HealthAI: Intelligent Healthcare Assistant

Introduction

Project Title:- HealthAI : Intelligent Healthcare

HealthAI is an innovative system designed to enhance healthcare delivery through the power of artificial intelligence. The project focuses on developing intelligent solutions that can assist in diagnostics, patient monitoring, personalized treatment recommendations, and administrative automation. By integrating machine learning and data-driven technologies, HealthAI aims to improve the efficiency, accuracy, and accessibility of healthcare services.

Key objectives of HealthAI include:

- Providing accurate diagnostic support using AI-driven algorithms.
- Enabling real-time health monitoring and alert systems.
- Offering personalized healthcare recommendations.
- Streamlining hospital operations through automation.

A major feature of **HealthAI** is its **AI-powered disease prediction system**, which analyzes patient-reported symptoms to provide possible diagnoses. The system is built using advanced machine learning techniques trained on extensive medical datasets. Here's how it works

Key Features:

- **Symptom Input Interface:** Patients or doctors can input observed symptoms through a simple and intuitive interface.
- **Natural Language Processing (NLP):** The system understands and processes natural language symptom descriptions.

- **Machine Learning Model:** Trained on thousands of cases, the model identifies patterns and associations between symptoms and diseases.
- **Probability Scoring:** It predicts a list of possible diseases ranked by likelihood, helping doctors narrow down diagnoses.
- **Decision Support:** The system supports early detection of conditions such as diabetes, cardiovascular diseases, infections, and more.

Example Workflow:

1. A patient reports symptoms like fever, cough, fatigue, and shortness of breath.
2. The system processes the input using NLP.
3. Based on symptom similarity and training data, it predicts likely diseases (e.g., influenza, COVID-19, pneumonia).
4. Results are displayed with a confidence score for each potential condition.
5. The doctor uses this information for further tests and confirmation.

Team Members and Roles:-

Names	Roles
S.Z.MD.IRFAN	Backend & API Developer
S.SULEMAN	Frontend & UI Developer
S.MUSTAKHEEM	Project Manager
S.SAMEER BASHA	Model Tester
S.MD.ABDUL ZAIB	QA Lead

Project Overview

Purpose

HealthAI is an intelligent healthcare assistant designed to help users receive AI poweredmedical insights and suggestions. It aims to simplify preliminary health diagnosis, provide personalized treatment plans, and allow users to interact with a medical assistant via chat, all while analyzing vital health data.

Features

- Symptom-based Disease Prediction
- AI-generated Treatment Plans
- Patient Chat Assistant using FLAN-T5
- Health Analytics with Vitals Visualization
- Modern and Clean UI with Streamlit

Architecture

Frontend (React-based Design)

- Originally developed in React (Bolt.AI exported Netlify project).
- Adapted and re-built using Streamlit for simplicity and fast deployment.
- Includes interactive tabs for disease prediction, treatment plan, analytics, and chat.
- Integrated with Python backend via REST APIs.

Backend (Node.js Equivalent -> Python FastAPI)

- Built using FastAPI (Python) instead of Node.js for ML integration ease.
- Uses transformers pipeline for Google FLAN-T5 model.
- Exposes REST endpoints for all functionalities.

Setup Instructions

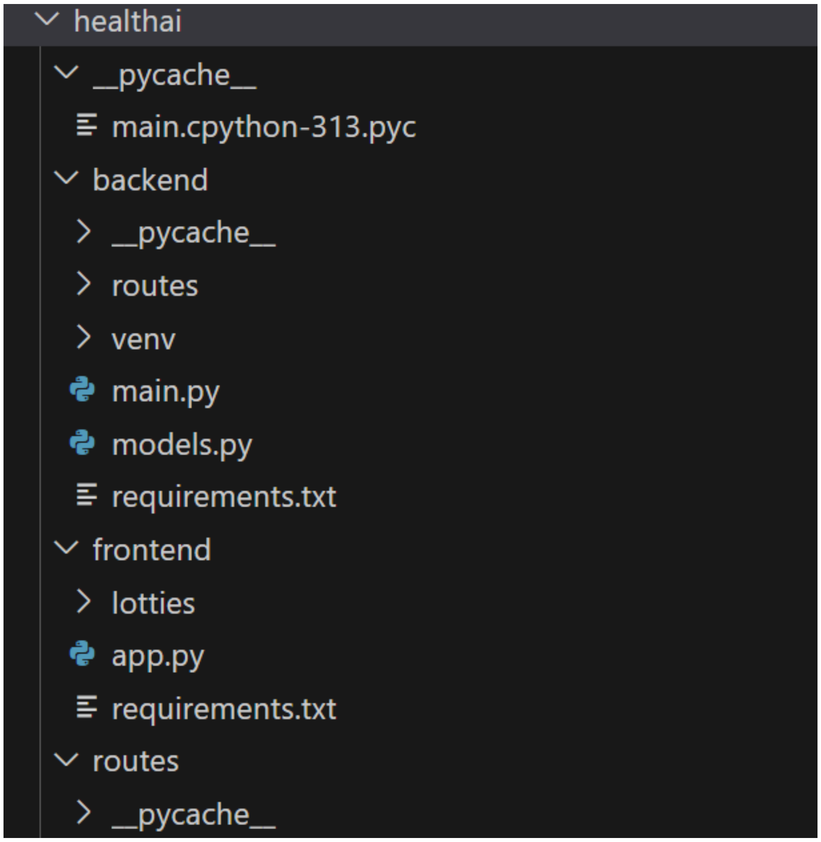
Prerequisites

- Python 3.10+
- Pip
- Git
- Hugging Face Token

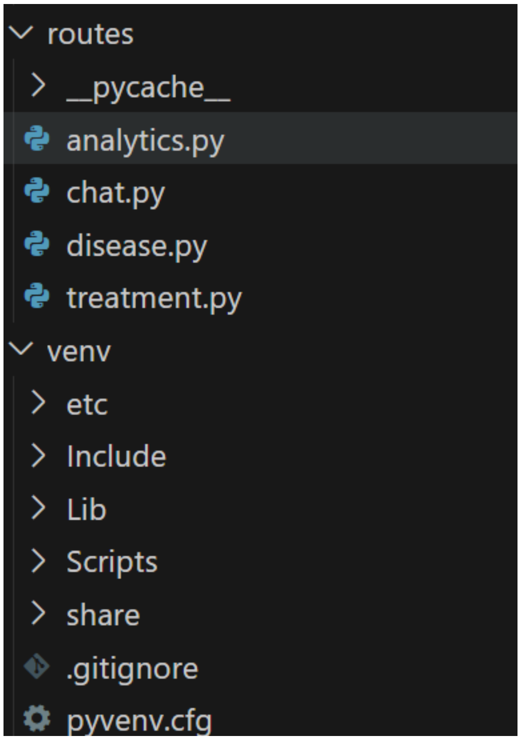
Installation

```
# To Clone the repo

$ git clone https://github.com/z-irfan/health-ai.git

$ cd health-ai(file name)

# To Download model run this below command in file directory command prompt

$ python download_flan_t5.py
```

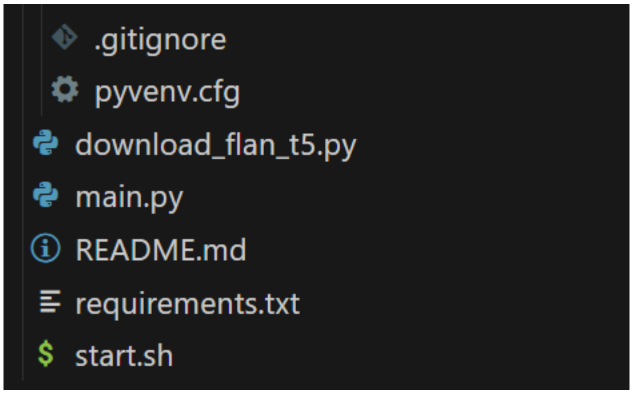
Folder Structure



```
healthai
  __pycache__
    main.cpython-313.pyc
  backend
    __pycache__
    routes
    venv
    main.py
    models.py
    requirements.txt
  frontend
    lotties
    app.py
    requirements.txt
  routes
    __pycache__
```



```
˅ routes
  › __pycache__
  analytics.py
  chat.py
  disease.py
  treatment.py
˅ venv
  › etc
  › Include
  › Lib
  › Scripts
  › share
  .gitignore
  pyvenv.cfg
```



```
    .gitignore
    pyvenv.cfg
  download_flan_t5.py
  main.py
  README.md
  requirements.txt
  $ start.sh
```

Running the Application

# To Start the backend

$ cd backend

$ uvicorn main:app --reload

# In new terminal, run the frontend

$ streamlit run app.py

API Documentation

Endpoint	Description	Request Format	Response
/disease/predict	Predict disease based on symptoms	{ "symptoms": "fever, cough" }	{ "prediction": "flu" }
/treatment/plan	Generate treatment plan	{ "condition": "diabetes" }	{ "plan": "Eat healthy, take metformin..." }
/analytics/vitals	Analyze health metrics	{ "heart_rate": [...], "bp": [...], ...}	{ "status": "ok" }
/chat/message	Patient chat response	{ "message": "What to do for cold?" }	{ "reply": "Drink warm fluids..." }

Authentication

Currently, no user authentication is implemented.

- Future versions may use JWT tokens or OAuth for secure login.
- Session tracking is also a potential feature.

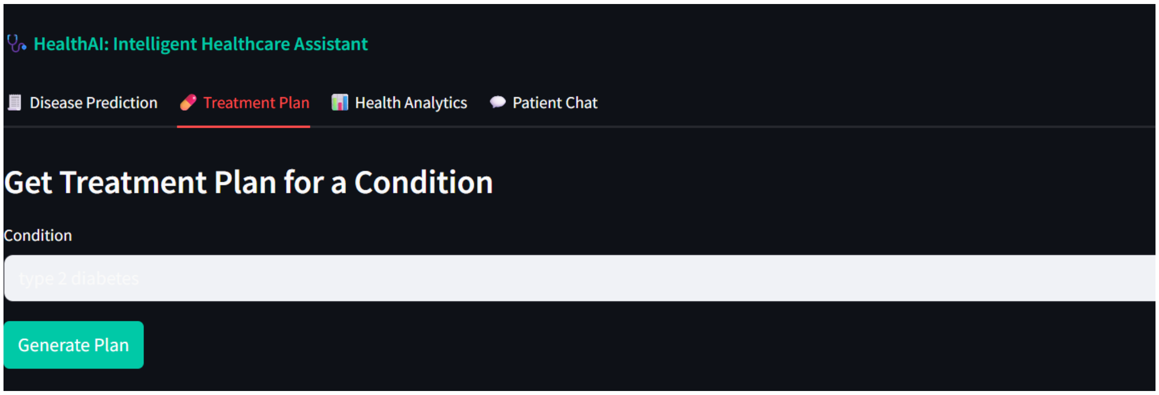
User Interface

- Built using Streamlit Tabs.
- Clean light theme with wide layout.
- Vitals entered as text input are visualized using st.line_chart().
- All interactions are real-time and responsive.

Testing

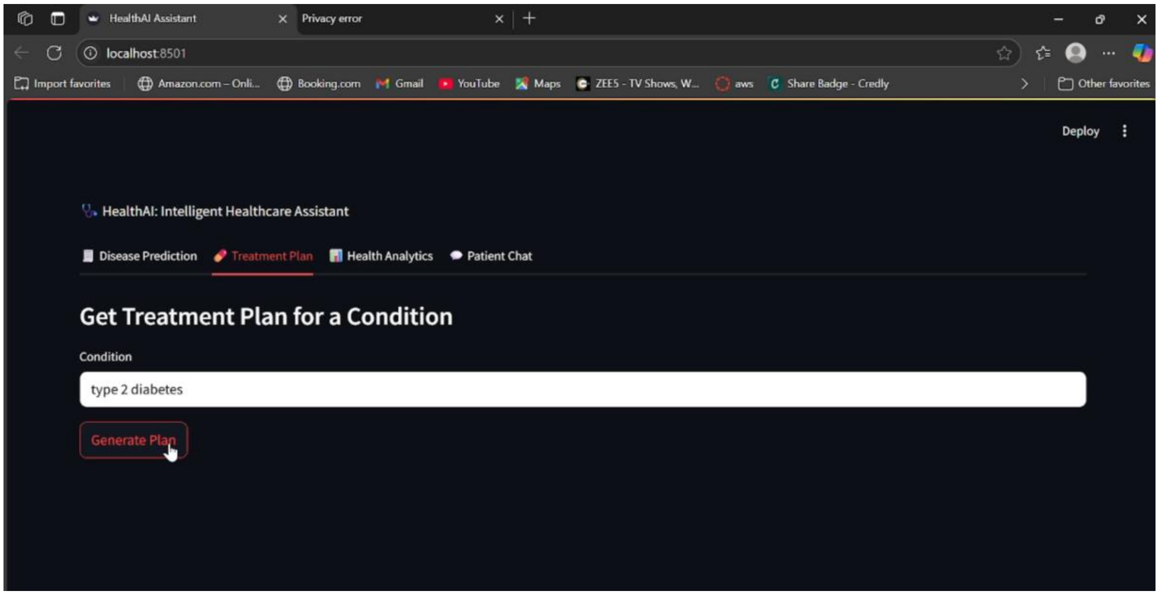
- Manual testing via Streamlit UI.
- Backend endpoints tested via curl and Postman.
- Future scope includes pytest for automated tests.

Screenshots / Demo



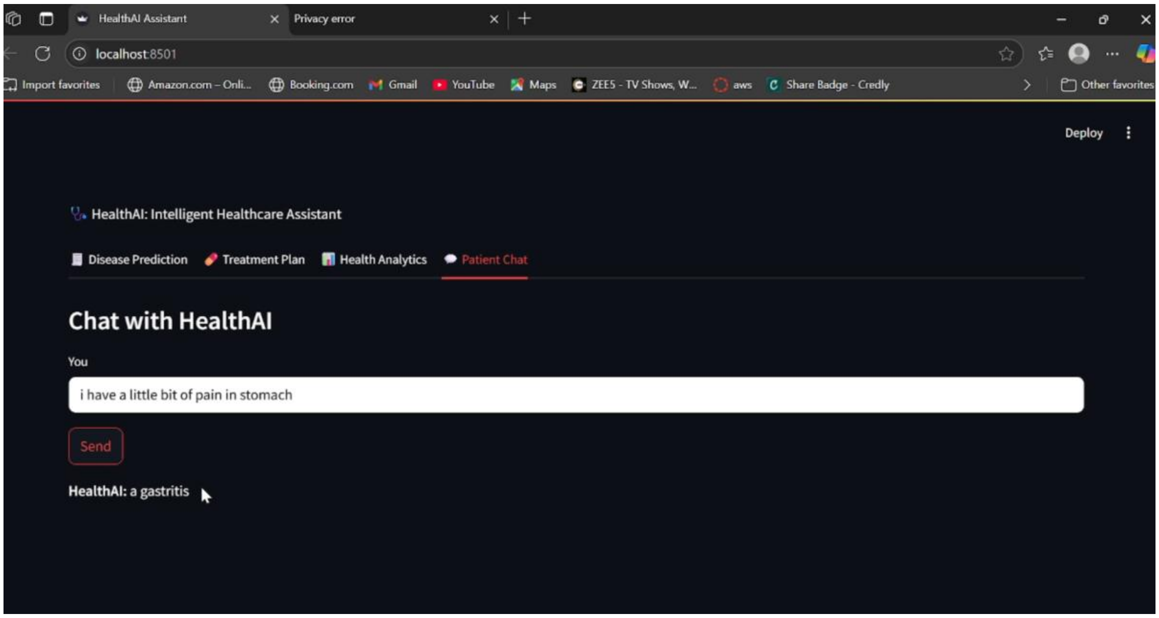
Get Treatment Plan for a Condition

Condition: type 2 diabetes



Get Treatment Plan for a Condition

Condition: type 2 diabetes



HealthAI: Intelligent Healthcare Assistant

Chat with HealthAI

You

i have a little bit of pain in stomach

HealthAI: a gastritis

Demo video link:
https://drive.google.com/file/d/140BEIlf47li3kDMIjwM74yNKPzh6rujM/view?usp=sharing

Known Issues

- No authentication or login flow.
- Input validation is minimal.

Future Enhancements

- Add user login and secure session
- Support for audio/voice symptoms input
- Medication reminders
- Emergency alert system