**Task 1: Overview**

This task involves integrating OpenAI API calls into **askquestion.py** to generate one challenging question on topic chosen in dropdown and submit the question and generate the feedback.

**Task List**

1. **Understand the Boilerplate Code**:
   * Review the existing **askquestion.py** structure and understand the placeholders.
2. **Setup OpenAI API Integration**:
   * Replace the placeholder code in **/ask-question** end point with the provided OpenAI API call logic.
   * Replace the placeholder code in **/question-feedback** with the provided OpenAI API call logic.
3. **Integrate API Details**:
   * Use the specified model **(gpt-4o-2024-05-13**) and configuration (temperature: 0.2) initially and play around with various temperatures.
4. **Define System and User Roles**:
   * Implement the system role to set the context as a technical interview expert.
   * Implement the user role to feed the topic or prompt for question generation and feedback.
5. **Debug and Optimize**:
   * Ensure proper handling of errors (e.g., invalid inputs, API issues).
   * Optimize for response accuracy and performance.

**Task Solution**

Update boilerplate **askquestion.py** code with two endpoints:

**Challenge 1: (**/ask-question)

1. Write the appropriate prompt to generate interview question for the selected topic
   * Generate only one question
   * Ensure that question is prefixed with “Question:” eg: “Question: What is .NET”
   * Ensure that the returned question doesn't have answer included.
2. Call OpenAI API to execute the prompt.

**Challenge Solution 1.a:**

Replace

prompt = f""

with

prompt = f"Compose a challenging {topic} interview question without answer. \n Question:"

**Challenge Solution 1.b:**

**Replace**

response = { }

**with**

response = client.chat.completions.create(

        model="gpt-4o-2024-05-13",

        temperature=0.2,

        messages=[

            {"role": "system", "content": "You are an expert in technical interviews, generating relevant questions with concise (1-2 sentence) answers."},

            {"role": "user", "content": prompt},

        ],

    )

**Challenge 2: (**/question-feedback)

a) Generate the prompt to submit both question and answer and also get feedback along with rating between 1 and 10

b) Call OpenAI API to get the response.

**Challenge Solution 2.a**

**Replace**

prompt = f"""

                      """

**with**

prompt = f"""

Evaluate the following question and answer and rate the candidate's response on a scale of 1-10:

question: {question}

 answer: {answer}

"""

**Challenge Solution 2.b**

**Replace**

    response = {}

**with**

    response = client.chat.completions.create(

        model="gpt-4o-2024-05-13",

        temperature=0.2,

        messages=[

            {

                "role": "system",

                "content": "You are an expert in technical interviews"

            },

            {"role": "user", "content": prompt},

        ],

    )