# Medical AI Assistant Project Documentation

## 1.Introduction

- Project Title: Health AI Intelligent Healthcare Assistant Using IBM Granite
- **Team Members:**

1. Team member : Nandhini D
2. Team member:  Mahalakshmi A
3. Team member: Navia paulin J

## 2. Project Overview

The Medical AI Assistant is designed to provide intelligent healthcare support using Generative AI. It can analyze user-provided symptoms, suggest possible health conditions, and provide general treatment guidelines. Additionally, it generates simple, personalized treatment plans based on age, gender, and medical history. This project leverages the IBM Granite model and integrates it with a user-friendly Gradio interface. The system emphasizes that all recommendations are for informational purposes only and encourages users to consult licensed healthcare professionals.

## 3. Architecture

• Frontend (Gradio): Provides an interactive web interface for disease prediction and treatment plan generation.
• Backend (Python + Transformers): Handles AI model loading, prompt engineering, and response generation.
• LLM Integration (IBM Granite 3.2 2B): Processes user queries and generates intelligent outputs.
• Hosting: The app can be launched locally or shared via Gradio's share feature.

## 4. Setup Instructions

- Prerequisites:
  • Python 3.8+
  • pip and virtual environment
  • IBM Watsonx API key

Installation Process

1. Clone the repository or copy project files.
2. Create a virtual environment and activate it.
3. Install required dependencies using requirements.txt.
4. Run the script using: python app.py
5. Access the Gradio app through the provided local or share URL.

## 5. Folder Structure

project-root
app/- FastAPI backend logic including chat,prediction,and analytics modules

ui/ – Streamlit frontend components for dashboards and health visualization

app.py – Entry script to run the main Streamlit interface

granite_llm.py – Handles IBM Granite model interactions

prediction_engine.py – Implements disease prediction logic

treatment_planner.py – Generates treatment recommendations

health_dashboard.py – Visualizes health data and insights

## 6. Running the Application

1. Open terminal in the project directory.
2. Run: python app.py
3. Gradio will provide a local URL and an optional public share link.
4. Use the interface to enter symptoms or request a treatment plan.
5. View AI-generated outputs in real-time.

## 7. API Documentation

Since this implementation is built with Gradio, the interface primarily runs on the frontend. However, the internal functions can be exposed as APIs if integrated with FastAPI or Flask.

Functions available in this project:
• disease_prediction(symptoms): Returns possible conditions and general advice.
• treatment_plan(condition, age, gender, history): Generates a basic treatment plan with remedies.

## 8. Authentication

Currently, this project does not implement authentication since it is a prototype. In production, JWT-based authentication or API key protection should be added for secure access.

## 9. User Interface

- Tab-based navigation in Gradio
- Symptom input for disease prediction
- Condition, age, gender, and history inputs for treatment plan generation
- Text outputs displaying generated AI responses
- Disclaimer highlighting the need to consult a medical professional

## 10. Testing

- Unit Testing: Validate AI functions and prompt responses.
- Manual Testing: Ensure UI components work as expected in Gradio.
- Edge Cases: Tested with incomplete inputs and unusual symptom descriptions.

## 12. Known Issues

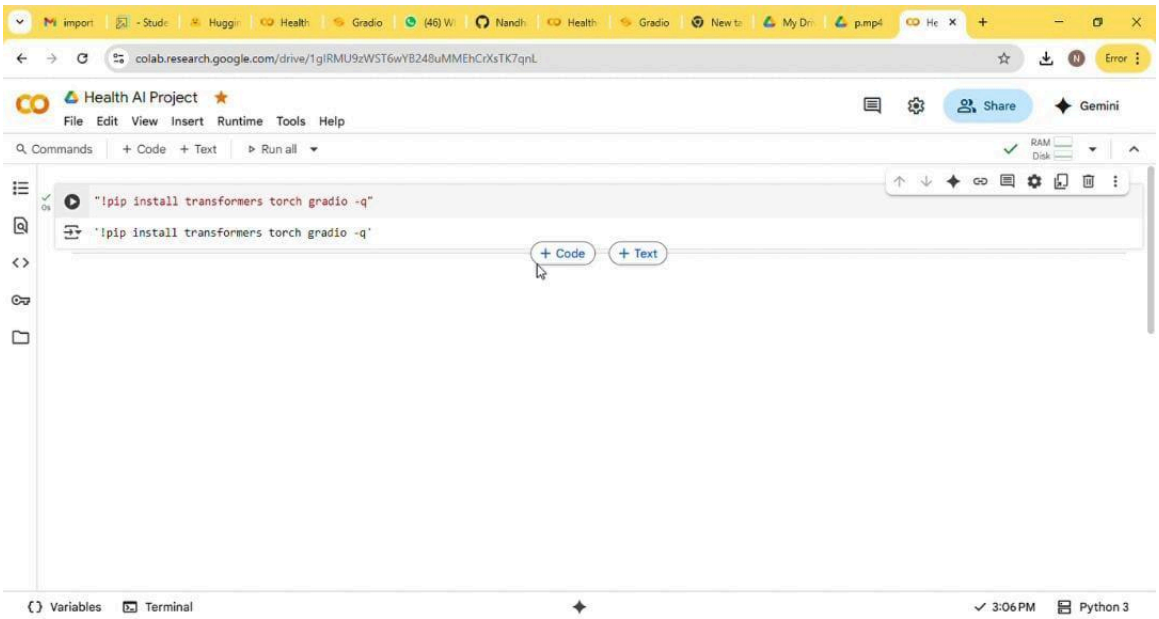
- Limited medical condition coverage due to model constraints.
- Responses may vary depending on phrasing of symptoms.
- Requires stable internet for model loading.
- Not a substitute for professional healthcare advice.

## 13. Future Enhancements

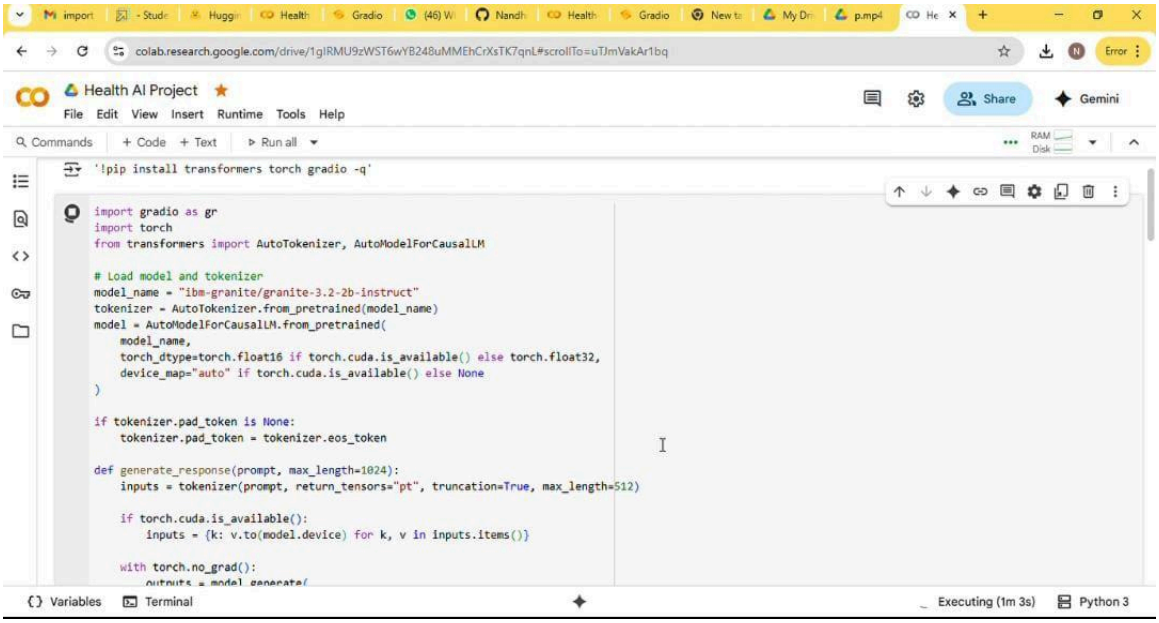
- Integration with FastAPI or Flask for API deployment.
- Expansion of disease prediction knowledge base.
- Multi-language support for broader accessibility.
- Secure authentication mechanisms.
- Improved accuracy with fine-tuned healthcare datasets.

## 14. ScreenSlot

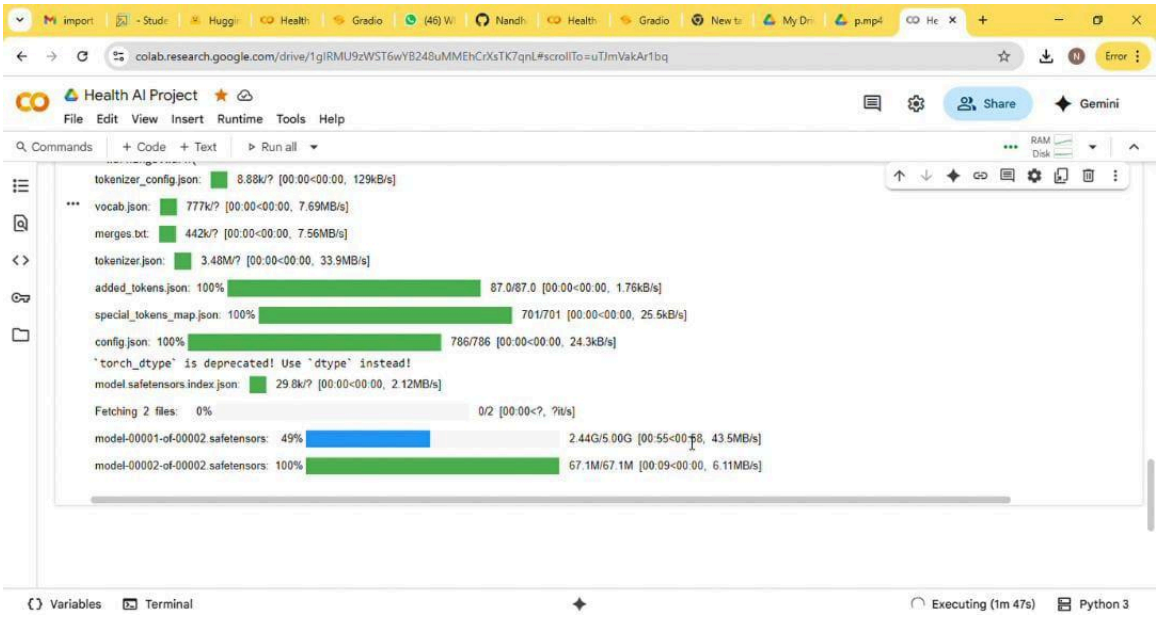
# Screen1:



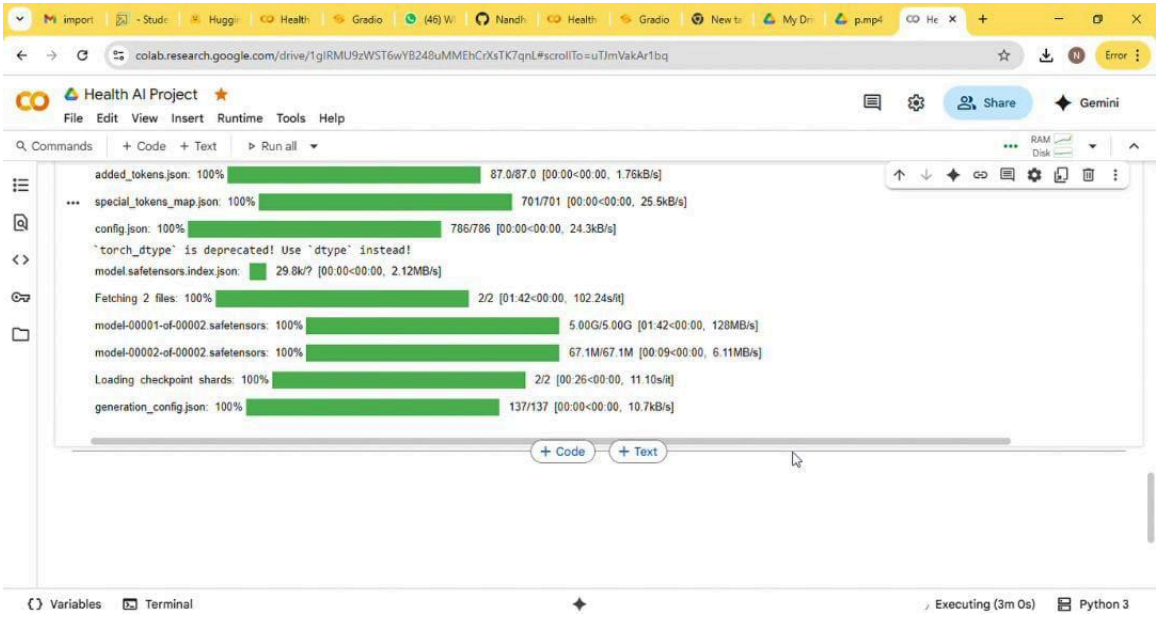

# Screen2:



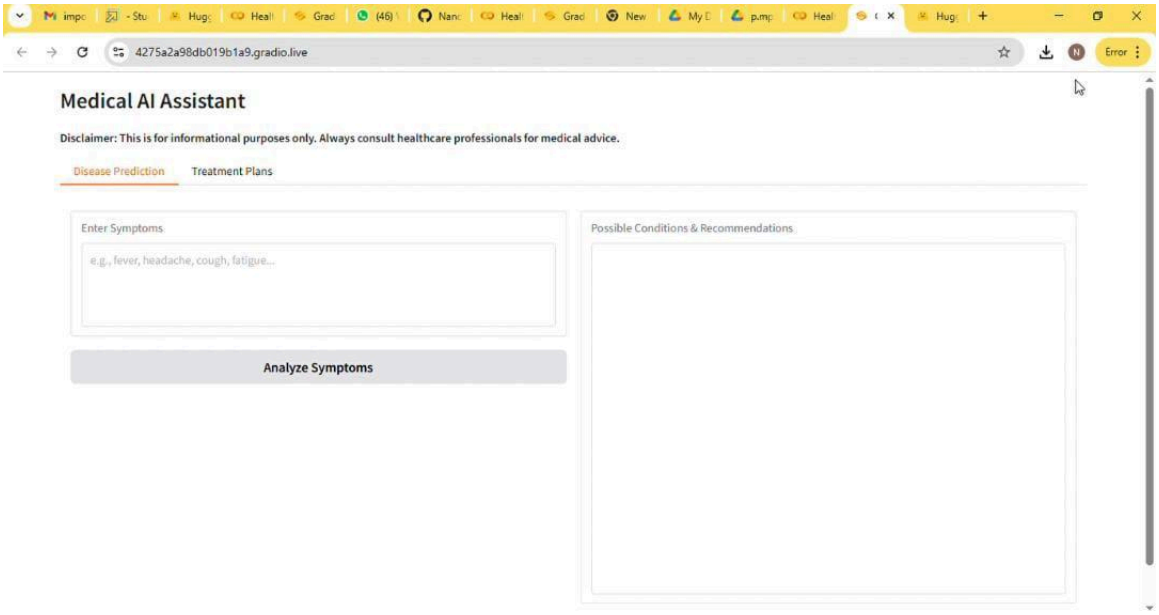
## Screen3:




## Screen4:



## Screen5:


4275a2a98db019b1a9.gradio.live
Error
Medical AI Assistant
Disclaimer: This is for informational purposes only. Always consult healthcare professionals for medical advice.
Disease Prediction
Treatment Plans
Enter Symptoms
e.g., fever, headache, cough, fatigue...
Analyze Symptoms
Possible Conditions & Recommendations