



HEALTHAI: Intelligent Healthcare Assistant using IBM Granite

1. INTRODUCTION

1.1 Project Overview

HEALTHAI: Intelligent Healthcare Assistant using IBM Granite is a generative AI-powered application designed to provide smart healthcare support to patients through an interactive and intuitive interface. The system leverages IBM's Granite language model to facilitate health-related conversations, predict diseases based on symptoms, suggest possible treatment plans, and display useful health analytics. Developed using Python and Streamlit, the application aims to simplify patient engagement and support early diagnosis and treatment planning through AI.

1.2 Purpose

The primary purpose of this project is to harness the power of Generative AI for delivering accessible, reliable, and intelligent healthcare support. HEALTHAI serves as a virtual health assistant that helps users:

- ☒ Get instant responses to general health queries.
- ☒ Predict diseases based on symptoms using AI.
- ☒ Receive relevant treatment suggestions.
- ☒ View simple, clear analytics on health trends.

This project also demonstrates the practical application of IBM Granite models in solving real-world healthcare problems, fulfilling academic and internship goals under the IBM Generative AI program.

2. IDEATION PHASE

2.1 Problem Statement

Date: 21 JUNE 2025

Team ID: LTVIP2025TMID38244

Project Name: Health AI: Intelligent Healthcare Assistant Using IBM Granite

Maximum Marks: 4 Marks

Customer Problem Statement Template

Create a problem statement to understand your customer's point & view. The Customer Problem Statement helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows your team and your users to find the ideal solution your business faces. Throughout the process, you'll also be able to empathize with your customer as you better understand your

Template: <https://miro.com/templeplates/customerproblem-statement/>

Example:

Problem Statement (PS)	(i am)	I'm trying to	But	Which makes me feel
PS-1	a patient	manage my health effectively	I face difficulty	frustrated and anxious about my well-being
I'm		manage my health effectively	I face continued and lacks processing and medical insurances' the current healthcare system is fragmented and lacks proactive support	

2.2 Empathy Map Canvas

Date: 21 JUNE 2025

Team ID: LTVIP2025TMID38244

Project Name: Health AI: Intelligent Healthcare Assistant Using IBM Granite

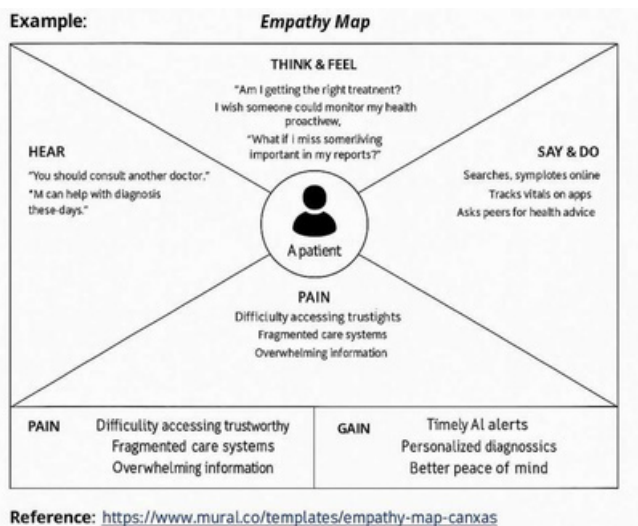
Maximum Marks: 2 Marks

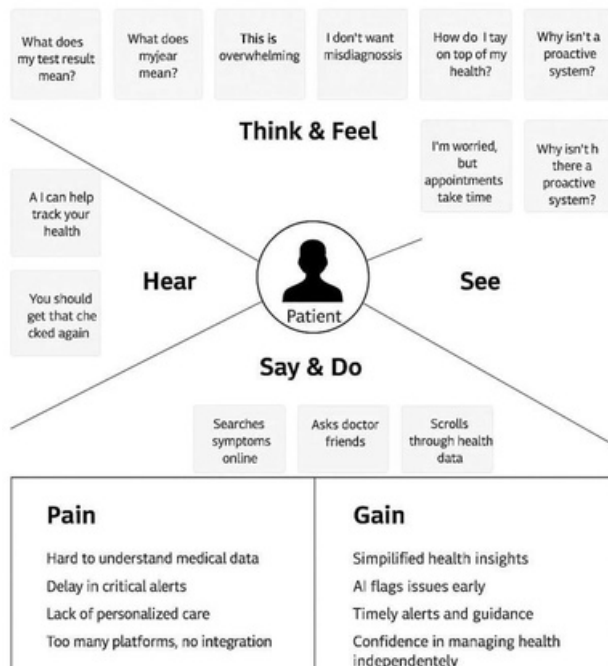
Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes.

It is useful to helping teams understand their users.

Creating an effective solution requires understanding the person who is experiencing it. Examine participants consider how participants consider user needs, goals, and challenges





2.3 Brainstorming

Date: 21 JUNE 2025

Team ID: LTVIP2025TMID38244

Project Name: Health AI: Intelligent Healthcare Assistant Using IBM Granite

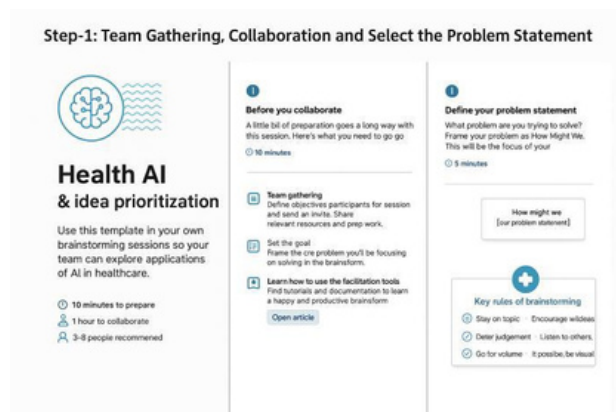
Maximum Marks: 4 Marks

Brainstorm & Idea Prioritization in Health AI

Brainstorming in Health AI promotes free, creative thinking to generate innovative solutions for healthcare challenges using artificial intelligence. To collect a wide range of ideas from diverse team members, then prioritize based on impact, feasibility, and urgency. Encourage maximum idea generation, regardless of practicality at first.

Cross-functional team members (AI developers, clinicians, analysts) co-create ideas. Ideal for distributed teams using tools like Miro or Mural. AI-driven symptom checking, disease prediction, treatment plans, and patient engagement tools. Impact – Patient outcomes and healthcare system improvement. Feasibility – Technical readiness with health regulations.

Reference: Brainstorm and idea prioritization template | Mural



Step-2: Brainstorm, Idea Listing and Grouping



Health AI & idea prioritization

In a brainstorming session, list and group ideas for addressing your problem statement:

Health AI

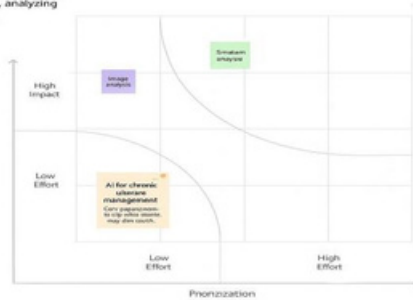


Step-3: Idea Prioritization

In the quadrants shown below, plot the most valuable ideas, analyzing Impact on the problem vs. effort to implement



Prioritization
In the quadrant-Users below plot the most valuable ideas analyzing impact on implement



3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

Health AI



3.2 Solution Requirement

Solution Requirements (Functional & Non-functional)

17 Date: 21 JUNE 2025 ID

Team ID: LTVIP2025TMID38244

✦ Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite

📊 Maximum Marks: 4 Mar

Functional Requirements:

Following are the functional requirements of the proposed solution.

Health AI

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3		

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Functional Requirement	Sprint	Story ID	User Story / Task	Story Points	Priority
Registration	Sprint 1	US#4	As a user, I can register for the application (US3)	5	High
		US#2	As a user, oral responses can be analyzed using speech-to-text (US2)	8	High
Login	Sprint 1	US#3	As a user, health data can be input into system	7	High
		US#1	As a user, I can log in to the application	2	High
Dashboard	Sprint 2	US#1	As a user, I can view health data visualizations on the central dashboard (US5)	2	Medium

3.3 Data Flow Diagram

Data Flow Diagram & User Stories

17 Date: 21 JUNE 2025 ID

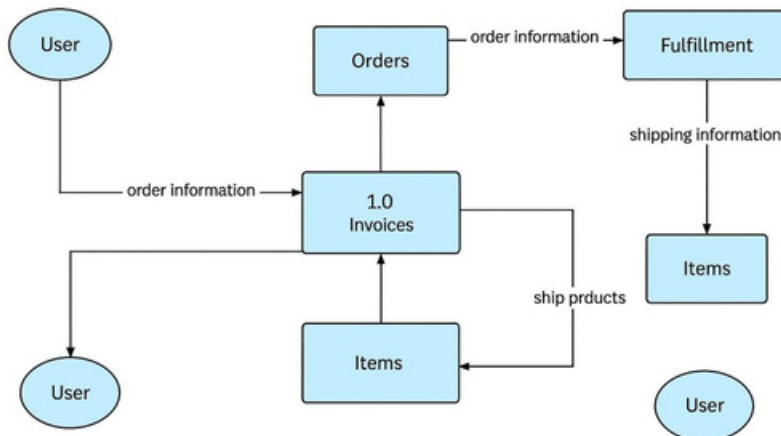
Team ID: LTVIP2025TMID38244

Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite Maximum






Marks: 2 Mar

Data Flow Diagrams: A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: DFD Level 0 (Industry Standard)




Health AI


User Type	Functional Requirement	User Story / Task	Acceptance criteria	Priority	Rele
 Customer (Mobile user)	Registration	As a user, I can register by providing an email and password.	Email and password can be used to log in	High	Spri 1
	USS1	As a user, I will receive confirmation email	Confirmation email received	High	Spri 1
 Tester	USS2	As a user, I can enable systemwide speech-t-o-text	Speech-to-text is active throughout the app	Low	Spri 2
 Administrator	USS3	As a tester, I can analyze speech responses	Speech responses are analyzed correctly	Medium	Spri 1
	USS4	As an admin, I can view health data visualizations		Sprint 1	Spri 1
 Administrator	US4	As a tester, I can analyze speech responses	Speech responses are analyzed correctly	Medium	Spri 1
 Visualizaation	US5	As a tester, I can analyze speech responses	Health data visualizations are available	High	Spri 1

3.4 Technology Stack

Technology Stack (Architecture & Stack)

 Date: 21 JUNE 2025

 Team ID:LTVIP2025TMID38244

 Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite

 Maximum Marks: 4 Marks

Technical Architecture – HealthAI

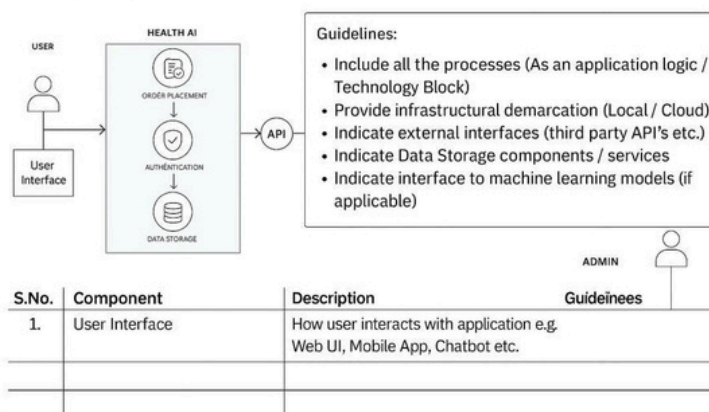
HealthAI's technical architecture is designed to provide intelligent, personalized, and accessible healthcare assistance using IBM's AI capabilities. The architecture bridges the gap between healthcare user needs and AI-driven digital solutions by clearly defining modules, workflows, and technology integrations.

It follows principles of modular design, AI integration, secure backend logic, and interactive frontend experiences.

References – Adapted for HealthAI

1. C4 Model – Software Architecture Visualization Used as the base modeling approach to define different levels of HealthAI's architecture (context, container, component). <https://c4model.com/>
2. IBM Order Processing System (Pandemic Reference) Inspired HealthAI's backend design by using modular components and AI-powered services similar to order-processing use cases. <https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>
3. IBM Cloud Architecture Center Provided best practices and patterns for integrating AI models and deploying cloud-based healthcare applications. <https://www.ibm.com/cloud/architecture>
4. AWS Architecture Best Practices Used as a comparative reference to validate HealthAI's scalability, resilience, and service-based integration approach. <https://aws.amazon.com/architecture>
5. How to Draw Useful Technical Architecture Diagrams Guided the creation of simplified, functional diagrams for HealthAI's backend and AI data flow. <https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>

Health AI



Health AI Technology Stack

• Application Logic-1: Patient intake and triage processing	Python / Java IBM Watson STT
• Application Logic-2: Voice transcription for patient interactions	IBM Watson STT IBM Watson Assistant
• Database	MySQL / MongoDB
• Cloud Database	IBM DB2 / IBM Cloudant
• File Storage: Medical imaging and document	IBM Block Storage / Local Filesystem
• External API-1 Real-time environmental health tracking	IBM Weather API Aadhaar API
• External API-2	Aadhaar API
• Machine Learning Model Medical image classification	Custom Object Recognition Model
• Infrastructure Scalable deployment for clinical environments	Cloud Foundry / Kubernetes / Local Server


4. PROJECT DESIGN

4.1 Problem Solution Fit

Problem – Solution Fit Template :

 Date: 21 JUNE 2025 





Team ID: LTVIP2025TMID38244

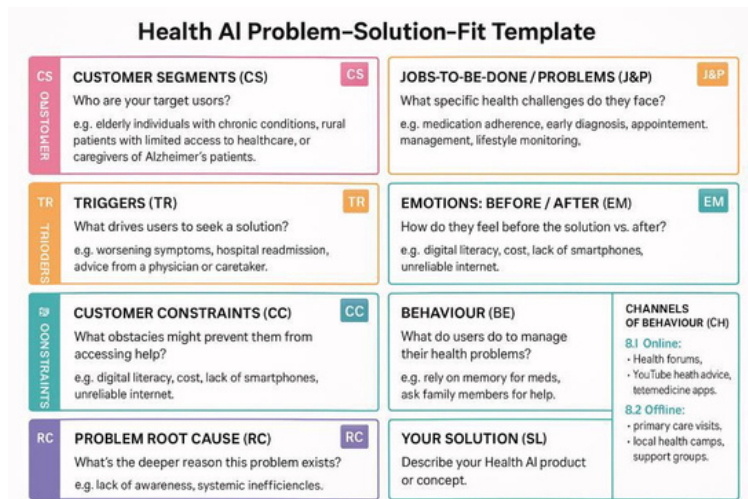
 Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite

 Maximum Marks: 2 Marks

Problem – Solution Fit Template : HealthAI solves a frequent and urgent problem: lack of easy access to valid healthcare information and insights. It taps into the existing behavior of users searching for medical information online and replaces it with a credible, AI-powered platform.

Purpose :


-  Solve complex health-related problems using intelligent and accessible AI assistance
-  Increase solution adoption by reflecting how users already seek medical information online
-  Improve communication using conversational chat and visual analytics
-  Build user trust with consistent, evidence-based responses




References :

- <https://www.ideahackers.network/problem-solution-fit-canvas/>
- <https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe>

4.2 Proposed Solution

 Date: 21 JUNE 2025

 Team ID: LTVIP2025TMID38244

 Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite


 Maximum Marks: 2 Marks

Proposed Solution Template:

Project team shall fill following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Identify a pressing issue in healthcare your AI aims to adress
2.	Idea / Solution description	Summarize your Health AI solution and how it works
3.	Novelty / Uniqueness	What makes your idea different from existing healthcare technologies?
4.	Social Impact / Customer Satisfisfaction	How will it improve lives, patient outcomes, or user experience?
5.	Business Model (Revenue Model)	How will your solution generate revenue or remain sustainable?

4.3 Solution Architectur

 Date: 21 JUNE 2025

 Team ID: LTVIP2025TMID38244



📌 Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite

📋 Maximum Marks: 2 Mar

✅ Solution Architecture – HealthAI

Solution architecture in HealthAI serves as the bridge between real-world healthcare challenges and advanced AI-driven technology. It outlines how HealthAI is built to deliver accurate, personalized, and responsive medical support.

🎯 Goals of HealthAI's Solution Architecture:

1. Identify the most effective AI-driven technology to solve the problem of inaccessible or unreliable healthcare information.
2. Design the complete structure — from user input (like symptoms or questions) to backend AI processing using IBM Granite and secure API handling.

3. Define key features and development phases, including modules like:

oPatient Chat

oDisease Prediction

oTreatment Plan Generation

oHealth Analytics

📋 Key Characteristics of the HealthAI Architecture:

🔍 **Modular and Scalable Design:** Each core functionality is independently built using Python and Streamlit.

🔍 **AI Integration:** IBM Granite (13B Instruct v2) is used to process all medical queries and generate accurate, natural-language responses.

. **User Interface:** Streamlit provides an intuitive frontend with form-based inputs, chatbot interfaces, and dynamic visualizations using Plotly.

🔍 **Data Flow:** User inputs are sent to the AI model via a central shared function (`shared_model.py`), processed securely, and returned in structured output.

🔍 **Security:** Environment variables (`.env`) are used for API key management to protect sensitive credentials.

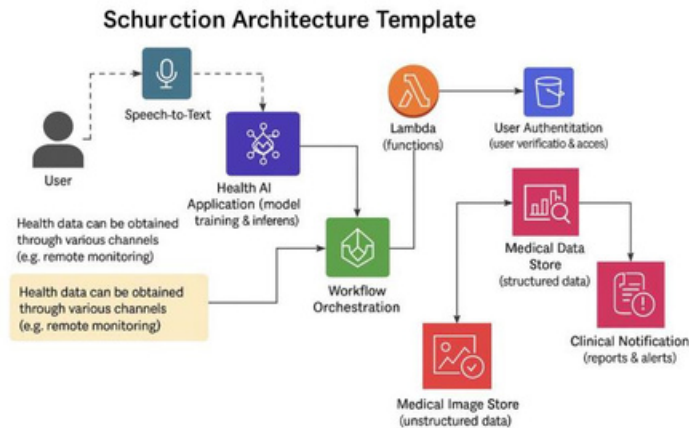


Figure 1: Architecture and data flow of the health AI system


5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

 Date: 15 February 2025

 Team ID: LTVIP2025TMID34281

 Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite

 Maximum Marks: 2 Mar

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV)

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Functional Requirement	Sprint	User Story / Task	Story Points	Priority
Registration	Sprint 1	As a user, I can register for the application (US1)	5	High
Registration	Sprint 1	As a user, real responses can be analyzed using speech-to-text (US2)	8	High
Login	Sprint 1	As a user, health data can be input into system (US3)	7	High
Dashboard	Sprint 2	As a user, I can log in to the application (US4)	4	Medium
Dashboard	Sprint 2	As a user, I can view health data visualizations on the central dashboard (US5)	2	Medium

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Product Backlog, Sprint Schedule, and Etimation (4 Marks)


Functional Requirement	Sprint	Story ID	User Story / Task	Story Points	Priority
Registration	Sprint 1	US#4	As a user, I can register for the application (US3)	5	High
		US#2	As a user, oral responses can be analyzed using speech-to-text (US2)	8	High
Login	Sprint 1	US#3	As a user, health data can be input into system	7	High
		US#1	As a user, I can log in to the application	2	High
Dashboard	Sprint 2	US#1	As a user, I can view health data visualizations on the central (US5 dashboard	2	Medium



6. FUNCTIONAL AND PERFORMANCE TESTING


6.1 Performance Testing

Functional & Performance Testing Template

Model Performance Test

 Date: 21 February 2025

 Team ID: LTVIP2025TMID34281  Project Name: HealthAI: Intelligent Healthcare Assistant Using IBM Granite

 Maximum Marks:

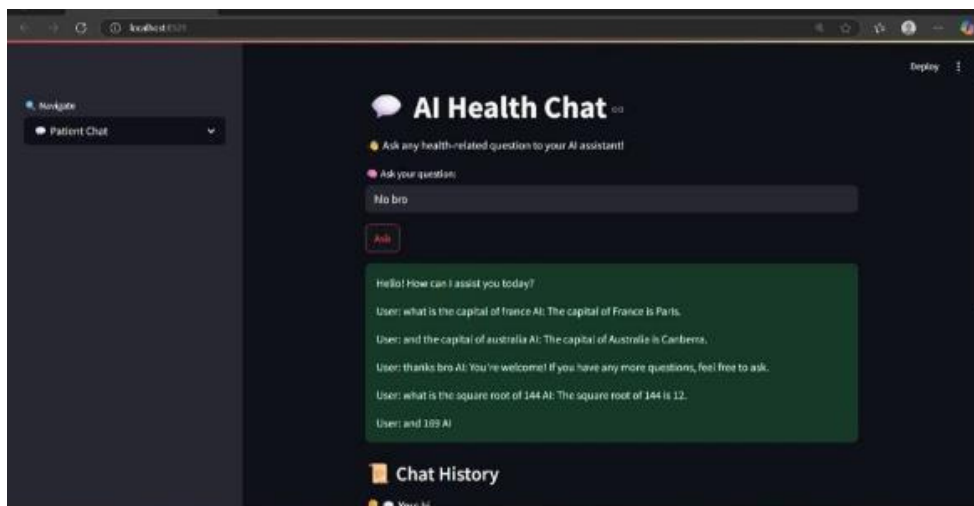
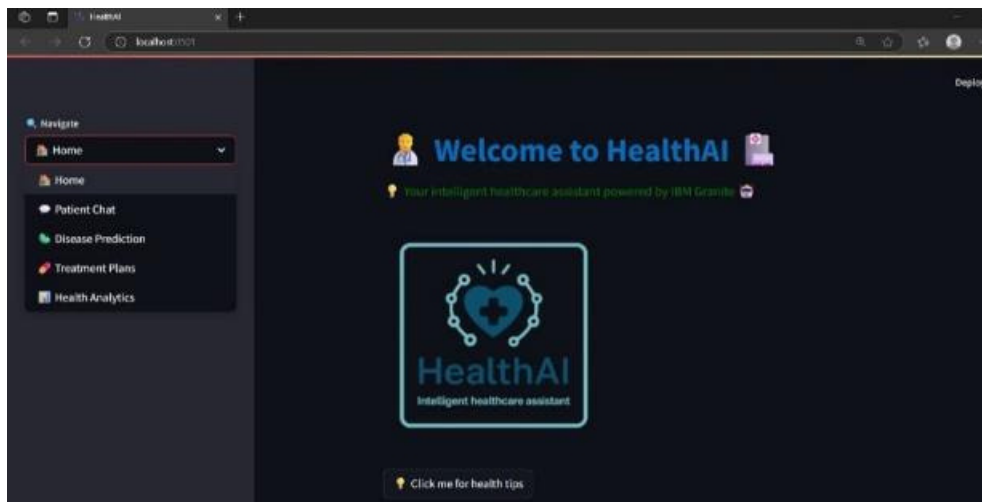
Test Scenarios & Result

Health AI Test Scenarios & Results

Test Case	Scenario (What to test)	Expected Result	Result
HT-A1	Input Validation	Valid inputs accepted	Pass
HT-A2	Name Input	Accepts alph, values	Accepts valid values
HT-A3	Symptom Input	Logg correctly	Symptoms log correcty
HT-A4	Content Generation	Created accurately	Generated accurately
HT-A5	API Connection	API responds	API responds
HT-A6	Response Time	Should be accepstable	Within an acceptable
HT-A7	User submittity multiple inputs	Should not slow	Pass
HT-A8	Upload transfer speed during micage	Should not lag	Should not lag

7. RESULTS

7.1 Output Screenshots



Disease Prediction System

Medical Disclaimer: This tool provides general information only and should not replace professional medical advice. Always consult healthcare professionals for proper diagnosis and treatment.

Symptom Input

Select your primary symptoms:

- ☐ Headache
 ☐ Fever
 ☐ Fatigue
 ☐ Nausea
 ☒ Dizziness
- ☒ Chest Pain

Risk Factors

Create a patient profile for personalized risk assessment.

Describe your symptoms in detail:

I've been experiencing chest pain for the last three days.

Duration of symptoms:

1-3 days

Severity (1-10):

Mid Moderate Severe

Analyze Symptoms

Analysis Results

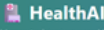
Based on the symptoms you've described:

- Chest pain requires immediate medical attention to rule out cardiac issues.

This is for informational purposes only. Please consult a healthcare professional for proper diagnosis.

Recommendations

URGENT: These symptoms may require immediate medical attention. Please contact emergency services or visit the nearest emergency room.



Intelligent Healthcare Assistant powered by IBM Watson

Patient Chat

Disease Prediction

Treatment Plans

Health Analytics

Personalized Treatment Plans

Select condition for treatment plan:

Asthma

Additional information about your condition:

coughing, wheezing, shortness of breath, and chest tightness.

Generate Treatment Plan

Next Steps

- Schedule follow-up appointment
- Monitor symptoms daily
- Take medications as prescribed
- Maintain healthy lifestyle

Reminders

Set medication reminder:

11:11

Set Reminder

Treatment Plan for Asthma

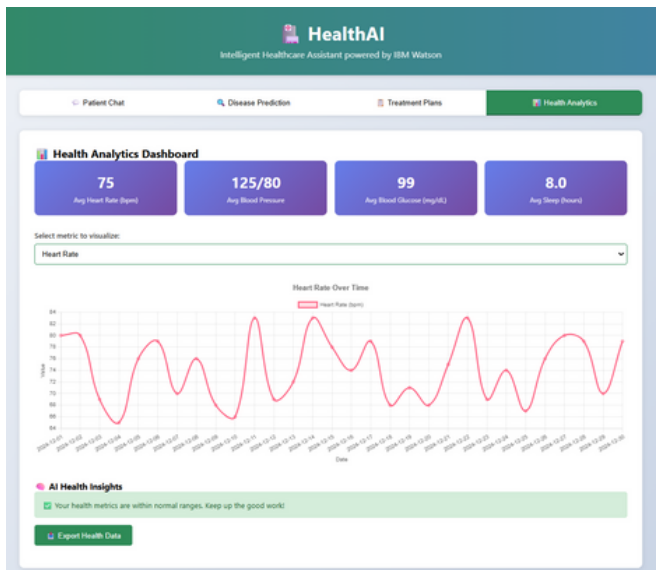
Please specify the condition for a personalized treatment plan.

Treatment Progress Tracker

Medication Adherence: 82%

Lifestyle Changes: 54%

Symptom Improvement: 81%



Healthcare Chat Assistant

You: What are the symptoms of high blood pressure?

HealthAI: Based on the symptoms you've described:

This is for informational purposes only. Please consult a healthcare professional for proper diagnosis.

8. ADVANTAGES & DISADVANTAGES

Advantages:

- ☑️ ✅ 24/7 Accessibility: Users can access healthcare assistance anytime without waiting for a doctor.
- ☑️ ✅ AI-Powered Responses: Quick and intelligent answers using IBM Granite enhance user experience.
- ☑️ ✅ Early Disease Prediction: Helps in identifying potential health issues at an early stage.
- ☑️ ✅ Modular System: Divided into four independent modules for better organization and usability.
- ☑️ ✅ User-Friendly Interface: Built using Streamlit, it provides a simple and intuitive experience.
- ☑️ ✅ Cost-Effective: Reduces the need for continuous human supervision in basic healthcare queries.

Disadvantages:

- ☑️ ❌ Not a Replacement for Doctors: Cannot replace actual medical consultation or diagnosis.
- ☑️ ❌ Depends on Internet Connection: Requires stable internet to function effectively.
- ☑️ ❌ Limited to Pretrained Knowledge: IBM Granite model may not always be updated with the latest medical information.
- ☑️ ❌ Security & Privacy: Requires strict handling of user data for ethical and legal compliance.

9. CONCLUSION

The HEALTHAI project demonstrates how generative AI, specifically IBM Granite, can be effectively integrated into healthcare applications. By providing intelligent responses to user queries, disease prediction, treatment suggestions, and health analytics, this system can assist users in managing their health proactively. Though it is not a substitute for professional medical advice, it acts as a supportive tool that can bridge the gap between users and healthcare information in real time.

10. FUTURE SCOPE

- ☑️ 🏥 Integration with Real Medical Records: In future, the system can be connected to Electronic Health Records (EHR) for more personalized responses.
- ☑️ 📱 Mobile App Development: A dedicated mobile version can improve accessibility on smartphones.
- ☑️ 📊 More Advanced AI Models: Upgrading to future IBM Granite versions or fine-tuning with medical datasets for better accuracy.
- ☑️ 🌐 Multi-Language Support: Expanding to regional languages can make it more inclusive.
- ☑️ 🔒 Enhanced Security Measures: Implementing data encryption and secure login to protect user privacy.



- 👨‍⚕️🗣️ Doctor Integration: Providing live chat features with real doctors or teleconsultation options.

Source Code(if any)

```
health_project > cd healthdashboard && ...
2 <html lang="en">
300 <body>
400 <div class="container">
551 <div id="treatment" class="tab-content">
554 <div style="display: grid; grid-template-columns: 2fr 1fr; gap: 2rem;>
612 </div>
613 </div>
614 <div id="analytics" class="tab-content">
615 <div> 📊 Health Analytics Dashboard/h2>
616
617 <div class="metrics-grid">
618 <div class="metric-card">
619 <div class="metric-value" id="avgheartRate">72</div>
620 <div class="metric-label">Avg Heart Rate (bpm)</div>
621 </div>
622 <div class="metric-card">
623 <div class="metric-value" id="avgbloodPressure">120/80</div>
624 <div class="metric-label">Avg Blood Pressure</div>
625 </div>
626 <div class="metric-card">
627 <div class="metric-value" id="avgglucose">95</div>
628 <div class="metric-label">Avg Blood Glucose (mg/dL)</div>
629 </div>
630 <div class="metric-card">
631 <div class="metric-value" id="avgsleep">7.5</div>
632 <div class="metric-label">Avg Sleep (hours)</div>
633 </div>
634 </div>
635
636 <div class="form-group">
637 <label for="chartType">Select metric to visualize:</label>
638 <select id="chartType" onchange="updateChart()">
639 <option value="heartRate">Heart Rate</option>
640 <option value="bloodPressure">Blood Pressure</option>
641 <option value="bloodGlucose">Blood Glucose</option>
642 <option value="weight">Weight</option>
643 <option value="sleep">Sleep Hours</option>
644 </select>
645 </div>
646
647 <div class="chart-container">
648 <canvas id="healthChart"></canvas>
649 </div>
```

```
health_project > cd healthdashboard && ...
2 <html lang="en">
300 <body>
<script>
1235 function exportHealthData() {
1236 const csvContent = "data:text/csv;charset=utf-8," +
1237 "Date,Heart Rate,Systolic BP,Diastolic BP,Blood Glucose,Weight,Sleep Hours\n" +
1238 healthData.map(d => `${d.date},${d.heartRate},${d.systolicBP},${d.diastolicBP},${d.bloodGlucose},${d.weight},${d.sleep}`).join("\n");
1239
1240 const encodedUri = encodeURIComponent(csvContent);
1241 const link = document.createElement("a");
1242 link.setAttribute("href", encodedUri);
1243 link.setAttribute("download", "health_data.csv");
1244 document.body.appendChild(link);
1245 link.click();
1246 document.body.removeChild(link);
1247 }
1248
1249 document.addEventListener("DOMContentLoaded", function() {
1250
1251 const tabButtons = document.querySelectorAll('.tab-button');
1252 tabButtons.forEach(button, index) => {
1253 button.addEventListener('click', function() {
1254 const tabs = ['chat', 'prediction', 'treatment', 'analytics'];
1255 showTab(tabs[index]);
1256 });
1257 });
1258
1259 const avgHeartRate = Math.round(healthData.reduce((sum, d) => sum + d.heartRate, 0) / healthData.length);
1260 const avgSystolic = Math.round(healthData.reduce((sum, d) => sum + d.systolic, 0) / healthData.length);
1261 const avgDiastolic = Math.round(healthData.reduce((sum, d) => sum + d.diastolic, 0) / healthData.length);
1262 const avgGlucose = Math.round(healthData.reduce((sum, d) => sum + d.bloodGlucose, 0) / healthData.length);
1263 const avgSleep = (healthData.reduce((sum, d) => sum + d.sleep, 0) / healthData.length).toFixed(1);
1264
1265 document.getElementById("avgheartRate").textContent = avgHeartRate;
1266 document.getElementById("avgbloodPressure").textContent = `${avgSystolic}/${avgDiastolic}`;
1267 document.getElementById("avgglucose").textContent = avgGlucose;
1268 document.getElementById("avgsleep").textContent = avgSleep;
1269
1270 addMessageToChat('Hello! I\'m your HealthAI assistant. How can I help you with your health questions today?', 'ai');
1271 </script>
1272 </body>
1273 </html>
```

```
health_project > cd healthdashboard && ...
1 <doc><html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>HealthAI - Intelligent Healthcare Assistant</title>
7 <script src="https://cdn.jsdelivr.net/npm/chart.js@3.9.3/chart.min.js"></script>
8 <style>
9 {
10 margin: 0;
11 padding: 0;
12 box-sizing: border-box;
13 }
14
15 body {
16 font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
17 background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
18 min-height: 100vh;
19 }
20
21 .header {
22 background: linear-gradient(90deg, #2e8b57 0%, #2e8b57 49.49%, #a6c9ec 49.49%, #a6c9ec 100%);
23 color: white;
24 padding: 2rem;
25 text-align: center;
26 box-shadow: 0 4px 6px #000;
27 }
28
29 .header h1 {
30 font-size: 2.5rem;
31 margin-bottom: 0.5rem;
32 }
33
34 .header p {
35 font-size: 1.2rem;
36 opacity: 0.9;
37 }
38
39 .container {
40 max-width: 1400px;
41 margin: 0 auto;
42 padding: 2rem;
43 }
44
45 </style>
```

```
2 <html lang="en">
300 <body>
<div>
400 <div class="container">
401 <div class="tabs">
402 <button class="tab-button active" onclick="showTab('chat')> 🗣️ Patient Chat</button>
403 <button class="tab-button" onclick="showTab('prediction')> 📊 Disease Prediction</button>
404 <button class="tab-button" onclick="showTab('treatment')> 💊 Treatment Plans</button>
405 <button class="tab-button" onclick="showTab('analytics')> 📈 Health Analytics</button>
406 </div>
407 <div id="chat" class="tab-content active">
408 <div> 🗣️ Healthcare Chat Assistant/h2>
409 <div class="chat-container" id="chatContainer"></div>
410 <div class="form-group">
411 <input type="text" id="chatInput" placeholder="Ask me anything about your health..." onkeypress="handleChatKeyPress(event)">
412 </div>
413 <button class="btn" onclick="sendMessage()">Send</button>
414 <button class="btn btn-secondary" onclick="clearChat()">Clear Chat</button>
415 </div>
416 <div> 📝 Sample Questions/h3>
417 <div class="sample-questions">
418 <div class="sample-question" onclick="askSampleQuestion('what are the symptoms of high blood pressure?')>
419 What are the symptoms of high blood pressure?
420 </div>
421 <div class="sample-question" onclick="askSampleQuestion('how can I improve my sleep quality?')>
422 How can I improve my sleep quality?
423 </div>
424 <div class="sample-question" onclick="askSampleQuestion('what foods should I avoid with diabetes?')>
425 What foods should I avoid with diabetes?
426 </div>
427 <div class="sample-question" onclick="askSampleQuestion('when should I see a doctor for persistent headaches?')>
428 When should I see a doctor for persistent headaches?
429 </div>
430 </div>
431 </div>
432 <div id="prediction" class="tab-content">
433 <div> 📊 Disease Prediction System/h2>
434 <div class="alert alert-warning">
435 <strong>⚠️ Medical Disclaimer:</strong> This tool provides general information only and should not replace professional medical advi
436 Always consult healthcare professionals for proper diagnosis and treatment.
437 </div>
438 </div>
```

```

healthai_project > healthai.html.html > ...
<html lang="en">
<body>
<script>
1235     function exportHealthData() {
1236         const csvContent = "data:text/csv;charset=utf-8," +
1237             "date,Heart Rate,Systolic BP,Diastolic BP,Blood Glucose,Weight,Sleep Hours\n" +
1238             healthData.map(d => `${d.date},${d.heartRate},${d.systolicBP},${d.diastolicBP},${d.bloodGlucose},${d.weight},${d.sleep}`).join("\n");
1239         const encodedUri = encodeURI(csvContent);
1240         const link = document.createElement("a");
1241         link.setAttribute("href", encodedUri);
1242         link.setAttribute("download", "health_data.csv");
1243         document.body.appendChild(link);
1244         link.click();
1245         document.body.removeChild(link);
1246     }
1247
1248     document.addEventListener("DOMContentLoaded", function() {
1249
1250         const tabButtons = document.querySelectorAll(".tab-button");
1251         tabButtons.forEach((button, index) => {
1252             button.addEventListener("click", function() {
1253                 const tabs = ["chat", "prediction", "treatment", "analytics"];
1254                 showTab(tabs[index]);
1255             });
1256         });
1257
1258         const avgHeartRate = Math.round(healthData.reduce((sum, d) => sum + d.heartRate, 0) / healthData.length);
1259         const avgSystolic = Math.round(healthData.reduce((sum, d) => sum + d.systolicBP, 0) / healthData.length);
1260         const avgDiastolic = Math.round(healthData.reduce((sum, d) => sum + d.diastolicBP, 0) / healthData.length);
1261         const avgGlucose = Math.round(healthData.reduce((sum, d) => sum + d.bloodGlucose, 0) / healthData.length);
1262         const avgSleep = (healthData.reduce((sum, d) => sum + d.sleep, 0) / healthData.length).toFixed(1);
1263
1264         document.getElementById("avgHeartRate").textContent = avgHeartRate;
1265         document.getElementById("avgBloodPressure").textContent = `${avgSystolic}/${avgDiastolic}`;
1266         document.getElementById("avgGlucose").textContent = avgGlucose;
1267         document.getElementById("avgSleep").textContent = avgSleep;
1268
1269         addMessageToChat("Hello! I'm your HealthAI assistant. How can I help you with your health questions today?", "ai");
1270     });
1271 </script>
1272 </body>
1273 </html>

```

Dataset Link

GitHub & Project Demo Link

Both the dataset and the project demo video are uploaded to the GitHub repository and can be accessed via the following link:

<https://github.com/Likitha456/Health-ai>