**HelpMate AI – System Documentation**

**Objective**

HelpMate AI is designed to provide intelligent, accurate, and efficient assistance by answering user queries about insurance policy documents. The chatbot system is capable of:

* Understanding natural language queries related to insurance policies.
* Retrieving relevant information from policy documents using semantic search.
* Generating clear, informative, and concise responses.
* Providing source citations to support its answers.
* Optimizing performance with a caching mechanism for faster response times.

**System Architecture Overview**

HelpMate AI is built using a modular, scalable architecture, consisting of the following main components:

1. **Data Ingestion and Preprocessing**
2. **Embedding and Indexing**
3. **Query Processing**
4. **Response Generation**
5. **Caching Mechanism**
6. **User Interface**

A high-level workflow is shown below:

**PDF Document** ➔ **Text Chunking** ➔ **Embedding and Indexing** ➔ **Semantic Search** ➔ **Response Generation** ➔ **Caching** ➔ **User Interaction**

**Components Detail**

**1. Data Ingestion and Preprocessing**

* **Input**: Insurance policy document (PDF format).
* **Process**:
  + Utilize pdfplumber to extract:
    - Raw text
    - Headings
    - Tables
  + Split the extracted text into smaller overlapping chunks to preserve context.
  + Annotate each chunk with metadata, such as:
    - Page number
    - Section heading
* **Output**: Structured dataset containing text chunks and corresponding metadata.

**2. Embedding and Indexing**

* **Input**: Preprocessed structured text data.
* **Process**:
  + Generate embeddings for each text chunk using **OpenAI’s text-embedding-ada-002** model.
  + Store embeddings, text, and metadata within a **ChromaDB** collection for efficient semantic retrieval.
* **Output**: Indexed and searchable collection in ChromaDB.

**3. Query Processing**

* **Input**: Natural language user query.
* **Process**:
  + Generate an embedding for the query using the same **text-embedding-ada-002** model.
  + Search ChromaDB to find the most semantically relevant text chunks.
  + Rerank the search results using a **cross-encoder model** to further optimize relevance and precision.
* **Output**: Ranked list of relevant text chunks corresponding to the user query.

**4. Response Generation**

* **Input**: User query + Ranked text chunks.
* **Process**:
  + Pass the query and relevant text chunks as **context** to a **GPT-4o-mini** model.
  + Design prompts carefully to instruct the model to:
    - Generate an accurate, complete response.
    - Include citations (e.g., page numbers, sections) from the source document.
* **Output**: A natural language answer, properly cited and tailored to the user’s query.

**5. Caching Mechanism**

* **Input**: User queries and corresponding responses.
* **Process**:
  + Store both the query and its generated response in a separate ChromaDB collection dedicated to caching.
  + Before performing a new query search, check the cache:
    - If a similar query is found, retrieve the cached response.
    - Otherwise, process the query as usual.
* **Output**: Reduced latency and improved response times for repeated or similar queries.

**6. User Interface**

* **Input**: User-submitted queries.
* **Process**:
  + Present an intuitive chatbot interface for users to interact with HelpMate AI.
  + Send user queries to the Query Processing module.
  + Display AI-generated responses in a user-friendly format, including citations and links where applicable.
* **Output**: Smooth, interactive, real-time conversational experience.

**Technologies Used**

| **Component** | **Technology** |
| --- | --- |
| Text Extraction | pdfplumber |
| Embedding Model | OpenAI text-embedding-ada-002 |
| Vector Database | ChromaDB |
| Reranking Model | Cross-Encoder (custom or HuggingFace) |
| Response Generation | OpenAI gpt-4o-mini |
| Programming Language | Python |
| Frontend | (To be decided, e.g., Streamlit, React) |
| Backend APIs (optional) | FastAPI, Flask |

**Key Features**

* **Semantic Search**: Understands intent beyond keywords.
* **Citations**: Provides trustworthy, traceable answers.
* **Efficiency**: Caching for faster repeat responses.
* **Adaptability**: Easily extendable to different types of documents.
* **Scalability**: Modular components support easy scaling.

**Future Enhancements (Optional Section)**

* Implement **multi-document support**.
* Add **user feedback learning loop** to continually improve accuracy.
* Integrate **multi-language support** for broader accessibility.
* Optimize **chunk size dynamically** based on query type.

**Conclusion**

HelpMate AI provides an innovative, efficient solution for navigating complex insurance documents. By combining state-of-the-art semantic search, accurate language modeling, and performance-focused caching, it ensures users get precise, explainable answers quickly and reliably.