

1. Introduction

ProjectTitle:IntelligentHealthcare AI Assistant

Team leader :YASAR ARAFAT H

Team member:MOHAMED JAVITH J

Team member:RAJESH S

Team member:VISHVA J

Team member:HARISH P

2. Project Overview

Purpose: The purpose of the Intelligent Healthcare AI Assistant is to enhance patient care, support medical professionals, and optimize healthcare operations. By leveraging AI and real-time health data, the assistant provides decision support, predictive insights, and personalized health guidance. It improves diagnostic efficiency, simplifies complex medical documentation, and empowers patients with accessible health advice. For hospitals and healthcare providers, it serves as a strategic partner—helping forecast patient demand, identify risks, and streamline workflows. Ultimately, this assistant bridges technology, medicine, and patient engagement to foster healthcare systems that are more proactive, inclusive, and effective.

Features:

- Conversational Interface – Enables patients and doctors to ask medical questions, access records, and receive guidance.
- Medical Report Summarization – Converts lengthy medical documents, diagnostic reports, or research into concise summaries.
- Disease & Risk Forecasting – Identifies early health risks using historical and real-time patient data.
- Personalized Health Recommendations – Provides lifestyle, diet, and treatment suggestions tailored to individuals.
- Patient Feedback Loop – Collects and analyzes patient feedback to improve care quality and hospital services.
- Clinical KPI Forecasting – Projects patient inflow, resource needs, and treatment outcomes for administrators.
- Anomaly Detection in Health Data – Flags unusual patient vitals or lab results to detect emergencies.
- Multimodal Input Support – Accepts EHRs, lab reports, PDFs, and CSVs for analysis.
- Streamlit or Gradio UI – Provides dashboards for doctors, patients, and administrators.

3. Architecture

Frontend (Streamlit): Interactive UI for dashboards, patient records, reports, chat, feedback, and clinical visualizations. Backend (FastAPI): Manages APIs for medical report summarization, chat, forecasting, anomaly detection, and recommendations. LLM Integration (IBM Watsonx Granite): Produces clinical summaries, patient-friendly explanations, and diagnostic insights. Vector Search (Pinecone): Stores and retrieves medical documents and guidelines using semantic search. ML Modules: Forecasts patient inflow, disease risks, and detects anomalies in health metrics.

4. Setup Instructions

Prerequisites: - Python 3.9 or later - pip and virtual environment tools - API keys for IBM Watsonx and Pinecone - Access to healthcare datasets
Installation Process: - Clone the repository - Install dependencies - Configure credentials in .env file - Run backend server with FastAPI - Launch frontend via Streamlit - Upload health data and interact with modules.

5. Folder Structure

- app/ – Backend logic
- app/api/ – API routes for chat, reports, feedback, and data processing
- ui/ – Streamlit frontend components
- health_dashboard.py – Main dashboard entry script
- granite_llm.py – Handles Watsonx Granite interactions
- document_embedder.py – Embeds medical documents
- kpi_health_forecaster.py – Forecasts patient flow, disease trends
- anomaly_health_checker.py – Detects irregular health values
- report_generator.py – Generates AI-based medical summaries and recommendations

6. Running the Application

1. Start the FastAPI backend 2. Run Streamlit frontend 3. Navigate sidebar (Chat, Health Reports, Forecasting, Feedback) 4. Upload medical documents 5. Receive outputs like insights, summaries, forecasts in real-time.

7. API Documentation

- POST /chat/ask – User queries → AI response
- POST /upload-doc – Upload and embed documents
- GET /search-docs – Retrieve related medical references
- GET /get-health-tips – Personalized preventive care suggestions
- POST /submit-feedback – Stores patient feedback

8. Authentication

Supports token-based authentication (JWT/API keys), OAuth2 with provider credentials, and role-based access (doctor, patient, admin, researcher). Future updates include session tracking & health record history.

9. UserInterface

Includes sidebar navigation, KPI dashboards, chat tabs, medical summaries, forecasting visualizations, real-time input forms, and downloadable health reports.

10. Testing

Testing included unit testing for summarization and forecasting, API testing (Swagger/Postman), manual testing for uploads and chat, and edge case handling for malformed records or missing data.

11. Screenshots

To be added.

12. Known Issues

- Limited datasets in demo mode - Accuracy depends on quality of EHRs/reports - Requires medical validation by licensed professionals

13. Future Enhancements

- Integration with wearable devices and IoT health trackers
- Multilingual patient support

- Advanced disease progression models
- Telemedicine integration for remote consultations