HealthAI – Intelligent Healthcare Assistant using IBM Granite

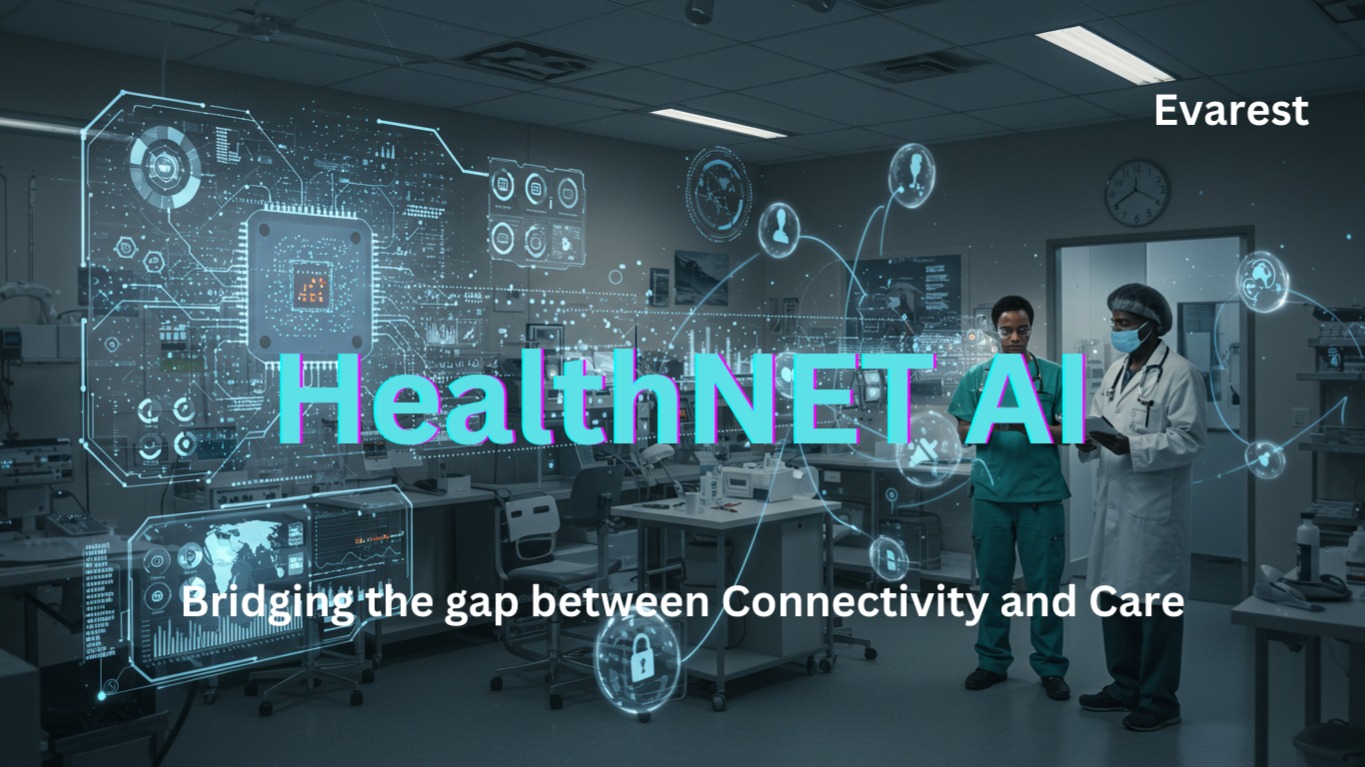
Submitted by: Bangaru Tejaswini

Reg No: 22KP1A0509

College: NRI Institute of Technology

Internship Platform: SmartInternz with IBM

Project Guide(s): Rithvik / Aravind Reddy



# *1. Abstract*

HealthAI is a generative AI-based healthcare assistant built using IBM Granite models. It helps users interact with a chatbot to ask medical questions, predict conditions based on symptoms, and receive treatment guidance. The system leverages Python, Streamlit, and Hugging Face API to offer accessible health services with a simple user interface.

HealthAI is a lightweight, intelligent healthcare assistant that harnesses the power of IBM’s Granite large language model via Hugging Face. The system is designed to assist patients by answering common health-related queries, predicting potential health conditions based on symptoms, and suggesting evidence-based treatment plans. HealthAI aims to provide preliminary support to users—especially those in rural or underserved areas—where access to immediate medical care is limited. Built using Python and Streamlit, the solution ensures accessibility through a simple user interface and real-time AI interaction. While it is not intended to replace professional medical advice, it can help users make informed decisions and encourage timely consultations with healthcare providers. This project showcases how generative AI can be applied responsibly in healthcare, offering users empathetic, informative, and ethical digital assistance.

# *2. Introduction*

With the rising demand for AI in healthcare, there is a growing need for intelligent assistants that can provide timely and context-aware health responses. HealthAI bridges this gap by delivering three core functionalities: symptom-based prediction, patient Q&A, and treatment planning through a lightweight Streamlit app.

The integration of Artificial Intelligence (AI) in healthcare has brought transformative changes to the way medical services are delivered, accessed, and experienced. With the rise of generative AI models like IBM Granite, it has become feasible to provide intelligent, real-time responses to user queries, thus enhancing accessibility and engagement in health services. However, many users, especially in remote or resource-limited settings, still lack access to instant medical guidance for basic health concerns.

HealthAI addresses this gap by offering a conversational AI-based web application that acts as a virtual healthcare assistant. It provides three main functionalities: (1) answering user health-related questions, (2) predicting possible health conditions based on symptoms, and (3) generating customized treatment recommendations. Built using Python, Streamlit, and IBM Granite APIs via Hugging Face, the application emphasizes ease of use, accessibility, and responsible AI practices.

This project demonstrates how generative AI can be used for public good in the healthcare domain by enhancing user awareness, promoting preventive care, and supporting better decision-making in non-emergency situations.

# *3. Problem Statement*

Many people delay consulting doctors for minor doubts. In rural or underserved areas, access to qualified practitioners is limited. There is no easy, AI-driven solution for initial health support. HealthAI was developed to act as a first-level intelligent health assistant.

Access to timely and accurate health information remains a significant challenge for many individuals, particularly those living in rural or underserved communities. In most cases, people are unsure about whether their symptoms require medical attention or not. As a result, they either delay visiting a doctor or rely on unverified online sources that may provide misleading or even dangerous advice.

Traditional healthcare systems are often overwhelmed and cannot always provide real-time responses to every patient query. Additionally, minor symptoms or general doubts are frequently ignored due to the effort, time, or cost required to consult a physician. This can lead to worsening of conditions or unnecessary panic. Moreover, the shortage of qualified healthcare professionals in many regions further exacerbates this problem.

In today’s digital world, where mobile and internet access is rapidly increasing, there is a critical need for an intelligent system that can offer basic health support in an accessible and user-friendly manner. Such a system should be capable of understanding patient input, offering helpful guidance, and most importantly, encouraging users to seek medical care when necessary—all while ensuring the responses are safe, ethical, and evidence-based.

The HealthAI project aims to solve this gap by building a Generative AI-powered assistant that can provide instant, conversational responses to users. It is not a replacement for doctors, but rather a supportive tool that empowers users with information, bridges communication gaps, and supports better decision-making. With a focus on simplicity, ethical AI use, and public access, HealthAI can make intelligent healthcare guidance more widely available and impactful

# *4. Objectives*

- Build a simple and intuitive healthcare assistant app  
- Integrate IBM Granite via Hugging Face API  
- Predict health conditions from symptoms  
- Generate treatment guidance based on AI  
- Offer an interactive chatbot-style experience

The main goal of the HealthAI project is to design and develop an AI-powered healthcare assistant that can provide users with accurate and timely health-related support through a conversational interface. The specific objectives are:

* **To build a simple and intuitive healthcare assistant application**  
  The application should have a clean and user-friendly interface so that even non-technical users can easily interact with it. The design must focus on simplicity, responsiveness, and accessibility across devices (laptops, mobiles, etc.).
* **To integrate IBM Granite Large Language Model via Hugging Face API**  
  Leverage the powerful IBM Granite generative AI model hosted on Hugging Face to generate high-quality, natural-language responses. This integration allows the app to process user input and deliver reliable, informative, and context-aware outputs.
* **To predict health conditions based on user symptoms and personal data**  
  Users can input their symptoms, age, gender, medical history, and vitals. The system uses AI prompting to analyze this data and predict the top three most likely health conditions, with likelihood levels and explanations.
* **To generate personalized treatment guidance using AI**  
  Based on the predicted or provided health condition, the system generates a treatment plan that includes medication suggestions, dietary advice, lifestyle modifications, follow-up testing, and mental health considerations. These plans are non-diagnostic and designed to offer general awareness and direction.
* **To offer an interactive chatbot-style experience for better user engagement**  
  Create a conversational environment where users feel like they are interacting with a helpful assistant. The chatbot interface must be interactive, empathetic, and clear in communication, helping to simulate a basic consultation experience.

# *5.* 🛠️*Tools and Technologies Used*

IBM Granite (Hugging Face), Python, Streamlit, Pyngrok, Google Colab, GitHub

| **Technology / Tool** | **Purpose & Where It Was Used** |
| --- | --- |
| **IBM Granite (via Hugging Face API)** | IBM Granite is a state-of-the-art generative AI model. It was used to generate natural, accurate, and evidence-based responses. Integrated during the **Implementation** phase through Hugging Face API calls using requests and custom prompt templates. |
| **Hugging Face Inference API** | Hosted the IBM Granite model. Used to securely send prompts and receive AI-generated replies in JSON format. This was applied in **Implementation** and **Testing** phases. |
| **Python** | The core programming language for the backend logic. Used in all phases including **Design**, **Implementation**, and **Testing** to build the logic for symptom analysis, prompt creation, and response handling. |
| **Streamlit** | A fast and simple open-source Python framework used to build the web application UI. Applied in the **Design**, **Implementation**, and **Deployment** stages to create an interactive user interface. |
| **Pyngrok** | Used for creating a temporary **public URL** so the Streamlit app could be accessed and tested by anyone online. Mainly used during the **Testing** and **Deployment** stages. |
| **Google Colab** | An online IDE used to write, run, and test the code without local setup. It was especially useful during the **Implementation** and **Debugging** phases when checking API responses and deploying app components. |
| **GitHub** | Used for version control, sharing the project repository, uploading the code, presentation, and report. Used at the **Deployment** and **Documentation** stages for project submission. |

# *6. System Architecture*

User Input --> Streamlit Interface --> Prompt Template --> Hugging Face API --> IBM Granite Model --> Response to User

The HealthAI system follows a straightforward yet effective architecture that connects user input to a powerful generative AI model, processes the data intelligently, and provides a meaningful output through a web-based interface. The architecture flow is as follows:

User Input → Streamlit Interface → Prompt Template → Hugging Face API → IBM Granite Model → Response to User

Here’s how each stage works:

1. **User Input**
   * The user provides input through the web interface. This could be a health-related question, a list of symptoms, or a known medical condition along with personal details like age, gender, or medical history.
2. **Streamlit Interface**
   * Streamlit is used to design the interactive web UI. It captures the user’s input through text boxes, dropdowns, and buttons and displays the output in real-time. This part ensures the app is easy to use for all users, including non-technical people.
3. **Prompt Template**
   * Based on the type of query (ask question, predict condition, or get treatment plan), a custom prompt is created. These prompts are written in natural language and structured to extract accurate, safe, and helpful responses from the AI model.
   * Example:  
     For prediction – “As a healthcare AI assistant, predict potential health conditions based on the following patient data…”
4. **Hugging Face API**
   * The prompt is sent securely to the Hugging Face Inference API, which hosts the IBM Granite model. The API handles the communication between the app and the AI model, ensuring smooth, real-time interaction.
5. **IBM Granite Model**
   * This is the core intelligence of the system. The Granite model reads the prompt, understands the medical context, and generates an empathetic, evidence-based, human-like response. It ensures answers are informative yet cautious (without making diagnoses).
6. **Response to User**
   * Finally, the response received from the model is shown back to the user via the Streamlit interface. The output is clearly formatted and easy to understand, including recommendations or suggestions for seeking further medical attention if needed.

# *7. 🔧 Methodology*

7.1 SDLC Applied:  
- Requirement Analysis  
- Design  
- Implementation  
- Testing  
- Deployment

The development of the HealthAI application followed the **Software Development Life Cycle (SDLC)** model to ensure structured, phased, and efficient project completion. The methodology is divided into two parts: the SDLC approach and the key functionalities implemented.

**7.1 SDLC Applied:**

**1. Requirement Analysis**  
At this stage, the needs and expectations of the end users were studied. We identified that users needed:

* A way to ask health questions and receive AI-generated advice
* A method to input symptoms and receive likely conditions
* A system that suggests treatments in a clear, understandable way  
  We finalized that the assistant must be simple, text-based, and accessible from any device.

**2. Design**  
The structure of the web app was planned, including:

* The user interface (tabs for each feature)
* The flow of user input to AI output
* The API communication with Hugging Face  
  Prompt templates were also carefully designed for each feature to ensure accurate model responses.

**3. Implementation**  
The entire logic was developed in Python using Google Colab. The IBM Granite model was integrated through the Hugging Face Inference API. Streamlit was used to build the user interface, and functionalities were split into three major sections: Q&A, prediction, and treatment.

**4. Testing**  
Each feature was tested with real and sample inputs to ensure the outputs were relevant, ethical, and clear. Edge cases like incomplete inputs, excessive symptoms, and unusual queries were tested. Ngrok was used to test the app on public links.

**5. Deployment**  
The app was successfully deployed using pyngrok, making it accessible via a public link. The project was also uploaded to GitHub for submission and future reference.

**7.2 Functionalities:**

**1. Ask Health Question**  
This feature allows users to type any health-related question. The AI responds with an empathetic and educational answer based on current medical knowledge. It avoids giving direct diagnoses and advises users to consult doctors when necessary.

**2. Predict Condition**  
Users enter symptoms, age, gender, medical history, and vitals. The AI analyzes this data and predicts the top 3 most likely health conditions. It provides a brief explanation and categorizes likelihood as High, Medium, or Low.

**3. Treatment Plan Generator**  
When a user provides a known condition (e.g., Asthma), the app generates a detailed treatment suggestion. This includes possible medications, diet advice, physical activity recommendations, and mental health tips—tailored to the user's profile

# *8. Screenshots*

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# *9. 📋 Sample Output*

Condition: Diabetes  
AI Suggestion:  
- Medication: Metformin 500mg  
- Diet: Low sugar, high fiber  
- Exercise: 30 mins walk daily  
- Monitoring: HbA1c every 3 months

To demonstrate how the HealthAI assistant works, we present a **realistic simulation** of its response when a user inputs a common chronic health condition – **Type 2 Diabetes Mellitus** – along with basic personal information.

**🩺 Input to the System**

* **Condition**: Diabetes Mellitus Type 2
* **Age**: 45
* **Gender**: Female
* **Medical History**: Hypertension
* **Additional Info**: Mild fatigue and frequent urination

**🤖 AI-Generated Response by IBM Granite Model**

**1. Medication Guidance**  
The AI recommends the most widely prescribed initial therapy for Type 2 Diabetes:

* **Metformin 500mg**, taken once daily with meals
* This drug helps reduce liver glucose production and improves insulin sensitivity
* The AI also reminds the user to monitor for side effects such as nausea or stomach upset, which are common during the initial dose period
* If blood glucose levels remain uncontrolled, the user is advised to consult a physician for combination therapy or insulin

**2. Dietary Recommendations**  
The assistant provides diet advice tailored to diabetic care:

* **Low-sugar, high-fiber diet** is emphasized to control blood sugar spikes
* Includes whole grains (like oats, brown rice), legumes, green leafy vegetables, and nuts
* Recommends avoiding foods with high glycemic index like white bread, sweets, sugar-heavy beverages, and processed snacks
* Encourages regular meal timings and portion control for blood glucose stability

**3. Physical Activity Guidelines**  
Exercise is a core component of diabetes management, and the AI recommends:

* **Brisk walking for 30–45 minutes daily**, ideally post meals
* Suggests incorporating moderate activities like cycling, yoga, or swimming
* Advises consistency rather than intensity and reminds that even light walking can improve insulin sensitivity
* Warns to avoid overexertion, especially in older users with additional conditions

**4. Monitoring and Medical Follow-Up**  
To prevent complications, the AI includes:

* **Regular blood glucose monitoring**: Fasting and post-meal sugar checks at home
* **HbA1c testing every 3 months** to track long-term glucose control
* Encourages the user to maintain a **health log or diary** of food, medication, and blood sugar readings
* Advises regular eye check-ups and kidney function tests to monitor for diabetic complications

**5. Mental and Lifestyle Considerations**  
The AI includes lifestyle wellness suggestions, showing its empathetic design:

* **Stress management** through breathing exercises, mindfulness, or light yoga
* Promotes **7–8 hours of quality sleep**, as poor sleep worsens insulin resistance
* Recommends avoiding smoking and excessive alcohol, both of which interfere with blood sugar control
* Encourages social interaction and regular routines to boost emotional health

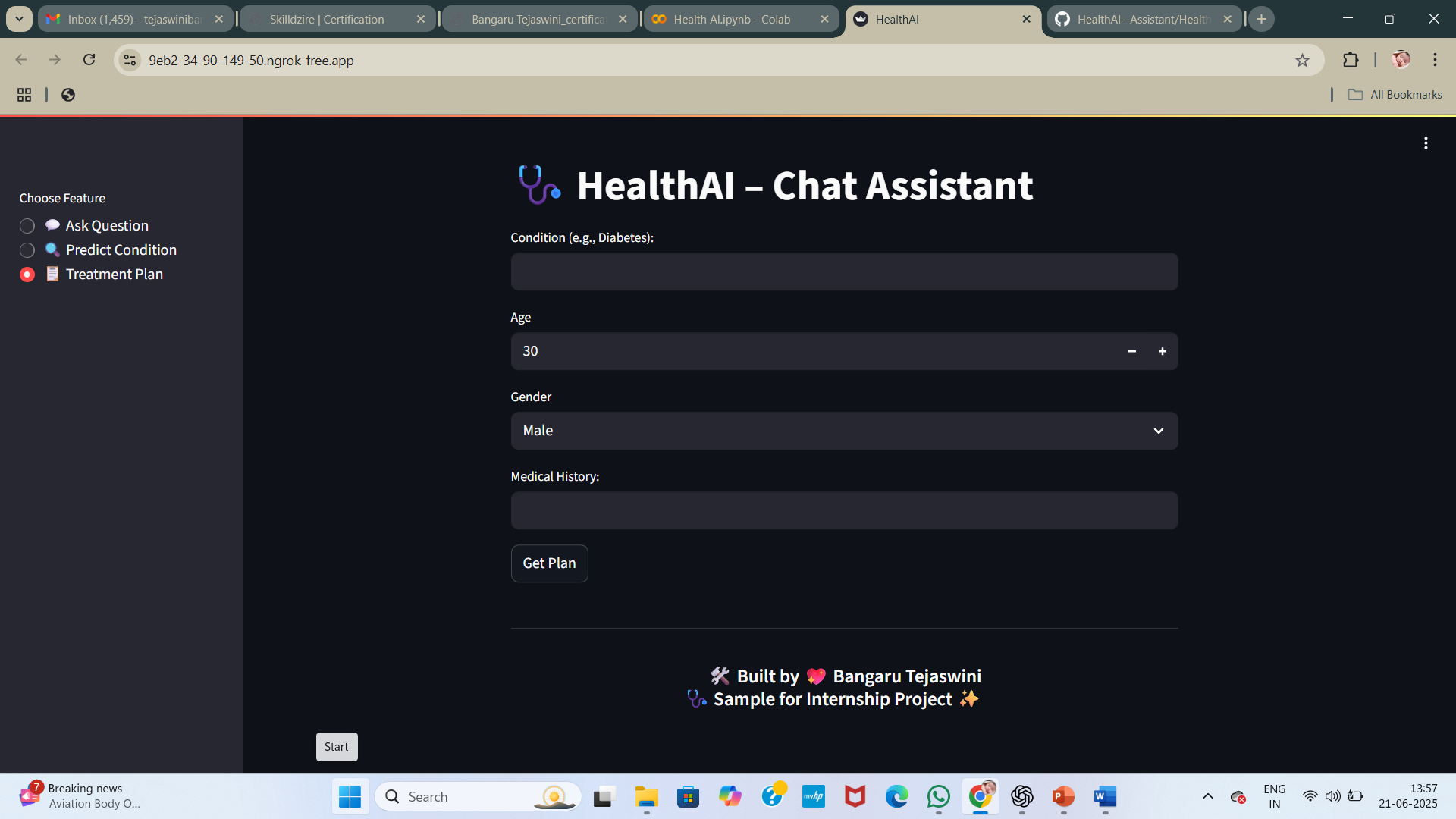
**⚠️ Disclaimer (Ethical Response)**

“This AI assistant provides general guidance and educational suggestions. It is **not a substitute for a licensed physician**. For medication prescriptions or health concerns, always consult your doctor or a certified healthcare provider.”

This level of output shows that the HealthAI assistant does not just generate short replies, but can produce **structured, helpful, and medically aligned** guidance that promotes **awareness and informed decision-making**.

A screenshot of a computer

AI-generated content may be incorrect.



# *10. Advantages*

- Always available (24x7)  
- Easy to use  
- Works from mobile/desktop  
- Non-technical users can understand output  
- Powered by latest IBM Granite model

**✅ 10. Advantages (Detailed Explanation)**

The HealthAI system brings several benefits, especially in the context of modern healthcare needs, where speed, accessibility, and reliability are crucial. Below is a breakdown of its main advantages:

**1. 24x7 Availability – Always Ready to Assist**  
Unlike traditional healthcare services that operate within specific hours, HealthAI is **available anytime** to users across the globe. Whether it's early morning or late night, the system is accessible and responsive—making it highly reliable in urgent or off-hour scenarios.

**2. Simple and User-Friendly Interface**  
The application is built using **Streamlit**, which ensures a clean and minimalistic interface. Users can easily interact with the system without any technical training. All options are neatly organized into tabs (Ask Question, Predict Condition, Treatment Plan), making navigation smooth even for first-time users.

**3. Device Independent – Access from Mobile or Desktop**  
HealthAI is hosted via a web interface, meaning it can be accessed through any device with an internet connection—**laptops, tablets, or smartphones**. There’s no need to install any software, which adds to its portability and convenience.

**4. Designed for Non-Technical Users**  
One of the key design goals was inclusivity. All AI-generated outputs are written in **easy-to-understand language**. The system avoids technical medical jargon and presents explanations in simple, conversational terms. This ensures that even rural or less-educated users can benefit from the app.

**5. Powered by IBM Granite – Trusted AI Engine**  
At the core of HealthAI is the **IBM Granite 13B model**, hosted securely on **Hugging Face**. This foundation ensures:

* High-quality, evidence-based responses
* Ethical and responsible replies
* Fast and scalable inference using state-of-the-art generative AI  
  This brings **credibility and intelligence** to every interaction.

**6. Encourages Preventive Health Behavior**  
By making health information accessible, the app encourages users to seek medical help when needed. It also provides awareness about symptoms, diseases, and treatments in an early stage—helping reduce future complications.

**7. Lightweight and Fast Deployment**  
Since the app runs on Streamlit and uses cloud APIs, it doesn't require heavy infrastructure or high-end systems to function. This makes it easy to **deploy, share, and maintain**, especially in academic or pilot-scale healthcare projects.

# *11. ⚠️ Limitations*

- Not a replacement for professional medical advice  
- Limited to text-based input/output  
- Depends on API token limits and internet access

**⚠️ 11. Limitations (Detailed Explanation)**

While HealthAI offers many advantages, it also has a few important limitations that must be considered:

**1. Not a Replacement for Certified Medical Professionals**  
The AI assistant is designed only to provide **general awareness and guidance**. It cannot replace a real doctor’s diagnosis, clinical tests, or prescriptions. All responses are based on publicly available medical knowledge and may not be specific to the user’s medical history. Therefore, it is crucial that users consult a certified healthcare provider before making any medical decisions.

**2. Text-Based Interface Only**  
Currently, the application only supports **text-based input and output**, which may limit interaction for users who prefer voice communication or have literacy challenges. Features like voice input/output or multi-language support are not yet available, but are proposed as future enhancements.

**3. Dependent on API Limits and Internet Connectivity**  
The system depends on external APIs (Hugging Face for IBM Granite), which come with **token usage limits** in free or academic tiers. Once limits are reached, the application cannot generate further responses unless upgraded. Additionally, the app requires a **stable internet connection**, making it less accessible in areas with poor connectivity.

**4. Limited Medical Context**  
The AI does not have access to **real-time health records or patient history**, which can limit the accuracy of its predictions or suggestions. It relies only on the information the user provides at runtime.

**5. Temporary Deployment via Ngrok**  
In this prototype version, the app is hosted using **Ngrok**, which provides only temporary public links. A permanent web domain and secured hosting would be needed for real-world deployment.

These limitations help clarify that while HealthAI is a powerful supportive tool, it must be used responsibly and with awareness of its boundaries.

# *12.🚀 Future Scope*

- Voice-based interaction  
- Multilingual support (including Telugu)  
- EHR integration  
- Personal user history

**🚀 12. Future Scope (Detailed Explanation)**

HealthAI, in its current prototype form, demonstrates the potential of generative AI in the healthcare sector. However, there are several promising directions for future enhancements that can make the system more powerful, inclusive, and medically impactful.

**1. Voice-Based Interaction**  
Introducing **voice input and output** would significantly improve the app's usability—especially for elderly users, people with disabilities, and users with low literacy levels. Integrating speech-to-text and text-to-speech capabilities would create a more natural and conversational experience.

**2. Multilingual Support (Including Telugu)**  
To reach a wider audience, especially in regional and rural areas, the assistant can be extended to support **multiple Indian languages**, starting with Telugu. This would break language barriers and help users understand medical suggestions in their native language, increasing both trust and accessibility.

**3. Integration with EHR Systems (Electronic Health Records)**  
Future versions of HealthAI can be enhanced to **read and interpret patient health records** (with consent). This would enable the system to provide more **personalized and context-aware advice**, improving its relevance and accuracy in real-world medical situations.

**4. Secure User History and Profile Management**  
Currently, the app does not store any user data. Future enhancements could include a **secure user profile system** where individuals can save their health queries, symptoms, vitals, and previous suggestions for future reference. This feature would enable **long-term health monitoring** and follow-up assistance.

**5. Integration with Appointment Booking Systems**  
HealthAI can be linked to local clinics, hospitals, or telemedicine platforms to allow users to **book appointments directly** after receiving AI guidance—bridging the gap between awareness and action.

**6. AI Model Fine-Tuning for Healthcare Domain**  
The AI model can be further **fine-tuned using medical datasets** to increase its accuracy, empathy, and compliance with ethical guidelines. This could improve the assistant’s clinical relevance while ensuring safe, bias-free communication.

These enhancements would not only make HealthAI more intelligent and responsive but also more useful in **real-world public health applications**.

# *13. 🧾 Conclusion*

HealthAI successfully demonstrates the use of generative AI in delivering quick and helpful health support. It empowers users with knowledge and guidance without replacing medical professionals. The combination of IBM Granite and Streamlit creates an efficient and scalable platform.

**🧾 13. Conclusion (Detailed and Comprehensive)**

The **HealthAI – Intelligent Healthcare Assistant using IBM Granite** project represents a significant step forward in applying generative AI technology to the field of digital healthcare. Throughout the course of this internship project, the application was carefully designed, developed, and tested using a structured SDLC (Software Development Life Cycle) approach to ensure clarity, quality, and completeness.

The project started by identifying a common and critical problem in society: **lack of timely and accessible health guidance** for individuals—especially in rural, semi-urban, or underserved communities. Many people often hesitate to approach doctors for minor doubts or health concerns due to time, cost, or accessibility issues. In response to this gap, HealthAI was envisioned as a **user-friendly, AI-powered virtual assistant** that could provide **basic health support, condition predictions, and treatment guidance** in a conversational style.

The development process was powered by cutting-edge tools and technologies including **Python, Streamlit, Hugging Face Inference API**, and most importantly, the **IBM Granite 13B generative AI model**. Each of these tools played a vital role in shaping the architecture and functionality of the system:

* Python for core logic and API handling
* Streamlit for building the interactive front-end
* Hugging Face for accessing AI model endpoints
* Pyngrok for temporary hosting and public testing
* Google Colab for seamless development
* GitHub for project documentation and submission

The system was divided into **three major functional modules**:

1. **Ask Health Question** – allows users to ask any health-related question and receive AI-generated responses that are empathetic, informative, and easy to understand.
2. **Predict Health Condition** – takes symptoms, age, gender, and vitals as input and predicts the top three likely health conditions using prompt-based AI reasoning.
3. **Treatment Plan Generator** – generates a complete treatment recommendation for a given condition, including medications, lifestyle tips, dietary guidelines, and mental wellness considerations.

Each module was tested for accuracy, ethical response generation, and user readability. Throughout this project, great care was taken to ensure that the AI does not act as a medical authority but rather **supports decision-making and encourages consultation with real doctors**.

The application proved to be:

* **Accessible** – as it works on any device (mobile or desktop)
* **Simple to use** – with a clear layout and no technical expertise required
* **Powerful** – due to its use of IBM Granite’s large language capabilities
* **Scalable** – with possibilities to extend voice support, user history, and EHR integration

We also identified a few **limitations** such as the need for internet access, reliance on API limits, and lack of voice/multilingual features—which were noted for future scope. Enhancements like **voice-based interaction**, **regional language support (Telugu)**, **EHR system integration**, and **secured user profile tracking** can make HealthAI more useful for larger-scale deployment.

In conclusion, the HealthAI project not only fulfilled its original objectives but also successfully showcased the **power of responsible generative AI in healthcare**. It demonstrates how AI, when combined with thoughtful UI and ethical design, can offer valuable solutions to everyday problems. This project was a strong learning experience in both **technical implementation and real-world problem solving**, and stands as a complete and functional prototype ready for further enhancement and deployment.

# *14. References*

- IBM Granite on Hugging Face: https://huggingface.co/ibm  
- Streamlit Documentation: https://docs.streamlit.io/  
- Hugging Face API Guide: https://huggingface.co/docs/api-inference  
- Pyngrok: https://github.com/alexdlaird/pyngrok  
- SmartInternz Internship Portal

1. **IBM Granite on Hugging Face**  
   <https://huggingface.co/ibm>  
   Used as the core generative AI model to handle symptom analysis, Q&A, and treatment suggestions.
2. **Streamlit Documentation**  
   <https://docs.streamlit.io/>  
   Referred for building the web interface, managing tabs, forms, and displaying model outputs.
3. **Hugging Face API Guide**  
   <https://huggingface.co/docs/api-inference>  
   Used to understand how to send prompts, handle API tokens, and receive AI-generated responses in real time.
4. **Pyngrok**  
   <https://github.com/alexdlaird/pyngrok>  
   Used for deploying the Streamlit app with a temporary public URL for live demonstration and testing.
5. **SmartInternz Internship Portal**  
   Platform used for project submission, task tracking, and communication with mentors.