# LLM FITNESS CHAT BOT

**Overview of the project**

fitness chatbot is an AI-powered virtual assistant designed to help individuals achieve their health and fitness goals. Whether you're looking to lose weight, build muscle, or adopt a healthier lifestyle, our chatbot provides personalized guidance, support, and motivation every step of the way

**Motivation**

I built this fitness chatbot to address the common challenges and barriers people face when striving to lead a healthier lifestyle. With busy schedules, conflicting information, and lack of personalized guidance, many individuals struggle to stay motivated and on track with their fitness goals

**Audience**

The intended users of our fitness chatbot are individuals of all fitness levels and backgrounds who are looking to improve their health and well-being. Whether you're a beginner just starting your fitness journey or an experienced athlete seeking new challenges, our chatbot is designed to meet you where you are and provide personalized support and guidance to help you succeed. Whether you're looking for workout routines, nutritional advice, accountability, or motivation, our chatbot is here to support you on your journey to a healthier, happier you.

**Summary**

A streamlit application built using streamlit, langchain, and the open AI API.

The application has three main features including:

1. **Workout plan generator:**

Provide some information and submit the form to generate a workout plan.

1. **Answer fitness questions:**

Select a common fitness and health related question and submit to recieve an

answer.

1. **Diet Plan generator:**

Customize diet plan

**Workout of the day:** Select a preference, and a random workout of the day will be provided to you.

**Installation and Setup**

python 3.11.5 python -m venv venv cd venv/scripts activate

pip install -r requirements.txt

streamlit run 1\_Workout\_Plan\_Generator.py

**API key management: OPEN AI API KEY**

There are two ways to use the app.

1 - Use the public version hosted on streamlit cloud.

2 - Clone the repository, and provide the key in the secrets.toml file which should

be placed in the .streamlit folde or store a key in vairable name “open\_ai\_ key”

**Architecture**

The architecture of the chatbot can be broken down into several components:

**1.Streamlit Frontend:**

The interface is built using Streamlit, a Python library for creating web applications with minimal effort.

It provides elements like buttons, text inputs, number inputs, and form submission functionality to create an interactive user interface.

**2.Backend Interaction**

The backend is responsible for handling user inputs, validating API keys, and generating workout plans.

It interacts with OpenAI's GPT-3.5 model through the `langchain` package, particularly using `ChatOpenAI` and `LLMChain` classes.

**3.OpenAI GPT-3.5 Model**

The `ChatOpenAI` class encapsulates interactions with the OpenAI GPT-3.5 model for generating text-based responses to user prompts.

It requires an API key to authenticate with the OpenAI service.

**4.Prompt Template**

A `PromptTemplate` object is used to define the structure of prompts to be passed to the GPT-3.5 model.

It contains placeholders for variables that will be filled with user-provided data.

**5. LLMChain**

The `LLMChain` class manages the chaining of prompts and responses for generating coherent and context-aware output.

-It takes the instantiated `ChatOpenAI` object and a prompt template as inputs.

**6. User Input Handling**

User inputs such as workout parameters (e.g., program duration, number of sessions per week, maximum lift weights, etc.) are collected through Streamlit elements.

These inputs are then passed to the GPT-3.5 model for generating personalized workout plans same for diet plan

**7. Output Presentation**

Generated workout plans are displayed back to the user through the Streamlit interface.

Output is presented in a readable format, likely as text, indicating the weekly workout plans.

**9. State Management**

Streamlit's session state management is used to store and manage the state of the application, including whether the program generation process has been initiated.

Overall, the architecture enables users to input their workout preferences, personalized workout plans generated by the GPT-3.5 model, with the interaction facilitated through a Streamlit-powered web interface.

**AI and NLP Techniques**

1. **Prompt-based Generation**

The core of the chatbot's functionality lies in generating responses based on prompts. Prompts are predefined text templates that guide the AI model in understanding the user's inputs and generating relevant responses. The `PromptTemplate` class is used to define these prompts.

1. **Language Model (GPT-3.5)**

The chatbot employs a pre-trained language model (LLM) based on OpenAI's GPT-3.5 architecture. This model is responsible for understanding the prompts and generating appropriate workout plans based on the provided inputs.

1. **Entity Recognition**

The chatbot extracts relevant information from the user's input, such as the duration of the workout plan (weeks), the number of sessions per week, 1-rep max values for different exercises (squat, bench press, deadlift), and the number of accessory exercises per session. These extracted entities are used to personalize the generated workout plan.

1. **Dialogue Management**

The chatbot manages the dialogue flow by sequentially generating workout plans for each week based on the user's inputs. It keeps track of the current week and adjusts the program's difficulty level as requested by the user.

**Data sources and APIs**

**1. OpenAI API (GPT-3.5)**

This is the primary API used for natural language processing and generation.

The `openai` Python package is used to interact with the OpenAI API.

The API key is obtained from the user via the Streamlit interface.

The API key is validated by making a call to the OpenAI API using the

**2. Streamlit**

Streamlit is used as the framework for building the interactive web application.

It provides various components for building user interfaces, including buttons, text inputs, number inputs, and forms.

Streamlit's session state is utilized to store information such as the API key and user input across different interactions.

**3. External Data Sources**

There are no external data sources used. The chatbot generates workout plans purely based on user input and the capabilities of the GPT-3.5 model.

**4. Data Retrieval, Processing, and Usage**

User input is collected via Streamlit components such as text inputs, number inputs, and select boxes.

The collected input parameters are then used to construct prompts/templates for the GPT-3.5 model.

The GPT-3.5 model generates workout plans based on the provided prompts and input parameters.The generated workout plans are displayed to the user via the Streamlit interface.