RecipeMate - Conversational Recipe Innovation Bot

Objective

Build a conversational AI system that interacts naturally with users to create innovative recipes based on their inputs.

The system should:

1. Understand the user's needs through dialogue (not intent mapping).
2. Ask clarifying questions when needed to gather context or resolve ambiguities.
3. Provide clear explanations of its capabilities and limitations.
4. Gently guide the user back on track if the conversation diverges from the system’s goal (recipe innovation).

Key Features

1. **Conversational Understanding**
    - Utilize natural language processing (NLP) to:
        - Parse user inputs.
        - Detects missing information (e.g., user says, "I want to cook something," but doesn't provide ingredients).
        - Maintain context over the course of the conversation (e.g., remembering previously mentioned ingredients).

2. **Dynamic Questioning**
    - Ask follow-up questions to gather the necessary information:
        - Ingredients available: “What ingredients do you have on hand?”
        - Preferences: “Do you have a preference for vegetarian or non-vegetarian recipes?”
        - Equipment: “Do you have an oven, stove, or other specific tools available?”
    - Adapt questions based on the user’s responses to ensure clarity and completeness.

3. **Guidance on Scope**
    - Gently redirect the user if they go off-topic:
        - User: “Can you book a restaurant for me?”
        - System: “I’m here to help you create recipes! If you’d like, I can suggest recipes that are similar to restaurant dishes.”
    - Handle vague inputs:
        - User: “I don’t know what to cook.”
        - System: “No problem! Just tell me what ingredients you have, and I’ll help you brainstorm recipes.”

Core Workflow

1. **Understand User Intent Through Conversation**
    - Extract context dynamically without rigid intent mapping:
        - User: “I want to cook something Italian.”
        - System: “Great! Do you have any ingredients in mind? Or should I suggest based on popular Italian dishes?”
        - User: “I have chicken, garlic, and tomatoes.”

2. **Ask Clarifying Questions**
    - Gather essential information for recipe generation:
        - “Do you have pasta or rice available?”
        - “Are there any dietary restrictions I should consider?”

3. **Generate Recipes**
    - Use the **Spoonacular API** to:
        - Fetch recipes matching user inputs.
        - Retrieve cooking techniques, preparation times, and ingredient pairings.

    - Innovate new recipes by:
        - Combining traditional methods with creative twists.
        - Suggesting ingredient substitutions or new approaches.

4. **Reflect and Validate**
    - Cross-check generated recipes with **Spoonacular API** to ensure:
        - Accurate cooking times and temperatures.
        - Logical ingredient combinations.

Expected Implementation

1. Frontend (Streamlit):
    - Chat-style interface.
    - Dynamic response display.
    - Step-by-step recipe generation.
2. Backend (FastAPI):
    - Routes for:
        - Ingredient understanding.
        - Recipe generation using Spoonacular.
        - Dynamic question handling.
3. Spoonacular Integration:
    - Recipe search and filtering
    - Ingredient substitution lookup
    - Nutritional information retrieval
    - Recipe analysis and validation

Expected Structured Output

```
{
  "recipe_title": "Garlic Chicken Pasta in Tomato Sauce",
  "ingredients": [
    {
      "ingredient": "2 chicken breasts",
      "amount": "2",
      "unit": "pieces"
    },
    {
      "ingredient": "3 tomatoes",
      "amount": "3",
      "unit": "pieces"
    },
    {
      "ingredient": "4 garlic cloves",
      "amount": "4",
      "unit": "cloves"
    },
    {
      "ingredient": "250g pasta",
      "amount": "250",
      "unit": "grams"
    }
  ],
  "cooking_time": "30 minutes",
  "cooking_temperature": "Medium heat",
  "steps": [
    {
      "step_number": 1,
      "instruction": "Boil water for pasta.",
      "explanation": "Pasta needs plenty of water to cook properly, so we start with this to ensure it’s ready when the chicken
and sauce are done."
    },
    {
      "step_number": 2,
      "instruction": "Cut chicken into even pieces.",
      "explanation": "Cutting the chicken into even pieces ensures they cook at the same rate, preventing under or overcooked
portions."
    },
    {
      "step_number": 3,
      "instruction": "Crush garlic.",
      "explanation": "Crushing garlic releases more flavor compared to chopping, helping the sauce have a deeper flavor profile."
    },
    {
      "step_number": 4,
      "instruction": "Cook chicken on medium heat until golden and cooked through.",
      "explanation": "Cooking the chicken at medium heat prevents it from burning while ensuring it cooks evenly inside."
    },
    {
      "step_number": 5,
      "instruction": "Add tomatoes and cook down into a sauce.",
      "explanation": "The tomatoes release moisture as they cook, forming a natural sauce base, which complements the chicken."
    },
    {
      "step_number": 6,
      "instruction": "Add cooked pasta to the sauce and mix.",
      "explanation": "Combining the pasta with the sauce allows the pasta to absorb the flavors and marry the dish together."
    }
  ],
  "tips": [
    "Salt pasta water generously to enhance flavor.",
    "Don’t overcrowd the chicken when cooking, as it may prevent even cooking.",
    "Let the sauce simmer to thicken and concentrate flavors."
  ]}
```

Expected Deliverables

01. Source Code - readme, requirements, Environment config files
02. A video showcasing the interaction with chatbot

Timeline

- 07 days from the date of receipt