HealthAI Project Documentation (FastAPI Version)

Team Members

Team Leader: Tallam Neeraja Sree Vyshnavi

Team Member: Uppu Prathyusha

Team Member: Yeruva Lakshmi Gayathri

Project Overview

Purpose:

Health AI provides a lightweight and intelligent healthcare assistant that predicts possible diseases based on user symptoms and suggests natural home remedies. The core model used is IBM Granite 3.3B Instruct, accessed via Hugging Face Transformers.

Key Features:

1. Disease Prediction Users input symptoms, and the AI predicts the most likely disease using natural language understanding.

2. Home Remedy Suggestion Given a disease, the AI suggests safe and natural remedies.

3. Web-based Interface Built using Fast API and Jinja2, with a clean HTML/CSS front end.

Health AI Architecture Diagram

Overview Diagram (Conceptual):

User Inputs symptoms or disease
Frontend (HTML/CSS + Jinja2 Templates)
- Collects input
- Displays predictions
Backend (Fast API)
- Routes input
- Calls prediction/remedy functions
Model Layer (Hugging Face Transformers + IBM Granite)
- Prompt generation
- Response parsing
Response Sent to User
- Remedy suggestions
- Disease prediction

Component Mapping

Component	Technology Stack
Frontend	HTML, CSS, Jinja2
Backend	Fast API
Model Interface	Hugging Face Transformers
AI Model	IBM Granite 3.3B Instruct

HealthAI Project Documentation (FastAPI Version)

- Uses causal language modeling to return health predictions and remedy suggestions.

Setup Instructions

Prerequisites:

- Python 3.8+
- Git
- A virtual environment manager (optional but recommended)
- Access to Hugging Face with model 'ibm-granite/granite-3.3-2b-instruct'

Installation Steps

```
# Clone the repository
git clone <your-repo-url>
cd HealthAI

# Create and activate virtual environment
python -m venv venv
source venv/bin/activate  # On Windows: .\venv\Scripts\activate

# Install dependencies
pip install fastapi uvicorn transformers torch jinja2
```

Folder Structure

```
HealthAI/
 app.py                # FastAPI application
 model_utils.py        # ML logic (prompting, decoding)
 templates/
   index.html          # Jinja2 HTML frontend
 static/
   style.css           # Custom UI styling
 venv/                 # Python virtual environment (optional)
 requirements.txt      # Dependencies list
```

Running the Application

HealthAI Project Documentation (FastAPI Version)

uvicorn app:app --reload

Open browser and go to: http://127.0.0.1:8000

Example Prompts Used

Disease Prediction Prompt:

User has the following symptoms: fever, chills, headache. Predict the most likely disease.

Remedy Prompt:

Suggest a natural home remedy for the disease: dengue.

Features in Detail

1. Disease Prediction

Function: predict_disease(symptoms: str)

Model: IBM Granite via Hugging Face

Prompt-based generation using user-provided symptoms.

Returns: Disease name and brief explanation.

2. Remedy Suggestion

Function: get_remedy(disease: str)

Returns: A home remedy suggestion in a safe and concise way.

User Interface

Developed with Jinja2 HTML Templates and CSS.

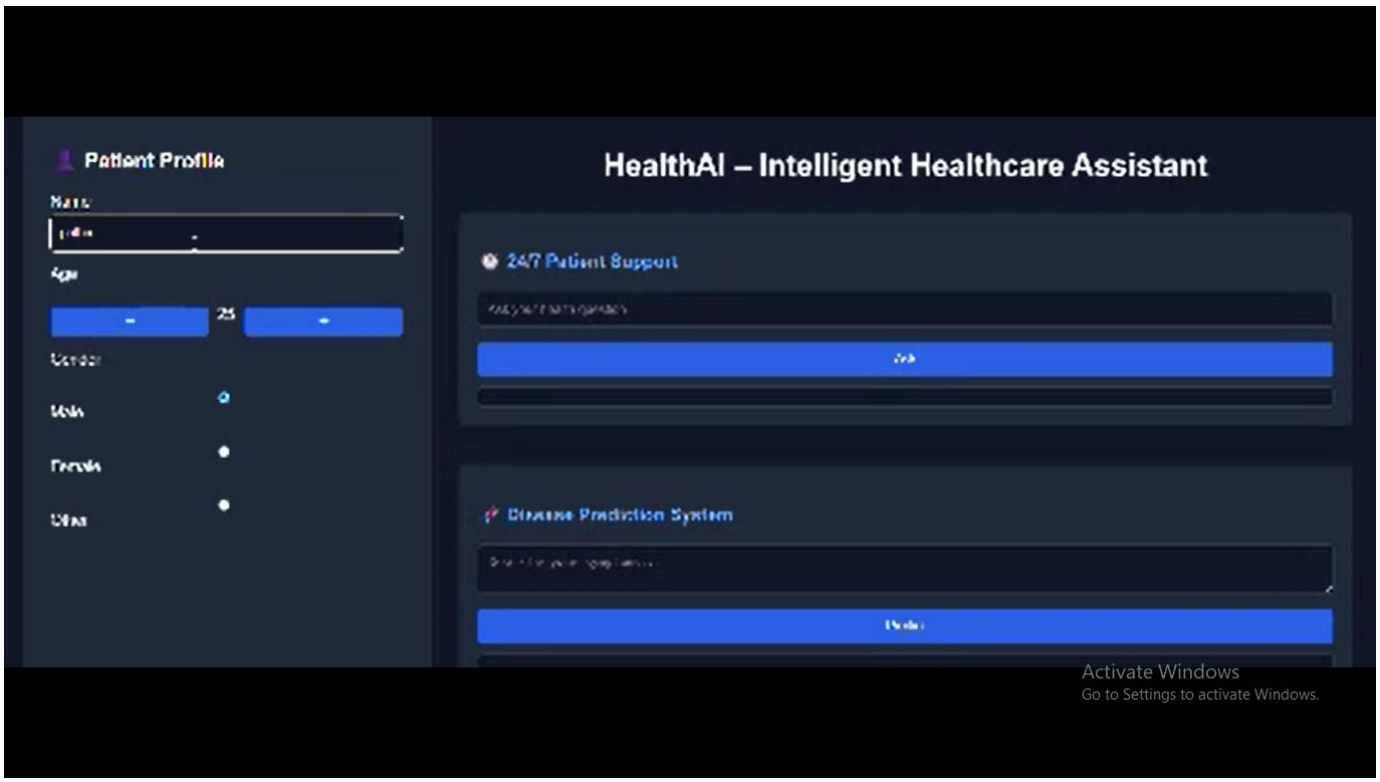
Responsive form layout:

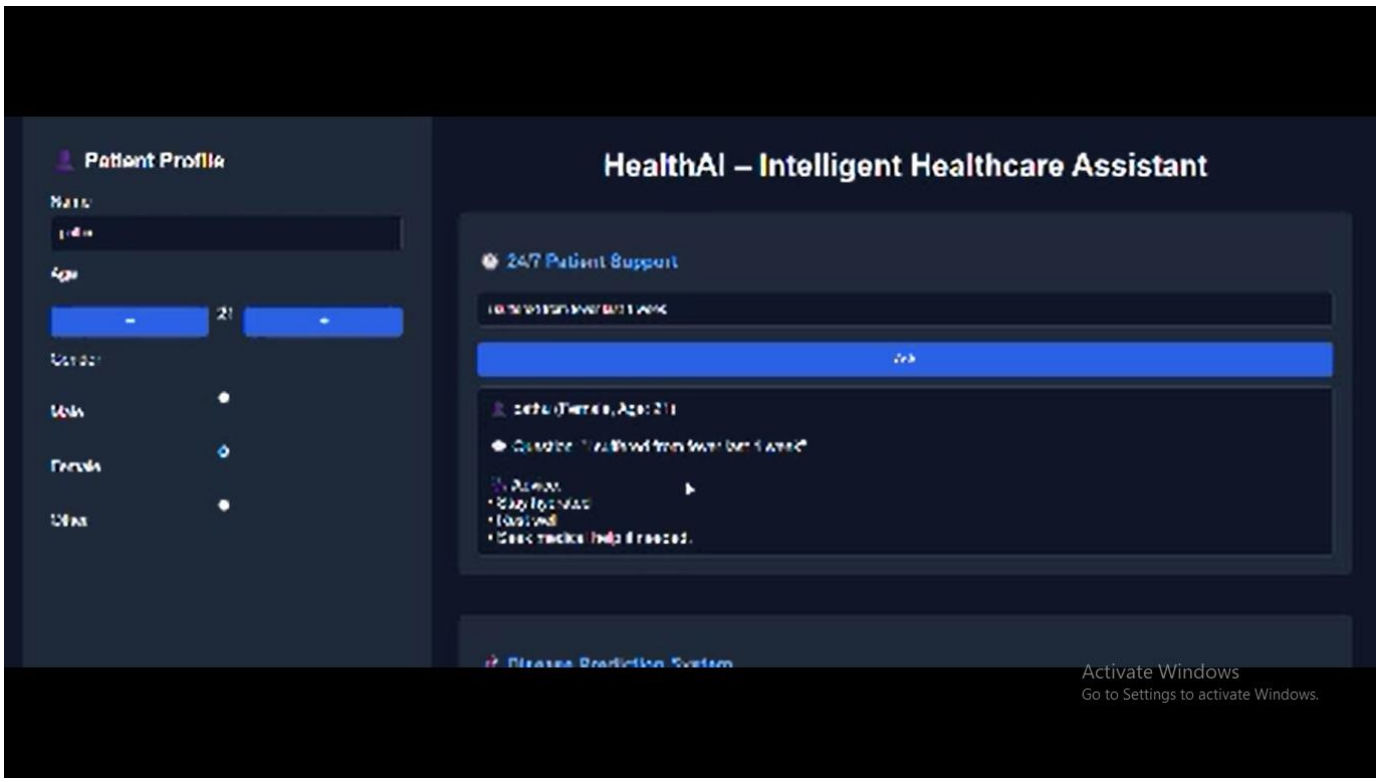
- Symptom input disease prediction

- Disease input home remedy suggestion

Screenshots

HealthAI Project Documentation (FastAPI Version)





-

Testing Strategy

Unit Testing: Basic tests of the predict_disease() and get_remedy() functions.

Integration Testing: Ensures FastAPI correctly connects the frontend, backend, and model.

User Testing: Manual testing for form interactions and prompt outputs.

Error Handling: Minimal input sanitization and default fallbacks.

Known Limitations

- No persistent data storage or user session memory.
- Model outputs may be general-purpose and not fully clinically validated.
- No user authentication or protection against spam/bot inputs.

Future Enhancements

- Add database to store user queries and health profiles.
- Enable user login and personalized dashboards.
- Integrate voice input or chatbot UI.
- Extend model to include more detailed diagnosis and treatments.
- Host the application on Hugging Face Spaces or Render for public access.