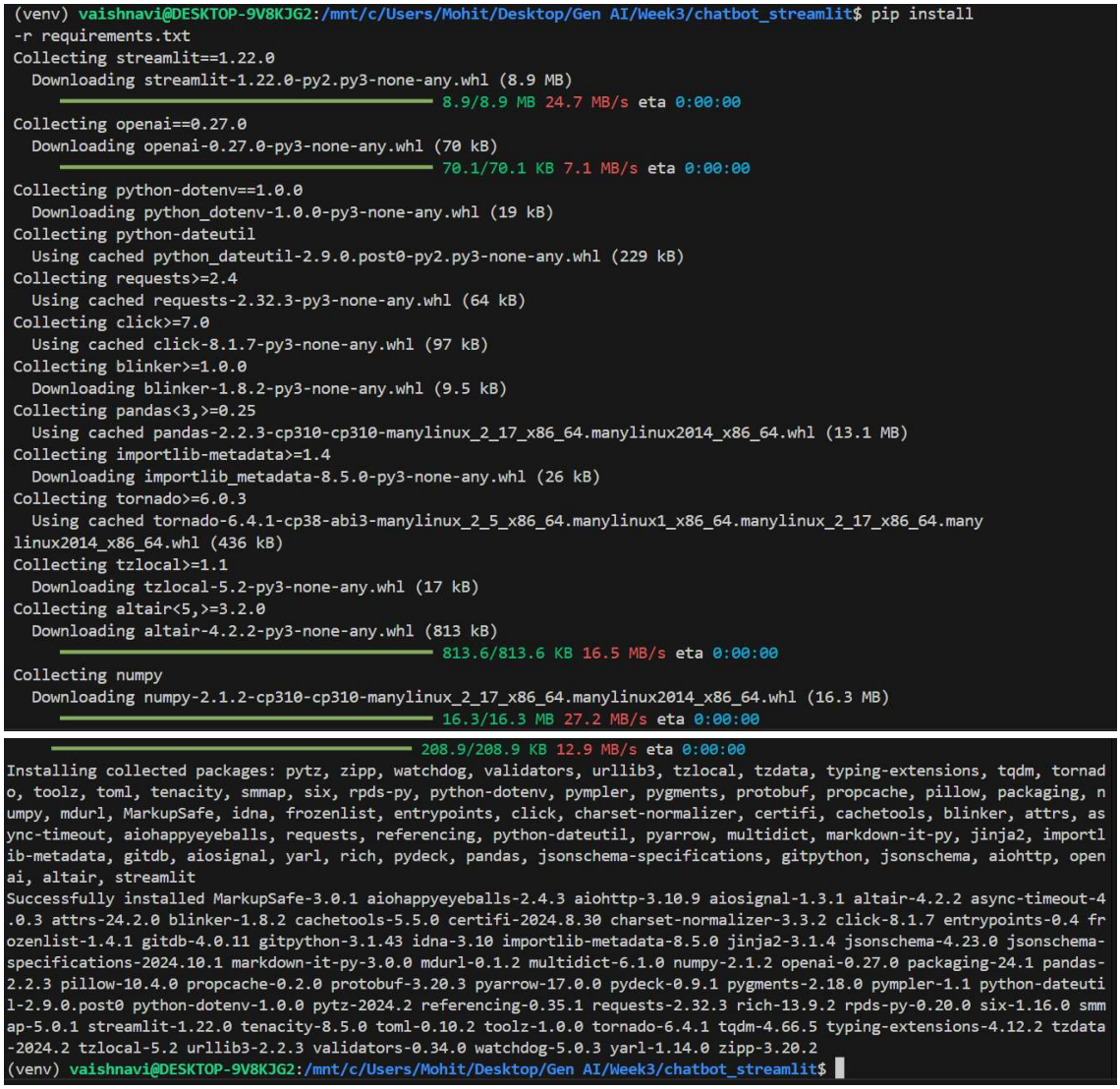
      I am using Ubuntu so run **source venv\bin\activate**

```
vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$ source venv/bin/act
ivate
(venv) vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$
```

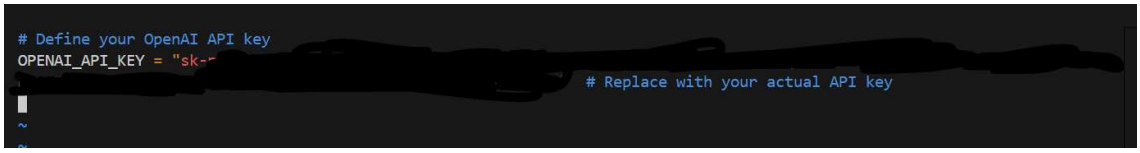
      You should now see (venv) in your terminal, indicating that the virtual environment is active.

   c. Install Dependencies

Once the virtual environment is activated, install the dependencies from the requirements.txt file:



d. Set up environment variables:



e. Deactivate the virtual environment (when done) using **deactivate**

2. **Main Application Code**

After all the requirements are installed and virtual environment is set, lets move with the creation of chatbot using gpt-4o-mini model and streamlit.

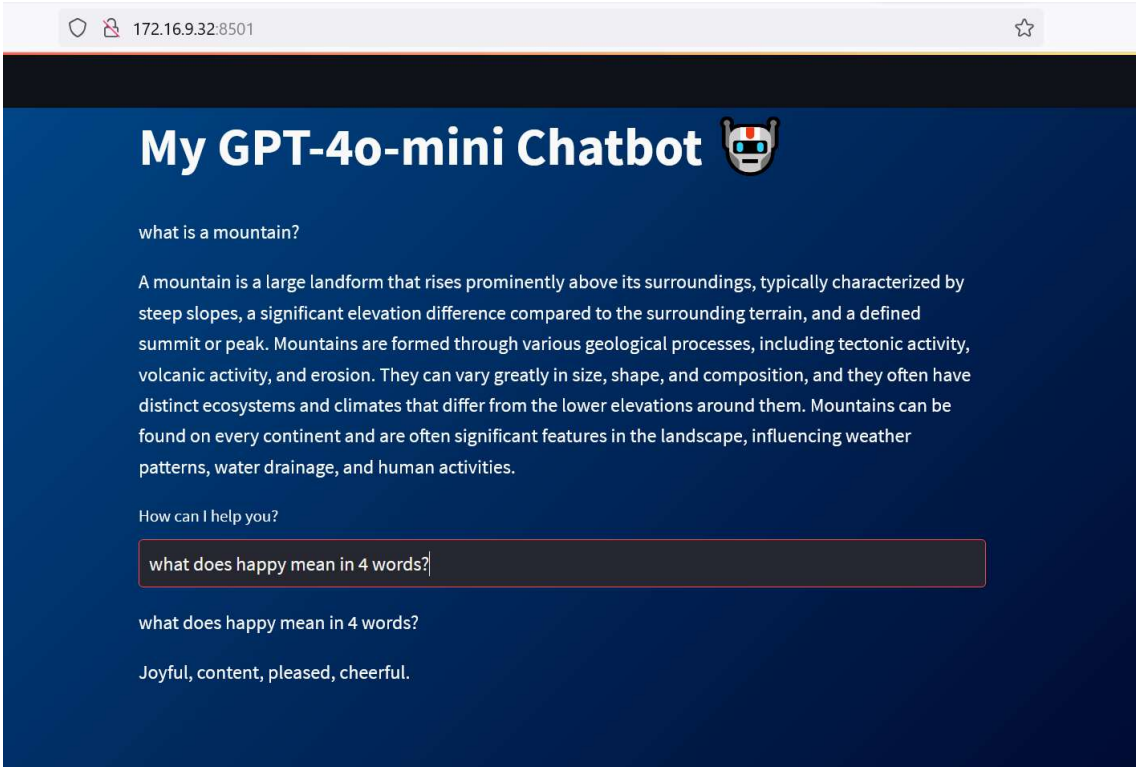
```python
import streamlit as st  # type: ignore
import openai, os
from dotenv import load_dotenv

# Load environment variables for OpenAI API key
load_dotenv()
openai.api_key = os.getenv("OPENAI_API_KEY")  # Set the API key

# Title of the chatbot
st.title("My GPT-4o-mini Chatbot 🤖")

# CSS for full-page shaded blue gradient background
st.markdown(
    """
    <style>
    /* Apply the gradient to the whole page */
    html, body, .stApp {
        height: 100%;
        background: linear-gradient(135deg, #004e92, #000428); /* Shaded blue gradient */
        color: white; /* Text color for readability */
        /* Footer style */
    }
    </style>
    """,
    unsafe_allow_html=True
)

# Initialize messages in the session state
if "messages" not in st.session_state:
    st.session_state.messages = []

# Display messages
for message in st.session_state["messages"]:
    with st.text(message["role"]):
        st.markdown(message["content"])

# Handle user input and OpenAI response
if user_prompt := st.text_input("How can I help you?"):
    st.session_state.messages.append({"role": "user", "content": user_prompt})

    # Display user message
    with st.text("user"):
        st.markdown(user_prompt)

    # Assistant response
    with st.text("assistant"):
        chatbot_msg = st.empty()
        full_response = ""
        stream = openai.ChatCompletion.create( # type: ignore
            model="gpt-4o-mini",
            messages=[
                {"role": msg["role"], "content": msg["content"]}
                for msg in st.session_state["messages"]
            ],
            temperature=0,
            stream=True,
        )

        # Stream the response
        for chunk in stream:
            token = chunk.choices[0].delta.get("content")
            if token is not None:
                full_response = full_response + token
                chatbot_msg.markdown(full_response)

        chatbot_msg.markdown(full_response)

    # Store assistant's response in session
    st.session_state.messages.append({"role": "assistant", "content": full_response})
```

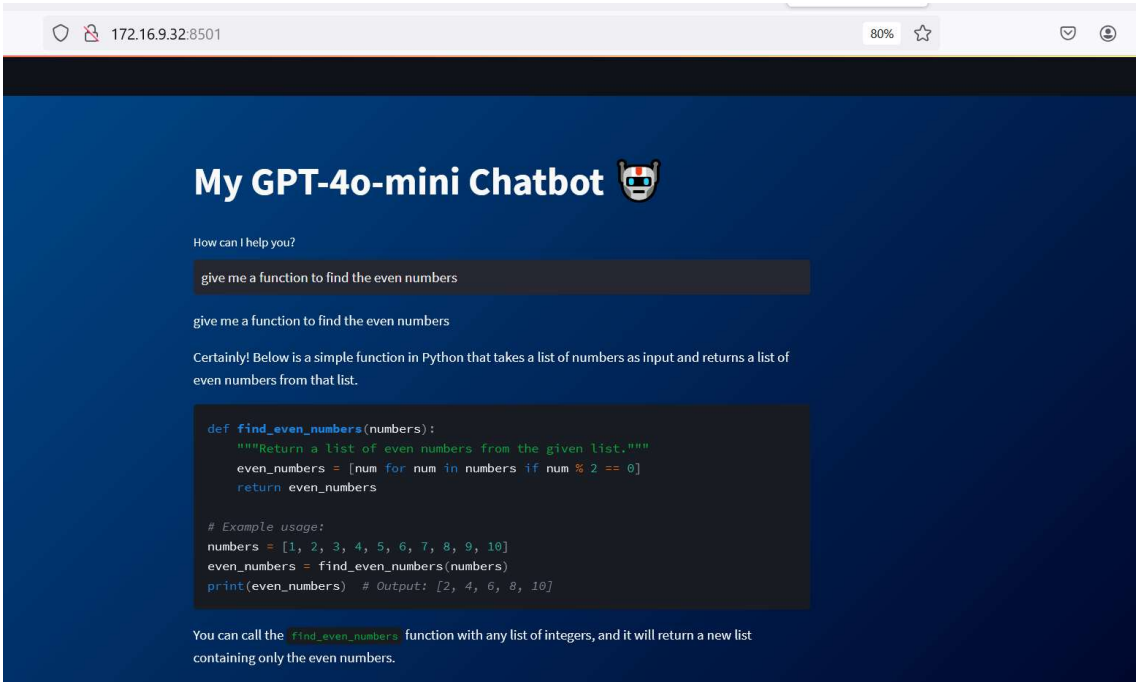
Running the Application

To run the Streamlit app, use the command: **streamlit run chatbot.py**

```
vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$ source venv/bin/act
ivate
(venv) vaishnavi@DESKTOP-9V8KJG2:/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit$ streamlit ru
n chatbot.py
/mnt/c/Users/Mohit/Desktop/Gen AI/Week3/chatbot_streamlit/venv/lib/python3.10/site-packages/numpy/_core/
getlimits.py:548: UserWarning: Signature b'\x00\xd0\xcc\xcc\xcc\xcc\xcc\xcc\xfb\xbf\x00\x00\x00\x00\x00\
x00' for <class 'numpy.longdouble'> does not match any known type: falling back to type probe function.
This warnings indicates broken support for the dtype!
  machar = _get_machar(dtype)

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.


  You can now view your Streamlit app in your browser.

  Network URL: http://192.168.1.76:8501
  External URL: http://107.193.140.97:8501
```

Then follow the link shown




172.16.9.32:8501
80%
My GPT-4o-mini Chatbot
How can I help you?
give me a function to find the even numbers
give me a function to find the even numbers
Certainly! Below is a simple function in Python that takes a list of numbers as input and returns a list of even numbers from that list.
def find_even_numbers(numbers):
"""Return a list of even numbers from the given list."""
even_numbers = [num for num in numbers if num % 2 == 0]
return even_numbers
# Example usage:
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_numbers = find_even_numbers(numbers)
print(even_numbers) # Output: [2, 4, 6, 8, 10]
You can call the find_even_numbers function with any list of integers, and it will return a new list containing only the even numbers.