**Homework Assignment: Basic Order Status Chatbot**

**Overview:**

You’ll create a simple backend system that simulates an order-status chatbot using the OpenAI GPT API with function calling. The goal is to allow a user to inquire about an order’s status or request that an order be canceled. You’ll use a provided CSV file as a mock database of orders and implement basic REST endpoints to handle chat interactions.

**What We Provide:**

**CSV File (orders.csv):**  
Contains a list of orders. Each row includes at least:

* + order\_id
  + status (e.g., "shipped", "processing", "canceled")
  + item (the product name)

Example CSV:

order\_id,status,item

12345,shipped,Sneakers

23456,processing,Laptop

34567,canceled,Headphones

**Requirements:**

1. **Project Goal:**  
   Implement a minimal backend system that:
   * Maintains conversation state for a user.
   * Uses OpenAI GPT API in the conversation to:
     + Lookup an order by order number in the CSV (mock DB).
     + “Cancel” an order (just update the CSV or store the updated state in memory—no actual file write required, just simulate).
2. **API Endpoints:** Implement the following endpoints (format and programming language are up to you):
   * POST /conversations  
     *Start a new conversation.*  
     Returns a conversation\_id that can be used in subsequent requests.
   * GET /conversations/{conversation\_id}/messages  
     *Get the message history for a given conversation.*  
     Should return a list of messages (user and assistant).
   * POST /conversations/{conversation\_id}/messages  
     *Send a new user message.*  
     The request body should include the user’s message.
3. **Conversation Handling:**  
   You don’t need a real database for conversations. Storing in memory is fine. The data might look like:

{

"conversation\_id": "some-uuid",

"messages": [

{"role": "user", "content": "Hi, can you check my order 12345?"},

{"role": "assistant", "content": "Sure, one moment..."}

]

}

Basic, in-memory storage is acceptable. Don’t worry about persistence across server restarts.

1. **Notes:**
   * No fancy front-end required. Just the JSON endpoints.
   * Keep it simple and minimal. We just want to see how you structure the code and integrate with OpenAI’s API.
   * Focus on correctness and clarity rather than polish.
   * Don’t worry too much about authentication or error handling—just keep it reasonable.
   * Assume all requests come in from a single user for this exercise; no need for user accounts.

**Expected Deliverables:**

* A small codebase (a single file or a few files) that:
  + Sets up a minimal server with the specified endpoints.
  + Integrates with OpenAI GPT function calling.
  + Loads the CSV into memory on startup for lookups.
  + Handles conversation flow as described.
* Instructions on how to run the server (dependencies, environment variables for the API key, etc.).

**Timebox:**

Spend no more than one hour on this. We are looking for how you approach the problem, not a production-ready solution.

**Good Luck!**