Task 2- Overview

This task involves integrating OpenAI API calls into **descriptive.py** to generate 3 challenging questions on topic chosen in dropdown and submit the questions and generate the feedback for all questions.

Task List

1. **Understand the Boilerplate Code**: Review the pre-existing structure and endpoints.
   - Understand the JSON response flow.
2. **Implement Prompt Formatting**: Write logic to create effective prompts for OpenAI API calls.
3. **Integrate OpenAI API**: Add OpenAI API integration for generating and evaluating responses.
4. **Test and Validate**: Ensure endpoints function correctly and validate all JSON responses.

Task Description

Update boilerplate **descriptive.py:**

Challenge 1: (/generate-questions)

Write a prompt to generate following o/p

1. 3 questions with properties "Id", "Question" and "ExpectedAnswer"

2. Json o/p

3. Answer in 2 or 3 sentences.

```
[
 {
  "Id": 1,
  "Question": ____,
  "ExpectedAnswer": ____"
 }
]
```

a) Call OpenAI API to execute the prompt. Choose an appropriate System Message. Temperature=0.2

b) Debug and find the response text and remove all the special characters (if any) to ensure that has only JSON object.

Challenge 2 (/submitdescriptivequestions)

a) Write a prompt instructing to evaluate all the questions in variable **user_message**
   - You will need to debug and find the format of data in user_message variable.
   - Get feedback for each question.
   - Overall rating should be between 1 and 10

b) Call OpenAI API to get response

Tips for Success

- Always test your prompts using simple examples.

- Use tools like Postman to test your API endpoints.

Challenge 1 Solution

Challenge Solution 1.a:

Replace

```
prompt = f""
```

with

```
prompt = f"""
  Create 3 challenging {topic} interview questions with concise, ideal answers (1-2 sentences). Output in JSON format using this schema:
  [
    {{
      "Id": "Question Number",
      "Question": "Question text",
      "ExpectedAnswer": "Ideal answer"
    }}
  ]
"""
```

Challenge Solution 1.b:

Use the following steps to complete the integration:

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},
    ],
  )
```

Challenge Solution 1.c: Find the response text and remove all the special characters to match the response as in below JSON format

```
questions_json_str = re.sub(r"^```json|```$", "", questions_text.strip())
```

Challenge 2 Solution

(/submitdescriptivequestions)

c) Write a prompt instructing with following specification

   ○ Answer should be rated between 1 and 10

d) Call OpenAI API to get response

Challenge Solution 2.a

Replace

```
  prompt = f"""
            """
```

with

```
  prompt = f"""
            Evaluate the following candidate's answers to technical interview questions and provide:
            - An overall rating out of 10.
            - Specific feedback on each answer, highlighting strengths and weaknesses.

            JSON Data:
            { user_message }
            """
```

Challenge Solution 2.b

Use the OpenAI API to process the evaluation prompt:

Replace

```
feedback_response = {}
```

with

```
feedback_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, providing feedback on answers."},
```

```
        {"role": "user", "content": prompt}
    ]
)
```