**INTELLIGENT HEALTHCARE**

**ASSISTANT USING IBM GRANITE’S**

**Project documentation format**

1. **Introduction**

🏷 Project Title: Intelligent Healthcare Assistant (IBM Granite)

Team ID: LTVIP2025TMID31490

👥 **Team Members**: B.Tejasri

A.Baba Fakruddin

B .Badrinath

---

1. **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)

1. **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)

1. **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

1. **Folder Structure**

/health\_assistant\_project

│

├── main.py # Main app script

├── model\_utils.py # IBM Granite wrapper

├── requirements.txt

└── README.md

1. **Running the Application**

🧪 CLI Version:

python main.py

🖥 Streamlit Version:

streamlit run app.py

1. **API Documentation (Optional if using FastAPI)**

**Endpoint:** /analyze-symptoms

**Method**: POST

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

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

1. **Authentication**

None required for CLI prototype

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

1. **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

1. **Known Issues**

Model may return general rather than specific diagnoses

Requires internet for model loading

Ethical concerns: Not a substitute for a doctor

1. **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)