

INTELLIGENT HEALTHCARE
ASSISTANT USING IBM GRANITE'S
Project documentation format

1. Introduction

 Project Title: Intelligent Healthcare Assistant (IBM Granite)

 Team Members:

Team ID: LTVIP2025TMID31494

Team Size: 4

Team Leader: Thunga Ashok Kumar Reddy

Team member: Akepati Narasimha Reddy

Team member: Ammireddy Hema Sundar Reddy

Team member: A Reddy Sai

2. Project Overview

 Purpose:

To provide AI-driven health support for users by analyzing symptoms and suggesting possible conditions using IBM's Granite LLM.

 Features:

Natural language chatbot

Symptom-based condition analysis

Time-stamped medical advice record


Friendly and interactive CLI interface (Streamlit optional)

3. Architecture

 Frontend: Streamlit or CLI (for prototype)

 Backend: Python with transformers + torch

 AI Model: IBM Granite (granite-3.3-2b-instruct)

 Database (optional): MongoDB or PostgreSQL for storing interactions

 APIs: Potential for RESTful endpoints (FastAPI)

4. Setup Instructions

 Prerequisites:

Python 3.8+

transformers, torch

IBM Granite model access via Hugging Face

Streamlit (optional)

 Installation:

`pip install transformers torch streamlit`

 Environment Setup:

Hugging Face token if needed

Optional: Set up .env with API keys

5. Folder Structure

`/health_assistant_project`

|

```
|— main.py          # Main app script
|— model_utils.py   # IBM Granite wrapper
|— requirements.txt
|— README.md
```

6. Running the Application

 CLI Version:

```
python main.py
```

 Streamlit Version:

```
streamlit run app.py
```

7. API Documentation (Optional if using FastAPI)

Endpoint: /analyze-symptoms

Method: POST

Request: {"symptoms": "fever, cough"}

Response: {"conditions": ["flu", "COVID-19", ...]}

8. Authentication

None required for CLI prototype

Future enhancement: Add OAuth2 or token-based access for medical privacy

9. User Interface

CLI prompts and responses (currently)

Optional upgrade to Streamlit-based UI with:

Symptom entry box

AI result cards

Health timeline panel

10. Testing

Unit Tests: mock inputs for tokenizer and model

CLI testing with unittest or pytest

11. Screenshots or Demo

CLI output sample:

Symptoms: fever, sore throat

AI Suggested Conditions: Flu, Strep throat, COVID-19

12. Known Issues

Model may return general rather than specific diagnoses

Requires internet for model loading

Ethical concerns: Not a substitute for a doctor

13. Future Enhancements

Streamlit UI

Voice input/output using Whisper + TTS

Patient history tracking via database

Multilingual support

Integration with wearable data (e.g., Fitbit, Apple Health)

Would you like me to create this as a downloadable PDF or Word doc for easy sharing with your team?