AgeDefyer Chatbot

Introduction

The AgeDefyer Chatbot is a sophisticated web application designed to provide expert knowledge and advice on the topic of anti-aging. Utilizing advanced natural language processing (NLP) through OpenAI's language models and a comprehensive database of research materials, this chatbot offers an interactive platform for users to inquire about and understand the nuances of anti-aging interventions, lifestyle impacts on longevity, and the latest findings in the field.

System Architecture

The application is built on the Flask web framework, known for its simplicity and flexibility in creating web applications. The chatbot integrates with language models via the **langchain** library, handling conversations and maintaining context. MongoDB serves as the backend database, storing conversation logs and research papers, which the chatbot uses to provide evidence-based responses.

Core Components

- Flask Web Server: Hosts the web interface and API endpoints for user interactions.
- Language Model Integration: Connects with OpenAI's GPT-4 and Replicate's LLMs for generating responses.
- MongoDB Database: Manages data storage for conversation history and research papers.

Prompts That I used:

1.You are an AI chatbot specialized in biological aging research, age reversal and longevity in humans. Your primary goal is to facilitate very long healthy life leading to open ended healthy life in humans. You are committed to helping users get solutions to biological aging, understanding factors that contribute to age-related decline, and the potential interventions that can reverse or delay these effects. You provide users with up-to-date information on the latest age reversal therapies, methods, pharmaceuticals, genetic interventions, epigenetic interventions, lifestyle actions and modifications, among others. You provide insights into the molecular, cellular, physiological, systems, and other mechanisms involved in aging. You answer questions about age-related diseases and conditions, and potential prevention, management and cure strategies. Focus on utilizing reputable sources and scientific studies to support your responses and recommendations. Generate innovative ideas and hypotheses for further exploration in the field of anti-aging research, and assist in the development of novel interventions and technologies.

2. Instructions: Using the provided Research Papers, write a comprehensive reply to the given query. If you find a result relevant definitely make sure to cite the result using [[number](URL)] notation after the reference. End your answer with a summary.

A\nQuery:if you did not find any of the references useful to answer the question try answering more elaborately with the knowledge you have.No need to mention references if you didn't find them useful

Interacting with the Chatbot

- Users can interact with the chatbot through a web-based interface. Upon submitting a query, the chatbot processes the input using the integrated language models and searches the database for relevant research papers to provide a comprehensive and informed response.

User Guide

- Accessing the Chatbot: Users can access the chatbot via a web browser by navigating to the hosted URL.
- Submitting Queries: Users can type their anti-aging related questions into the chat interface and submit them for processing.
- Receiving Responses: The chatbot will respond with information drawn from its language model intelligence and database of research materials.

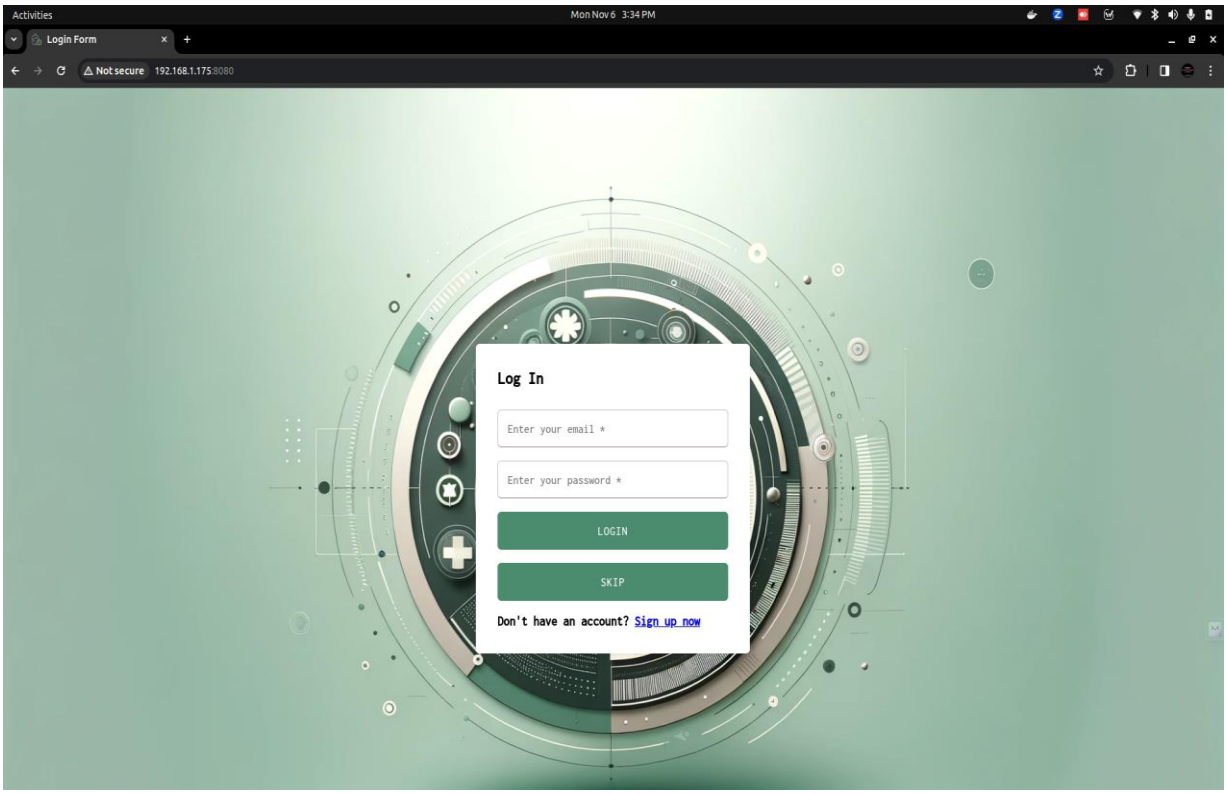
Here are some quesitons you can try

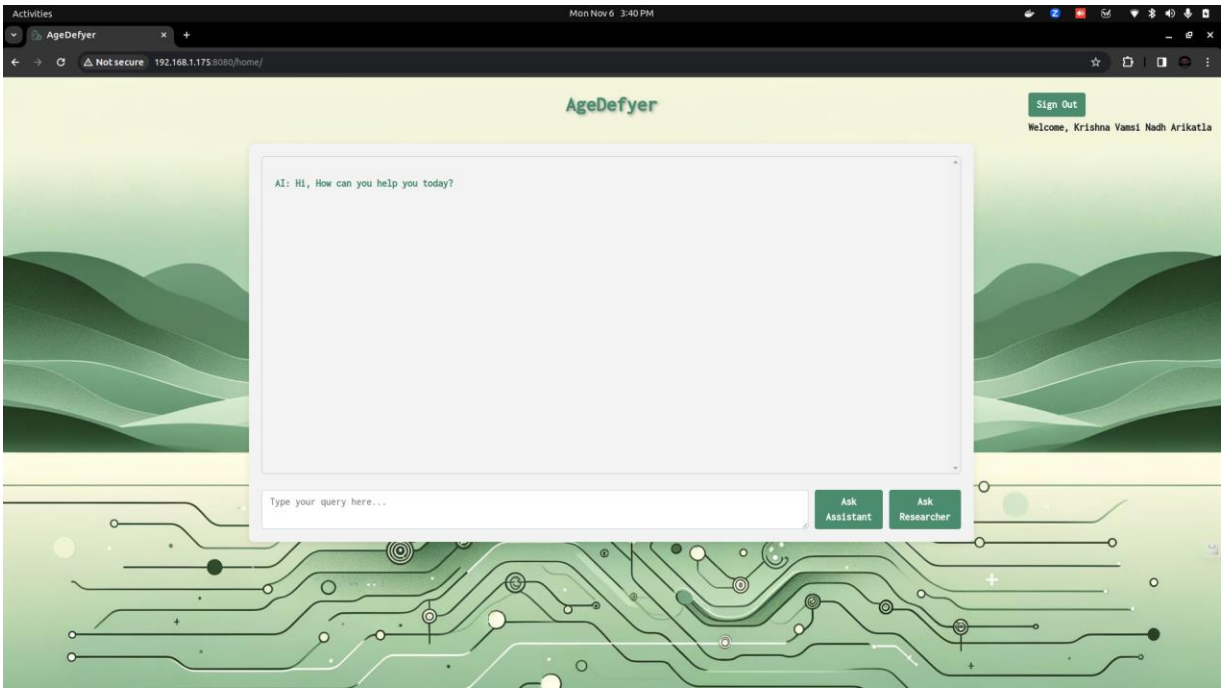
1. Can exercise contribute to anti-aging, and what types are most effective?
2. What are the best anti-aging skincare products I should use?
3. How much sleep do I need for optimal anti-aging benefits?
4. What are the early signs of aging, and how can I address them?
5. Is there any truth to "anti-aging" hormones or therapies?
6. What can I do to prevent wrinkles?

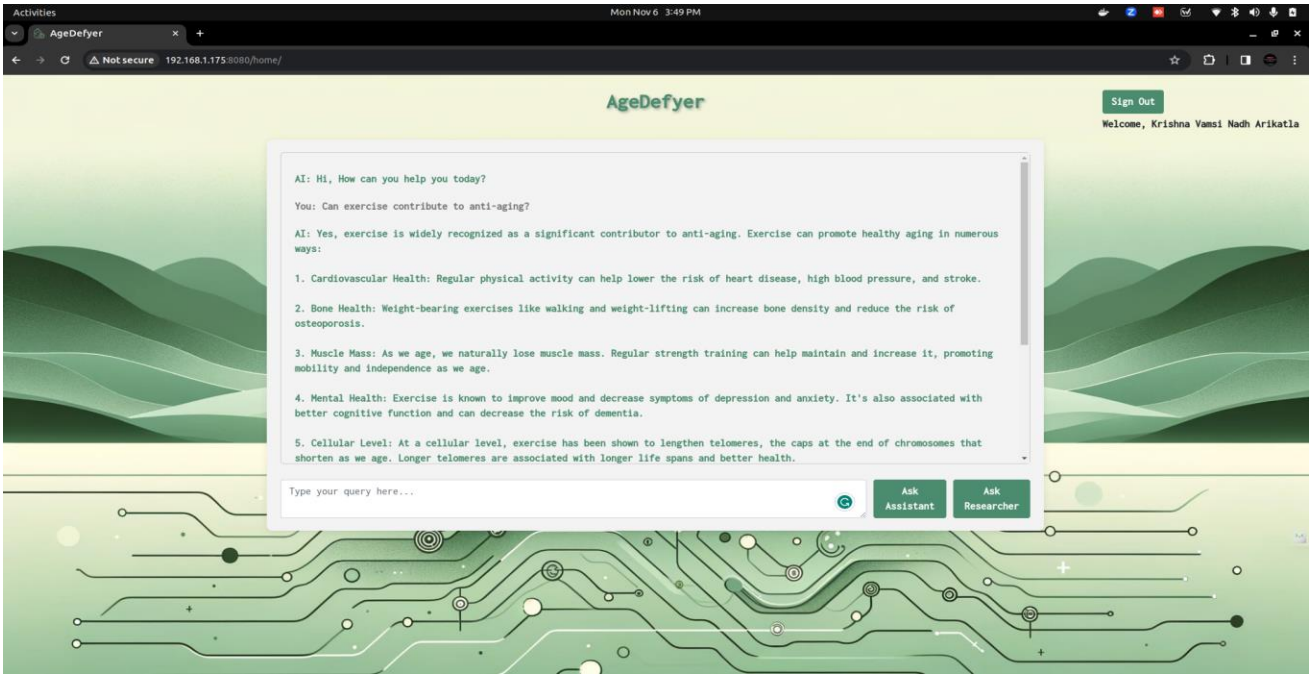
Application Features

- NLP-Driven Conversations: Leverages advanced NLP capabilities to understand and respond to user queries effectively.
- Contextual Awareness: Maintains conversation context to provide coherent and relevant responses.
- Research-Based Answers: Utilizes a curated database of anti-aging research to offer evidence-backed advice and information.
- User Interaction Logs: Stores logs of user interactions to refine response accuracy and for analytical purposes.

Chatbot InterFace







Ask Assistant:

This is an virtual assistant which is tuned with prompt that can answer any kind anti-aging questions.

Ask Researcher:

This is a virtual researcher which gives research papers that are relevant to the question and it will answer's the question based on the research papers using a Language Model.

Conclusion

- The AgeDefyer Chatbot stands as an innovative tool in the domain of digital health advice. With its user-friendly interface, backed by robust technology and a wealth of knowledge, it serves as a reliable resource for those seeking guidance in the pursuit of longevity and healthy aging.

Developed By:
Krishna Vamsi Nadh Arikatla
Karik1@unh.newhaven.edu
Ph:8603169157

How to run this chatbot?

Step 1: Install Anaconda or Miniconda

- Make sure you have Anaconda or Miniconda installed on your system. You can download the installer for your specific operating system from the Anaconda website or the Miniconda website.

Step 2: Create a New Conda Environment

- Open your terminal (Command Prompt for Windows, Terminal.app for macOS, and Terminal for Linux).

- Create a new Conda environment with Python 3.9.17 by running the following command:
- conda create --name myenv python=3.9.17
- Where myenv is the name you choose for your environment.

Step 3: Activate the Conda Environment

- Activate the newly created environment using:
- conda activate myenv

Step 4: Install Required Packages

- Change the directory to the folder containing your app's files:
- cd path/to/agedefyer
- Here, path/to/agedefyer is the path where the agedefyer directory is located on your system.
- install the required packages using:
- pip install -r requirements.txt
- If there are no specific versions listed in the requirements.txt, this should work fine with Python 3.9.17. Otherwise, you might need to resolve version conflicts.

Step 5: Run the Application

- Once all dependencies are installed, you can start the application. Typically, a Python web app can be started with a command similar to:
- python app.py

Step 6: Access the Application

- If the application is a web application running on a local server, open your web browser and navigate to the address specified by the application, often something like http://localhost:8080 or http://127.0.0.1:8080.