

# Recipe Recommendation Chatbot

## Introduction:

The Cooking Recipes Chatbot is designed to provide users with personalized recipe recommendations based on their culinary preferences and available ingredients. By leveraging Retrieval-Augmented Generation (RAG) and a vector database, the chatbot can efficiently retrieve and suggest recipes that match the user's specific requirements, ensuring a tailored and engaging cooking experience.

## Scope:

The chatbot will focus on delivering personalized cooking recipe recommendations. It will cater to various user preferences, such as dietary restrictions, cuisine types, available ingredients, and cooking time. The chatbot will offer a user-friendly interface for natural language interaction, leveraging the capabilities of LangChain for managing the flow of interactions and storing data in a vector database.

## Objective:

The primary objective is to create a chatbot that can:

1. Understand and process user preferences and needs through a conversational interface.
2. Generate accurate and relevant recipe recommendations using RAG.
3. Ensure that all recommendations are based solely on the pre-existing dataset.
4. Provide intelligent and context-aware responses to user queries.
5. Utilize a vector database for efficient storage and retrieval of recipe data.

## Approach Taken

### Data Collection and Preparation

The initial step involved gathering a comprehensive dataset of recipes across four cuisines: Italian, Mexican, Thai, and Indian. This data was collected from pre-existing PDF files, ensuring a rich source of information for the chatbot to draw from. The PDFs were then processed to extract textual content using PyMuPDF, with a fallback to pdfminer for robustness.

### Vector Database Implementation

Pinecone was chosen as the vector database due to its efficient indexing and retrieval capabilities. The extracted text from the PDFs was chunked into manageable pieces using the RecursiveCharacterTextSplitter and then embedded using OpenAIEmbeddings. These embeddings were stored in Pinecone, creating a vector index that could be queried for relevant information.

### Retrieval-Augmented Generation (RAG) Pipeline

The core of the chatbot's functionality lies in its RAG pipeline. This pipeline involves querying the vector database for relevant information based on the user's input and using the LangChain framework to generate responses. The chatbot maintains context across conversations, enhancing its ability to provide coherent and relevant responses.

### Handling Out-of-Scope Queries and Unavailable Data

To ensure the chatbot remains focused on its domain, mechanisms were implemented to handle out-of-scope queries and unavailable data. When a query falls outside the chatbot's scope, it provides a polite response redirecting the user to relevant resources. Similarly, if the requested data is not available, the chatbot informs the user and offers assistance with other queries.

## Challenges Faced

### Integration with Pinecone

One of the significant challenges was integrating Pinecone for efficient data retrieval. Ensuring that the vector index was accurately created and queried required careful handling of the data embedding and indexing processes.

### Context Management

Maintaining conversation context across multiple interactions was crucial for providing relevant responses. This involved implementing a robust session management system to track and recall user queries and responses within the same session.

### Handling Out-of-Scope Queries

Detecting and appropriately responding to out-of-scope queries was another challenge. This required defining a set of criteria to identify such queries and crafting responses that kept the chatbot focused on its intended domain while guiding the user effectively.

## Limitation –

4 cuisines only for now – Indian, Thai, Italian and Mexican (40 total)

## Type of Data:

1. **Recipe Titles:**
  - Short, descriptive titles of the recipes.
2. **Ingredients:**
  - Detailed lists of ingredients required for each recipe, including quantities and any specific notes (e.g., organic, substitute options).
3. **Cooking Instructions:**
  - Step-by-step instructions on how to prepare the dish, including preparation and cooking times.
4. **Cuisine Type:**
  - Categories indicating the cuisine type (e.g., Italian, Mexican, Asian, Vegan).
5. **Dietary Information:**
  - Information on dietary restrictions and suitability (e.g., vegetarian, gluten-free, keto-friendly).
6. **Nutritional Information:**
  - Details on the nutritional content of the recipes (e.g., calories, protein, fat, carbohydrates).
7. **User Ratings and Reviews:**
  - Optional user ratings and reviews to provide additional context and feedback on the recipes.
8. **Preparation and Cooking Time:**
  - Estimated preparation and cooking times to help users plan their meals effectively.
9. **Serving Size:**
  - Information on the number of servings each recipe yields.
10. **Additional Notes:**
  - Any additional notes or tips related to the recipe (e.g., storage suggestions, serving ideas).

# Overcoming Challenges

## Efficient Data Chunking

To handle large PDF files efficiently, an asynchronous method was implemented to load and chunk the data. This approach significantly improved response times by processing data in parallel and reducing bottlenecks.

## Enhanced Prompts

Using detailed and structured prompts was key to guiding the language model in generating accurate and context-aware responses. These prompts were crafted to provide clear instructions to the model, ensuring that it adhered to the intended scope and context.

## Error Handling and User Guidance

Incorporating checks for out-of-scope queries and unavailable data enhanced the user experience. The chatbot was programmed to provide polite and informative responses, guiding users towards relevant queries or suggesting alternative approaches.

## Conclusion

The Recipe Recommendation Chatbot successfully demonstrates the potential of a RAG pipeline in providing personalized and context-aware recipe recommendations. By leveraging advanced data retrieval and language generation techniques, the chatbot delivers a seamless and engaging user experience. The challenges faced during development were effectively addressed through innovative solutions, resulting in a robust and efficient system.

## Future Enhancements

- **User Profiles:** Implementing user profiles to store preferences and history for more personalized recommendations.
- **Enhanced Data Features:** Including more detailed recipe metadata, such as nutritional information, difficulty levels, and cooking times.
- **Community Features:** Allowing users to rate and review recipes, and submit their own recipes for a richer community-driven experience.