Health AI Project Documentation

# 1. Introduction

• Project Title: Health AI with IBM

• Team Members:

* 1. Rajalakshmi. T
* 2. Vishnudevi. R
* 3. Vinothini. M
* 4. Umamaheshwari. S
* 5. Swathi. R

# 2. Project Overview

HealthAI uses the Granite model from Hugging Face to deliver smart, easy-to-understand healthcare help. It includes Patient Chat, Disease Prediction, and Treatment Plans. The project is deployed in Google Colab using Granite for fast, accessible, and secure medical guidance.

# 3. Pre-requisites

• Gradio Framework Knowledge

• IBM Granite Models (Hugging Face)

• Python Programming Proficiency

• Version Control with Git

• Google Colab’s T4 GPU Knowledge

# 4. Project Workflow

• Activity-1: Exploring Naan Mudhalvan Smart Interz Portal.

• Activity-2: Choosing an IBM Granite Model from Hugging Face.

• Activity-3: Running Application in Google Colab.

# 5. Architecture

Frontend (Gradio): Provides a user interface for patient chat, disease prediction, and treatment plans.

Backend (Google Colab + Python): Executes AI models, processes patient data, and generates responses.

Model (IBM Granite - Hugging Face): Core AI providing healthcare guidance and predictions.

# 6. Setup Instructions

• Search for 'Google Colab' and open a new notebook.

• Change runtime type to T4 GPU.

• Run the command: !pip install transformers torch gradio -q

• Run the Health AI code provided in the guided project.

• Click the URL generated to launch the Gradio Application.

# 7. Folder Structure

The project is organized in Colab and GitHub as:

• health\_ai.ipynb – Main Colab notebook

• requirements.txt – List of dependencies

• healthai.py – Application script

• README.md – Documentation file

# 8. Running the Application

➢ Launch the Colab Notebook.  
➢ Install dependencies.  
➢ Run the notebook cells.   
➢ Open the Gradio link provided.  
➢ Interact with Health AI for patient chat, predictions, and treatment advice.

# 9. API Documentation

The project runs on Gradio and does not expose standalone APIs.

Key functionalities include:

• Patient Chatbot

• Disease Prediction

• Treatment Plan Suggestions

# 10. Authentication

The demo version runs openly in Colab.

For secure deployments:

• Use API key-based authentication

• Apply OAuth2 for secure access

• Limit user roles (patients, doctors, admins)

# 11. User Interface

The interface is created with Gradio. Features include:

• Input fields for patient details (age, condition, history)

• Output displaying predictions and treatment plans

• Simple, clear layout suitable for non-technical users

# 12. Testing

Testing includes:

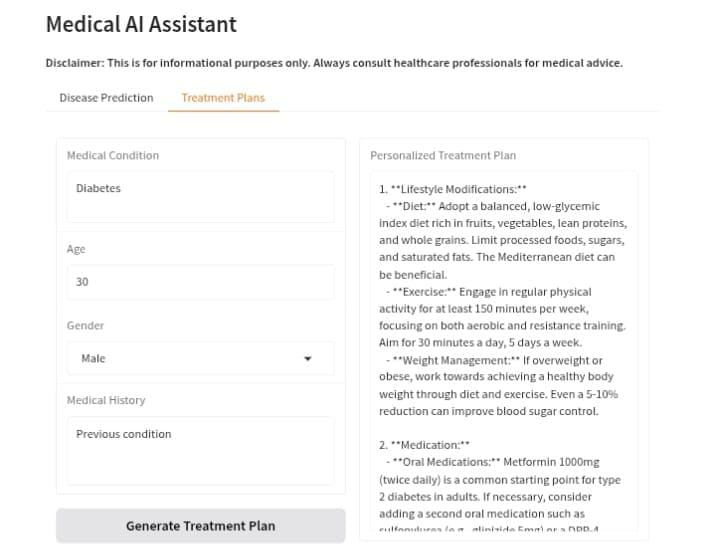
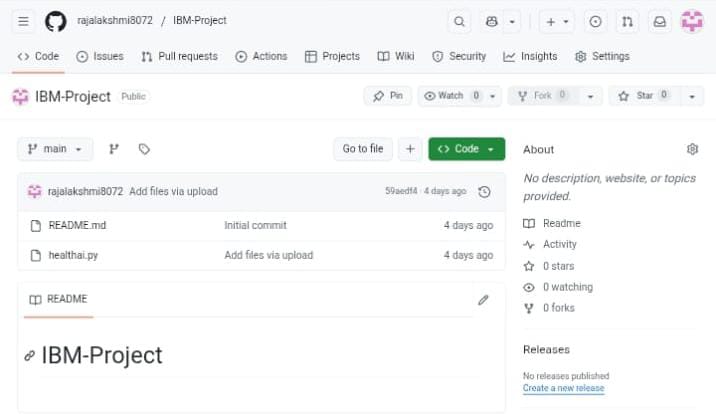
• Unit testing AI functions

• Manual testing of chatbot and predictions

• Validating outputs against known datasets

• Handling invalid or incomplete inputs

# 13. Screenshots



# 14. Known Issues

• Limited accuracy for rare diseases

• GPU resource limitations in Google Colab

• Response latency during peak usage

# 15. Future Enhancements

• Expanding disease database for predictions

• Adding multilingual support for patients

• Deploying as a standalone healthcare platform

• Integration with hospital databases for real-time updates