**Abstract**

The Healthcare FAQ Chatbot is an intelligent, web-based system designed to provide users with **informational guidance about common diseases, symptoms, medicines, and precautions**. By leveraging a structured disease database and natural language input processing, the chatbot can identify potential diseases based on user queries or described symptoms. It also provides recommendations on medicines and highlights conditions that may require immediate medical consultation. This system aims to enhance awareness and preliminary self-assessment in healthcare, while emphasizing that it does **not replace professional medical advice**.

**Problem Statement**

In today’s fast-paced world, people often lack immediate access to healthcare professionals for minor symptoms or general health queries. Existing solutions may provide fragmented information or require searching multiple sources. The challenge is to create a **user-friendly, interactive system** that allows users to input symptoms, disease names, or treatment-related queries and receive **accurate, concise, and reliable information**. The system should:

1. Allow users to search for diseases or symptoms.
2. Provide symptom details, suggested treatments, and precautions.
3. Identify serious conditions and recommend consulting a relevant specialist.
4. Be flexible enough to handle free-text input and common variations in queries.

**System Design**

**1. Architecture**

* **Frontend:** Web interface built with HTML/CSS and optional JavaScript for interactivity.
* **Backend:** Python Flask app handling user queries, processing input, and returning responses.
* **Database:** A structured **disease dictionary** (healthcare\_data.py) containing:
  + Disease names and aliases
  + Symptoms
  + Suggested medicines/treatments
  + Precautions
  + Seriousness flag & specialist recommendations
* **Optional External Resource:** Wikipedia summary fallback for queries not in the local database.

**2. Flow Diagram**

User Query (disease/symptoms)

↓

Input Processing

(tokenization, keyword matching)

↓

Disease Identification

(local database match or symptom analysis)

↓

Response Generation

(symptoms, medicines, precautions, specialist advice)

↓

Display Response on Web

**3. Key Modules**

1. **Input Processing:**
   * Tokenizes user input.
   * Handles disease names, aliases, and free-text symptom descriptions.
2. **Disease Matching:**
   * Direct name or alias match.
   * Symptom-based matching using keyword overlap.
   * Optional fuzzy matching for typos.
3. **Response Module:**
   * Formats output with disease details.
   * Provides serious condition warning and recommended specialist.
   * Optionally fetches short web summaries if local database fails.
4. **Frontend Interface:**
   * Simple and clean web interface for query input.
   * Shows chatbot responses with structured information and warnings