
Project Documentation – Healthcare AI Assistant

1. Introduction

Project Title: Healthcare AI Assistant

Team Members:

• Team Leader : AINUL JARIYA K A

• Team Member 1 : RABIYATHUL ALMAS M A

• Team Member 2 : RAHMATH NISHA N

• Team Member 3 : SHANFIYA FATHIMA T

---

2. Project Overview

Purpose:

The purpose of the Healthcare AI Assistant is to empower healthcare professionals and patients by leveraging artificial intelligence and real-time data. It optimizes medical workflows, supports decision-making, and provides personalized health advice. The assistant helps improve patient outcomes, reduce medical errors, and streamline administrative processes while enabling patients to better manage their own health.

Features:

Conversational Interface

Key Point: Natural language interaction

Functionality: Allows doctors and patients to ask questions, get medical updates, and receive personalized advice in plain language.

Medical Document Summarization

Key Point: Simplified medical records

Functionality: Converts lengthy medical records and research papers into concise, actionable summaries.

Predictive Health Analytics

Key Point: Early disease detection

Functionality: Estimates the risk of diseases based on patient history, genetic factors, and lifestyle using predictive models.

Personalized Health Tips

Key Point: Behavior guidance

Functionality: Recommends daily actions and lifestyle changes to improve health outcomes based on individual patient profiles.

Patient Feedback Loop

Key Point: Continuous care improvement

Functionality: Collects patient-reported outcomes to inform treatment adjustments and service improvements.

Clinical KPI Forecasting

Key Point: Hospital performance monitoring

Functionality: Projects key performance indicators such as patient admission rates and resource usage to help administrators plan ahead.

Anomaly Detection in Medical Data

Key Point: Early warning system

Functionality: Identifies abnormal patterns in patient vitals or lab results to flag potential health risks.

Multimodal Input Support

Key Point: Flexible data handling

Functionality: Accepts text, PDFs, medical images (DICOM), and structured data (CSV) for document analysis and diagnostics.

User-friendly Interface

Key Point: Intuitive dashboard

Functionality: Offers healthcare professionals and patients an easy-to-use web interface to interact with the assistant.

---

3. Architecture

Frontend (Streamlit):

Interactive web UI with dashboards, file uploads, chat interface, feedback forms, and medical report viewers.

Backend (FastAPI):

Manages document processing, chat interactions, health tip generation, medical report creation, and embedding patient records.

LLM Integration (IBM Watsonx Granite):

Utilized for summarization of medical documents, generating health recommendations, and assisting in diagnostic reasoning.

Vector Search (Pinecone):

Embeds medical research papers and patient records to allow semantic search using natural language queries.

ML Modules (Predictive Analytics & Anomaly Detection):

Lightweight ML models (using Scikit-learn) for predicting patient risk factors and detecting anomalies in health data.

---

4. Setup Instructions

Python 3.9+

Pip and virtual environment

API keys for IBM Watsonx and Pinecone

Internet access

Installation Steps:

Clone the repository

Install dependencies

Configure credentials in a .env file

Start FastAPI backend server

Launch Streamlit frontend

Upload medical data and interact with features

---

5. Running the Application

Start FastAPI server

Launch Streamlit dashboard

Upload medical documents or CSV patient data

Interact with the AI assistant for summaries, tips, forecasts, and anomaly detection

---

6. API Documentation

POST /chat/ask: Ask health-related questions

POST /upload-doc: Upload patient records or research papers

GET /search-docs: Search similar medical cases or research

GET /get-health-tips: Get personalized health tips

POST /submit-feedback: Submit patient feedback

---

7. Authentication (Future Work)

JWT-based authentication

OAuth2 integration

Role-based access (Doctor, Patient, Admin)

---

8. User Interface

Minimalist design focusing on clarity and accessibility:

Sidebar navigation

Patient health KPIs displayed in cards

Tabs for chat, health tips, forecasts

PDF medical report download capability

---

9. Testing

Unit testing for health recommendation logic

API testing with Postman and Swagger

Manual testing for file uploads and chat responses

Edge case handling (e.g., malformed medical file)