Appointment Scheduling

Problem Statement:

Creates a chatbot assistant specifically for scheduling appointments. It interacts with users in a natural way, collecting details like name, email for contact, and preferred time. Finally, it confirms all the appointment details .

Functionalities:

- Interact with users to schedule appointments.
- Collect essential details like user name, desired date/time, and contact information.
- Confirm the appointment and schedule it.

Technical Approach:

Library Installation:

- LangChain for conversation flow management.
- OpenAI library to interact with GPT-3.5-turbo model.
- Langchain.agents library for handling agents.

Environment Setup:

- Secure access to OpenAI using environment variables with API keys.

Template Creation:

- Define a conversation template outlining the information to be collected:
  - User name
  - Date and time preferences
  - Email address
- Emphasise adherence to the template to avoid redundant questions and unsolicited responses.

Prompt and Model Initialization

- Create prompts for user and system messages using LangChain's ChatPromptTemplate.
- Initialise the ChatOpenAI model with a specific temperature setting for response randomness.
- Utilise ConversationBufferMemory to track conversation history.

Chat Chain Creation:

- Develop a function to generate responses using LangChain.
- This function:
  - Takes user queries as input.
  - Processes them through the chat model.
  - Returns appropriate responses aligned with the conversation template.

Agent Chain Initialization:

- Initialise an agent to leverage Zapier functionalities .
- This agent operates in zero-shot learning mode, functioning without extensive training data.

Information Extraction:

- Create a function to extract relevant information from the conversation using regular expressions.
- This function identifies details like:
  - User name
  - Date and time
  - Email address

Main Conversation Loop:

- Continuously handle user input, generate responses, and extract information.
- Check if all necessary details are collected.

- Upon complete information gathering:
  - Schedule the appointment
  - Save the conversation for record-keeping.

Output:

Attached the screenshot of the output, how it works

```
Human: Need an appointmnet with Dvid at 10 AM on 15/06/24
/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:119: LangChainDeprecationWarning:
  warn_deprecated(
/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:119: LangChainDeprecationWarning:
  warn_deprecated(
Great! I have successfully scheduled an appointment for you with David at 10 AM on June 15, 2024.

Please provide me with your details so I can finalize the booking.

Full Name: [Your Full Name]
Email Address: [Your Email Address]

Thank you for choosing us.
Human: Dharshini , dharshini@gmail.com
Great! Here are the details of your appointment:

Full Name: Dharshini
Date and Time: 10 AM on June 15, 2024
Email Address: dharshini@gmail.com

Thank you for choosing us.
Human: bye
Thank you for choosing us.
bye
```