**Health AI - Intelligent Healthcare Engagement Platform**

# INTRODUCTION

* **PROJECT TITLE : Health AI – Intelligent Healthcare Engagement Platform**
* **TEAM MEMBER : Sree Chitra Lekha S**
* **TEAM MEMBER : Keerthana S**
* **TEAM MEMBER : Hemamalini V**
* **TEAM MEMBER : Rekha V**

# PROJECT OVERVIEW

* **PURPOSE :**

The purpose of Health AI is to revolutionize the way patients, doctors, and healthcare providers interact with medical services by offering an intelligent, seamless, and accessible digital platform. Leveraging IBM’s Granite AI models, the system provides:

* Real-time patient query resolution
* Medical report summarization
* Symptom analysis and triage guidance
* Healthcare analytics for hospitals and administrators

The platform acts as both a **patient support system** and a **decision-support tool for medical staff**, enhancing diagnosis, improving patient satisfaction, and streamlining hospital operations.

**KEY FEATURES:**

* **Conversational Medical Assistant:**

Patients can ask health-related questions in natural language and receive AI-generated responses.

* **Medical Report Summarization:**

Simplifies complex reports and prescriptions into patient-friendly summaries.

* **Symptom Checker & Triage:**

Provides preliminary health guidance and suggests doctor consultations.

* **Appointment & Service Handling:**

Patients can book appointments, request lab tests, and track service status.

* **Decision Support for Doctors:**

Offers insights from patient data, trends, and medical literature.

* **Patient Feedback Loop:**

Collects and analyses patient reviews to enhance healthcare services.

* **Anomaly Detection:**

Detects unusual patterns in patient records, prescriptions, or diagnostic results for early warnings.

* **Multilingual & Accessible Interface:**

Ensures inclusivity for diverse populations.

* **Hospital Dashboard:**

Provides administrators with analytics on patient flow, treatment outcomes, and resource management.

# ARCHITECTURE

**FRONTEND**

The frontend is designed using **Gradio or Streamlit**, offering an interactive and user-friendly interface. It contains dedicated tabs such as **Patient Services**, **Doctor Support**, and **Hospital Analytics**. Patients can enter symptoms, upload reports, or ask medical queries, while doctors and administrators can access dashboards, review insights, and track healthcare analytics.

**BACKEND**

The backend is implemented in **Python** and acts as the bridge between the user interface and the AI model. It processes inputs, executes functions like symptom\_analysis(), medical\_summarization(), and patient\_interaction(), and ensures smooth communication between the patient, doctor, and the AI engine.

**AI/LLM CORE**

At the heart of the system lies the **IBM Granite large language model**, integrated through Hugging Face Transformers. This model performs natural language understanding, medical report summarization, and generates context-aware responses. It powers real-time patient query resolution and provides decision support for healthcare professionals.

**ML MODULES**

The system is enriched with specialized **machine learning modules**. These include **symptom categorization and triage** to identify possible conditions, **medical report summarization** to simplify prescriptions and test results, **feedback analysis** to monitor patient satisfaction, **anomaly detection** to flag unusual medical data patterns, and **hospital analytics** to provide insights on patient flow, treatment outcomes, and risk prediction.

**AUTHENTICATION AND SECURITY**

Security is ensured through **role-based access control**, allowing patients, doctors, and administrators to log in with appropriate permissions. The platform supports **token-based authentication** such as JWT or OAuth2, ensuring data confidentiality. It is also designed to comply with healthcare data protection standards like **HIPAA and GDPR**.

**DATA LAYER**

For advanced use cases, the platform can connect to **Electronic Health Records (EHRs)**, hospital databases, or **IBM Cloud services**. This integration enables real-time access to patient records, laboratory results, and other critical healthcare data, improving both accuracy and efficiency in patient care.

# SETUP INSTRUCTIONS

**PREREQUISITES:**

* **Programming Language:** Python 3.9 or later should be installed.
* **Package Manager:** pip or conda for managing Python libraries.
* **AI Libraries:** Hugging Face Transformers, PyTorch, and Gradio (for the user interface).
* **Hardware Requirements:** A system with a dedicated GPU and CUDA support is recommended for faster AI model inference.
* **Internet Connectivity:** Required to download IBM Granite models from Hugging Face or IBM Cloud repositories.
* **Virtual Environment Tools:** To keep dependencies organized and avoid conflicts.

**INSTALLATION STEPS**

1. **Clone the Repository**
   * Download or clone the Health AI project repository from GitHub (or your project source).
2. **Create a Virtual Environment**
   * Set up a virtual environment to manage dependencies.
3. **Install Dependencies**
   * Use the provided requirements.txt file to install all necessary libraries.
4. **Download IBM Granite Model**
   * Configure Hugging Face Transformers to fetch the **IBM Granite LLM**.
   * The model will be automatically downloaded on first use, but you can also pre-download it for faster setup.

**3. Running the Application**

Once the environment is set up and dependencies are installed, run the application:

**Launch the Main Script**

* + Execute the entry file

**Access the Web Interface**

* + The Gradio server will start automatically and display a local URL
  + Open this link in a web browser to access the Health AI interface.

**Use the Application**

* + Navigate between different tabs:
    - **Patient Services:** Enter symptoms, queries, or upload reports.
    - **Doctor Support:** Get AI-driven summaries and recommendations.
    - **Hospital Analytics:** Explore dashboards and trend insights.

**4. Optional Configuration**

* **Authentication Setup:** Enable token-based or role-based access control for secure deployments.
* **Cloud Integration:** Configure IBM Cloud services for scaling and EHR database connectivity.
* **GPU Optimization:** Adjust CUDA settings for improved inference speed.
* **Logging & Monitoring:** Enable system logs to track user activity and model performance.

# FOLDER STRUCTURE

* **app/**  
   This is the core application directory that contains the main logic of the Health AI system. It manages the Gradio interface, user interactions, and connects different modules to the backend and AI model.
* **app/modules/**  
   A subdirectory dedicated to modular scripts. Each file in this folder focuses on a specific healthcare functionality, such as **symptom analysis**, **medical report summarization**, **patient interaction**, or **data preprocessing**. This modular design ensures scalability and easy maintenance.
* **ui/**  
   Stores frontend-related resources, including **CSS stylesheets**, images, icons, or design files. These assets enhance the user experience and can be customized for better accessibility and branding.
* **health\_ai.py**  
   The **main entry point** of the project. Running this script launches the Health AI application, initializes the Gradio interface, and provides access to different tabs such as Patient Services, Doctor Support, and Hospital Analytics.
* **granite\_llm.py**  
   Handles integration with the **IBM Granite AI model**. This file includes functions for **loading the model**, **tokenization**, and **text generation**. It ensures that all user queries are processed with context-aware and reliable AI responses.
* **medical\_services.py**  
   Manages the **patient-facing services**. It processes health-related queries, performs medical report summarization, and provides AI-driven suggestions such as symptom guidance and triage support.
* **hospital\_analytics.py**  
   Dedicated to **administrative and clinical insights**. It generates dashboards for doctors and hospital administrators, showing patient trends, risk predictions, and performance metrics to support decision-making.
* **requirements.txt**  
   A dependency file that lists all the necessary Python libraries (e.g., PyTorch, Transformers, Gradio). This allows quick installation of required packages with a single command.
* **README.md**  
   A documentation file that provides an overview of the project, setup instructions, usage guidelines, and details about the system’s purpose and features. It serves as a quick reference for developers and users.

# RUNNING THE APPLICATION

 **To start the project:** Run the Gradio app script .

 **Gradio server launch:** The Gradio server will start automatically and display a local URL in the terminal.

 **Access the interface:** Open the URL in your web browser to access the Health AI platform.

 **Navigate between tabs:** Use the main sections – Patient Services and Hospital Analytics.

 **Patient Services tab:** Enter symptoms, upload medical reports, or type a health-related query to receive AI-generated summaries and guidance.

 **Hospital Analytics tab:** View dashboards, patient trends, and anomaly detection results to support healthcare management.

 **AI processing:** All outputs and insights are generated in real time using the IBM Granite model, integrated through Hugging Face Transformers.

**FRONTEND (Gradio)**

The frontend of Health AI with IBM is developed using Gradio Blocks and Tabs, which provide an interactive and easy-to-use interface. It includes multiple pages such as Patient Services and Hospital Analytics. Patients can enter symptoms, upload medical reports, or ask health-related questions, while doctors and administrators can view dashboards and insights. The layout is organized with rows and columns, ensuring a clean design and smooth user experience for healthcare interactions.

**BACKEND (Python + Hugging Face Transformers)**

The backend is implemented in Python, where Gradio also manages the web server functionality. Functions like symptom\_analysis() and medical\_interaction() process patient inputs and forward them to the IBM Granite LLM. The backend handles loading the model, tokenization, text generation, and formatting outputs before sending responses back to the user interface. This ensures real-time AI-powered feedback for patients, doctors, and healthcare administrators.

# AUTHENTICATION

The Health AI system runs in an open environment for demonstration. For secure healthcare deployments, the following methods can be integrated:

* Basic username/password authentication using Gradio’s built-in authentication feature.
* Token-based authentication (JWT or API keys) for secure API-level access.
* Auth2 integration with IBM Cloud or third-party healthcare identity providers.
* Role-based access (Patient, Doctor, Admin) to ensure controlled and secure usage.
* Planned enhancements include user sessions and chat history tracking, enabling personalized healthcare experiences and continuity of care.

# USER INTERFACE

The user interface of Health AI with IBM is designed to be simple, interactive, and patient-friendly using Gradio Tabs and Blocks. It provides different sections for patients, doctors, and administrators, ensuring easy navigation and accessibility.

* **Patient Services Tab:** Patients can enter symptoms, upload medical reports, or ask queries in plain language. The interface instantly displays AI-generated summaries, guidance, or triage suggestions.
* **Doctor Support Tab:** Provides medical professionals with quick report summaries, decision-support insights, and patient interaction history.
* **Hospital Analytics Tab:** Displays dashboards, graphs, and statistics about patient trends, anomalies, and hospital performance.

The design emphasizes clarity, speed, and accessibility, with multilingual and disability-friendly support to ensure inclusivity.

# TESTING

**Testing – Health AI with IBM**

Testing of the Health AI system was conducted in multiple phases to ensure accuracy, reliability, and smooth interaction:

* **Unit Testing:** Focused on functions like generate response(), symptom analysis(), and medical interaction() to verify that they generate correct and relevant outputs.
* **Interface Testing:** Checked Gradio input/output components, including textboxes, file uploads, buttons, and tab navigation, to confirm smooth interaction for patients, doctors, and administrators.
* **Manual Testing:** Conducted by entering symptoms, uploading medical reports, and submitting healthcare-related queries, then verifying the AI-generated responses against expected results.
* **Edge Case Handling:** Validated system performance with incomplete inputs, empty queries, very large medical records, unusual symptoms, and multilingual queries to test robustness.

Each function and interaction was thoroughly validated to ensure reliability, responsiveness, and consistency within the real-time Health AI interface.

# SCREEN SHOTS

Disease Prediction:

A screenshot of a computer

AI-generated content may be incorrect.

Treatment Plan:

A screenshot of a computer

AI-generated content may be incorrect.

# 11.KNOWN ISSUES

# 

* **Slow Response for Large Inputs:** Long or detailed medical reports may take extra time to process because of heavy model inference.
* **No Authentication in Demo Mode:** The current build runs in an open environment, meaning anyone with access can use the system without login.
* **Lack of Chat History:** Patient queries and doctor interactions are not stored, so there is no session continuity.
* **Handling of Unstructured Data:** Very complex, incomplete, or malformed medical inputs may sometimes lead to incomplete or less accurate responses.
* **No Knowledge Grounding:** The AI relies only on the IBM Granite model. It does not yet integrate with external medical databases or knowledge bases, which may limit accuracy for rare or complex cases.

# 12.FUTURE ENHANCEMENT

* Advanced Predictive Analytics – Early disease detection using multi-modal data (genomics, lifestyle, environment).
* Enhanced Natural Language Understanding – AI interprets complex clinical notes and doctor-patient conversations.
* Integration with Wearables and IoT Devices – Real-time monitoring for proactive healthcare.
* Personalized Treatment Planning – Precision medicine with tailored therapies based on genetics and lifestyle.
* Improved Interoperability and Data Sharing – Secure, standardized, and seamless health data exchange.
* AI-Driven Drug Discovery and Clinical Trials – Accelerated research and trial matching.
* Ethical and Explainable AI – Transparent AI models to build trust and regulatory compliance.
* Global Health and Pandemic Preparedness – Predictive modeling for outbreaks and rapid response.
* Integration of Virtual Care and Robotics – AI-assisted telemedicine, virtual assistants, and robotic surgeries.