**Health AI: Intelligent Healthcare Assistant Using IBM Granite.**

**Project Documentation.**

**Introduction:**

**Project Title:** **Health AI: Intelligent Healthcare Assistant Using IBM Granite.**

* **Team ID : NM2025TMID12522**
* **Team Leader:** LEANTO PRAKASH.J
* **Team member:** RAGUL.S
* **Team member:** AKASH
* **Team member:** KAVIRAJAN
* **Team member:** ARUN KUMAR.U

1. **Project Overview:**

Health AI is an intelligent healthcare assistant that leverages cutting – edge AI to offer services like disease prediction, personalized treatment plans, patient chat, and health data analytics. The project aims to make health advice accessible and insightful through a user – friendly interface. Its built using Python, with a focus on enhancing personalized health guidance.

1. **Architecture:**

* **Frontend (Stream lit)**
* Provides an interactive dashboard for patients and doctors.
* Displays real-time health trends, predictions, and chatbot interface.
* **Backend (Fast API**
* Acts as the middle layer handling API requests.
* Manages user data, authentication, and ML model execution.

1. **Setup Instructions:**

* Python 3.9+
* Pip or conda
* IBM Watsonx API key
* Pinecone API key
* Git

1. **Folder Structure:**

* app/ -- Contains all Fast API backend logic including routers, models, and integrations modules.
* app/api/ -- Subdirectory for modular API routes like chat, feedback, report, and document vectorization.
* ui/ -- Contains frontend components for Stream lit pages, card layouts, and from UIs.
* Smart\_dashboard.py – Entry scripts for launching the main Stream lit dashboard.

1. **Running the Application:**

* Launch the Fast API server to expose backend endpoints.
* Run the Stream lit dashboard too access the web interface.
* Navigates through pages via the sidebar.
* Upload documents or CSVs, interact with the chat assistant, and view outputs like reports, summaries, and predictions.
* All interactions are real – time and use backend APIs to dynamically update the frontend.

1. **API Documentation:**

* POST / chat / ask –Accepts a user query and responds with an AI – generated message.
* POST / upload – doc – Uploads and embeds documents in Pinecone
* GET / search – doc – Returns semantically similar policies to the input query
* GET / get – eco – tips – Provides Healthcare tips for selected topics like energy, water, or waste
* POST / submit – feedback – Stores patient feedback later review or analytics

Each endpoint is tested and documented in Swagger UI for quick inspection and trial during development.

1. **Authentication:**

* JWT – based authentication system
* User registration and login functionality
* Tokens required for accessing secured endpoints
* Role – based access (Patient or doctor)

1. **User – interface:**

* Stream lit – based dashboard
* Patient data input forms
* AI Chatbot for symptom queries
* Visualizations of forecasts and anomalies
* Simple and interactive design

1. **Testing:**

* **Unit Testing:** Core functionalities and models
* **Integration Testing:** API and frontend – backend communication
* **ML Validation:** Accuracy and performance of models
* **Tools Used**: Pytest, Postman

1. **Screenshot:**

Provide screenshot and Demo Link showcasing the application features and design. Here, The Project Documentation, demo link and screenshot have been added in the GitHub repository for better understanding of the systems.

1. **Known Issues:**

* Dependency on IBM Watsonx Granite availability
* Limited dataset size may affect predictions
* Performance variation with large medical records
* Vector database costs for large -scale usage
* LLM response may sometimes generate generic advice
* Requires stable internet for LLM & Pinecone integration
* Forecasting limited to structured datasets.

1. **Future Enhancements:**

* Add support for wearable IOT devices integration.
* Multi – language support for patients.
* Advanced anomaly detection with deep learning models.
* Integration with hospital database (FHIR/HL7).
* Mobile applications versions for easier patient access.
* Provide doctor and hospital recommendation system based on patient symptoms.
* Expand support for Electronic Health Records (EHR).

1. **Objectives & Scope:**

* **Objectives:**
* To Provide AI – driven healthcare supports.
* To make healthcare accessible and affordable.
* To assist doctors with quick insights.
* **Scope:**
* Covers patients, doctors, and healthcare institutions.
* Supports both preventive and diagnostic healthcare.

1. **Problem Statement:**

* Traditional healthcare systems face:
* Overloaded doctors and long waiting times.
* Limited access to quality medical guidance in rural areas.
* Difficulty in maintaining accurate medical records.
* Health AI solves these by offering intelligent, real – time, AI – based healthcare

1. **Use Case Scenarios:**
   1. **Patient Use Case –** A patient enters symptoms 🡪 Health suggests possible causes + next steps.
   2. **Doctor Use Case –** A doctor uploads medical reports 🡪 Health AI generates quick summaries.
   3. **Emergency Case –** Wearable device detects abnormal heart rate 🡪 Health AI sends alerts.
2. **Technology Stack:**

* **Frontend:** Stream lit
* **Backend:** Fast API
* **Database:** MongoDB / PostgreSQL (choose one)
* **AI / LLM:** IBM Watsonx Granite
* **Vector Search:** Pinecone

1. **Performance Metrics:**

* Response time of APIs (should be < 1 sec for queries).
* Accuracy of symptom checker (% correctness).
* Forecasting error rate (using RMSE / MAE).
* Scalability (number of concurrent users supported).

1. **Limitations:**

* Not a replacement for certified medical diagnosis.
* Accuracy depends on quality for training data.
* Requires internet access for AI / LLM queries.
* Free API limits (IBM Granite, Pinecone).

1. **References:**

* IBM Watsonx Granite official documentation,
* Pinecone Vector DB Documentation.
* Stream lit & Fast API developer docs.
* Research Papers on AI in healthcare.

1. **Conclusion:**

In conclusion, this project highlights how AI – Powered healthcare assistants can bridge the gap between patients and healthcare providers, offering a step forward towards smarter, more accessible, and efficient healthcare systems.