Technical Documentation for Restaurant RAG-based Chatbot

# System Architecture Explanation

The architecture of the Restaurant RAG-based Chatbot involves several key components:  
1. Data Collection: Data is fetched using a restaurant data scraper, which collects information about restaurants, such as location, cuisines, menu items, spice levels, offers, and other relevant details. This data is structured in a JSON format.  
2. Context Generation: The collected data is then used to generate contexts for each restaurant. This includes information about the restaurant itself (name, location, address, etc.) as well as detailed descriptions of the menu items, including their spice levels and gluten-free status.  
3. Embedding and Retrieval:  
 - The contexts are passed through the DPR Context Encoder (based on Facebook's DPR model) to generate dense vector embeddings.  
 - These embeddings are stored in a FAISS index for efficient similarity search.  
4. Query Processing: The user's query is encoded using the DPR Question Encoder to generate a query embedding, which is then used to retrieve the most relevant restaurant contexts from the FAISS index.  
5. Answer Generation: The retrieved contexts are passed along with the user's query to the Flan-T5 model (a transformer-based model) for answer generation.  
6. Interface: A Gradio-based web interface is used to interact with the chatbot, allowing users to ask questions about restaurants, which are answered using the retrieved contexts and generated answers.

# Implementation Details and Design Decisions

1. Initial Model: Initially, a simpler transformer model was used for context retrieval and question answering. However, after experimenting with various approaches, the system was upgraded to use the DPR (Dense Passage Retrieval) model for better accuracy and efficiency. The DPR model separates the process of encoding questions and contexts, allowing for more effective retrieval.  
2. Features Added:  
 - Spice Level: To enhance the chatbot's ability to compare restaurants, a spice level feature was added to the menu items. This allows the system to compare restaurants based on the spiciness of their menu items.  
 - Gluten-Free Items: Information about gluten-free items was also incorporated into the context, allowing users to specifically ask about such options.  
3. Libraries and Tools Used:  
 - Hugging Face Transformers: For the DPR-based model (DPR Context Encoder and DPR Question Encoder) and Flan-T5 for answer generation.  
 - FAISS: For creating an efficient index for fast context retrieval.  
 - Gradio: For building an interactive web interface for the chatbot.

# Challenges Faced and Solutions Implemented

1. Limited Query Understanding: Initially, the simple transformer model struggled with understanding and correctly answering complex or specific queries. This was mitigated by switching to the DPR-based retrieval approach, which improved both context retrieval and query matching accuracy.  
2. Data Inconsistencies: Some restaurants had missing or inconsistent data (e.g., missing spice level or allergens). Data preprocessing techniques were used to handle such issues and ensure that the system could function even with incomplete data.  
3. Spice Level Inference: Since some menu descriptions didn't explicitly mention spice levels, a rule-based system was created to infer the spice level from the description text (e.g., using keywords like 'spicy' and 'mild').  
4. Gluten-Free Items: Identifying gluten-free items was challenging due to inconsistent naming conventions. A simple approach of matching the term 'gluten' in both item names and descriptions was used, with manual checks for accuracy.

# Future Improvement Opportunities

1. NLP Enhancements: Future improvements could involve adding more sophisticated NLP techniques, such as:  
 - Named Entity Recognition (NER) to extract specific details (like ingredients, allergens, etc.) more reliably.  
 - Text Summarization to condense long restaurant descriptions into more readable and user-friendly summaries.  
2. Contextual Enhancements:  
 - Personalized Recommendations: Using machine learning to recommend restaurants based on user preferences or past interactions.  
 - Advanced Feature Extraction: Adding more features to the restaurant data (e.g., customer reviews, ratings, etc.) to enhance the recommendation system.  
3. Cross-restaurant Comparisons: Enabling users to compare multiple restaurants based on various features such as menu options, price, ratings, and more.  
4. User Feedback Integration: Incorporating user feedback to refine and personalize responses, ensuring the system continuously improves over time.