Project Report: Mr. HelpMate AI

1. Objective

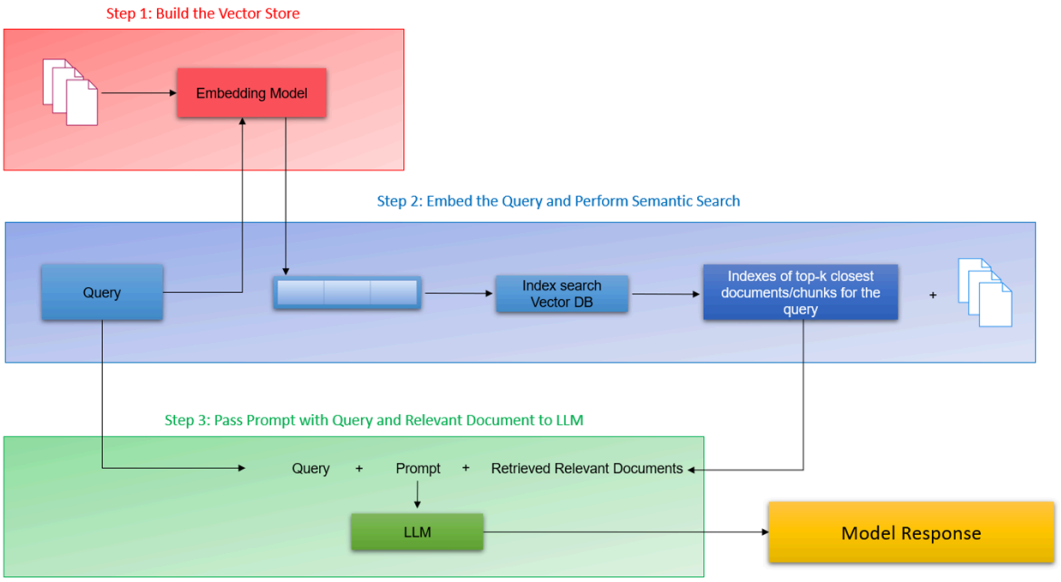
The objective of the Mr. HelpMate AI project is to design and implement an intelligent chatbot assistant for GetFlowersDaily.com. The chatbot, known as "Mr. HelpMate", is built to assist users with product information, availability, and delivery-related queries. It focuses on improving customer experience by offering instant and accurate responses about fresh pooja flowers, garlands, and temple items.

2. Design

The design of Mr. HelpMate AI revolves around leveraging advanced Natural Language Processing (NLP) technologies powered by OpenAI's GPT models. The chatbot integrates a Retrieval-Augmented Generation (RAG) framework to ensure responses are contextually accurate and domain-specific. The RAG approach allows the model to refer to a pre-curated knowledge base of GetFlowersDaily's products, services, policy, and delivery guidelines before generating the final user response.

Key design components include:

1. **User Query Interface**: Collects queries from customers through the website chat widget.
2. **Retriever Module**: Searches a local (ChromaDB) or cloud-based document store for relevant product and policy information.
3. **Generator Module (GPT)**: Uses OpenAI's GPT model to generate natural language responses by combining retrieved context with user input.
4. **Response Layer**: Formats and delivers the final answer to users in a friendly, conversational tone.



3. Implementation

The implementation was carried out using Python in a Colab environment. The project utilized the OpenAI API for response generation and the RAG framework for knowledge retrieval. A dataset containing GetFlowersDaily product details and frequently asked questions was embedded into a ChromaDB vector database for semantic search.
Implementation Steps:

1. **Data Preparation:** Product catalog and delivery policy documents were cleaned and stored as text data.
2. **Vector Database Setup:** ChromaDB was used to convert text documents into embeddings using OpenAI's embedding model.
3. **Retrieval Logic:** For every incoming user query, the top 3 most relevant chunks were retrieved from ChromaDB.
4. **Response Generation:** Retrieved data and user queries were passed to OpenAI GPT to generate an accurate, conversational response.

The architecture ensures scalability, allowing the chatbot to handle multiple queries simultaneously while maintaining accuracy and context relevance.

4. Challenges

During development, several challenges were encountered:

1. **Data Collection**: Generating PDF document for each product from live website.
2. **Model Prompting**: Designing effective system and user prompts was crucial to maintaining context and brand tone.
3. **Database Lock Issues**: Temporary database write errors (e.g., 'attempt to write a readonly database') occurred during ChromaDB integration.
4. **Relevance Filtering**: Some irrelevant retrievals initially reduced accuracy, which was later resolved using improved similarity thresholds.

Each challenge contributed to refining the final architecture and improving system robustness.

5. Lessons Learned

The project provided key insights into practical AI implementation using GPT and RAG architectures. Some of the major lessons learned include:

1. Effective prompt engineering is critical for maintaining a conversational tone and factual accuracy.
2. Combining GPT with a retrieval system (RAG) significantly improves reliability for domain-specific tasks. ChromaDB offers an efficient and lightweight vector store solution for local deployment. Maintaining clear separation between system logic (retrieval, generation, and response) simplifies troubleshooting and scaling.
3. Even a small business can leverage modern AI tools to provide enterprise-grade customer support experiences.

Overall, Mr. HelpMate AI demonstrates how generative AI can be applied to enhance customer engagement and operational efficiency in niche e-commerce environments.

6. Project Summary - Mr. HelpMate AI

Mr. HelpMate AI is an AI-powered chatbot designed for GetFlowersDaily.com to assist users with product and delivery-related queries. It uses OpenAI's GPT model and a Retrieval-Augmented Generation (RAG) approach to provide accurate, contextual, and customer-friendly responses. Challenges such as data structuring and database errors were overcome through effective prompt design and system optimization. The project demonstrates how small businesses can leverage AI for better customer experience and operational support.