Full Stack Development with MERN

Project Documentation format

1. Introduction

- **Project Title:** HealthAI – Intelligent HealthCare Assistant using IBM Granite.
- **Team Members:**

Name	Role
1.Adabala PujithaSri Naga GangaBhavani	Team Leader, Full Stack & LLM Integration using IBM Granite
2.Animireddy Sai Gowtham	Frontend Developer (Streamlit)
3.Datla Vijaya Durga Devi	Backend Developer (Python)
4.Navya Bindhu	Model Testing, Dashboard Data Aggregation.

2. Project Overview

- **Purpose:**

The purpose of HealthAI is to provide intelligent, AI-powered healthcare assistance. It offers medical information, disease predictions, personalized treatment plans, and health analytics, making healthcare insights easily accessible to users.

- **Features:**
  - AI-powered patient chat (using IBM Granite LLM)
  - Disease prediction based on symptoms
  - Personalized treatment plan generation
  - Health analytics dashboard with data visualizations
  - User profile management
  - Streamlit-based responsive web UI.

3. Architecture

- **Frontend:** Developed using Streamlit and Plotly for interactive health metric visualizations.
- **Backend:** Built with Python, handling user inputs, API calls to IBM Watson ML (Granite LLM), and AI response processing.
- **Database:** Currently, patient data and interactions are handled in-session; future versions may integrate a dedicated database for user profiles and health history.

4. Setup Instructions

- **Prerequisites:**

  - Python 3.x

  - Streamlit

  - IBM Watson Machine Learning API Key

  - dotenv for environment variable management

- **Installation:**

  1. Clone the repository.
  2. Create and activate a virtual environment.
  3. Install dependencies using: pip install -r requirements.txt
  4. Create a .env file with your IBM API credentials

5. Folder Structure

Main Project Folder:

- app.py (Main Streamlit app)
- modules/ (Feature-specific functions)
- static/ (Images, logos)
-.env (API keys)
- requirements.txt.

6. Running the Application

Run the following command to start the application:

streamlit run app.py

7. API Documentation

Endpoint	Method	Description
/patient_chat	POST	Sends patient query to IBM Granite for chat response
/predict_disease	POST	Processes symptoms and returns disease prediction
/treatment_plan	POST	Generates personalized treatment plan using AI
/health_analytics	GET	Fetches and displays patient health metrics dashboard

.

8. Authentication

API authentication is handled via IBM Watson API keys, stored securely using dotenv. No user login functionality is implemented in the current version.

9. User Interface

The UI is built with Streamlit, featuring sidebar navigation and separate pages for each functionality like Chat, Disease Prediction, Treatment Plans, and Analytics Dashboard.

10. Testing

Manual testing was performed for each module to ensure API response accuracy and frontend functionality.

11. Screenshots or Demo

Demo:
https://drive.google.com/file/d/1cY2BQlOaQupod8tOIjalVggDEOvegkve/view?usp=drivesdk

12. Known Issues

- Currently no persistent user database.
- Responses depend on IBM Granite model limits.
- No user login or session management in this version.

13. Future Enhancements

- Add user login and session tracking.
- Store patient history and chat logs in a database.
- Integrate voice-based inputs for patient chat.
- Expand with more advanced AI models for deeper diagnosis.