



Solution Architecture – HealthAI

This document outlines the solution architecture for HealthAI, a system designed to provide users with quick, accessible, and personalized health support through AI-powered disease prediction and guidance. It details the various components of the system, from the user interface to the AI model and data storage, and how they interact to deliver a seamless, intelligent, and secure healthcare experience.



Architecture Components & Data Flow

Here's how the HealthAI system works, layer by layer:

1. Frontend (Streamlit Web App)

- **Function:** Acts as the user interface.
- **Technology:** Built using Python's Streamlit library for rapid deployment.
- **Provides:**
 - Login/Register UI
 - Symptom form
 - Chatbot interface
 - Output display (predictions + treatment)

The frontend is the user's primary point of interaction with HealthAI. Streamlit's rapid development capabilities allow for quick iteration and deployment of the user interface. The login/registration UI ensures secure access to personalized health information. The symptom form provides a structured way for users to input their health concerns, while the chatbot interface offers a more natural language-based approach. Finally, the output display presents the AI's predictions and treatment suggestions in a clear and understandable format.

2. Authentication System (MySQL Database)

- **Function:** Stores registered user credentials securely.
- **Technology:** MySQL Database.
- **Handles:** Login and registration requests.
- **Links:** User profiles with their health inputs and prediction history.

The authentication system is crucial for maintaining user privacy and security. By storing user credentials in a MySQL database, HealthAI ensures that only authorized users can access their health information. The system handles login and registration requests, verifying user identities and creating new accounts. Furthermore, it links user profiles with their health inputs and prediction history, enabling personalized recommendations and tracking of health trends.

3. User Input Layer

- **Function:** Collects user input.
- **Methods:**
 - Symptom entry form (dropdowns/textboxes)
 - Chatbot (for natural language queries like "I have a headache and nausea")

The user input layer is responsible for gathering information about the user's health concerns. The symptom entry form provides a structured way for users to input their symptoms, using dropdowns and textboxes to ensure consistency and accuracy. The chatbot, on the other hand, allows users to describe their symptoms in natural language, making the interaction more intuitive and user-friendly.

4. Backend AI Engine (Python + Transformers)

- **Function:** Core processing unit of the system.
- **Technology:** Python + Transformers.
- **Key Modules:**
 - **Symptom Parser (NLP):** Uses basic NLP or a fine-tuned transformer to extract structured data from chat input.
 - **Disease Prediction Model:** Uses transformer-based models or rule-based logic to identify the most probable disease.
 - **Personalized Treatment Module:** Suggests treatment based on user's age, gender, medical history, and predicted disease.

The backend AI engine is the heart of the HealthAI system. It processes user input, predicts potential diseases, and suggests personalized treatments. The symptom parser uses NLP techniques, potentially leveraging fine-tuned transformer models, to extract structured data from the chatbot's natural language input. The disease prediction model employs transformer-based models or rule-based logic to identify the most likely disease based on the user's symptoms. Finally, the personalized treatment module considers the user's age, gender, medical history, and predicted disease to suggest appropriate treatment options.

5. Output Renderer (Streamlit UI)

- **Function:** Displays the results of the AI engine.
- **Technology:** Streamlit UI.
- **Displays:**
 - Predicted diseases
 - Confidence scores (optional)
 - Recommended steps or basic treatment advice
- **Goal:** Offers a clean, readable output to users.

The output renderer presents the AI engine's predictions and treatment suggestions to the user in a clear and understandable format. It displays the predicted diseases, along with optional confidence scores to indicate the certainty of the prediction. It also provides recommended steps or basic treatment advice, empowering users to take informed action regarding their health.

6. Feedback Collector [Future Scope]

- **Function:** Allows users to rate the usefulness of predictions.
- **Storage:** Stores feedback in the database for model improvement.

The feedback collector is an optional component that allows users to provide feedback on the accuracy and usefulness of the AI's predictions. This feedback is stored in the database and can be used to improve the model's performance over time through retraining and refinement.



Data Flow

