Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecte

DEVELOPMENT OF A MEDICAL CHAT BOT (VEDA AI)

Presented By

Agrim Kulshreshtha (01318002721)
Aditya Choudhary (01918002721)
Tejas Taneja (02518002721)
Haritik Anand (05118002721)

Under Guidance of

Dr. Seema Verma

Problem Definition

The rapid innovations in healthcare technology have paved the way for new age solutions that enhance the efficacy of patient care and streamline medical processes. One such advancement is the creation of medical chatbots, designed to assist patients by providing immediate access to health information and tailored suggestions. This project introduces a new medical chatbot developed using Flask for both the frontend and backend, aimed at recommending relevant medicines and treatments based on particular conditions. The chatbot serves as a virtual assistant, offering users a sort of preliminary guidance on medications while ensuring accessibility, swiftness, and ease in regard to both useability and comprehensiveness.

Access to accurate medication advice is often limited, leading to potential misuse or delayed treatment. This chatbot provides real-time, personalized medicine recommendations based on user symptoms, promoting safer and more informed healthcare decisions.

Feasibility Study

- **Technological Feasibility:** The chatbot is built using Flask, a lightweight and well-supported framework for web applications. Flask's flexibility allows a convenient and smooth integration of the frontend, backend, and necessary APIs (such as drug databases). Natural language processing (NLP) libraries like NLTK can be easily integrated for better symptom interpretation. This makes the project both technically feasible and scalable to a good extent**.**
- **Economic Feasibility:** The cost of developing this chatbot is relatively low compared to hiring additional healthcare professionals for similar advisory tasks. Open-source libraries, cloud infrastructure, and Flask's simplicity mitigates development and maintenance costs**.**
- **Operational Feasibility:** Once developed, the chatbot can be easily deployed and used with minimal training required. It can run 24/7 and serve a large number of users without any exponential or problematic operational overhead.

Need and Significance

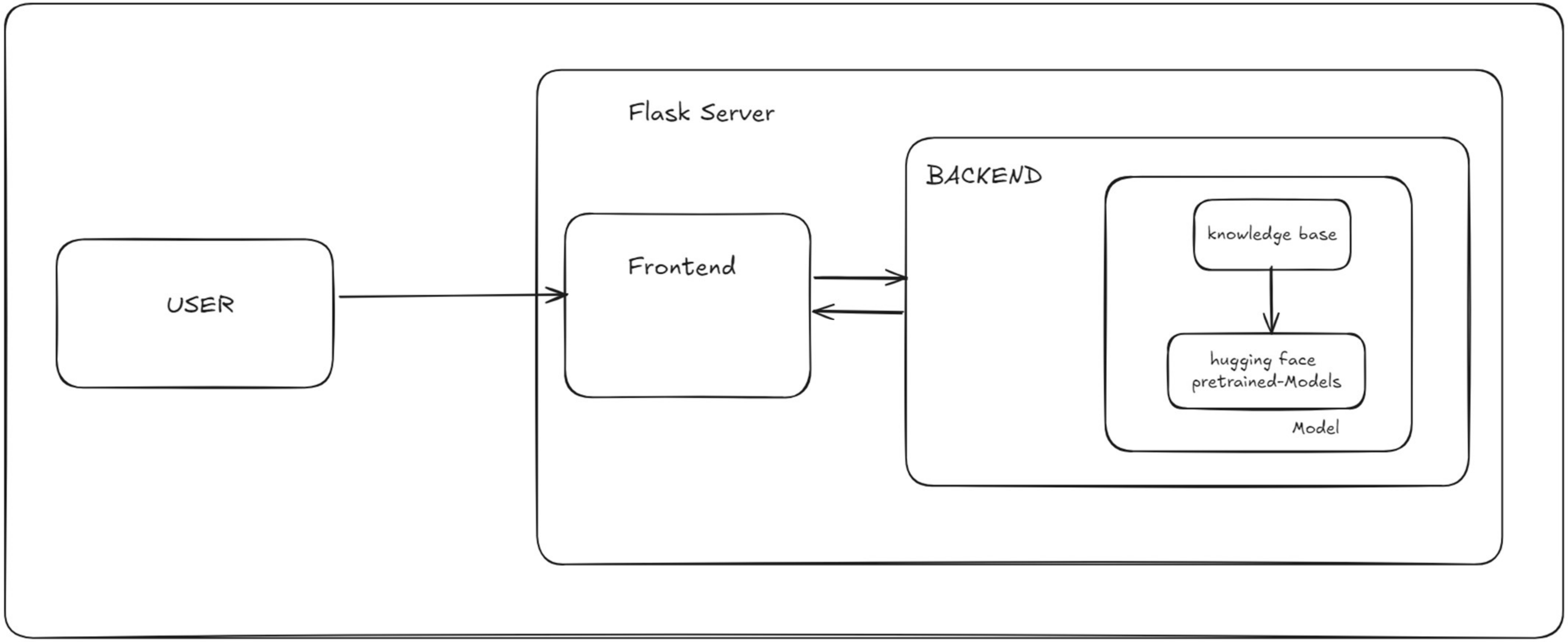
- **Increased Healthcare Demand:** With steadily rising global healthcare needs and stretched medical resources, patients often face long waiting times for consultations. A medical chatbot can bridge the gap by providing fundamental or preliminary advice quickly, reducing the burden on healthcare providers**.**
- **Access to Information:** Many individuals, especially in remote or underserved areas, have a very limited access to medical professionals. This project serves as an accessible tool for obtaining necessary and quick advice and medication recommendations, improving patient care by a significant measure.
- **Self-medication Risks**: Many people resort to self-medicating without having the proper guidance, which can lead to multiple complications. A chatbot providing precise, symptom-specific recommendations can recede this risk by guiding users towards suitable treatments.

Objectives

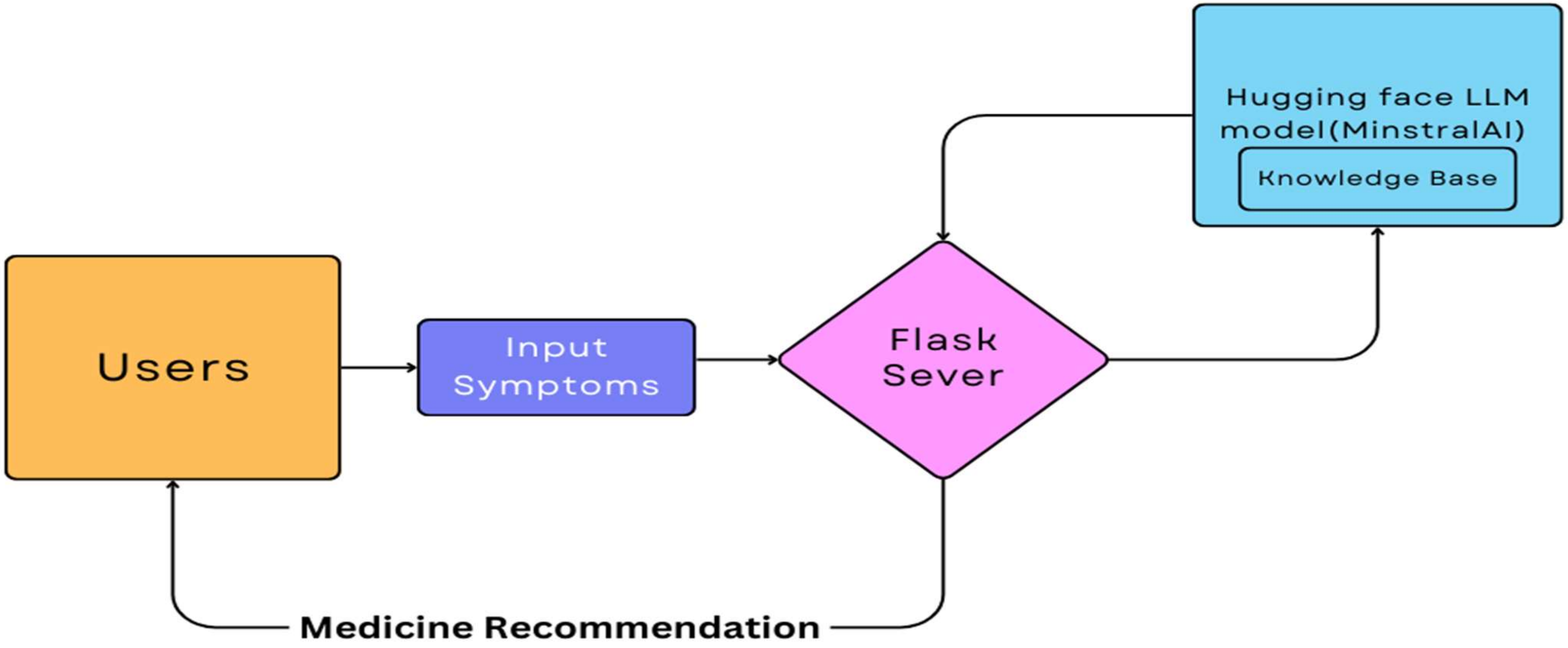
Enhance Access to Medical Guidance:
The Flask-based chatbot offers a user-friendly interface, allowing individuals to input symptoms and receive reliable medical advice quickly, without needing medical expertise or extensive searching.

Provide Accurate Medication Recommendations:
The chatbot delivers precise medicine suggestions based on user-reported symptoms, ensuring relevant guidance through a robust Flask backend, aligned with standard medical guidelines.

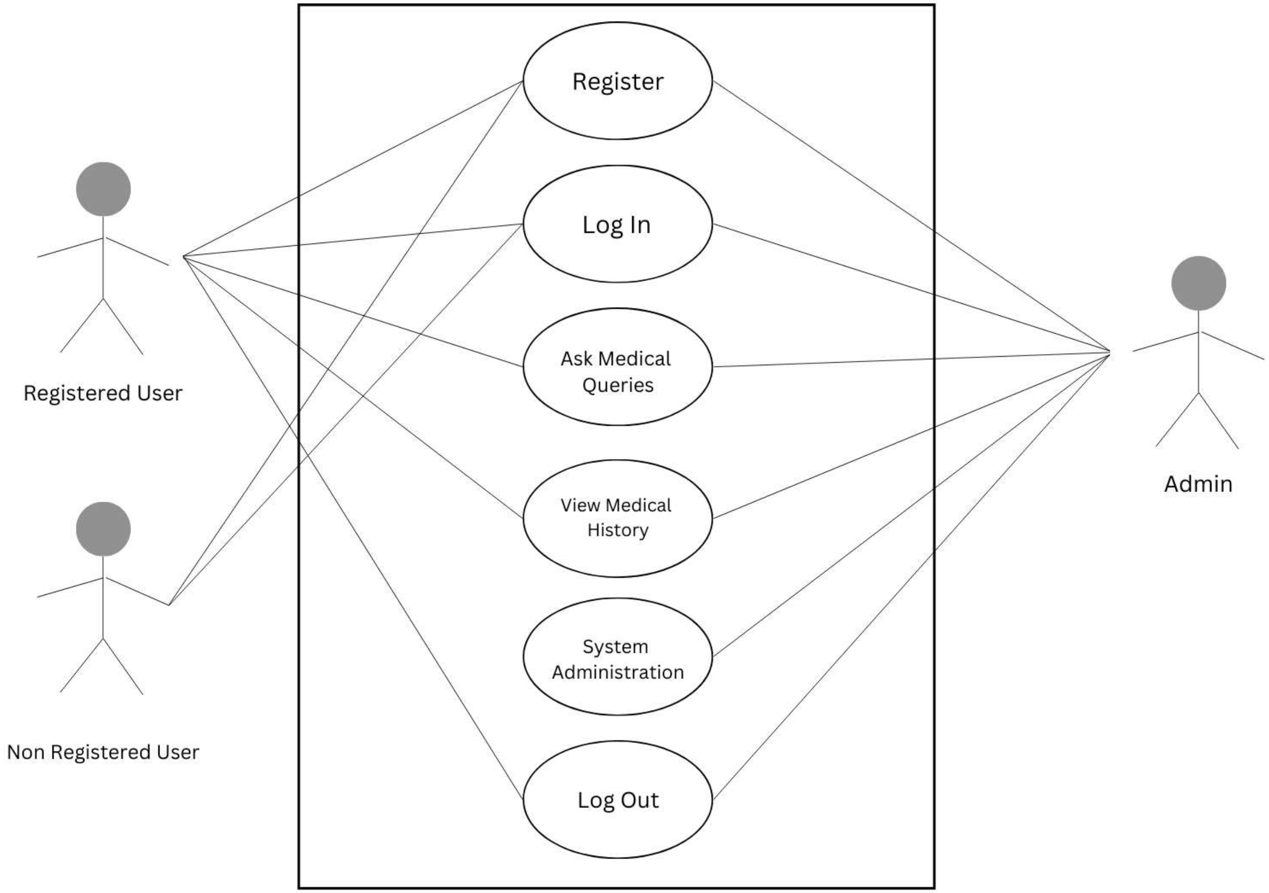
System Architecture



Flow Chart


Hugging face LLM model(MinstralAI)
Knowledge Base
Users
Input Symptoms
Flask Sever
Medicine Recommendation

Use Case



Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecture

Affiliated to GGSIP University, New Delhi
Approved by AICTE & Council of Architecture

Software Requirements

Frontend

- **HTML :** HTML is the standard markup language for structuring content on the web, defining the layout and elements of the chatbot's interface.
- **CSS :** CSS styles HTML elements, controlling the visual presentation and design of the chatbot, including colors, fonts, and spacing.
- **JavaScript :** JavaScript adds interactivity and dynamic functionality to the chatbot, handling user interactions and updating content in real-time.

Software Requirements

Backend

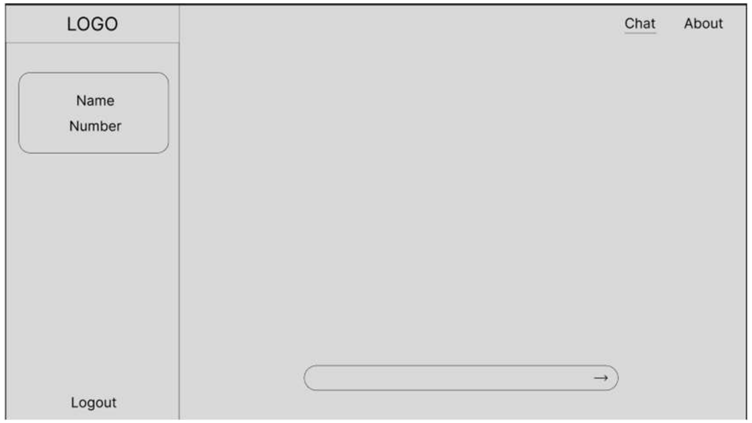
- **Flask (Python) for Establishing Relation with Knowledge Base :** Flask is a lightweight Python framework used to connect the chatbot with its knowledge base, handling user inputs, processing queries, and retrieving relevant information to generate accurate responses.
- **Database (PostgreSQL/MongoDB for Storing User Queries and Recommendations)**
  - **a. PostgreSQL** is a robust relational database used for storing structured data such as user queries and recommendations, ensuring data integrity and complex querying.
  - **b. MongoDB** is a flexible NoSQL database that stores unstructured data in JSON-like documents, ideal for dynamic and scalable storage of diverse user interactions and recommendations.

Software Requirements

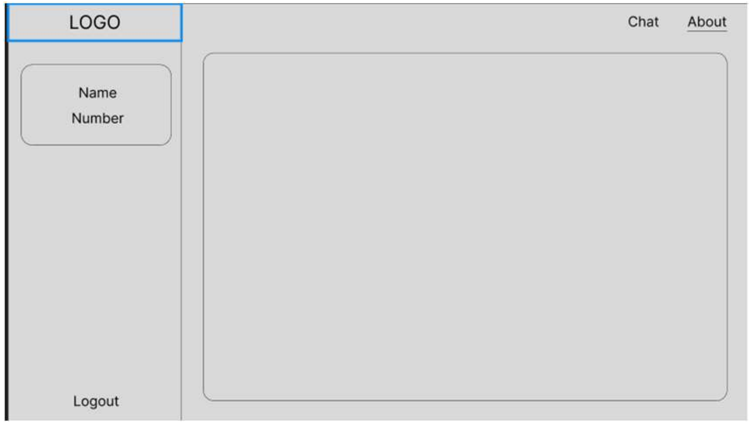
LLM model and Tools

- **Hugging Face Transformers Pre-trained Model :** Provides advanced NLP models like BERT and GPT to enhance the chatbot's text understanding and response accuracy.
- **Python Libraries**
  - **Keras:** Simplifies deep learning model creation and training.
  - **Numpy:** Handles numerical operations on arrays.
  - **Pandas:** Manages and analyzes data with DataFrames.
  - **Seaborn:** Creates attractive statistical visualizations.
- **Jupyter Notebook :** An interactive tool for writing code, visualizing data, and documenting the development process.

Proposed Model (Wireframe)

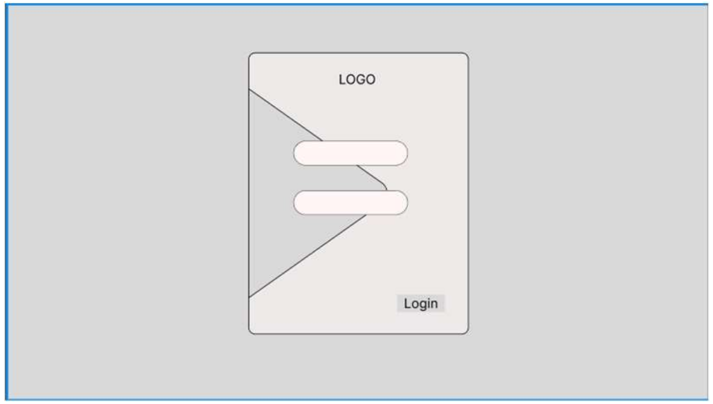


Chat Page



About Page

Login Page



Scope of Project

- **Symptom-based Medicine Recommendations:** Provides medicine suggestions based on user-reported symptoms with dosage and side effect information.
- **Drug Database Integration:** Connects to reliable databases for accurate, up-to-date medication recommendations.
- **User Authentication:** Enables user accounts for personalized advice and tracking of medical history.
- **Conversational Interface:** Uses NLP for natural, conversational symptom input and guidance.
- **Emergency Alerts:** Detects serious symptoms and advises seeking professional care when necessary.
- **Scalability and Future Expansion:** Designed for future integrations like diagnostics, telemedicine, and wearable devices.
- **Multiplatform Access:** Accessible via a web interface, with potential expansion to mobile platforms.
- **Compliance:** Adheres to healthcare data privacy standards like HIPAA or GDPR.