PROJECT DOCUMENTATION

**HEALTH** **AI**

TEAM ID:NM2025TMIDO2762

TEAM LEADER

V BHUVANESHWARI

Gmail:poojabhuvi1117@gmail.com

TEAM MEMBERS

1.AKSHAYA A (Gmail:akshayaanand00@gmail.com)

2.SADHANA K (Gmail:sadhanakandhan10@gmail.com)

3.NITHYA S (Gmail:nithyadoon008@gmail.com )

Project overview

Purpose:

The Medical AI Assistant provides AI-powered disease prediction and treatment suggestions based on user input. It leverages state-of-the-art language models to generate possible conditions, home remedies, and general treatment guidelines.

Features:

Symptom-based disease prediction

Personalized treatment plan generation

User-friendly interface with Gradio

Built-in medical disclaimer for safety

Extendable API endpoints

3. Architecture

Frontend: Gradio (Python-based UI)

Backend (Optional): Node.js & Express.js for API handling and integration

AI Model: Hugging Face Transformers (Granite 3.2 Instruct)

Database (Optional): MongoDB for storing user interactions and medical history

4. Setup Instructions

Prerequisites:

Python 3.9+

Node.js (optional for backend)

Git

Hugging Face Transformers

PyTorch

Gradio

Installation Steps:

# Clone the repository

git clone <repo-url>

cd MedicalAI-Assistant

# Install Python dependencies

pip install -r requirements.txt

# (Optional) Install server dependencies

cd server

npm install

5. Folder Structure

MedicalAI-Assistant/

│-- ai-model/ # Gradio + AI model

│ └── app.py

│-- client/ # (Optional React frontend)

│-- server/ # Node.js backend (if needed)

│ └── index.js

│-- docs/ # Documentation

│ └── ProjectDoc.pdf

6. Running the Application

AI Model:

cd ai-model

python app.py

Access at: http://localhost:7860

Backend (optional):

cd server

npm start

Access at: http://localhost:5000

7. API Documentation

Disease Prediction:

POST /api/predict

Body: { "symptoms": "fever, cough, fatigue" }

Treatment Plan:

POST /api/treatment

Body: { "condition": "diabetes", "age": 40, "gender": "Male", "history": "hypertension" }

8. Authentication

Currently open-access.

Can be extended with JWT-based authentication for secure API endpoints.

9. User Interface

Disease Prediction Tab: Users enter symptoms → AI suggests possible conditions.

Treatment Plan Tab: Users provide condition, age, gender, history → AI generates personalized plan.

Disclaimer Banner: Prominent warning about non-medical nature of suggestions.

10. Test

Manual testing of prompts and responses

Tools:

Postman (for API testing)

Chrome DevTools (frontend testing)

11. Screenshots or Demo

(To be added after deployment)

-12. Known Issues

AI may generate inaccurate or overly general suggestions

No real-time doctor validation

Limited to text input/output

13. Future Enhancements

Add multilingual support

Enable history storage in MongoDB

Doctor verification workflow

Mobile-friendly UI