**Project Documentation**

1. Introduction:

Project Title: Health AI: Intelligent Healthcare Assistant

Team Leader:Deepan M

Team Member:Kalaiyarasan S

Team Member:Balaji A

Team Member:Gogul S

2. Project Overview:

Purpose:

The purpose of the Health AI Assistant is to provide intelligent, AI-driven support for patients, doctors, and healthcare staff. It assists with symptom checking, medical record management, appointment scheduling, and personalized health recommendations. The system improves decision-making for healthcare providers and offers patients accessible, reliable guidance for better health outcomes.

Features:

Conversational Interface:

Key Point: Natural language health assistant

Functionality: Patients can ask health-related questions and receive AI-powered answers in plain language.

Symptom Checker:

Key Point: Early guidance

Functionality: Provides preliminary health advice based on user symptoms and medical data.

Medical Record Summarization:

Key Point: Simplified patient history

Functionality: Converts lengthy health records into concise, actionable summaries for doctors.

Appointment Management:

Key Point: Streamlined scheduling

Functionality: Allows patients to book, reschedule, or cancel appointments through the assistant.

Medication Reminder:

Key Point: Improved adherence

Functionality: Sends reminders for prescriptions, dosages, and follow-ups.

Anomaly Detection:

Key Point: Early health alerts

Functionality: Detects unusual patterns in patient vitals or reports and alerts caregivers.

Multimodal Input Support:

Key Point: Flexible health data handling

Functionality: Accepts text, PDFs, medical reports, and wearable data for analysis.

User-Friendly Dashboard (Streamlit/Gradio)

Key Point: Easy interaction

Functionality: Provides a simple, interactive dashboard for patients and healthcare providers.

3. Architecture:

Frontend (Streamlit/Gradio): Interactive UI with patient portal, chat interface, medical uploads, and dashboards.

Backend (FastAPI): REST API for handling queries, records, and scheduling.

LLM Integration (IBM Watsonx / OpenAI): For summarization, Q&A, and health guidance.

Database (MongoDB / PostgreSQL): Stores patient data, reports, and appointments.

ML Modules: For anomaly detection, symptom prediction, and trend forecasting.

Vector Search (Pinecone/FAISS): For quick retrieval of medical documents and reports.

4. Setup Instructions:

Python 3.9+

pip and virtual environment

API keys for LLM & database

Clone repo → install dependencies → configure .env → run FastAPI backend → launch Streamlit dashboard

5. Folder Structure:

app/ → FastAPI backend (routers, models, API logic)

app/api/ → API routes (chat, records, appointments, feedback)

ui/ → Streamlit frontend (dashboards, forms, reports)

medical\_embedder.py → Handles medical report embedding/search

symptom\_checker.py → Predicts possible health conditions

appointment\_manager.py → Manages scheduling system

report\_generator.py → Generates patient summaries

6. Running the Application:

1. Start FastAPI backend

2. Run Streamlit/Gradio frontend

3. Upload patient records, interact via chat, check symptoms, and generate reports

7. API Documentation:

POST /chat/ask – Ask health-related questions

POST /upload-record – Upload patient medical reports

GET /search-records – Retrieve patient history summaries

POST /book-appointment – Book or reschedule appointments

POST /submit-feedback – Collect patient feedback

8. Authentication:

JWT or API key authentication

OAuth2 for hospital system integration

Role-based access (Doctor, Patient, Admin)

9. User Interface:

Sidebar navigation

Patient dashboard with health summaries

Doctor dashboard with reports and alerts

Symptom checker and chat interface

PDF report download option

10. Testing:

Unit testing for symptom checker & summarization

API testing via Swagger/Postman

Manual testing for chat & scheduling

Edge cases: invalid symptoms, large medical files

11. Screenshots:

(Add UI/flow screenshots here)

12. Known Issues:

Requires stable internet for cloud LLM

Limited to supported medical terminologies

13. Future Enhancements:

Integration with wearable devices (Fitbit, Apple Watch)

Multi-language support for patients

AI-powered diagnosis support for doctors

Offline mode for rural/low-network areas