## HEALTH AI: INTELLIGENT HEALTHCARE ASSISTANT USING IBM GRANITE

## 1. INTRODUCTION

Project Title: Health AI Assistant

Team Member: SRUTHI K

Team Member: JAYASRI V

Team Member: DHARANI SREE L

Team Member: KEERTHI P

## 2. PROJECT OVERVIEW

Purpose:  
The Health AI Assistant is designed to provide AI-powered support for medical information, symptom analysis, and health guidance. It leverages natural language processing and predictive modeling to help users understand potential health conditions, generate general advice, and offer awareness resources. The system is not a replacement for professional medical consultation but serves as an accessible tool for preliminary information, education, and awareness.

Features:

* Conversational Interface  
  Key Point: Natural language interaction  
  Functionality: Provides AI-powered answers to health-related queries through an intuitive chat interface.
* Symptom Analysis  
  Key Point: Predictive health insights  
  Functionality: Accepts symptom inputs and suggests possible conditions for user awareness.
* Medical Information Retrieval  
  Key Point: Reliable knowledge support  
  Functionality: Uses AI and knowledge bases to provide accurate medical information.
* Preventive Care Tips  
  Key Point: Health awareness  
  Functionality: Delivers lifestyle, nutrition, and preventive care tips to promote healthy living.
* User-Friendly UI (Gradio)  
  Key Point: Accessible healthcare assistant  
  Functionality: Offers a simple, browser-based interface without complex setup.

## 3. ARCHITECTURE

Frontend (Gradio): Provides an interactive interface for users to input symptoms, queries, or health-related concerns. Built using Gradio Blocks/Tabs for smooth interaction.

Backend (Python + Hugging Face Transformers): Handles model inference, input tokenization, and response generation. Integrated with IBM Granite LLM for conversational health guidance.

LLM Integration (IBM Granite): Adds natural language understanding, enabling the assistant to interpret queries, generate contextual responses, and suggest possible health information.

ML Modules:  
- Symptom-to-Condition Prediction: Suggests possible health conditions.  
- Health Advice Generation: Provides preventive care recommendations.

## 4. SETUP INSTRUCTIONS

Prerequisites:  
- Python 3.9+  
- pip and virtual environment tools  
- Hugging Face Transformers  
- PyTorch (with CUDA if GPU available)  
- Gradio

Installation Steps:  
1. Clone the Repository  
2. Create and Activate Virtual Environment  
3. Install Dependencies (pip install -r requirements.txt)  
4. Configure the Granite Model  
5. Run the Application (python app.py)  
6. Access the Gradio Interface in Browser

## 5. FOLDER STRUCTURE

- app/ – Main Gradio application logic  
- modules/ – Symptom analysis, health tips, helper scripts  
- granite\_llm.py – Handles communication with IBM Granite model  
- ui/ – UI assets (future enhancements)  
- health\_ai.py – Entry point for launching the Gradio interface  
- requirements.txt – Lists dependencies  
- README.md – Setup, usage, and project details  
- docs/ – Extra documentation and diagrams

## 6. RUNNING THE APPLICATION

Steps to Run:  
- Execute the Gradio app script  
- Open the provided local URL in browser  
- Interact via tabs:  
 \* Symptom Analysis → Input symptoms and get possible conditions  
 \* Health Tips → Receive lifestyle and preventive care advice  
 \* General Queries → Ask health-related questions

Frontend (Gradio): Tabs for Symptoms, Tips, Queries; Textboxes with placeholder guidance; Buttons for query submission; Real-time responses displayed.

Backend: Input forwarded to Granite LLM; Model generates structured responses; Output returned in formatted style.

## 7. AUTHENTICATION

Current demo is open-access.  
For secure deployment:  
- Username/Password login  
- Token-based authentication (JWT / API keys)  
- OAuth2 with cloud IAM providers  
- Role-based access:  
 \* Admin → Full control  
 \* Doctor/Analyst → Access to AI-generated insights  
 \* User → Restricted interaction

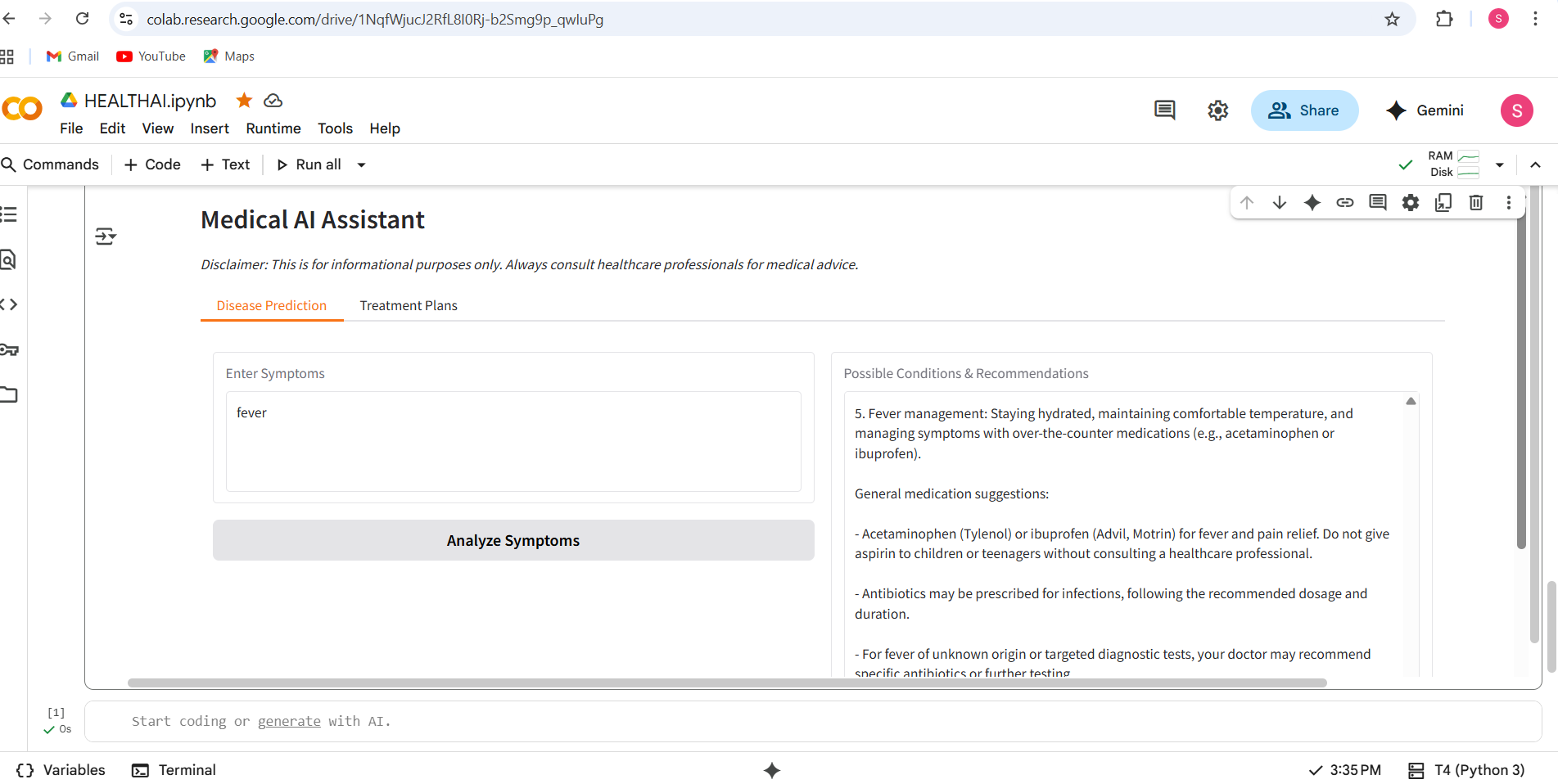
## 8. USER INTERFACE

- Tabbed Layout – Symptoms | Tips | Queries  
- Textboxes – Guided placeholders for user input  
- Buttons – For submission and reset  
- Real-time AI Responses – Displayed in output areas  
- Simple, Accessible, Mobile-friendly Design

## 9. TESTING

- Unit Testing: Verified health query functions individually  
- Manual Testing: Checked symptom inputs and AI responses  
- Interface Testing: Ensured smooth Gradio tab navigation  
- Edge Case Handling: Tested with empty, malformed, and long inputs

## 10. SCREENSHOTS



## 

## 11. KNOWN ISSUES

- Large model may cause slow response on CPU  
- Internet required to load Hugging Face models  
- Basic error handling (needs improvement for medical edge cases)  
- Not a substitute for professional medical advice

## 12. FUTURE ENHANCEMENTS

- Add medical image support (X-ray, MRI analysis)  
- Provide multilingual support  
- Integrate with hospital databases for appointment scheduling  
- Improve inference speed with optimized models  
- Add persistent chat history and user health records  
- Enable wearable device integration for real-time monitoring.