Health AI – Project Documentation

1. Introduction

Project Title: Health AI – Intelligent Healthcare Assistant

Team Member 1: Dhanalakshmi S (Leader) 23RCN30

Team Member 2: Gowri K 23RCN32

Team Member 3: Gowsalya L 23RCN33

Team Member 4: Jayasri S 23RCN34

2. Project Overview

Purpose:
Health AI is a generative AI-powered healthcare assistant that provides smart, easy-to-understand guidance for patients. It supports patient chat, disease prediction, treatment suggestions, and other medical assistance features.

The project is implemented in Google Colab for cloud-based execution, with IBM Granite models from Hugging Face for natural language processing. The source code and documentation are hosted on GitHub for version control.

Features:
- Patient Chat – Interactive conversational AI for answering health-related questions.
- Disease Prediction – Predicts possible conditions based on symptoms.
- Treatment Suggestions – Provides general treatment guidance (non-clinical).
- Colab Integration – Fully deployed in Google Colab with GPU support.
- Gradio UI – Simple and user-friendly interface for patients.
- GitHub Hosting – All source code maintained on GitHub.

3. Architecture

Google Colab Environment:
- Used for running the application with GPU (T4).
- Installs libraries, loads IBM Granite model, and launches the Gradio app.

Frontend (Gradio):
- Provides tabs for patient chat, disease prediction, and treatment suggestions.
- Runs inside Colab and generates a shareable public link.

Backend (Python in Colab):
- All logic is coded in Colab notebooks.
- Handles disease prediction, treatment generation, and conversational responses.

LLM Integration (IBM Granite):
- Model used: granite-3.2-2b-instruct (fast and lightweight).

Data Storage:
- GitHub used for source code and project files.
- Outputs can be downloaded from Colab.

4. Setup Instructions

Prerequisites:
- Google Account with Colab access
- Hugging Face account for IBM Granite models
- GitHub repository for project hosting
- Internet connection

Installation in Colab:

```
!pip install transformers torch gradio -q
```

5. Folder Structure (GitHub Repository)

```
health-ai/
│── health_ai.ipynb       # Main Colab notebook
│── requirements.txt      # Dependencies
│── model_integration.py  # Granite model loading
│── gradio_ui.py          # Gradio app interface
│── utils/                # Helper functions (chat, prediction)
│── README.md             # Documentation
```

6. Running the Application

1. Open Google Colab.
2. Clone GitHub repo:
```
  !git clone https://github.com/your-username/health-ai.git
  %cd health-ai
```
3. Install dependencies.
4. Run all cells in health_ai.ipynb.
5. A Gradio link will be generated.
6. Open the link and interact with the app (chat, prediction, treatment).

7. API Documentation

Internal functions (executed inside Colab):
- patient_chat(input_text) → Returns conversational response.
- predict_disease(symptoms) → Suggests possible diseases.
- treatment_plan(condition) → Provides general treatment suggestions.

8. Authentication

- Hugging Face API key required to access Granite models.
- GitHub account for uploading final project files.

9. User Interface

- Tab 1 – Patient Chat (ask health-related questions).
- Tab 2 – Disease Prediction (enter symptoms).
- Tab 3 – Treatment Plans (get general advice).

Runs inside Colab with a Gradio shareable link.

10. Testing

- Unit Testing: Checked prediction and chat responses.
- Manual Testing: Verified end-to-end in Colab with Gradio UI.
- Edge Cases: Tested blank inputs, rare symptoms, invalid queries.

11. Screenshots

- Colab notebook showing installation.
- Gradio link generated in Colab.
- Patient Chat demonstration.
- Disease Prediction result.
- Treatment Plan output.

12. Known Issues

- Colab sessions auto-expire after ~12 hours.
- Requires continuous internet for Hugging Face models.
- Predictions are not a replacement for medical advice.

13. Future Enhancements

- Add integration with real medical datasets for accuracy.
- Provide PDF reports for patients.

- Expand to multi-language support.
- Build a mobile app version.
- Add secure authentication for doctors and patients.