We are offering an offline coding challenge, where you are tasked with creating a chat interface using an **agent framework**, such as [Autogen](https://github.com/microsoft/autogen), along with mocked data and APIs to meet the following requirements:

1. Objective: Facilitate an SAP employee in booking round-trip flights for any SAP event through a chat interface. You can mock data and APIs / tools if necessary.

2. Suggested agents:

- Travel Agent: Manages travel-related tasks.

- Search Agent: Retrieves SAP internal information.

3. The user information must be retrieved from a table of at least five employees. Note that users are only eligible to depart from airports near their home office. Your demo should showcase how the chat interface displays distinct chat streams under the same query, for different internal users.

4. Demo test query: "Book me a flight to Sapphire."

- Note: Sapphire is a specific SAP event occurring in Orlando from June 3, 2024, at 9 AM to June 5, 2024, at 5 PM. You can mock data of at least two additional SAP events.

- The agent-to-agent conversation log should be trackable in the backend. At some point, it should show multiple flight options from the user’s location to the event’s location.

- You can decide the criteria of user intervention.

An OpenAI API key will be provided for this project, with a usage limit of $20. This key is to be used responsibly and exclusively for this project. You are suggested to use GPT-4-turbo or models other than GPT-4.

A Python Notebook is an acceptable format for the demo, which should include an input window for typing queries. Creating an end-to-end chat interface with a frontend tool like Streamlit is an ambitious goal but not mandatory.

Prepare for a 25-minute demo presentation in the next interview round, followed by a 15-minute session for follow-up questions. You may be asked to type queries beyond the specified scope. Your agent framework should be capable of handling routine conversations.

Evaluation Metrics:

1. Inclusion of the required components in the agent framework.

2. The range of functionalities demonstrated.

3. The ability to handle minor changes to prompts.

4. Code quality.

5. Clarity of the presentation.