YourMedicAI: Healthcare Chatbot with Lab Report Analysis

Abstract

This document provides an overview of the **YourMedicAI: Healthcare Chatbot with Lab Report Analysis** project. The system helps users interact with a healthcare assistant, manage their medical history, and upload lab reports for analysis and medical history updates.

1 Introduction

YourMedicAI is an advanced healthcare chatbot that assists users with medical-related inquiries and allows them to upload lab reports for analysis. The system leverages artificial intelligence to understand the user's medical history and provide relevant health information. It also supports uploading and analyzing lab reports, enabling users to keep track of their medical data.

The system is powered by:

- **FastAPI** for building the backend API.
- **JavaScript, HTML, and CSS** for the frontend interface.
- **Python (with libraries like OpenCV and PIL)** for handling lab report analysis.

Users can interact with the chatbot, get personalized responses, and upload lab reports to update their medical history.

2 Features

- **User Authentication and Medical History:** Users can input and manage their name, age, and medical history during the login phase. This information is used throughout their interactions with the chatbot.
- **Chatbot Interaction:** Users can ask healthcare-related questions, and the chatbot provides responses based on the user's medical history and health-related queries.
- **Lab Report Upload and Analysis:** Users can upload lab reports (such as PDFs or images), which are then analyzed by the system. The analysis results are used to update the user's medical history, helping them maintain an up-to-date record.

3 Project Setup

3.1 Requirements

The following software is required to run the project:

- Python 3.8 or higher
- FastAPI for building the backend
- Uvicorn to run the FastAPI app
- JavaScript, HTML, and CSS for frontend development
- Libraries for lab report analysis (e.g., OpenCV, PIL)
- Any compatible web browser for running the frontend

3.2 Installation

To set up the project, follow these steps:

1. Clone the repository:

```
git clone <repository-url>
```

2. Navigate to the project directory:

```
cd YourMedicAI
```

3. Create and activate a Python virtual environment:

```
python -m venv gen_env
source gen_env/bin/activate  \text{(Linux/Mac)}
gen_env\Scripts\activate \text{(Windows)}
```

4. Install project dependencies:

```
pip install -r requirements.txt
```

3.3 Running the Application

To start the FastAPI backend, use the following command:

```
uvicorn chatapi:app --reload
```

Once the backend is running at `http://localhost:8000`, you can interact with the frontend by opening the `index.html` and `chatbot.html` files in your web browser.

4 Project Structure

- `chatapi.py` : FastAPI backend code that handles the chatbot logic and lab report uploads.
- `index.html` : Login page where users input their name, age, and medical history.
- `chatbot.html` : The chatbot page where users interact with the chatbot and upload lab reports.
- `styles.css` : CSS file for styling the HTML pages.
- `script.js` : JavaScript file for handling frontend logic, such as user inputs and AJAX requests to the backend.
- `requirements.txt` : A file listing all the dependencies needed to run the backend.

5 Usage

5.1 User Flow

- **Step 1:** The user opens the `index.html` file, enters their name, age, and medical history, and submits the form.
- **Step 2:** The user is redirected to the `chatbot.html` page, where they can interact with the chatbot.
- **Step 3:** The user can upload lab reports using the "Upload Lab Report" button. After uploading, the system analyzes the report and updates their medical history.

6 Lab Agent Functionality

The **Lab Agent** feature enables users to upload lab reports (PDF or image formats). The backend analyzes the lab report to extract key medical information and update the user's medical history. This helps in tracking the user's health over time.

The lab report analysis process follows these steps:

- The user uploads a lab report via the frontend.
- The backend uses various Python libraries to analyze the report.
- Based on the analysis, the user's medical history is updated with the extracted data.

7 Conclusion

YourMedicAI provides a unique and personalized healthcare experience. It combines a chatbot interface for healthcare queries with an advanced lab report analysis feature to keep users' medical histories up-to-date. By leveraging FastAPI, JavaScript, and AI-based analysis techniques, the system offers a robust solution for managing healthcare data.

This project showcases how modern technologies can empower individuals to take control of their health by simplifying interactions with healthcare systems and improving the accuracy of medical records.