Generative AI Solution for Zomato Restaurant Chatbot

1. System Architecture Overview

The system integrates web scraping, data preprocessing, a semantic search engine, and a language model to build a Retrieval-Augmented Generation (RAG) chatbot. It comprises:

- A Selenium-based scraper to collect data from Zomato.
- A Pandas and regex-powered data loader for preprocessing.
- A SentenceTransformer-based semantic vector DB for retrieval.
- A Llama 3 model served via Groq for response generation.
- A command-line interface or potential Gradio/Streamlit app for interaction.

2. Web Scraping Strategy and Data Schema

The scraper uses Selenium and BeautifulSoup to dynamically scroll and extract details from Zomato listings.

Collected attributes include:

- Restaurant name and URL
- Cuisine type
- Rating
- Price for one

Data is saved into 'restaurant_data.csv' with fields:

[links, names, ratings, price for one, cuisine, location]

3. Knowledge Base Creation

The knowledge base is constructed by preprocessing the scraped CSV data. A new column 'search_text' is created containing concatenated restaurant info. Locations are parsed from URL slugs.

The SentenceTransformer model ('all-MiniLM-L6-v2') is used to encode this information into embeddings, which are stored along with metadata for later retrieval.

4. RAG-Based Chatbot Design

The chatbot uses a vector DB to retrieve top relevant entries from the knowledge base.

A formatted prompt combining the query and results is fed to the LLM (Llama3-8b-8192 via Groq API).

Query types supported:

- Menu availability and cuisine details
- Feature comparisons (e.g., spice level, rating)
- Price range
- Vegetarian or allergen filters

It includes safeguards against hallucination and only responds using context from the knowledge base.

5. Challenges and Solutions

- Extracting reliable information from Zomato required scrolling and dynamic rendering.
- Data consistency (e.g., some restaurants lacking fields) was handled with try-except blocks.
- Avoiding API cost: Groq and Hugging Face APIs were selected as free-tier services.
- Ensuring strict context-based generation to prevent hallucination.

6. Future Improvements

- Add restaurant menus, images, contact info, and opening hours (where legally permitted).
- Use a more advanced retrieval system (e.g., FAISS with filtering).
- Expand to multiple cities and support multi-turn dialogue.
- Add Streamlit/Gradio UI for enhanced user experience.

- Support Hindi/vernacular queries.