

PHASE-1
BRAINSTROMING AND IDEATION

Project Overview

Project Title: Health AI: Intelligent Healthcare Assistant using IBM Granite

Description: Health AI: Intelligent Healthcare Assistant using IBM Granite is an advanced

AI-driven healthcare platform designed to revolutionize patient interaction and clinical decision making. By harnessing the capabilities of IBM Granite—a cutting-edge large language model architecture—this assistant delivers personalized, context-aware support across various healthcare functions.

From symptom triage and appointment scheduling to medication reminders and health education, the assistant integrates seamlessly into clinical workflows while maintaining stringent data privacy and compliance standards. Its natural language processing capabilities allow it to understand patient inquiries, generate insightful responses, and assist healthcare providers with documentation and diagnostics.

Team ID: LTVIP2025TMID30267

Team Members:

(1) K Venkatesh

(2) P Chandrasekhar Reddy

(3) Gurikari Damodhar

(4) K Hareesh Kumar Reddy

(5) P Nagaraj

Project Considerations

1. Objectives

- **Problem Statements:**

The current healthcare system often struggles with timely diagnosis, patient data management, and continuous monitoring due to a shortage of medical professionals, inconsistent data handling, and lack of intelligent support tools.

- **Purpose and Impact of the Project:**

HealthAI aims to leverage IBM Granite's AI capabilities to assist in real-time patient monitoring, preliminary diagnosis, and efficient data management. The project seeks to enhance healthcare accessibility, reduce the burden on healthcare providers, and deliver faster, more accurate insights—ultimately improving patient outcomes and supporting medical staff.

2. Key Points

i.Problem Statement:

Healthcare systems today face challenges in timely diagnosis, patient monitoring, and personalized care due to increasing patient loads, limited staff, and fragmented data systems.

ii.Solution Proposed:

HealthAI will integrate IBM Granite's advanced AI capabilities to provide a smart healthcare assistant that can analyze patient data, offer preliminary diagnostics, monitor health conditions continuously, and assist doctors and patients with relevant insights and alerts.

iii.Target Users:

Hospitals, clinics, doctors, healthcare workers, and patients—especially in rural or underserved areas where access to quality healthcare is limited.

iv.Expected Outcome:

An intelligent, scalable healthcare assistant that enhances medical decision-making, streamlines patient care workflows, and improves healthcare access and quality.

PHASE-2
REQUIREMENT ANALYSIS

Key Points:

Technical Requirements:

- Languages: Python, IBM Cloud
- Frameworks: Google Colab for backend, React for frontend
- Tools: IBM Granite, IBM Watson, Cloud services (IBM Cloud), GitHub

Functional Requirements:

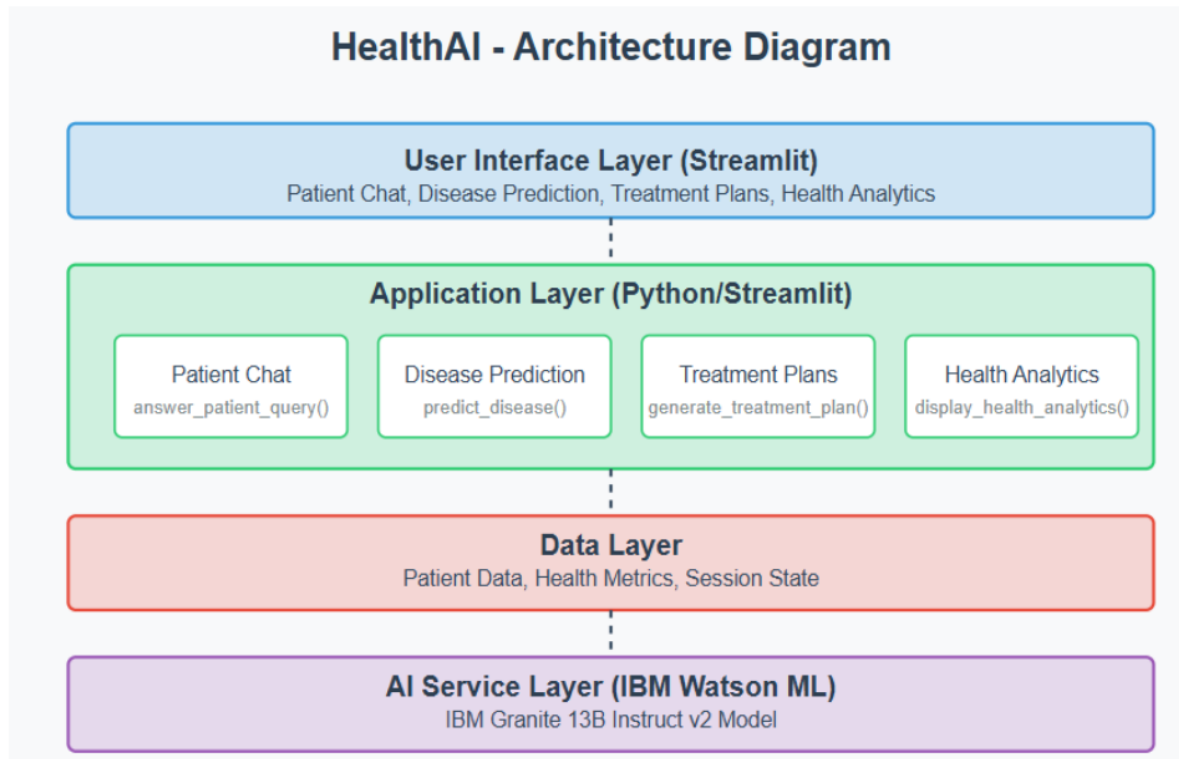
- Patient data input and management
- AI-driven health monitoring and suggestions
- Real-time alerts and notifications
- Role-based access (patient, doctor)

Constraints & Challenges:

- Ensuring data privacy and compliance with health regulations (e.g., HIPAA)
- Integrating with existing hospital systems
- Achieving high accuracy in AI diagnostics
- Network and hardware limitations in remote areas

PHASE-3
PROJECT DESIGNING

Architecture Diagram:



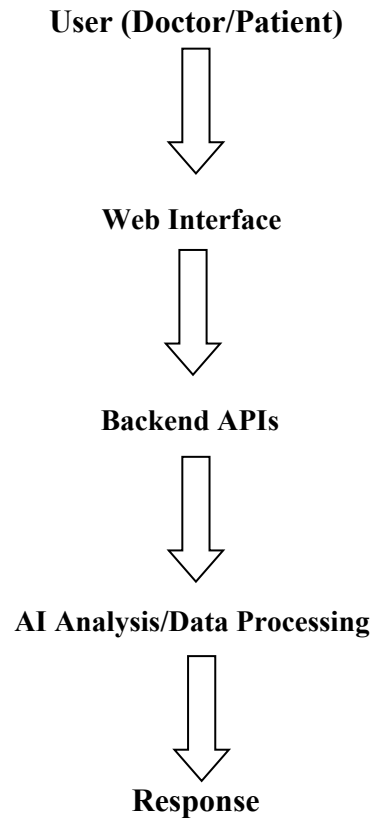
Key Points:

i. System Architecture Diagram

The system architecture consists of the following components:

- **Frontend Interface:** Built with React.js for user interaction (patients/doctors)
- **Backend Server:** Google Collab APIs hosted on Hugging face to handle requests, logic
- **AI Layer:** IBM Granite for natural language processing and intelligent diagnostics
- **Cloud Integration:** IBM Watson for additional AI services and chatbot interaction

ii .Flow Diagram



iii. User Flow

- **User Login:** Patients and doctors can easily login.
- **Dashboard Access:** Role-based access directs users to their respective dashboards.
 - Patients: View health records, enter symptoms, chat with assistant
 - Doctors: Monitor patient data, AI-generated alerts, and suggestions
- **Interaction with AI Assistant:** Users input symptoms/questions and receive AI responses.
- **Real-Time Monitoring:** System tracks key indicators and notifies users of concerns.
- **Data Updates & Reports:** Users can view reports, history.

- **Logout:** Ends secure session.

PHASE-4

PROJECT PLANNING (AGILE METHODOLOGY)

Sprint planning & Task allocation:

(i).Technical Architecture & Pre-requisites:

- ✓ P.Chandrasekhar Reddy
- ✓ P.Nagaraj
- ✓ Gurikari Damodhar

(ii).Milestone 1:Model Selection & Architecture:

- ✓ K.Venkatesh

(iii)Milestone 2:Core Functionalities Development:

- ✓ K.Hareesh Kumar Reddy

(iv)App.py Development:

- ✓ K.Venkatesh

(v)Design & Developing the user interface:

- ✓ K.Venkatesh
- ✓ P.Chandrasekar Reddy

(vi)Deployment & Conclusion:

- ✓ K.Venkatesh
- ✓ K.Hareesh Kumar Reddy
- ✓ P.Chandrasekar Reddy
- ✓ P.Nagaraj
- ✓ Gurikari Damodhar

PHASE-5
PROJECT DEVELOPMENT

TOOLS AND TECHNOLOGY USED:

Various tools and technology were used during our internship training. Some of them are listed below:

Python:

It is a high-level, general-purpose programming language widely used for developing web applications, automation scripts, AI/ML models, data analysis, and more.

Gradio:

It is an open-source Python library that allows you to quickly create and share interactive user interfaces for machine learning models, data science workflows, or AI applications — all in just a few lines of code.

IBM Cloud:

IBM Cloud is a comprehensive suite of cloud computing services offered by IBM. It provides infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) to help businesses and developers build, deploy, and manage applications and services over the internet.

- necessary libraries

```
• pip install transformers gradio
```

- Load the IBM Granite Model from Hugging

```
•  
• from transformers import AutoTokenizer, AutoModelForCausalLM, pipeline  
•  
• model_name = "ibm-granite/granite-3.3-2b-instruct"  
• tokenizer = AutoTokenizer.from_pretrained(model_name)  
• model = AutoModelForCausalLM.from_pretrained(model_name)  
• # Create a text generation pipeline  
• generator = pipeline("text-generation", model=model, tokenizer=tokenizer,  
• max_new_tokens=512)
```

- Functionalites:

```
• def patient_chat_assistant(user_query):
```

```
• prompt = f"You are a helpful and safe AI healthcare assistant. Do not  
provide diagnoses or prescriptions. Patient says: '{user_query}'. Respond  
clearly and responsibly."  
• result = generator(prompt)[0]['generated_text']  
• return result[len(prompt):].strip()  
• def disease_prediction(symptoms):  
• prompt = f"Patient presents with the following symptoms: {symptoms}.  
What possible non-diagnostic conditions could be considered? Provide  
general insights only."  
• result = generator(prompt)[0]['generated_text']  
• return result[len(prompt):].strip()  
• def treatment_plan(age, gender, history, condition):  
• prompt = (f"Patient profile: Age: {age}, Gender: {gender}, Medical  
History: {history}. "  
• f"Condition: {condition}. Suggest general home remedies,  
over-the-counter guidance, and wellness tips. "  
• f"Do not prescribe medication.")  
• result = generator(prompt)[0]['generated_text']  
• return result[len(prompt):].strip()
```

- Gradio interface:

```
• import gradio as gr  
•  
• with gr.Blocks(title="HealthAI: Intelligent Healthcare Assistant") as app:  
• gr.Markdown("## 🩺 HealthAI: Intelligent Healthcare Assistant")  
•  
• with gr.Tab("🗨 Patient Chat Assistant"):  
• user_input = gr.Textbox(label="Ask a health-related question (non-  
diagnostic)")  
• chat_output = gr.Textbox(label="Response")  
• chat_btn = gr.Button("Get Response")  
• chat_btn.click(fn=patient_chat_assistant, inputs=user_input,  
outputs=chat_output)  
•  
• with gr.Tab("🔍 Disease Prediction"):  
• symptoms_input = gr.Textbox(label="Enter symptoms (comma-  
separated)")  
• prediction_output = gr.Textbox(label="Possible Conditions")  
• pred_btn = gr.Button("Predict")  
• pred_btn.click(fn=disease_prediction, inputs=symptoms_input,  
outputs=prediction_output)  
•  
• with gr.Tab("📋 Personalized Treatment Plan"):
```

```
•     age = gr.Number(label="Age")
•     gender = gr.Dropdown(["Male", "Female", "Other"], label="Gender")
•     history = gr.Textbox(label="Medical History")
•     condition = gr.Textbox(label="Current Condition")
•     plan_output = gr.Textbox(label="Suggested Plan")
•     plan_btn = gr.Button("Generate Plan")
•     plan_btn.click(fn=treatment_plan, inputs=[age, gender, history,
condition], outputs=plan_output)
•
•     app.launch()
```

Key Points:

i. Technology Stack Used:

- **Frontend:** React.js
- **Backend:** T4 GPU in google collab
- **AI Model:** IBM Granite for NLP and diagnostics
- **Chatbot/NLP Service:** IBM Watson
- **Hosting/Infrastructure:** IBM Cloud, Hugging Face
- **Tools:** GitHub

ii. Development Process:

- Defined the project scope and created wireframes for the user interface.
- Developed the frontend using React.js with role-based UI components.
- Built backend APIs in Python to handle data requests, connected with Google Colab and IBM services.
- Integrated IBM Granite for generating AI responses to health queries.

- Set up IBM Watson Assistant for NLP-based chatbot interactions.
- Tested interactions between components and deployed to IBM Cloud

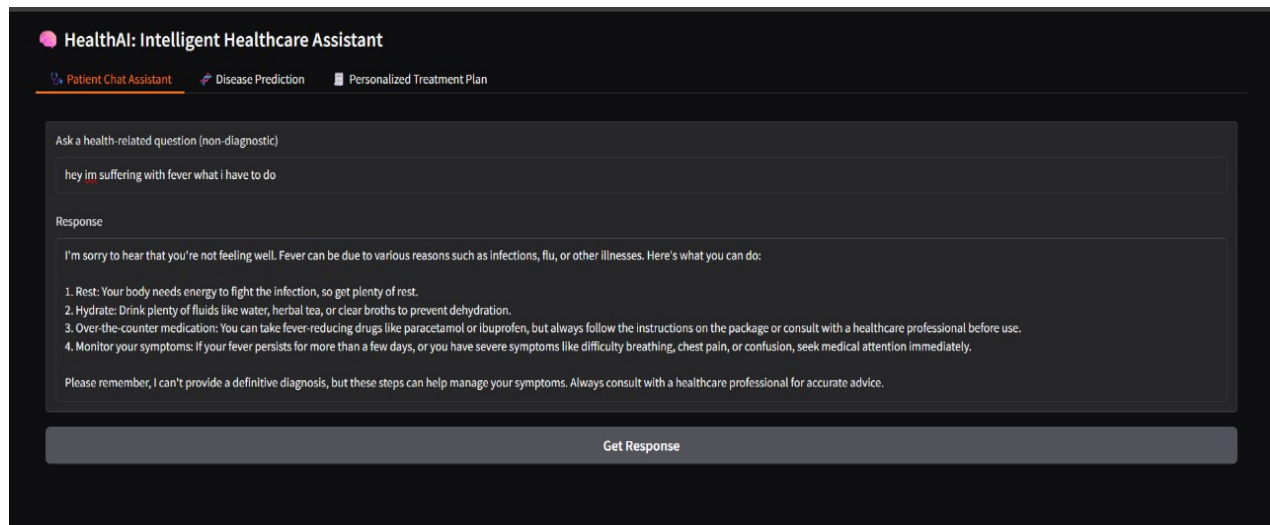
iii.Challenges and Fixes:

- **Challenge:** Integrating IBM Granite with real-time backend processing.
Fix: Used intermediary layers and asynchronous processing to streamline API response times.
- **Challenge:** Ensuring accurate health insights without false positives.
Fix: Added rule-based filters to refine AI-generated suggestions.

PHASE-6
FUNCTIONAL & PERFORMANCE TESTING

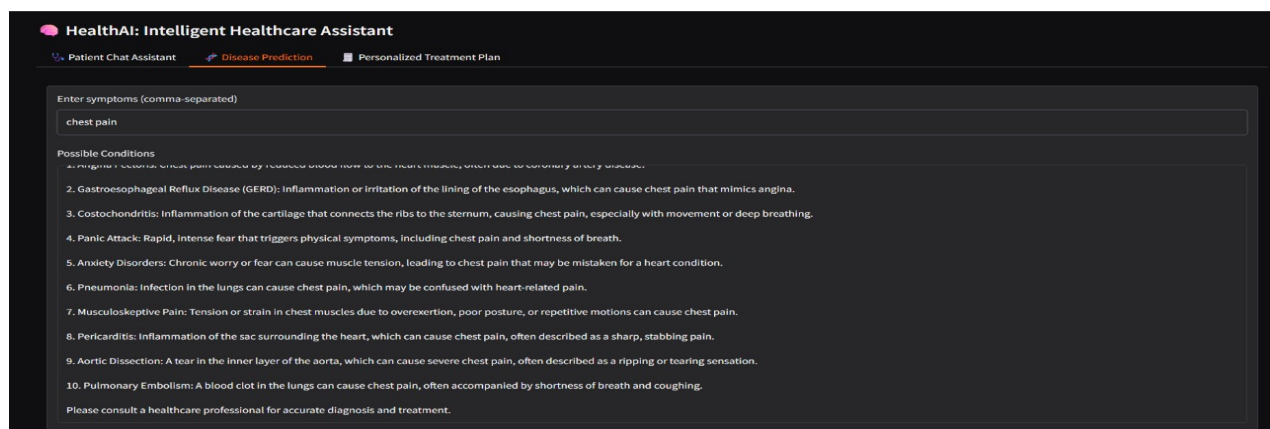
Test Case Scenarios:

Scenario 1: A user has a health-related question. Through the Patient Chat interface, they can ask any medical query and receive a clear, empathetic response that includes relevant medical facts, acknowledges limitations, and suggests when to seek professional medical advice.



The screenshot shows the 'HealthAI: Intelligent Healthcare Assistant' interface. At the top, there are three tabs: 'Patient Chat Assistant' (selected), 'Disease Prediction', and 'Personalized Treatment Plan'. Below the tabs, there is a text input field with the placeholder 'Ask a health-related question (non-diagnostic)'. The user has entered 'hey im suffering with fever what i have to do'. Below the input field, there is a 'Response' section. The response text reads: 'I'm sorry to hear that you're not feeling well. Fever can be due to various reasons such as infections, flu, or other illnesses. Here's what you can do:'. This is followed by a numbered list of four items: 1. Rest: Your body needs energy to fight the infection, so get plenty of rest. 2. Hydrate: Drink plenty of fluids like water, herbal tea, or clear broths to prevent dehydration. 3. Over-the-counter medication: You can take fever-reducing drugs like paracetamol or ibuprofen, but always follow the instructions on the package or consult with a healthcare professional before use. 4. Monitor your symptoms: If your fever persists for more than a few days, or you have severe symptoms like difficulty breathing, chest pain, or confusion, seek medical attention immediately. Below the list, there is a disclaimer: 'Please remember, I can't provide a definitive diagnosis, but these steps can help manage your symptoms. Always consult with a healthcare professional for accurate advice.' At the bottom of the interface, there is a 'Get Response' button.

Scenario 2: A user inputs their symptoms into the Disease Prediction system, describing issues like persistent headache, fatigue, and mild fever. The system analyzes the symptoms along with the patient's profile and health data to provide potential condition predictions, including likelihood assessments and recommended next steps.



The screenshot shows the 'HealthAI: Intelligent Healthcare Assistant' interface. At the top, there are three tabs: 'Patient Chat Assistant', 'Disease Prediction' (selected), and 'Personalized Treatment Plan'. Below the tabs, there is a text input field with the placeholder 'Enter symptoms (comma-separated)'. The user has entered 'chest pain'. Below the input field, there is a 'Possible Conditions' section. This section contains a numbered list of ten items: 1. Angina Pectoris: Chest pain caused by reduced blood flow to the heart muscle, often described as a squeezing or heavy sensation. 2. Gastroesophageal Reflux Disease (GERD): Inflammation or irritation of the lining of the esophagus, which can cause chest pain that mimics angina. 3. Costochondritis: Inflammation of the cartilage that connects the ribs to the sternum, causing chest pain, especially with movement or deep breathing. 4. Panic Attack: Rapid, intense fear that triggers physical symptoms, including chest pain and shortness of breath. 5. Anxiety Disorders: Chronic worry or fear can cause muscle tension, leading to chest pain that may be mistaken for a heart condition. 6. Pneumonia: Infection in the lungs can cause chest pain, which may be confused with heart-related pain. 7. Musculoskeletal Pain: Tension or strain in chest muscles due to overexertion, poor posture, or repetitive motions can cause chest pain. 8. Pericarditis: Inflammation of the sac surrounding the heart, which can cause chest pain, often described as a sharp, stabbing pain. 9. Aortic Dissection: A tear in the inner layer of the aorta, which can cause severe chest pain, often described as a ripping or tearing sensation. 10. Pulmonary Embolism: A blood clot in the lungs can cause chest pain, often accompanied by shortness of breath and coughing. Below the list, there is a disclaimer: 'Please consult a healthcare professional for accurate diagnosis and treatment.'

Scenario 3: A user needs personalized treatment recommendations for a diagnosed condition. By entering their condition in the Treatment Plans generator, the AI processes the information

Health AI: Intelligent Healthcare Assistant using IBM Granite

along with patient data to create a comprehensive, evidence-based treatment plan that includes medications, lifestyle modifications, and follow-up testing.

HealthAI: Intelligent Healthcare Assistant

🗨️ Patient Chat Assistant

🔮 Disease Prediction

📋 Personalized Treatment Plan

Age

25

Gender

Male

Medical History

shortness of breath

Current Condition

anxiety ,stress

Suggested Plan

1. **"Deep Breathing Exercises"**: Deep breathing can help reduce anxiety and lower blood pressure. Try the 4-7-8 technique: inhale for 4 seconds, hold the breath for 7 seconds, and exhale for 8 seconds. Repeat this cycle 4-6 times.

2. **"Regular Physical Activity"**: Engage in at least 30 minutes of moderate-intensity exercise, like brisk walking, cycling, or swimming, most days of the week. Regular exercise can help reduce anxiety and improve mood.

3. **"Healthy Diet"**: Maintain a balanced diet rich in fruits, vegetables, lean proteins, and whole grains. Limit intake of caffeine, alcohol, and processed foods, which can exacerbate anxiety.

4. **"Adequate Sleep"**: Prioritize 7-9 hours of quality sleep each night. Establish a consistent sleep schedule and create a relaxing bedtime routine to improve sleep quality.

5. **"Mindfulness and Meditation"**: Practice mindfulness techniques, such as yoga or guided meditation, to help manage stress and anxiety. There are numerous apps and online resources available to guide you through these practices.

6. **"Stress Management Techniques"**: Incorporate stress-reduction techniques like progressive muscle relaxation, deep breathing, or autogenic training into your daily routine.

7. **"Stay Hydrated"**: Drink plenty of water throughout the day to support overall health and well-being.

8. **"Social Connections"**: Maintain strong relationships with friends and family. Social support can help buffer the effects of stress and anxiety.

9. **"Limit Screen Time"**: Excessive screen time, especially before bed, can contribute to anxiety. Set boundaries for device usage and establish tech-free times during the day.

10. **"Consider Professional Help"**: If symptoms persist or worsen, consider consulting a mental health professional, such as a therapist or counselor, for personalized guidance and support.

Conclusion:

The HealthAI project effectively demonstrates the potential of AI in revolutionizing healthcare assistance. By integrating IBM's Granite language model, the platform enables users to receive personalized health insights through Patient Chat, Disease Prediction, Treatment Plan Generation, making healthcare information more accessible.

Utilizing IBM Watson Machine Learning, the application ensures accurate health question answering, detailed disease prediction, personalized treatment recommendations. The structured development process—spanning model selection, core feature implementation, backend and frontend development, and deployment—led to the creation of an interactive, user-friendly platform.

Built with Gradio, HealthAI facilitates seamless visualization of health data and AI-generated insights, ensuring an efficient and responsive experience. This project highlights how targeted AI models and a well-structured framework can enhance healthcare accessibility. With future scalability in mind, HealthAI has the potential to expand its capabilities, incorporating more advanced diagnostics and broader medical applications.