

Chatbot Architecture

Frontend Technology: Flutter (Dart)

Purpose: The user interface allows users to interact with the chatbot, inputting queries and receiving responses.

Backend Technology: FastAPI (Python)

Purpose: The middleware that processes user queries, interacts with databases, APIs, and the chatbot engine, and returns appropriate responses to the frontend.

Chatbot Engine Technology: Python

Purpose: The engine handles the intent analysis and response generation, using a rule-based approach to determine the user's intent based on keyword matching.

Steps for Processing User Queries:

Step 1: Language Detection and Translation to English

Language Detection: The model identifies the language of the query.

Query Translation: The detected language is translated into English for further processing.

Step 2: Query Pre-processing

- **Tokenization:** Split the user's query into individual words or tokens.
- **Stop-word Removal:** Remove common words such as "the," "is," "and" that do not add meaning.

- **Noise Filtering:** Eliminate irrelevant or ambiguous words that may interfere with intent detection.

Tools for Implementation:

- Libraries like **spaCy** or **NLTK** for efficient query pre-processing.

Step 3: Keyword-based Intent Recognition

Purpose: Analyze user queries by matching predefined keywords with various sections like hotels, transportation, weather, etc.

- **Keyword Matching:** Use predefined lists of keywords to match queries with intents:
- **Hotels/Accommodation:** Keywords like hotel, stay, accommodation.
- **Transportation:** Keywords like train, bus, transport, schedule.
- **Weather:** Keywords like weather, forecast, temperature.
- **Emergency Help:** Keywords like help, emergency, police, ambulance.
- **General Information:** Keywords like Kumbh Mela, history, event details.
- **Intent Mapping:** Once a keyword is detected in the query, the system will map it to a specific intent.

- ⇒ If the query contains the word "hotel" or "stay," the intent is classified as Hotel Info.
- ⇒ If the query contains the word "train" or "bus," the intent is classified as Transportation Info.
- ⇒ If the query contains the word "weather" or "forecast," the intent is classified as Weather Info.

Entity Extraction (Optional): Extract important information like location or date if necessary (for example, "hotels near Ganga").

Step 4: Response Generation

Database Querying: Based on the recognized intent, query specific data from PostgreSQL (e.g., hotel info, emergency numbers).

API Interactions:

- **Weather Info:** Use a weather API to get real-time or forecast data.
- **Transportation Info:** Query transport APIs for train schedules.
- **General Information:** Use ChatGPT API to generate responses for broader Kumbh Mela-related queries.

Response Formatting: Generate a user-friendly, natural language response based on the intent and the retrieved data.

****Additional Features: ****

- ⇒ **Contextual Understanding:** Track conversation context across multiple turns, enabling the bot to follow up and provide context-aware answers.

Step 5: Translation to Original Language

The response is then translated back into the user's original language (if not English) using the same API before being sent back to the user to ensure clear and effective communication.

Future Scope:

1: ML-Based Intent Classification:

- ⇒ **Model:** Use of pre-trained models like BERT to classify intents in the future. The rule-based system can gradually transition to ML-based intent classification once sufficient data is collected.
- ⇒ **Training Data:** As users interact with the chatbot, queries and intents can be logged to create a dataset for training machine learning models.
- ⇒ **Multi-intent Handling:** Once ML-based intent classification is in place, the chatbot can handle multiple intents within a single query (e.g., "Show me hotels and the weather forecast for Allahabad").

2: AI-based User Behaviour Prediction: Add a module that predicts user needs based on past interactions. For example, if a user has previously asked about hotel info, the bot might suggest transportation options automatically.

3: Offline Capability: Preload essential information like emergency contacts, basic event info, and schedules within the app to provide offline functionality in areas with poor connectivity.

Security and Optimization:

Data Privacy: Ensure sensitive user data is stored securely and in compliance with privacy laws.

Rate Limiting: Apply rate limiting for API requests to prevent overloading external services.