


---

## Generative AI Project using IBM Cloud – HEALTHAI

### Project Documentation Format






---

#### 1. Introduction

- **Project Title: HEALTHAI: Intelligent Healthcare Assistant using IBM Granite (Generative AI with IBM Cloud)**  **Team Members:**
  - **Divya Siva nagaMalleswari Dontuboina (Team Leader – Development & Integration):**  
Led the complete development of the HEALTHAI application, including IBMGranite integration, Streamlit-based UI design, module creation, and model API handling,project deployment.
  - **Akula Sujana (Model Interaction & Testing):**  
Contributed by assisting in prompt design, testing the AI model outputs across modules like Disease Prediction and Health Chat, and refining interactions with IBM Granite.
  - **Sri Vidya Lakshmi Althy (UI Structuring & Feature Enhancement):**  
Supported in designing user flow, organizing the Streamlit interface across all modules, and suggesting improvements in user interaction and feature behavior.

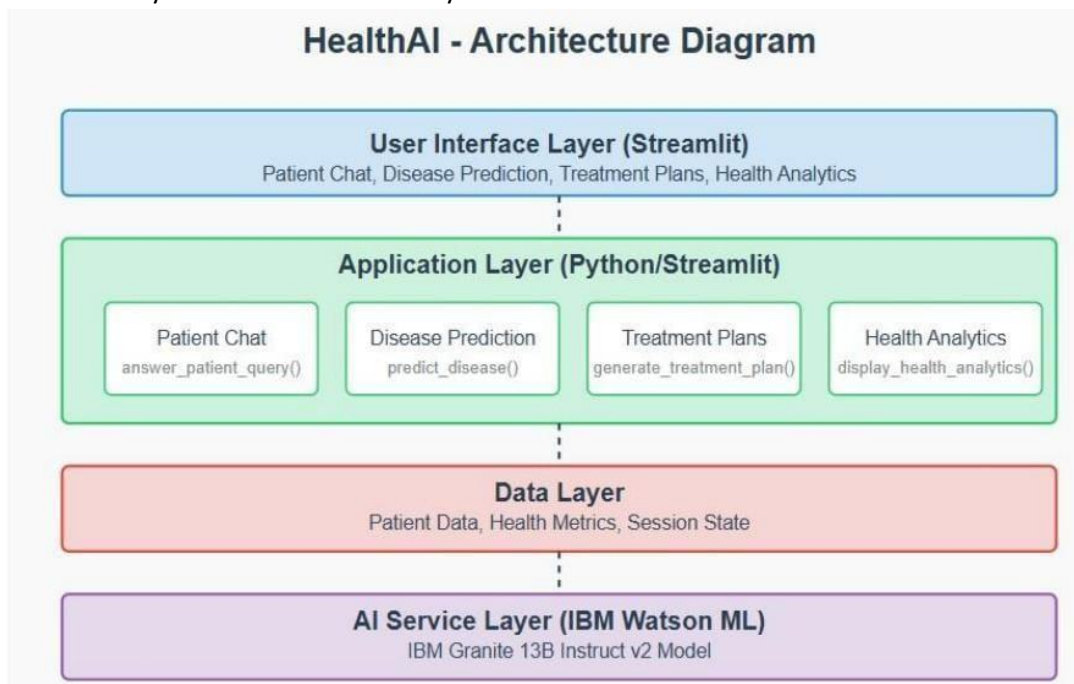
---

#### 2. Project Overview

- **Purpose:**  
To build a Generative AI-based healthcare assistant using IBM Granite, capable of answering health queries, predicting diseases, suggesting treatments, and displaying analytics.
  - **Features:**
    -  AI Health Chat using IBM Granite
    -  Disease Prediction from user symptoms
    -  Treatment Plan Suggestions
    -  Health Analytics Dashboard
    -  Centralized shared model for performance optimization
-

### 3. Architecture

- **Frontend:**  
Built using **Streamlit** for a clean and responsive web interface. Each feature is modularized for easy navigation via sidebar.
- **Backend & Model:**
  - No traditional backend. All logic handled in Streamlit using Python.
  - Uses **IBM Granite 3-2b Instruct model** from IBM Watsonx: ibm-granite/granite-3.2b-instruct
  - Supports both API and **local model loading** (granite/ folder).
- **Shared Model Loader:**  
The shared\_model.py file centrally loads and shares the AI model across modules to prevent memory crashes and redundancy.



### 4. Setup Instructions

#### Prerequisites

- Python 3.10+
- pip
- IBM cloud account and Streamlit Community Cloud account



- Installed model files if using local (granite/ folder)

## Installation

git clone: <https://github.com/divyareddy011/Health-ai.git>

cd Health-ai pip install -r requirements.txt

## Environment Variables

Create a .env file in the root folder:

IBM watsonx\_API\_Key= ESdPIiW78JY8LM32Vp6ujw\_EVP1dySvqb5ZQ5lc7K4wi

✓ .env file must be excluded in .gitignore.

---

## 5. Folder Structure

Health-ai/

```
├── app.py          # Main entry point
├── shared_model.py  # Shared AI model instance
├── patient_chat.py  # AI Health Chat module
├── disease_prediction.py # Disease Prediction logic
├── treatment_plans.py # Treatment Plan suggestions
├── health_analytics.py # Analytics module
├── requirements.txt # Python dependencies
├── .env            # API token (not pushed to GitHub)
├── granite/        # [Optional] Local model folder
└── assets/         # Logos and screenshots
```

---

## 6. Running the Application

**For IBM watsonx API:**

streamlit run app.py

**For Local Model:**



Ensure granite/ folder contains the downloaded model and tokenizer files.

In shared\_model.py, update: model\_path = "./granite"

---

## 7. API Documentation

**Endpoint:** <https://eu-de.ml.cloud.ibm.com/ml/v1/text/chat?version=2023-05-29>

### Headers:

```
{  
  "Authorization": "Bearer <IBM _API_Key>",  
  "Content-Type": "application/json"  
}
```

### Example Request:

```
{  
  "inputs": "What are the symptoms of diabetes?"  
}
```

### Example Response:

```
{  
  "generated_text": "Common symptoms of diabetes include frequent urination..."  
}
```

---

## 8. Authentication

- Streamlit cloud Secrets is securely stored the .env credentials.
- .env is excluded via .gitignore
- App is currently public and stateless (no user login)
- HuggingFace or Firebase Auth can be added in future

---

## 9. User Interface

- Built entirely with **Streamlit**
- Sidebar for navigation
- Text/chat inputs for interaction

- Visual graphs and health tips in Analytics
- Centralized theme and branding

## 10. Testing

- ☒ Manual testing across all modules
- ☒ Model tested with varied prompts and edge cases
- ☒ Handled errors for invalid inputs and model timeouts

## 11. Screenshots or Demo

```

app (2).py  app (6).py  app.py  X
C:\Users\Divya Siva Naga > OneDrive > Desktop > FST > healthai_project_updated > healthai > app.py
1 import streamlit as st
2 import pandas as pd
3 from utils.ai import get_ai_response
4 from utils.core import predict_disease, generate_treatment_plan
5 from utils.visualizer import display_health_analytics
6
7 # ---- Page Config ----
8 st.set_page_config(page_title="HealthAI", layout="wide")
9 st.markdown("<h1 style='text-align: center;'>% HealthAI - Intelligent Healthcare Assistant</h1>", unsafe_allow_html=True)
10
11 # ---- Sidebar Navigation ----
12 with st.sidebar:
13     st.image("https://cdn-icons-png.flaticon.com/512/3774/3774299.png", width=80)
14     menu = st.radio("Navigation", ["Patient Chat", "Disease Prediction", "Treatment Plan", "Health Analytics"])
15     st.markdown("----")
16     st.markdown("Developed by Divya Reddy")
17     st.caption("Data Secure - IBM Watsonx Powered")
18
19 # ---- Session for Patient Profile ----
20 if "profile" not in st.session_state:
21     st.session_state.profile = {}
22
23 # ---- Left Panel: Patient Profile Form ----
24 col1, col2 = st.columns([1, 2])
25
26 with col1:
27     st.markdown("## Patient Profile")
28     with st.form("profile_form"):
29         name = st.text_input("Name", st.session_state.profile.get("name", ""))
30         age = st.number_input("Age", min_value=0, max_value=120, step=1, value=st.session_state.profile.get("age", 30))
31         gender = st.selectbox("Gender", ["Male", "Female", "Other"], index=["Male", "Female", "Other"].index(st.session_state.profile.get("gender", "Male")))
32         history = st.text_area("Medical History", st.session_state.profile.get("history", ""))
33         medications = st.text_area("Current Medications", st.session_state.profile.get("medications", ""))

```

```

app (2).py  app (6).py  app.py  X
C:\Users\Divya Siva Naga > OneDrive > Desktop > FST > healthai_project_updated > healthai > app.py
26     with col1:
27         with st.form("profile_form"):
33             medications = st.text_area("Current Medications", st.session_state.profile.get("medications", ""))
34             allergies = st.text_input("Allergies", st.session_state.profile.get("allergies", ""))
35
36             submitted = st.form_submit_button("📄 Save Profile")
37             if submitted:
38                 st.session_state.profile = {
39                     "name": name,
40                     "age": age,
41                     "gender": gender,
42                     "history": history,
43                     "medications": medications,
44                     "allergies": allergies
45                 }
46                 st.success("✅ Profile saved!")
47
48             if st.session_state.profile:
49                 st.markdown("#### 📄 Saved Info")
50                 st.json(st.session_state.profile)
51
52 # ---- Right Panel: Functional Modules ----
53 with col2:
54
55     profile = st.session_state.profile
56     profile_summary = f"""
57     Patient Info:
58     - Age: {profile.get('age')}
59     - Gender: {profile.get('gender')}
60     - History: {profile.get('history')}
61     - Medications: {profile.get('medications')}
62     - Allergies: {profile.get('allergies')}
63     """

```

```

app (2).py  app (6).py  app.py  X
C:\Users\Divya Siva Naga > OneDrive > Desktop > FST > healthai_project_updated > healthai > app.py
53 with col2:
54
55 # ---- Patient Chat ----
56 if menu == "🗨️ Patient Chat":
57     st.subheader("👤 Patient Chat Assistant")
58     user_input = st.text_area("📝 Ask a medical question:")
59
60     if st.button("💡 Get AI Response"):
61         if user_input.strip():
62             with st.spinner("Consulting HealthAI..."):
63                 prompt = f"You are a healthcare assistant. Respond based on the following profile and question:\n\n{profile_summary}\n\nQuestion: {user_input}"
64                 response = get_ai_response(prompt)
65                 st.success(response)
66         else:
67             st.warning("⚠️ Please enter a question.")
68
69 # ---- Disease Prediction ----
70 elif menu == "🔍 Disease Prediction":
71     st.subheader("📋 Disease Prediction Based on Symptoms & Profile")
72     symptoms = st.text_input("📝 Enter symptoms (comma-separated):")
73
74     if st.button("🔍 Predict Disease"):
75         if symptoms.strip():
76             with st.spinner("Analyzing symptoms..."):
77                 prompt = f"Given the patient profile:\n\n{profile_summary}\n\nSymptoms: {symptoms}\n\nPredict possible diseases and next steps."
78                 result = get_ai_response(prompt)
79                 st.success(result)
80         else:
81             st.warning("⚠️ Please enter symptoms.")
82
83 # ---- Treatment Plan ----
84 elif menu == "💡 Treatment Plan":
85     st.subheader("📋 Generate Personalized Treatment Plan")
86     disease = st.text_input("📝 Enter diagnosed condition:")


```

```

app (2).py  app (6).py  app.py  X
C:\Users\Divya Siva Naga > OneDrive > Desktop > FST > healthai_project_updated > healthai > app.py
93
94 elif menu == "💡 Treatment Plan":
95     if st.button("📋 Get Treatment Plan"):
96         if disease.strip():
97             with st.spinner("Creating treatment guidance..."):
98                 prompt = f"Generate a treatment plan for this patient:\n\n{profile_summary}\n\nCondition: {disease}"
99                 plan = get_ai_response(prompt)
100                 st.success(plan)
101         else:
102             st.warning("⚠️ Enter a condition to continue.")
103
104 # ---- Health Analytics ----
105 elif menu == "📊 Health Analytics":
106     st.subheader("📈 Patient Health Analytics")
107
108     try:
109         df = pd.read_csv("data/patient_data.csv")
110         metric = st.selectbox("📊 Select a metric to visualize", df.columns[1:])
111         st.line_chart(df.set_index(df.columns[0])[metric])
112
113         if st.button("💡 Generate AI Insight"):
114             summary = df[metric].describe().to_string()
115             prompt = f"Given the patient's profile:\n\n{profile_summary}\n\nAnd health data summary:\n\n{summary}\n\nProvide analysis and recommendations."
116             ai_insight = get_ai_response(prompt)
117             st.info(ai_insight)
118     except FileNotFoundError:
119         st.error("❌ data/patient_data.csv not found.")
120
121 # ---- Footer ----
122 st.markdown("⚠️ unsafe_allow_html=True")
123 st.markdown("🏠<small> 2025 HealthAI - Built with ❤️ using Streamlit and IBM Watsonx</small></center>", unsafe_allow_html=True)

```

## • OUTPUT :



Navigation


- ☐ Patient Chat
- ☐ Disease Prediction
- ☐ Treatment Plan
- ☒ Health Analytics

Developed by Divya Reddy

Data Secure - IBM Watsonx Powered

## HealthAI - Intelligent Healthcare Assistant

Deploy



### Patient Profile


Name: rani

Age: 30

Gender: Female

Medical History: prior illnesses, surgeries, hospitalizations, and other relevant health events

Current Medications: current health status, including any




### Patient Chat Assistant

Ask a medical question:

in which scenarios get the headache feeling?

Get AI Response

- while urinating
- while defecating
- when you sneeze
- when you get a sunburn
- when you're about to sneeze



Navigation


- ☐ Patient Chat
- ☒ Disease Prediction
- ☐ Treatment Plan
- ☐ Health Analytics

Developed by Divya Reddy

Data Secure - IBM Watsonx Powered

## HealthAI - Intelligent Healthcare Assistant

Deploy



### Patient Profile


Name: rani

Age: 30

Gender: Male

Medical History: prior illnesses, surgeries, hospitalizations, and other relevant health events

Current Medications: a detailed account of a patient's current health problems, including




### Disease Prediction Based on Symptoms & Profile

Enter symptoms (comma-separated):

fatigue, swelling in the extremities, changes in urination, foamy or bloody urine, and itchy skin

Predict Disease

The patient presented with fatigue, swelling in the extremities, changes in urination, foamy or bloody urine, and itchy skin. The symptoms have developed over time. The patient is most likely suffering from a chronic condition, such as kidney disease or diabetes, and they should consult with a physician as soon as possible to receive a proper diagnosis and treatment.



Navigation


- ☐ Patient Chat
- ☐ Disease Prediction
- ☒ Treatment Plan
- ☐ Health Analytics

Developed by Divya Reddy

Data Secure - IBM Watsonx Powered

## HealthAI - Intelligent Healthcare Assistant

Deploy



### Patient Profile


Name: rani

Age: 30

Gender: Male

Medical History: prior illnesses, surgeries, hospitalizations, and other relevant health events

Current Medications: a detailed account of a patient's current health problems, including



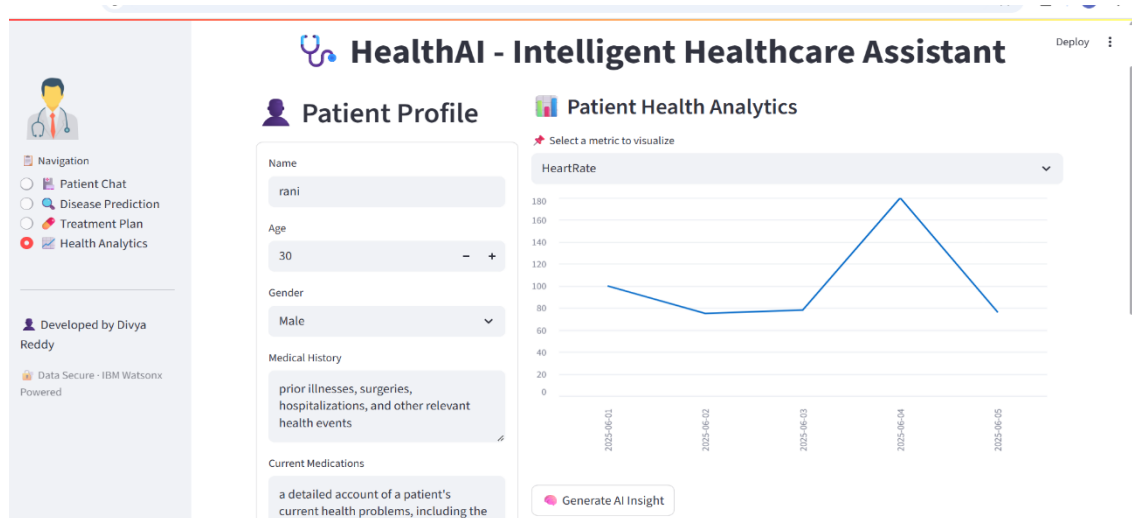
### Generate Personalized Treatment Plan

Enter diagnosed condition:

common illnesses like the flu, hypertension, and appendicitis, as well as more complex condi

Get Treatment Plan

for a patient with a chronic condition. I will focus on the patient's history, allergies, and current medications. I will also consider the patient's age, gender, and other relevant factors. The goal of treatment is to improve the patient's quality of life and help them manage their chronic condition. I will recommend a treatment plan that includes lifestyle changes, medication, and possibly surgery or other procedures. I will also encourage the patient to be active and participate in their own care.



## 12. Known Issues

- Generic model outputs due to lack of medical domain fine-tuning
- Internet dependency when using IBM watsonx.
- No data persistence (currently stateless app)

## 13. Future Enhancements

- ✓ Add user authentication and patient record storage
- ✓ Multilingual prompt support
- ✓ Mobile version of the app
- ✓ Integrate with real-time health APIs or EHRs