Health AI with IBM

**PROJECT DOCUMENTATION**

RAMYA.A ( TEAM LEADER )

RENITA AMALA RANI.J ( TEAM MEMBER )

RAMYA.B ( TEAM MEMBER ) RAMYA.S ( TEAM MEMBER )

Prepared by: Student Team

Institution: \_\_\_\_\_\_\_\_\_\_\_\_

Date: September 2025

# Table of Contents

1. Project Overview
2. Features
3. Architecture
4. Setup Instructions

5.Running the Application

1. User Interface
2. Testing
3. Authentication

# Project Overview

The Health AI with IBM project is designed to provide intelligent healthcare assistance using IBM Granite models from Hugging Face. It integrates Generative AI into healthcare workflows, enabling features such as patient chat, disease prediction, and treatment recommendations. The system is deployed on Google Colab with GPU acceleration, ensuring accessibility, scalability, and fast processing.

Objectives:

* Assist patients with conversational AI for quick medical queries.
* Provide predictive insights into diseases and health conditions.
* Generate AI-assisted treatment plans.
* Ensure secure, fast, and user-friendly deployment via Colab and GitHub.

# Features

* Patient Chat Interface – Conversational support with AI responses.
* Disease Prediction – Predicts potential illnesses from symptoms.
* Treatment Plan Suggestions – Generates tailored recommendations.
* Gradio Interface – Lightweight, easy-to-use UI for healthcare.
* Model Flexibility – Supports multiple IBM Granite models.
* Cloud Deployment – Runs on Google Colab with GPU support.
* Version Control – Integrated with GitHub for collaboration.

# Architecture

* Frontend (Gradio): Interactive UI for chat and predictions.
* Backend (Python + Transformers + Torch): Handles model execution.
* LLM Integration (IBM Granite): Pre-trained models from Hugging Face.
* Colab Environment: Provides GPU acceleration (T4 GPU).
* GitHub Repository: Stores project source code.

# Setup Instructions

Pre-requisites:

* Python 3.9 or above
* Google Colab account
* Hugging Face account with IBM Granite model access
* GitHub account for version control Steps:
* Open Google Colab and create a new notebook.
* Change runtime to T4 GPU (Runtime > Change runtime type). • Install required libraries using: !pip install transformers torch gradio -q
* Load IBM Granite model from Hugging Face.
* Build Gradio interface for chat, predictions, and treatments.
* Save and test the notebook.
* Download project files (.py) and upload to GitHub.

•

# Running the Application

* Launch the Google Colab notebook.
* Install dependencies (requirements.txt). • Run main.py or execute notebook

cells.

* Colab generates a Gradio link for the app.
* Open the link in browser to interact with the system.

# User Interface

* Chat Window: Enables conversation with AI.
* Symptom Input Box: Patients enter symptoms for predictions.
* Treatment Output Panel: Displays AI-suggested guidance.
* Minimal design using Gradio.

# Testing

* Unit Testing – Validate prediction and treatment logic.
* API Testing – Verify Hugging Face integration.
* UI Testing – Ensure Gradio interface responsiveness.
* Performance Testing – Speed check on GPU.
* Error Handling – Manage invalid inputs and failures.

# Authentication

The current version runs in open demo mode for education. For secure deployments:

* API key authentication for Hugging Face.
* Private GitHub repositories for source control.
* Optional user login for patient access.
* Role-based access control (Patient, Doctor, Researcher).