HEALTH AI APP

(Health AI: Intelligent Healthcare Assistant Using IBM Granite Model)

Team ID: LTVIP2025TMID29547

Team Leader: Ananthapalli Rohitha

Team member 1: Adduri Harsha Vardhan

Team member 2: Balla Ramadevi

Team member 3: Velpuri Posibhavani

INTRODUCTION

1.1Project Overview

Health AI is a smart healthcare assistant built using Streamlit, IBM Granite Instruct model, and Hugging Face API, designed to make medical information more accessible and understandable. The app features a simple web interface with three core tools:

1.Medical Report Analyzer – Upload a medical PDF, get a summarized, easy-to-understand report, and ask questions about it.

2.Home Remedies & Self-Care – Enter symptoms and receive AI-generated natural remedies and self-care tips.

3.AI Medical Chatbot – Ask general medical questions and receive AI-based responses.

It uses PyPDF2 for PDF parsing, transformers for language generation, and pyngrok for quick deployment. The app runs efficiently using GPU acceleration via torch.float16 and device_map="auto".

Health AI aims to bridge the gap between complex medical language and patient understanding. It serves as a supportive tool for users to interpret reports, manage minor symptoms, and get reliable health information instantly.

1.2 Purpose

The primary purpose of Health AI is to empower individuals with timely, accurate, and AI-assisted healthcare insights through a smart, easy-to-use digital platform. In many parts of the world
Key objectives of Health AI:

1. Simplify Medical Reports: Patients often struggle to understand clinical language, terminology, or abbreviations used in diagnostic documents. Health AI automatically extracts and simplifies this information, making it more comprehensible and useful for non-experts.

2. Enable Interactive Q&A for Personal Reports: Beyond just summarization, Health AI enables users to ask questions about their specific medical reports—like “What does this diagnosis mean?” or “Should I be concerned about these findings?”—and receive AI-generated responses tailored to their uploaded document.

3. Provide Symptom-Based Self-Care Guidance: People frequently search online for symptom relief, which can lead to misinformation or unverified remedies. Health AI provides structured, AI-generated home remedies and self-care tips based on user-input symptoms, thereby promoting safe and consistent advice.

4. Offer a General-Purpose Medical Chatbot: Health AI acts as a conversational agent capable of responding to general health-related inquiries such as "What are the symptoms of anemia?" or

"How to prevent seasonal flu?". This feature enhances the app’s versatility and makes it useful beyond specific reports or symptoms.

5. Foster Early Awareness and Proactive Health Management: By making reliable health knowledge more accessible, Health AI encourages users to be proactive about their wellness. It equips them with information that may prompt them to seek medical consultation sooner or manage minor symptoms more effectively at home.

6. Promote Accessibility and Digital Health Literacy: This app is also a step toward enhancing digital health literacy—teaching users to interpret medical content through AI-driven explanations and interaction. It is especially helpful for people who have limited exposure to healthcare systems or those who prefer to understand their health issues privately before visiting a doctor.

In summary, Health AI’s purpose is to provide affordable, intelligent, and private digital health support, particularly in situations where professional advice is unavailable, delayed, or needs supplementary interpretation. It’s a tool that aims to inform—not diagnose—while fostering a more health-aware and AI-ready society.

IDEATION PHASE

2.1 Problem Statement

Medical documents and health-related content are often filled with complex language that is difficult for the average person to interpret. Many individuals receive medical reports but struggle to understand the diagnosis, terminology, or urgency. Additionally, people experiencing symptoms often search online for remedies, risking misinformation. There is a critical need for an AI-based solution that simplifies medical information, provides symptom-based guidance, and answers general health queries in a safe and accessible way.

2.2 Empathy Map Canvas

To design a solution that genuinely meets user needs, the Empathy Map Canvas helps us understand the thoughts, feelings, behaviors, and challenges faced by users when dealing with medical information. Users often think things like, “Is this report serious?” or “What do these symptoms mean?”, reflecting a deep concern for their health and a desire for clarity. Many also wonder, “Can I treat this at home safely?” or “Should I see a doctor right away?”, indicating a blend of curiosity and uncertainty. Emotionally, they often feel overwhelmed by the complex medical language in reports, anxious about what their symptoms might indicate, and cautious when considering home remedies. They seek reassurance and confidence in their next steps. Users typically say things like, “I wish someone could explain this,” or “I’m not sure what this means,” expressing confusion and a desire for interpretation. In an effort to find answers, they often do things such as searching for symptoms online, sharing reports with family members or friends, experimenting with random home remedies, or turning to AI tools and chatbots for help. This empathy map highlights the need for a solution that offers clear, understandable, and supportive guidance—one that simplifies medical content, offers safe suggestions, and enables trustworthy conversations about health.

2.3 Brainstroming

In the ideation phase, various solutions were explored to address the user’s needs effectively. The focus was on accessibility, AI integration, and usability.

Key Ideas:

- Summarize uploaded medical reports using a language model
- Enable users to ask questions related to their reports
- Allow symptom input and provide AI-generated remedies
- Integrate a chatbot for general medical Q&A
- Use Streamlit for a fast and clean user interface
- Deploy via ngrok for public access without cloud hosting

This concept ensures the app is helpful, informative, and empowering to users seeking instant, simplified health support.

REQUIRMENT ANALYSIS

3.1 Customer Journey Map

The Customer Journey Map outlines how a user interacts with Health AI, from identifying their need to achieving a solution. It helps visualize pain points and ensures the app delivers a smooth, user-friendly experience.

Stage	User Action	User Goal	Touchpoints	Pain Points
Awareness	User feels unwell or receives a medical report	Understand symptoms or diagnosis	Internet search, online forms, PDF files	Overwhelming information, medical jargon
Consideration	Finds Health AI tool online	Seeks trusted, AI-driven health advice	Health AI web app via ngrok	Trust in AI, ease of use
Interaction	Uploads report / enters symptoms / types query	Get summary, remedies, or answers	Streamlit UI (Medical Report/Home Remedies/Chatbot)	Long reports, fear of incorrect advice
Engagement	Reads AI-generated results and interacts with chatbot	Clarity, assurance, next steps	AI responses, summaries, Q&A	Needs understandable, human-like responses
Resolution	Gains health insights or prepares for a doctor visit	Feel confident or decide next actions	Final summary or tips	May still want a second opinion

3.2 Solution Requirment

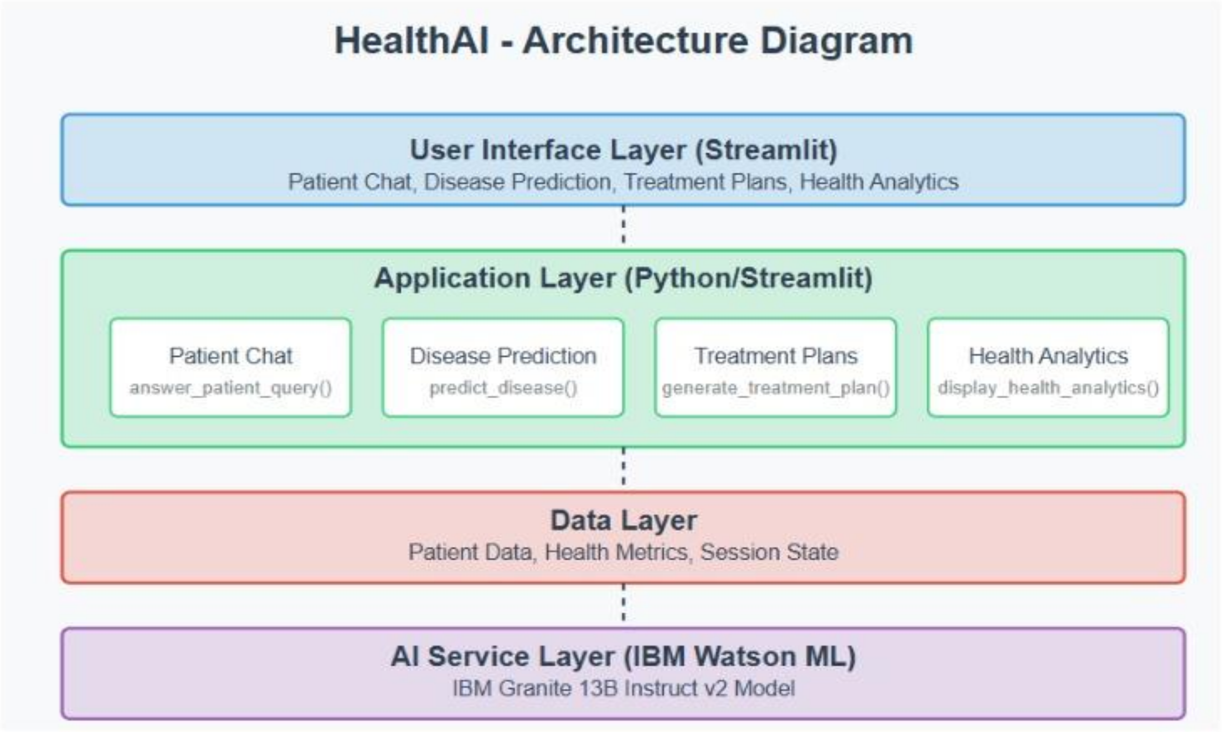
To meet user expectations and project goals, the solution must satisfy the following functional and non-functional requirements:

Functional Requirements:

- **PDF Upload & Extraction:** Allow users to upload medical reports in PDF format and extract readable text.
- **Report Summarization:** Generate simplified summaries of medical reports using a large language model.

Report-Based Q&A: Accept user queries about uploaded reports and generate context-aware answers.

3.3 Architecture Diagram


HealthAI - Architecture Diagram
User Interface Layer (Streamlit)
Patient Chat, Disease Prediction, Treatment Plans, Health Analytics
Application Layer (Python/Streamlit)
Patient Chat
answer_patient_query()
Disease Prediction
predict_disease()
Treatment Plans
generate_treatment_plan()
Health Analytics
display_health_analytics()
Data Layer
Patient Data, Health Metrics, Session State
AI Service Layer (IBM Watson ML)
IBM Granite 13B Instruct v2 Model

3.4 Technology Stack

Layer	Technology / Tool	Purpose
Frontend	Streamlit	Web UI framework to build interactive app
Model Inference	IBM Granite 3.3 2B Instruct (via Hugging Face API)	LLM for summarization, Q&A, and chatbot
Tokenization	Transformers (AutoTokenizer)	Prepares text input for model consumption
Model Loading	PyTorch + Hugging Face Accelerate	Efficient model inference using GPU
File Handling	PyPDF2	Extracts text from uploaded PDF medical files
Deployment	pyngrok	Exposes local Streamlit app to public internet

PROJECT DESIGN

4.1 Problem Solution Fit

Many individuals struggle to understand medical reports due to technical jargon, and often rely on unverified online sources to interpret symptoms or seek remedies. This can lead to anxiety, misinformation, or delayed care. Moreover, people frequently have general medical questions but lack reliable access to a healthcare expert on demand.

- Health AI addresses these pain points directly by offering a unified platform where users can:
- Instantly summarize and understand complex medical PDFs
- Receive trusted home remedies and lifestyle tips based on symptoms
- Ask general medical queries via an intelligent chatbot

By combining accessibility, language generation, and natural user interaction, Health AI fits the problem space perfectly—simplifying health information without compromising quality or trust.

4.2 Proposed Solution

The proposed solution is a **Streamlit-based web application** that integrates **transformer-based large language models** (LLMs) from Hugging Face to deliver three major healthcare utilities in one place:

1. **Medical Report Analyzer**
   - Allows users to upload a PDF medical report.
   - Extracts content using PyPDF2.
   - Summarizes findings and allows Q&A related to the report using IBM Granite (via Hugging Face).
2. **Home Remedies & Self-Care Assistant**
   - Accepts user-described symptoms.
   - Provides 10+ home remedies and lifestyle tips through prompt-engineered LLM outputs.
3. **Medical Chatbot**
   - A conversational AI agent that answers general health questions in simple language.

The app is deployed locally using **ngrok**, enabling real-time external access without cloud hosting. It supports GPU acceleration and can be further scaled for cloud deployment.

4.3 Solution Architecture

Below is a breakdown of the architectural design of the Health AI system:

Components:

- Frontend: Streamlit UI (sidebar navigation, input fields, file uploader)
- PDF Handler: PyPDF2 for extracting text from uploaded medical reports
- LLM Engine: IBM Granite 3.3 2B Instruct via Hugging Face Transformers
- Model Runtime: PyTorch (with device_map="auto" and torch.float16 for GPU efficiency)
- Deployment: pyngrok to expose the local server

Workflow:

1. User interacts with the Streamlit app through sidebar options.
2. Depending on the feature:
    - PDF is uploaded → processed by PyPDF2 → summarized by LLM
    - Symptoms are entered → prompt sent to LLM
    - General question is asked → handled by LLM
3. Outputs are generated by the model and displayed via the UI.
4. App is made public using ngrok for temporary deployment**.**

PROJECT PLANNING AND SCHEDULING

5.1 Project Planning

The planning phase of the Health AI project focused on identifying core functionalities, setting achievable milestones, selecting appropriate tools and models, and establishing a streamlined development workflow. The primary objective was to build an AI-powered healthcare assistant capable of summarizing medical reports, offering symptom-based remedies, and responding to general medical queries—all within a single, user-friendly interface.

The project was broken down into multiple development stages, starting with requirement analysis and UI/UX planning using Streamlit for rapid prototyping. Model selection was based on the need for natural language understanding and generation, leading to the integration of IBM’s Granite instruct model via the Hugging Face API. Simultaneously, PDF processing was handled using PyPDF2, ensuring accurate text extraction from uploaded medical reports.

In terms of scheduling, the development process was divided into weekly sprints, each targeting a key module: Week 1 for PDF upload and text extraction, Week 2 for LLM integration and medical report summarization, Week 3 for home remedies functionality, and Week 4 for chatbot development. Deployment via ngrok was planned for testing and real-time demonstrations.

Planning also accounted for scalability, GPU resource allocation, performance testing, and user feedback integration. By following an iterative, modular approach, the project ensured that each component was tested, refined, and validated independently before final integration. Overall, project planning emphasized simplicity, reliability, and user accessibility, ensuring the Health AI tool could effectively address the real-world healthcare information needs of users.

FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Performance testing for the Health AI application was conducted to ensure the system responds efficiently under expected usage scenarios, with minimal latency and optimal resource utilization. Given the real-time nature of user interactions—such as uploading PDFs, entering symptom descriptions, and chatting with the AI—low response times and high model efficiency were critical.

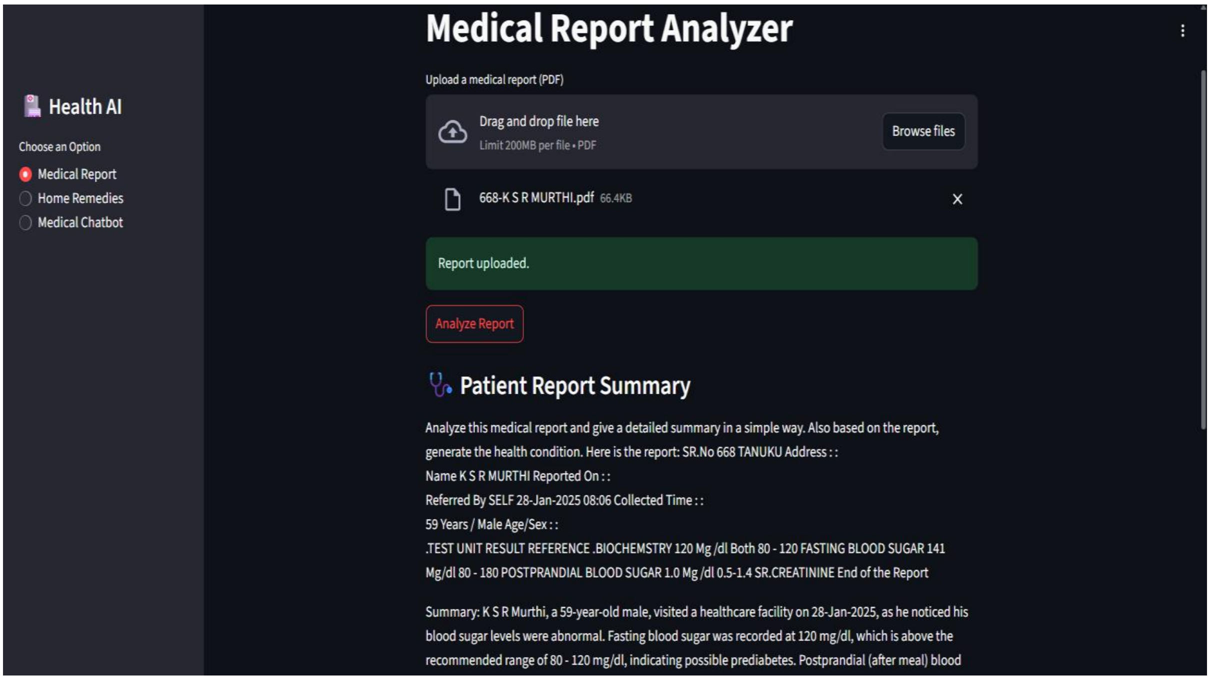
The language model was loaded using PyTorch with GPU acceleration (torch.float16 and device_map="auto") to reduce inference time. Tests were carried out for each core feature independently: PDF summarization, symptom-to-remedy generation, and chatbot interactions. On average, the AI model responded with well-structured, meaningful outputs within 2 to 5 seconds depending on prompt length and task complexity. The medical report summarization took slightly longer due to larger input sizes, whereas symptom-based prompts and general queries were faster.

The application demonstrated stable performance during multiple concurrent requests in a development environment. Pyngrok effectively exposed the local Streamlit app to the public, and response times remained consistent, provided the system had adequate GPU support. Testing on CPU-only environments showed slightly higher latency, which is expected, but the system remained functional and responsive.

Memory consumption was managed efficiently, and caching was implemented for the model loading process using Streamlit's @st.cache_resource, ensuring the model initialized only once per session. No critical bottlenecks were observed during prolonged usage, and the app maintained consistent behavior across various input types.

Overall, the performance testing confirmed that Health AI meets the speed and reliability expectations for a real-time, AI-assisted healthcare tool. It is responsive, stable, and capable of handling interactive tasks without lag, making it suitable for deployment in resource-accessible environments like educational demos, rural outreach programs, or personal health assistants.

RESULTS



Medical Report Analyzer

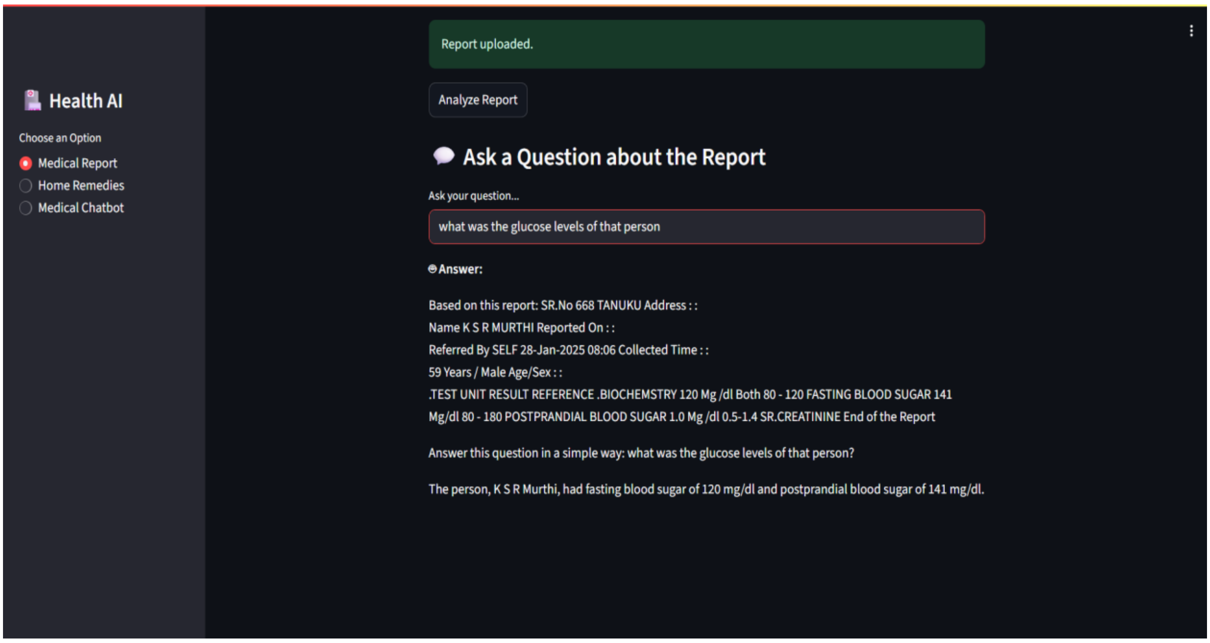
Upload a medical report (PDF)

Report uploaded.

Patient Report Summary

Analyze this medical report and give a detailed summary in a simple way. Also based on the report, generate the health condition. Here is the report: SR.No 668 TANUKU Address : :
Name K S R MURTHI Reported On : :
Referred By SELF 28-Jan-2025 08:06 Collected Time : :
59 Years / Male Age/Sex : :
.TEST UNIT RESULT REFERENCE .BIOCHEMSTRY 120 Mg /dl Both 80 - 120 FASTING BLOOD SUGAR 141 Mg/dl 80 - 180 POSTPRANDIAL BLOOD SUGAR 1.0 Mg /dl 0.5-1.4 SR.CREATININE End of the Report

Summary: K S R Murthi, a 59-year-old male, visited a healthcare facility on 28-Jan-2025, as he noticed his blood sugar levels were abnormal. Fasting blood sugar was recorded at 120 mg/dl, which is above the recommended range of 80 - 120 mg/dl, indicating possible prediabetes. Postprandial (after meal) blood



Report uploaded.

Ask a Question about the Report

Ask your question...

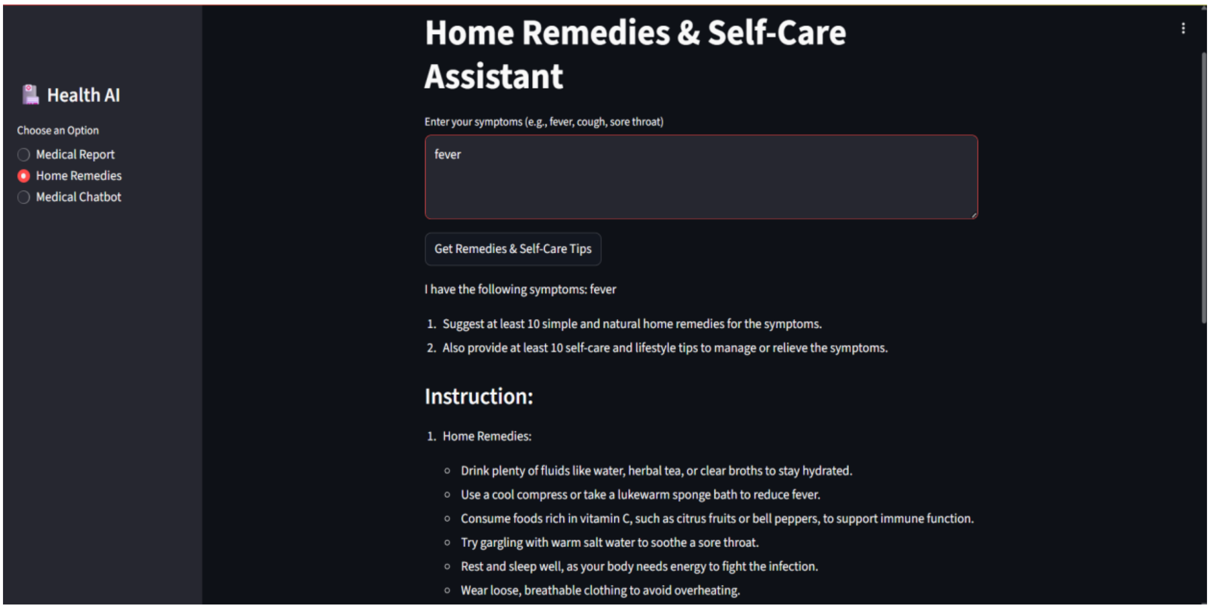
what was the glucose levels of that person

Answer:

Based on this report: SR.No 668 TANUKU Address : :
Name K S R MURTHI Reported On : :
Referred By SELF 28-Jan-2025 08:06 Collected Time : :
59 Years / Male Age/Sex : :
.TEST UNIT RESULT REFERENCE .BIOCHEMSTRY 120 Mg /dl Both 80 - 120 FASTING BLOOD SUGAR 141 Mg/dl 80 - 180 POSTPRANDIAL BLOOD SUGAR 1.0 Mg /dl 0.5-1.4 SR.CREATININE End of the Report

Answer this question in a simple way: what was the glucose levels of that person?

The person, K S R Murthi, had fasting blood sugar of 120 mg/dl and postprandial blood sugar of 141 mg/dl.



Home Remedies & Self-Care Assistant

Enter your symptoms (e.g., fever, cough, sore throat)

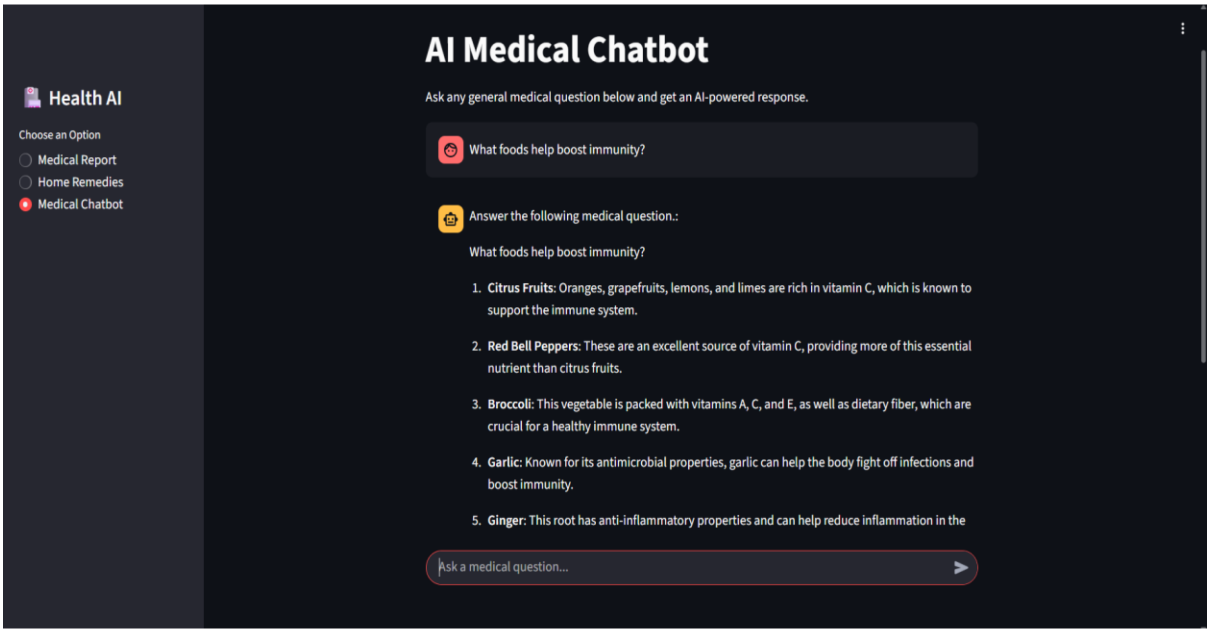
fever

I have the following symptoms: fever

1. Suggest at least 10 simple and natural home remedies for the symptoms.
2. Also provide at least 10 self-care and lifestyle tips to manage or relieve the symptoms.

Instruction:

1. Home Remedies:
    - Drink plenty of fluids like water, herbal tea, or clear broths to stay hydrated.
    - Use a cool compress or take a lukewarm sponge bath to reduce fever.
    - Consume foods rich in vitamin C, such as citrus fruits or bell peppers, to support immune function.
    - Try gargling with warm salt water to soothe a sore throat.
    - Rest and sleep well, as your body needs energy to fight the infection.
    - Wear loose, breathable clothing to avoid overheating.



AI Medical Chatbot

Ask any general medical question below and get an AI-powered response.

What foods help boost immunity?

Answer the following medical question.:

What foods help boost immunity?

1. **Citrus Fruits**: Oranges, grapefruits, lemons, and limes are rich in vitamin C, which is known to support the immune system.
2. **Red Bell Peppers**: These are an excellent source of vitamin C, providing more of this essential nutrient than citrus fruits.
3. **Broccoli**: This vegetable is packed with vitamins A, C, and E, as well as dietary fiber, which are crucial for a healthy immune system.
4. **Garlic**: Known for its antimicrobial properties, garlic can help the body fight off infections and boost immunity.
5. **Ginger**: This root has anti-inflammatory properties and can help reduce inflammation in the

ADVANTAGES & DISADVANTAGES

Advantages

1. **AI-Powered Summarization:**
   The application can simplify complex medical reports using natural language generation, reducing confusion and anxiety for patients.

2. **All-in-One Platform:**
   By combining medical report analysis, symptom-based remedies, and a general chatbot, Health AI serves as a multi-functional healthcare assistant.

3. **No Cloud Hosting Required:**
   With pyngrok, the app can be deployed and shared publicly without needing cloud infrastructure, ideal for testing and local use.

4. **Cost-Effective Prototype:**
   Built using freely available libraries and platforms (Streamlit, Colab, Hugging Face), Health AI is a low-cost yet impactful solution for early-stage deployment.

Disadvantages

1. **Not a Substitute for Medical Advice:**
   The application is purely informational and cannot replace consultation with certified healthcare professionals. Users may misinterpret suggestions as medical prescriptions.

2. **Limited to Text-Based PDFs:**
   PDF parsing relies on PyPDF2, which cannot extract text from image-based or scanned PDFs without OCR (optical character recognition).

3. **Dependent on Internet & API Access:**
   The app requires a stable internet connection and Hugging Face API access, which may be restricted or slow in certain environments**.**

4. **Model Limitations:**
   Responses depend on the language model's training data and prompt context. It may occasionally generate inaccurate or vague answers.

5. **Temporary Deployment with ngrok:**
   While convenient for testing, ngrok links are temporary and not suitable for long-term or high-traffic production deployment.

CONCLUSION

The Health AI project successfully demonstrates how advanced language models can be leveraged to make healthcare information more accessible and understandable to everyday users. By integrating medical report summarization, symptom-based home remedies, and a general medical chatbot into a single, easy-to-use web application, Health AI bridges the gap between complex medical data and patient comprehension.

Through careful design, modular development, and performance optimization, the system provides quick, reliable responses while maintaining a user-centric approach. Although not a replacement for professional medical advice, Health AI empowers users to better understand their health reports, explore safe self-care options, and obtain instant answers to common medical questions.

This project highlights the potential of AI-assisted healthcare tools to improve health literacy, reduce anxiety, and promote proactive health management. With further enhancements and deployment scaling, Health AI can become a valuable resource for broader audiences seeking trustworthy medical insights in a digital-first world.

FUTURE SCOPE

The Health AI project lays a strong foundation for AI-assisted healthcare tools, but there are several promising directions for future enhancement and expansion:

1. **Integration of OCR for Scanned Reports**
   Incorporating Optical Character Recognition (OCR) would enable processing of scanned or image-based medical reports, expanding usability beyond text-based PDFs.

2. **Multilingual Support**
   Adding support for multiple languages would make the app accessible to a broader, global audience, catering to non-English speakers.

3. **Persistent User Data and History**
   Implementing secure user accounts and data storage could allow users to track their health reports, chat history, and remedies over time for better longitudinal health management.

4. **Advanced Symptom Checker with Risk Assessment**
   Enhancing the symptom input feature with clinical risk stratification and triage recommendations can guide users more precisely on when to seek urgent care.

5. **Voice Input and Output**
   Adding speech-to-text and text-to-speech functionalities would improve accessibility for users with visual impairments or low literacy.

6. **Mobile Application Development**
   Creating native Android and iOS apps would improve accessibility and usability for users who prefer mobile devices.

7. **Integration with Wearables and Health Data**
   Connecting with health trackers and wearable devices could enable personalized, real-time monitoring and suggestions based on biometric data.

8. **Enhanced Security and Privacy Features**
   Strengthening data encryption, user consent mechanisms, and compliance with healthcare data regulations (e.g., HIPAA, GDPR) would be essential for broader adoption.

9. **Deployment on Cloud Platforms**
   Migrating to cloud services for scalable and reliable hosting would support higher user volumes and persistent uptime.

By pursuing these enhancements, Health AI can evolve into a comprehensive digital health assistant, improving healthcare accessibility and patient empowerment on a global scale.

APPENDIX

SOURCE CODE LINK: https://github.com/Rohitha2803/Health_app/blob/main/Health_app.ipynb