Task Title: Build a Simple Chatbot for Idea Suggestions

Objective: We want to see how well you can create a basic chatbot using backend coding that:

1. Generates **3 unique ideas** (intents) from a user's input.
2. Let the user pick **2 ideas** they like.
3. Give **detailed suggestions** for each of the selected ideas.

Task Description

Scenario: Imagine someone is looking for new ideas and asks for help. Your chatbot should:

- **Use this query** "What new app should I build?"
- Use AI to create **3 unique ideas** from that query.
- **Example Ideas Generated:**
    - 1. A meditation app with rewards.
    - 2. Grocery delivery for special diets.
    - 3. Subscription service for AI-generated art.
- Let the user pick **2 ideas** by typing numbers (like "1 and 3").
- Give a useful and detailed suggestion for each of the chosen ideas.

Core Requirements

1. **Generate Ideas (Intents):**
    - Use a **free AI API** like Hugging Face or OpenAI's free plan to create 3 ideas.
    - **Example Input:** "What new app should I build?"
    - **Example Ideas Generated:**
        - A meditation app with rewards.
        - Grocery delivery for special diets.
        - Subscription service for AI-generated art.
2. **Backend Setup:**
    - Use **Python (Flask)** or **JavaScript (Express.js) or whatever you want** to make a simple backend.
    - Integrate the AI API to generate the ideas.
    - **Handle API Keys Safely:** Store them securely in environment variables (like a `.env` file).
3. **API Integration:**
    - Use `requests` (in Python) or `axios` (in JavaScript) to call the AI API and get results or any solution you choose.
    - Include basic error handling to manage any API problems.
4. **User Interaction (Text-Based (Terminal)/ CLI (Command Line Interface)):**
    - Display the 3 ideas and ask the user to select 2 by number.

    - **Example Prompt:** “Please choose two ideas by typing their numbers (e.g., 1, 3).”
5. **Expand on Ideas:**
    - Give **detailed suggestions** for each chosen idea.
    - **Example for Meditation App Idea:**
        - “Add features like streak tracking, reward points, and badges for milestones. Market to mindfulness enthusiasts and promote through social media.”

Video Demonstration Requirement

Objective: The purpose of this video demonstration is to help us understand your approach and thought process in building the chatbot, as well as to ensure that the code works as intended. This also gives us a clear view of how the chatbot performs in real-time.

What to Include in Your Video Demo

1. **Code Walkthrough**:
    - Briefly walk us through your code.
    - Highlight the key sections, such as the API integration, error handling, and the logic used for generating and expanding on ideas.
    - Explain any important coding decisions or unique implementations you used.
2. **Chatbot Demonstration**:
    - Show your chatbot running in the terminal or command-line interface.
    - Demonstrate the chatbot's functionality by providing the required query: **“What new app should I build?”**
    - Ensure that the chatbot generates 3 unique ideas, lets you pick 2, and gives detailed suggestions for each.
    - If you implemented the bonus feature (Priority-Based Suggestions), demonstrate how it works and explain the ranking logic.

Technical Requirements for the Video

- **Length**: The video should be concise, ideally between 2 to 3 minutes.
- **Format**: The video should be in a common format (e.g., MP4, MOV) and easily accessible via a link (e.g., Google Drive, YouTube, or another sharing platform).
- **Quality**: Ensure the video and audio quality are clear enough for us to understand your explanations and see the chatbot in action.

Submission Instructions

- **Video File**: Upload your video to a file-sharing service and provide a link to access it.
- **Code**: Ensure your code is included in the submission (e.g., GitHub repository or zipped file) alongside the video link.

Evaluation Criteria for the Video Demo

1. **Clarity of Explanation**: How well you explain your code and the logic behind your implementation.
2. **Functionality Demonstration**: Whether your chatbot works as expected and follows the task requirements.
3. **Presentation Quality**: The clarity of the video and how effectively you present your solution.

This section will ensure that we can accurately assess your technical abilities, code quality, and how well your chatbot performs in a real scenario.

Bonus Task Description

1. **Priority Ranking of Intents:**
    - Implement a system where the 3 generated intents are ranked based on:
        - **Relevance:** How well each intent aligns with the original user query. You can use ai to check relevance of the insights
        - **Potential Impact:** The significance or expected ROI of each intent.
        - **Feasibility:** How easily the intent can be acted upon.
    - Each intent should display a **priority score** (e.g., 1-5, with 1 being the highest priority).
2. **User Interaction with Prioritized Intents:**
    - After generating and ranking the intents, present them to the user in order of priority.
    - Allow the user to either:
        - **Select from the ranked list**, or
        - **Request an explanation** for why a specific intent was given its priority score.
3. **Technical Implementation:**
    - **Scoring Logic:** Use simple heuristic rules to assign scores (e.g., using keywords or phrases to determine relevance and impact).
    - Ensure that the ranking system is intuitive and that explanations for each score are clear.

Bonus Submission Details:

1. **Updated Code:**
    - Include the code for the priority ranking feature.
2. **Explanation:**
    - A brief note (max 150 words) explaining the logic used to prioritize intents.

Evaluation Criteria for Bonus:

1. **Effectiveness of Ranking:**
    - Are the most relevant and impactful intents prioritized appropriately?
2. **User Understanding:**
    - Are the explanations for priority scores clear and logical?
3. **Technical Execution:**
    - Is the scoring system well-implemented and efficient?

This bonus section adds depth to the task and evaluates the candidate's ability to think strategically about user experience and intent relevance. Let me know if this aligns with your expectations!

If you complete this bonus section, make sure to let us know on the internshala chat.