Development assessment: Agents

Assessment

- This case is about a Intelligent Multi-Domain Chatbot with Information Retrieval

Project Goal

- **Objective:** Design and implement a chatbot capable of handling user queries across multiple distinct domains.
- **Core Idea:** Leverage different information sources (APIs, databases) to provide relevant answers.

Chatbot Functionality

The chatbot should be able to:

- Greet the user warmly.
- Handle questions about cities and weather.
- Handle questions about research subjects.
- Handle questions about specific products.

Functionality - Cities & Weather

- **If the user asks about a city:**
  - Use a remote API (e.g., Wikipedia API) to fetch general information.
- **If the user asks about weather in a specific city:**
  - Use an open-source weather API (candidate's choice).
  - Retrieve and present current weather conditions.

Functionality - Research Subjects

- **If a query appears to be about a research topic:**
  - Use a remote API (e.g., Semantic Scholar API) OR general web scraping (if API integration is complex).
  - Appropriate libraries should be used.
  - Present a concise summary of the findings.

Functionality - Specific Products

- **If the user asks about a product in the database:**
  - Utilize a PostgreSQL database containing information about specific products (name, description, price, etc.).
  - Connect to the database.
  - Retrieve relevant information using SQL queries.
  - Present the information to the user.

Key Requirements - Overview

- The implementation must meet several key technical requirements:
  - Chatbot Interface
  - Domain Identification
  - API Integration
  - PostgreSQL Integration
  - Information Extraction & Presentation
  - Error Handling
  - Modularity
  - Postman collection

Key Requirement 1: Chatbot Interface

- **Implementation:**
  - Basic conversational interface.
- **Nature:**
  - Text-based.
- **Framework Suggestion:**
  - FastAPI (for simplicity and speed).

Key Requirement 2: Domain Identification

- **Need:**
  - Intelligently determine the user's intent (city/weather, research, product).
- **Approach:**
  - Use supervisor to decide. Rule-based or a simple classification method is not acceptable.

Key Requirement 3: API Integration

- **Task:**
  - Demonstrate interaction with external APIs (Wikipedia, weather, research).
- **Process:**
  - Handle API requests and responses appropriately with fast Api rules.

Key Requirement 4: PostgreSQL Integration

- **Database Setup:**
  - Set up a PostgreSQL database (local instance or cloud-based free tier).
  - Design a simple schema for product information.
- **Application Logic:**
  - Implement necessary Python code to connect.
  - Execute SQL queries based on user input.
  - Handle database interactions effectively.

Key Requirement 5: Information Extraction & Presentation

- **Goal:**
  - Extract relevant information from API responses and database queries.
- **Output:**
  - Present the extracted information to the user in a clear and concise manner.

Key Requirement 6: Error Handling

- **Implementation:**
  - Basic error handling mechanisms.
- **Scope:**
  - API failures (e.g., timeouts, invalid responses).
  - Database connection issues.
  - Scenarios where requested information is not found.

Key Requirement 7: Modularity

- **Design Principle:**
  - Structure the chatbot code modularly.
- **Goal:**
  - Separate different functionalities logically.
- **Examples:**
  - Domain Identification logic.
  - API interaction modules.
  - Database interaction components.

GOOD LUCK!

- You have 2 day for this case.