Sapper Task

Overall Requirements:

1. For each task, the following steps should be included:
    - Read the question
    - Create a file/project
    - (optional) Search/view API
    - Write code
    - Test code: Please test strictly according to the provided test samples. Save the file once all tests are successful.
    - Save the file

    Note: There is no need to perfect the prompt; as long as the whole process is smooth and the answers are roughly reliable!
2. For each task, record a video from the beginning of reading the question to saving the file. Also, save the code or JSON file. Check the corresponding task checkbox after starting each recording. Check the corresponding checkbox after successfully completing a test.
    - File saving format: <ToolName>_task<ID>, e.g., sapperV1_taskA1.mp4, python_taskB3.py, sapperV2_taskC2.json
    - Put all saved files in the same folder (folder name <name>_round2); there is no need to separate them into multiple folders.
3. Before starting the user study, do not look up OpenAI's official documentation in advance. The official documentation needs to be searched on the spot. Do not copy the code from the previous user study.
4. During the user study, you can directly copy the API documentation webpage or newly written code found in previous tasks. You can also copy the code or test samples from the OpenAI official website or this document.
5. In this user study, three tools will be used to complete tasks: Python, SapperV1 (without Design View), and SapperV2 (with Design View). Different tools will be used for different tasks A/B/C.
6. When using the Design View for tasks, you can first generate the basic code with Design View each time, and then combine the code from the previous task with the newly generated code. Modify it to meet the requirements. It is not mandatory to use Design View for every task.
7. Please find your name, tool order, and corresponding tasks on the next page, and follow the order strictly. For example, V1-C, Python-B, V2-A means first use Sapper V1 to do Task C, then use Python to do Task B, and finally use SapperV2 to do Task A.
8. After completing all tasks, fill out the questionnaire at https://qfreeaccountssjc1.az1.qualtrics.com/jfe/form/SV_cAasiUScCFvJ7y6 and select "round2" for the current task.

Tasks

A. **Question-solving Robot: You need to develop a question-solving robot service, where users provide questions, and your service answers them for the user.**

☐**Start recording and check off**

TaskA1. The user asks a question, and the model outputs an answer using the text-davinci-003 model.

☐**Test:**

User input: Compare the political systems of the Roman Republic and the Roman Empire, listing their differences and similarities.

☐**Start recording and check off**

TaskA2. Based on the TaskA1 code, add a new requirement. Add a new inquiry asking the user whether it is a question about code or a general question. If it is a question about code, use the text-davinci-003 model, prompt 1; if it is a general question, keep the text-davinci-003 model as in task1, prompt 2. Prompt 1 and prompt 2 represent using different prompts. Finally, the user provides feedback.

Logic: Ask the user what type of question it is (code or general), and ask the user what the specific question is. If it is a code question, then...; if it is a general question, then... (use if else). User feedback.

☐**Test1:**

User input: General question

User input: Explain what a large language model is

Answer: ...

User input: I understand

☐**Test2:**

User input: Code question

User input: Write a function using Python, bubble sort method

Answer: ...

User input: The answer seems to lack some comments

☐**Start recording and check off**

TaskA3. Based on the TaskA2 code, add functionality. When the current round of answering is finished, ask the user if they want to continue. If they choose to continue, **automatically repeat tasks 1a-2a** using a while loop.

☐**Test:**

User input: General question

User input: Explain what a large language model is

Answer: ...

User input: I understand

Continue? (1 to continue, 2 to end)

User input: 1

User input: Code question

User input: Write a function, bubble sort method
User input: The answer seems to lack some comments
Answer: ...

Continue? (1 to continue, 2 to end)
User input: 2

☐**Start recording and check off**

TaskA4 Based on the TaskA3 code, add functionality. Use a history variable to record all conversations, recording in the following format (user feedback does not need to be recorded). After the user indicates they will not continue, use gpt-3.5-turbo to summarize all questions asked based on the conversation history.

The format of the History record should be as follows:

User: General question
User: Explain what a large language model is
Answer: ...
User: Code question
User: Write a function, bubble sort method Answer: ...

☐**Test:**

User input: General question
User input: Explain what a large language model is
Answer: ...
User input: I understand

Continue? (1 to continue, 2 to end)
User input: 1
User input: Code question
User input: Write a function, bubble sort method Answer: ...
User input: The answer seems to lack some comments

Continue? (1 to continue, 2 to end)
User input: 2
ChatGPT Summary: ...

Complete the questionnaire once after finishing all tasks:
https://qfreeaccountssjc1.az1.qualtrics.com/jfe/form/SV_cAasiUScCFvJ7y6 **and select "round2" for the current task.**

B. Mock Interview Robot

☐**Start recording and check off**

TaskB1. User inputs the job they are interviewing for (open-ended question), the model asks questions according to the requirements, using the text-davinci-003 model.

☐**Test:**

User input: I plan to interview for a senior programmer job.

☐**Start recording and check off**

TaskB2. Based on the TaskB1 code, add a new requirement. Add a new question, asking the user about their professional background (CS background or non-CS background). If it's a CS background, use the text-davinci-003 model to provide a Python code snippet with a bug for the user to fix; if it's a non-CS background, use the text-davinci-003 model to ask a question. The user then answers.

Use different prompts, different models

Logic: Ask the user what job they are interviewing for, ask if they have a CS-related background. If yes, then...; if no, then... (use if-else). Finally, the user inputs the answer, and it ends.

☐**Test 1:**

User input: I plan to interview for a senior programmer job.

User input: CS background

Question: ...

User input: The bug in this function is...

☐**Test 2:**

User input: I plan to interview for a senior programmer job.

User input: Non-CS background

Question: ...

User input: I participated in a large project...

☐**Start recording and check off**

TaskB3. Add functionality to the taskB2 code. After completing the current round of Q&A, ask the user if they want to continue the interview. If they want to continue, ask more questions and have the user answer.

☐**Test:**

User input: I plan to interview for a senior programmer job.

User input: CS background

Question: ...

User input: ...

Do you want to continue? (1 for continue, 2 for end)

User input: 1

Question: ...

User input: ...
Do you want to continue? (1 for continue, 2 for end)
User input: 2

☐Start recording and check off

TaskB4. Add functionality to the taskB3 code. Use a history variable to record all dialogues, with the following recording format. After the user indicates they will not continue, use the gpt-3.5-turbo model to provide improvement suggestions for the entire interview process based on the history.

The history should be recorded in the following format:

User: I plan to interview for a senior programmer job.
User: CS background
Question: ...
User response: ...
Question: ...
User response: ...

Note that the user's response to whether to continue or not does not need to be recorded.

☐Test:

User input: I plan to interview for a senior programmer job.
User input: CS background
Question: ...
User input: The bug is in the third line
Do you want to continue? (1 for continue, 2 for end)
User input: 1

Question: ...
User input: The bug is at the end
Do you want to continue? (1 for continue, 2 for end)
User input: 2

ChatGPT summarizes and provides feedback: ...

Complete the questionnaire once after finishing all tasks:
https://qfreeaccountssjc1.az1.qualtrics.com/jfe/form/SV_cAasiUScCFvJ7y6 **and select "round2" for the current task.**

C. Ad Slogan Generation and illustration

☐ **Start recording and check off**

TaskC1. User inputs a description of their product. The model comes up with an advertising slogan based on the requirements, using the text-davinci-003 model.

☐ **Test:**

User input: My product is a potato chip, it's especially delicious, everyone who tries it wants to buy more, we have some exclusive recipes.

☐ **Start recording and check off**

TaskC2. Based on TaskC1 code, add a new requirement. Add a new question, asking the target audience (children or adults). If it's for children, use the text-davinci-003 model; if it's for adults, keep using the text-davinci-003 model as in Task1. Use different prompts. After the slogan is output, ask about the style of the advertisement image (open-ended question).

Logic: Ask users about their requirements, ask users about the target audience. If it's for children, then...; if it's for adults, then... (use if-else). Ask about the style (open-ended question).

☐ **Test 1:**

User input: My product is a potato chip, it's especially delicious, everyone who tries it wants to buy more, we have some exclusive recipes.
User input: Children
Slogan: ...
Question: What style would you like?
User input: Cartoon

☐ **Test 2:**

User input: My product is a potato chip, it's especially delicious, everyone who tries it wants to buy more, we have some exclusive recipes.
User input: Adults
Slogan: ...
Question: What style would you like?
User input: Oil painting

☐ **Start recording and check off**

TaskC3. Add functionality to the taskC2 code. After the current round of Q&A is completed, ask the user if they are satisfied. If they are not satisfied, generate a new slogan. Move the style question from C2 to the end (only ask the style after the user is satisfied).

☐ **Test:**

User input: My product is a potato chip, it's especially delicious, everyone who tries it wants to buy more, we have some exclusive recipes.
User input: Children
Slogan: ...

Are you satisfied? (1 for satisfied, 2 for not satisfied)
User input: 2
Slogan: ...
Are you satisfied? (1 for satisfied, 2 for not satisfied)
User input: 1
Question: What style would you like?
User input: Cartoon

☐**Start recording and check off**

TaskC4. Add functionality to the taskC3 code. Use an additional "requirements" variable to record user requirements and the latest slogan, with the following recording format. After the user indicates satisfaction, use Dall-E to draw a picture based on the requirements.

The requirements should be recorded in the following format:

Requirement: My product is a potato chip, it's especially delicious, everyone who tries it wants to buy more, we have some exclusive recipes.
Target audience: Adults
Slogan: ...
Style: Cartoon

☐**Test:**

User input: My product is a potato chip, it's especially delicious, everyone who tries it wants to buy more, we havesome exclusive recipes.
User input: Children
Slogan: ...
Are you satisfied? (1 for satisfied, 2 for not satisfied)
User input: 2

Slogan: ...
Are you satisfied? (1 for satisfied, 2 for not satisfied)
User input: 1

Question: What style would you like?
User input: Cartoon

Dall-E creates an illustration based on the requirements. You need to open and view the generated image.

Complete the questionnaire once after finishing all tasks:
https://qfreeaccountssjc1.az1.qualtrics.com/jfe/form/SV_cAasiUScCFvJ7y6 **and select "round2" for the current task.**