Interview Question: Win analysis Agent for Pokémon Universe

Task:

Note: *This test is built to evaluate your Agent building and LLM skills. We're looking to see you write code that can do this!*

You need to build a **pokemon winner analysis agent** that takes in a user question and processes it in the following way:

Task Breakdown:

Given a user question, your task is to identify who the winner is between the two humans mentioned in the query/prompt. You MUST build an **agent** that can do this. The process to determine the winner is as follows

1. **Extract Human Names**:
    - The user will ask a question about **two specific humans** from the Pokémon universe . They must be any two of Ash, Misty, Brock, Jessie, or James. If the user provides names that are none of these or there are more than 2 of these names then immediately return "Incorrect question"
    - Your bot should correctly identify **which two humans are mentioned**.

2. **Read the Script**:
    - The script is stored in a **text file** called "pokemon_transcript.txt" (Provided to you already). Each line follows a structured dialogue format. This script is a sample conversation between few Pokemon Human characters about their respective pokemon
    - Your bot must **find all Pokémon** "owned" by the two humans in the given script.

3. **Retrieve Pokémon Weights**:
    - A function `get_pokemon_weight(pokemon_name)` is provided to you in a file called pokemon_api.py (**You are not allowed to edit this file**)
    - This function takes in the name of the pokemon and returns a dictionary in the following format:

```
{
    "text_description":
     "Charizard weighs 905 hectograms (90.5 kg)."
}
```

- Your bot must call this function for **each Pokémon** and **sort** them by weight for both humans.

4. **Generate a CSV File:**

    - The output should be a CSV file with **three columns**:

        - **Column 1:** Pokémon of Human 1 (sorted by weight)

        - **Column 2:** Pokémon of Human 2 (sorted by weight)

        - **Column 3:** Winner (for each row, the human whose Pokémon weighs more; if equal, mark as "Tie")
        - If one human has more Pokémon than the other, only the heaviest Pokémon from the larger team should be considered, ensuring both teams have an equal number of Pokémon for comparison. For example, if one human has 5 Pokémon and the other has 4, the first human should only use their 4 heaviest Pokémon, so that both teams are fairly matched.

5. **Determine the Overall Winner:**

    - Count the number of **wins per human** and decide the final outcome:
        - If one human wins more rounds, they are the overall winner.
        - If both have the same number of wins, It's a tie

6. **Final output: Print the overall Winner**

    - Your agent must return/print the final output on the screen as follows:

      "The winner of the duel between {human1} and {human2} is {winner}!
      Congratulations {winner}!!
      {winner} won against {loser} with a score of {winner_battles_won_number} to {loser_battles_won_number}!"

      If the competition was a tie then return:

      "The competition is a tie!"

SAMPLE EXECUTION:

If after step 2, you have mapped Ash with the pokemons “Pikachu” and “Charizard” and Misty with “Gyrados” and “Psyduck”, the next steps are as follows:

Step 3. After using the API, the weights of the pokemons are:

Ash:

- {'text_description': 'Charizard weighs 905 hectograms (90.5 kg).'}
- {'text_description': 'Pikachu weighs 60 hectograms (6.0 kg).'}

Misty:

- {'text_description': 'Gyarados weighs 2350 hectograms (235.0 kg).'}
- {'text_description': 'Psyduck weighs 196 hectograms (19.6 kg).'}

Therefore humans with their pokemons in order is:

- Ash: [Charizard(90.5),Pikachu(6)]
- Misty: [Gyarados(235), Psyduck(19.6)]

Step 4: Generate a CSV File that looks like this:

Ash	Misty	Winner
Charizard(90.5)	Gyarados(235)	Misty
Pikachu(6)	Psyduck(19.6)	Misty

Step 5: Determine the Overall Winner

Count the number of wins per human and decide the final outcome:
Misty wins both rounds (2 wins), and Ash has 0 wins. Therefore, Misty is the overall winner.

Step 6: Final output: Print the overall Winner

"The winner of the duel between *Ash* and *Misty* is *Misty*! Congratulations *Misty*!!
Misty won against *Ash* with a score of 2 to 0!"

Evaluation Criteria:

- The bot should **autonomously retrieve, process, and structure** the information.
- You **cannot** preprocess the pokemon_script file and then use it. All handling of the file must happen in real time.

- Even though the task is **sequential**, focus is on **agent-building capabilities** (e.g., tool building, agent file handling, prompt engineering, parsing LLM outputs, structured data generation, etc).

- The solution should be **scalable**, meaning it can easily adapt if new characters or Pokémon are added.
- The expected output is the CSV file and the final string

LLM Usage:

- You are provided with a google gemini API key already
- You can use the gemini (gemini-2.0-flash for example) model in any way you like. Like this for example:

```
from google import genai

client = genai.Client(api_key="YOUR_API_KEY")

response = client.models.generate_content(model="gemini-2.0-flash",
         contents="Explain how AI works in a few words"
)

print(response.text)
```

Good luck! 🚀