PROJECT REPORT

# Title: HealthAI: Intelligent Healthcare Assistant Using IBM Granite

**Team ID : LTVIP2025TMID33687**

1. **INTRODUCTION**

### Project Overview

**HealthAI** is an intelligent healthcare assistant powered by IBM Watson Machine Learning and Generative AI. It aims to enhance accessibility to reliable medical insights by offering users accurate, personalized, and data-driven guidance. Users can interact with HealthAI through a responsive single-page web application, gaining support for diagnosis, treatment, and daily health tracking.

### Purpose

The purpose of HealthAI is to serve as a virtual healthcare companion, helping users understand symptoms, receive predictive diagnoses, and access evidence-based treatment suggestions through a user-friendly interface.

# IDEATION PHASE

### Problem Statement

In today’s digital world, people often turn to the internet for health-related queries, but struggle with information overload, inconsistent sources, and lack of personalization. **HealthAI** addresses this gap by providing trusted, AI-powered medical insights tailored to the user’s inputs and conditions.

### Empathy Map Canvas

The empathy map helps understand the target users' mindset and expectations when interacting with a virtual health assistant like HealthAI.

| Category | Description |
| --- | --- |
| Says | "I want quick answers to my health questions.""Is this symptom serious?" |
| Thinks | "Can I trust this information?""Will this help me avoid a hospital visit?" |
| Does | Searches symptoms online. Asks friends or family for medical advice |
| Feels | Anxious about symptoms. Uncertain about next steps |
| Pains | Conflicting online information. Lack of access to immediate healthcare |
| Gains | Reliable AI suggestions. Quick advice at home. Health tracking insights |

### Brainstorming

The team explored various ideas such as mental health bots, COVID symptom trackers, and AI nutritionists before settling on an all-in-one intelligent assistant with chat, prediction, and analytics powered by IBM Watson.

# REQUIREMENT ANALYSIS

## Customer Journey Map

The customer journey in the HealthAI application follows a streamlined, user-friendly process:

* + 1. **Start Application** – The user opens the HealthAI web app (built using HTML, CSS, JavaScript).
    2. **Input Health Query** – The user types a symptom-related question or fills out a symptom form.
    3. **Receive Diagnosis** – The AI model (IBM Granite via FastAPI) responds with likely conditions and guidance.
    4. **Review Treatment Plan** – The app displays a structured, AI-generated treatment recommendation.
    5. **View Health Analytics** – Users visualize vital signs like heart rate, blood pressure, and glucose with Chart.js insights.
    6. **End/Next Action** – The user can reset the session, consult a doctor, or continue exploring HealthAI features.

### Session Requirements

* Real-time symptom input via chat
* Prediction based on user profile
* Personalized treatment plans
* Visualization of health metrics

### Data Flow Diagram

* 1. ***Technology Stack***
* **Frontend:** HTML, CSS, JavaScript (Single Page Application)
* **Backend:** Python (FastAPI)
* **AI Service:** IBM Watson ML (Granite 13B Instruct v2)
* **Visualization:** Chart.js
* **Environment Management:** virtualenv + .env (python-dotenv)

1. **PROJECT DESIGN**

### Problem-Solution Fit

People need quick, understandable, and trustworthy health information. HealthAI fulfills this by using medical LLMs for better accuracy.

### Proposed Solution

A layered web application consisting of a dynamic UI (HTML/JS), application logic (FastAPI), and an AI backend (IBM Granite). It guides users from symptom input to personalized treatment suggestions and visual analytics.

### Solution Architecture

* **UI Layer:** Chat interface, forms, and analytics dashboard (SPA using HTML/CSS/JS)
* **Application Logic:** main.py handles routing, input processing and API calls.
* **Helper Logic:** Utility functions for prompt formatting and session management
* **AI Layer:** IBM Granite 13B Instruct v2 connected via secure FastAPI endpoints using .env credentials

# PROJECT PLANNING & SCHEDULING

### Project Planning

| **Week Duration** | **Dates** | **Activities** |
| --- | --- | --- |
| Week 1 | June 14 – June 20 | Idea finalization, architecture planning, frontend UI (HTML/JS) |
| Week 2 | June 22 – June 29 | FastAPI backend + AI integration, testing, and documentation |

This two-week schedule allowed the team to focus on clear milestones and complete the HealthAI project within the planned timeline.

# FUNCTIONAL AND PERFORMANCE TESTING

### Performance Testing

* Unit Testing: Model loading, session data handling
* Integration Testing: End-to-end chat → prediction → plan → visualization
* Manual Testing: Validated each module with sample health queries
* Error Handling: Graceful handling of missing/invalid inputs, API key issues

# RESULTS

### Output Screenshots

* Screenshot of chat feature
* Screenshot of prediction result
* Screenshot of treatment plan generator
* Screenshot of treatment plan generator
* Screenshot of health tips generator



* Screenshot of health analytics chart

# ADVANTAGES & DISADVANTAGES

**Advantages:** - Easy to use - AI-powered recommendations - Visualization of health data

**Disadvantages:** - No real-time data integration - No authentication or user profiles - General- purpose AI model



# CONCLUSION

**HealthAI** successfully demonstrates the application of AI in healthcare by combining a simple and intuitive user interface with powerful backend intelligence using IBM WatsonX Granite. Developed as a modular web-based application, it offers chat-based health guidance, symptom- based predictions, and treatment suggestions. While currently a prototype, it provides a strong foundation for real-world deployment with future enhancements such as real-time health data integration, secure user authentication, and multilingual support.

# FUTURE SCOPE

* Add secure user login and profile management
* Use real anonymized patient databases for prediction accuracy
* Integrate with wearable devices like smartwatches for live health monitoring
* Fine-tune IBM Granite model on domain-specific medical data
* Add features like emergency alerts, appointment booking, and multilingual support

# APPENDIX

* **Source Code Files:** main.py,index.html,styles.css,app.js,.env

