**Project Title**

**Project Documentation**

# 1.Introduction

* **Project Title :**  Health AI
* Team Member : Saranya. S
* Team Member : Sevvanthi. K
* Team Member : Shakanaa. M.A
* Team Member : Shalini. P
* Team Member : Sharmila. S

# 2.project overview

* **Purpose** :

HealthAI is an AI-powered healthcare assistant designed to provide patients with 24/7 support through a conversational interface. It integrates IBM Granite (granite-3.2-2b-instruct) model for generating natural and context-aware responses. The system allows patients to interact with an AI assistant by entering their details (name, age, gender, medical history, medications, allergies) and asking health-related questions via a Gradio web interface.

* **Features** :
  + **Conversational Interface**
    - *Key Point*: Natural language interaction
    - *Functionality*: Patients can ask health-related questions and receive AI-generated responses in plain language
  + **Symptom Checker**
    - *Key Point*: Early diagnosis support
    - *Functionality*: Suggests possible conditions based on symptoms entered by the user
  + **Medical Record Management**
  + *Key Point*: Organized patient data
  + *Functionality*: Stores, retrieves, and summarizes patient health records
* **Prescription Reminder**
  + *Key Point*: Patient support
  + *Functionality*: Alerts patients to take medicines on time
* **Health Tips Generator**
  + *Key Point*: Preventive care
  + *Functionality*: Provides daily personalized tips to improve lifestyle and well-being
* **Doctor–Patient Interaction**
  + *Key Point*: Virtual consultation
  + *Functionality*: Enables patients to share symptoms, documents, and get feedback from doctors
* **Anomaly Detection**
  + *Key Point*: Alert system
  + *Functionality*: Identifies abnormal health patterns like unusual heart rate or blood sugar levels
* **Multimodal Input Support**
  + *Key Point*: Flexible data handling
  + *Functionality*: Accepts text, PDFs, or images (like prescriptions, lab reports) for analysis
* **User-Friendly Dashboard**
  + *Key Point*: Accessible interface
  + *Functionality*: Provides an intuitive UI for patients, doctors, and healthcare staff

# 3. Architecture

**. Frontend:** Gradio framework is used to build a simple, user-friendly interface with patient details input and chatbot window.

**. Backend:** Python-based application integrating Hugging Face Transformers. The Granite model is loaded for natural language generation.

**. Model Integration:** IBM Granite (granite-3.2-2b-instruct) from Hugging Face is used for generating context-aware health responses.

**. Deployment:** Runs locally or in Google Colab with T4 GPU for better performance.

## 4. Setup Instructions

• Install dependencies: `pip install transformers torch gradio`.

• Load IBM Granite model from Hugging Face.

• Run the Python script to start the Gradio application.

• Access the Gradio share link to interact with the chatbot.

## 5. Folder Structure

 **app/** – Backend logic with routers and services

 **app/api/** – APIs (chat, health records, reminders, anomaly detection)

 **ui/** – Streamlit pages and dashboards

 **health\_dashboard.py** – Launches main Streamlit dashboard

 **symptom\_checker.py** – Suggests possible conditions

 **record\_manager.py** – Stores and summarizes patient data

 **anomaly\_detector.py** – Detects unusual health values

 **reminder\_service.py** – Manages prescription reminders

 **report\_generator.py** – Generates health summary reports

## 6. Running the Application

1. Run the Python script in your environment or Google Colab.

2. After installation, the IBM Granite model will be downloaded.

3. Gradio will provide a local URL and an optional share link.

4. Open the link to access the chatbot and start interaction.

## 7. API Documentation

**POST /chat/ask** – AI health query responses

**POST /upload-record** – Upload patient health record **GET /search-records** – Retrieve patient history

**GET /get-health-tips** – Provides daily health tips

**POST /set-reminder** – Create prescription or appointment reminders

Documented with Swagger UI for testing

## . 8. Authentication

* Current version: Open environment (demo)
* Secure deployment options:

o Token-based authentication (JWT, API keys) o Role-based access (doctor, patient, admin) o Encrypted storage for sensitive data o Planned session & history tracking

## 9. User Interface

The interface is a **clean, user-friendly health platform** designed for accessibility and ease of use across all age groups. Key features include:

* **Sidebar Navigation**

A left-hand sidebar with simple input fields (name, age, gender, medical history, symptoms) for quick patient data entry.

* **Patient Dashboard with Health KPIs**

Displays a card-style summary of key health indicators (conditions, recommendations, therapy plans) with a clear typography hierarchy.

* **Tabbed Layout**

Tabs at the top or side for switching between:

o *Chat / AI Consultation* (for interactive health advice) o *Health Tips* (personalized lifestyle recommendations) o *Records* (medical history, prescriptions, reports)

* **Smart Notifications**

Reminder alerts for upcoming checkups, medications, or exercise sessions.

* **Downloadable PDF Summaries**

One-click option to export patient assessments or therapy plans into neatly formatted PDF files.

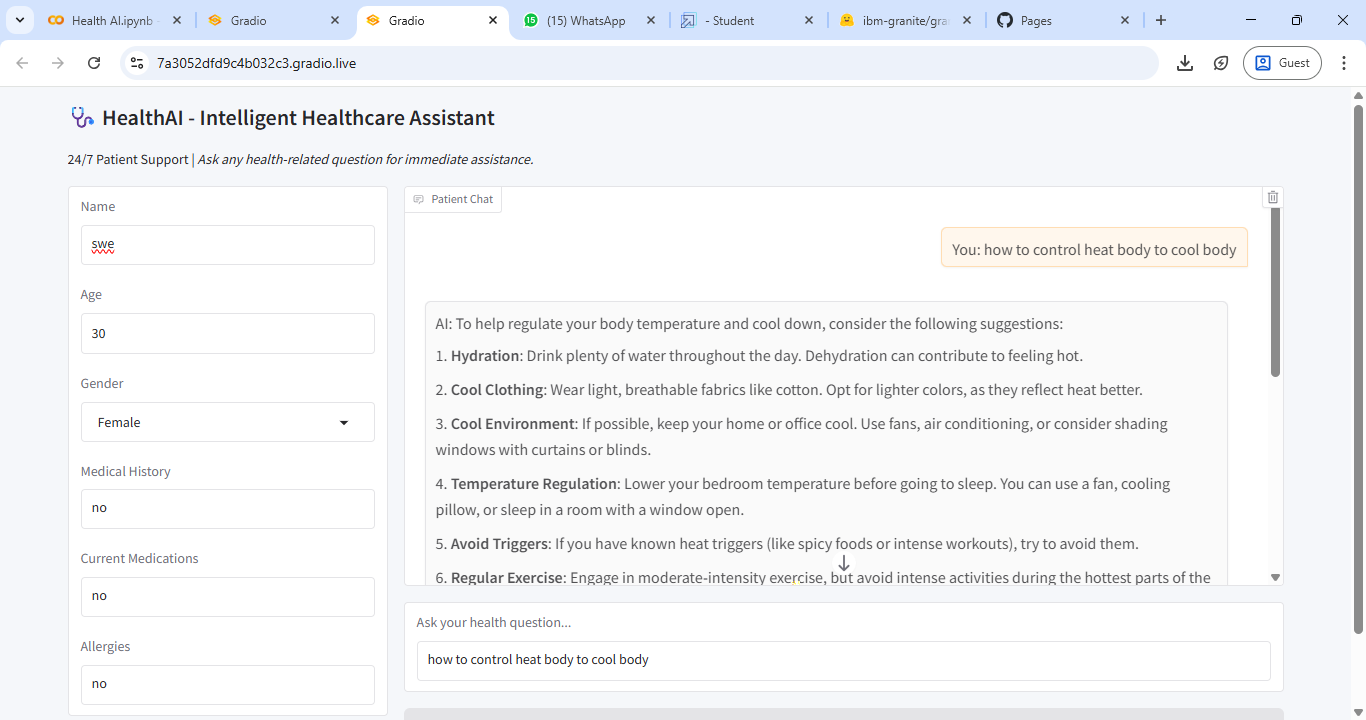
* **Minimalist, Accessible Design** o Large, high-contrast text for readability o Clear icons and spacing for easy navigation o Works seamlessly on desktop and mobile o Inclusive for both tech-savvy and non-technical users

## 10. Testing

Testing was done in multiple phases:

* **Unit Testing**: Symptom checker, anomaly detection, reminders
* **API Testing**: Swagger UI and Postman
* **Manual Testing**: Chat assistant, file uploads, record summaries
* **Edge Cases**: Malformed data, missing health records, invalid inputs

## 11.screen shots



## 12. Known Issues

* Requires reliable internet for AI queries
* Limited offline functionality
* High dependency on cloud APIs

## 13. Future enhancement

* Mobile app version for patients
* Integration with IoT devices (smartwatches, glucose monitors)
* AI-powered disease prediction models
* Multi-language support for regional users
* Blockchain for secure medical record storage