AI Virtual Doctor: Project Details & Documentation

Project Abstract

The AI Virtual Doctor is an interactive web application designed to provide preliminary health assessments based on user-reported symptoms and uploaded medical images. Built with Streamlit and powered by Google's Gemini AI, this application serves as a first-line health information tool, offering symptom analysis, potential condition identification, and care recommendations. The application emphasizes its supplementary role and includes clear disclaimers that it cannot replace professional medical advice or diagnosis. It features a chat-based interface where users can describe their symptoms, upload or paste images of visible symptoms, and receive AI-generated health assessments in a structured, informative format.

Key Packages & Technologies

Streamlit (`streamlit`)

• \*\*Purpose\*\*: Provides the web application framework and user interface components

• \*\*Usage\*\*:

• Creates the responsive layout and interactive elements

• Handles session state management to maintain conversation history

• Implements the chat interface with `st.chat\_input()` and `st.chat\_message()`

• Manages file uploads with `st.file\_uploader()`

• Renders markdown and HTML components

Google Generative AI (`google.generativeai`)

• \*\*Purpose\*\*: Powers the AI conversation and image analysis capabilities

• \*\*Usage\*\*:

• Connects to Google's Gemini 2.0 Flash model via API

• Processes both text input and image data

• Generates structured health responses following the system prompt guidelines

• Maintains chat history for contextual awareness in responses

Pillow (`PIL`)

• \*\*Purpose\*\*: Image processing and manipulation

• \*\*Usage\*\*:

• Handles uploaded and pasted image files

• Converts between different image formats

• Prepares images for display in the Streamlit interface

• Processes images before sending to the Gemini API

Base64 (`base64`)

• \*\*Purpose\*\*: Encoding and decoding for image data transfer

• \*\*Usage\*\*:

• Decodes pasted images from base64 format

• Handles the conversion between text-based image representation and binary data

• Enables clipboard image functionality

IO (`io`)

• \*\*Purpose\*\*: Handles binary stream operations

• \*\*Usage\*\*:

• Creates in-memory file-like objects with `BytesIO`

• Facilitates conversion between binary data and Pillow image objects

• Processes uploaded and pasted image data streams

HTML Component (`streamlit.components.v1`)

• \*\*Purpose\*\*: Enables custom HTML/JavaScript functionality

• \*\*Usage\*\*:

• Creates the image paste and drag-and-drop component

• Provides client-side interactivity for the image input

• Communicates between JavaScript and Python using Streamlit's component API

Core Features

1. Symptom-Based Health Assessment

• Users can describe their symptoms in natural language

• AI analyzes symptoms and provides structured responses

• Assessment includes possible conditions, care recommendations, and specialist suggestions

2. Multimodal Image Analysis

• Three methods to add images:

• Traditional file upload

• Direct clipboard paste (Ctrl+V/Cmd+V)

• Drag and drop from local storage

• Images can be analyzed independently or in conjunction with text descriptions

• AI provides visual assessment with appropriate cautions about limitations

3. Structured AI Responses

All AI responses follow a consistent format with sections for:

• Symptom analysis

• Possible conditions (listed by likelihood)

• Care recommendations (self-care or medical attention)

• Specialist recommendations

• Warning signs to watch for

• Limitations of the assessment

4. Conversational Memory

• Application maintains chat history throughout the session

• AI responses are contextually aware of previous exchanges

• Conversation style enhances user engagement and information retention

5. Medical Safeguards

• Clear disclaimers about limitations of AI medical advice

• System prompts designed to prioritize patient safety

• Automatic recommendation for professional care for serious symptoms

• Explicit warnings about the limitations of image-based analysis

6. User-Friendly Interface

• Clean, medical-themed UI with intuitive navigation

• Tabbed interface for different image input methods

• Real-time feedback for image uploads and processing

• Mobile-responsive design

Technical Implementation Details

System Architecture

• Frontend and backend integrated through Streamlit

• Stateless design with session-based persistence

• Client-side JavaScript for enhanced image handling

• Cloud-based AI processing via Google's Gemini API

AI Model Configuration

• Uses Gemini 2.0 Flash model for balance of speed and accuracy

• Specialized system prompt focusing on medical information provision

• Multi-turn conversation capabilities with history tracking

• Multimodal processing (text + images)

Data Flow

1. User inputs text and/or images via the interface

2. Images are processed, normalized, and converted to appropriate formats

3. Inputs are sent to the Gemini API along with conversation history

4. AI generates structured response following medical guidelines

5. Response is displayed to user and added to conversation history

Security & Privacy Considerations

• No permanent storage of user medical information

• Session-based data handling (cleared on browser refresh)

• No user authentication required (anonymous usage)

• Client-side image processing where possible

• Medical disclaimer emphasizing limitations

Use Cases & Limitations

Intended Uses

• First-line health information for common symptoms

• Educational tool for understanding potential conditions

• Triage guidance for deciding whether to seek medical care

• Assistance in remote or underserved areas with limited healthcare access

Explicit Limitations

• Not a diagnostic tool or substitute for professional medical care

• Limited accuracy for image-based analysis

• Cannot access personal medical records or history

• No capability for physical examination

• Limited to information available in the AI's training data

• Potential for misunderstanding complex symptom descriptions

Future Enhancement Possibilities

• Integration with telemedicine platforms

• Multi-language support for global accessibility

• Improved image analysis with specialized medical imaging models

• Optional user accounts for maintaining medical history

• Integration with health monitoring devices and wearables

• Expanded specialist information and referral capabilities

Technical Requirements

• Python 3.7+

• Streamlit 1.10+

• Google Generative AI API access (requires API key)

• Web browser with JavaScript enabled

• Internet connection for API communication