**FitSync AI: Real-Time Fitness Adjustments with LLaMA3**

**Project Description:**

Fitness journeys are deeply personal, and finding a program that adapts to your evolving needs can be challenging. Many fitness enthusiasts struggle to maintain motivation or see progress because their routines become stagnant or fail to meet their individual goals. There is a growing need for a solution that can provide real-time adjustments and personalized guidance to ensure that fitness routines remain effective and engaging. FitSync AI, powered by LLaMA3, addresses these challenges by offering an adaptive fitness solution that responds to individual progress and preferences, ensuring a dynamic and tailored experience for every user.

**Scenario 1: Real-Time Adjustments for Casual Fitness Enthusiasts**

FitSync AI revolutionizes the fitness experience for casual users by providing real-time adjustments to their workout routines. Users can select their preferred fitness goals, such as weight loss, muscle gain, or overall wellness. The AI continuously monitors their progress, adapting workouts to their current fitness level and preferences. This dynamic approach keeps users engaged and motivated, as they experience consistent progress without the risk of plateauing. FitSync AI ensures that casual fitness enthusiasts always have an effective and enjoyable workout plan tailored to their needs.

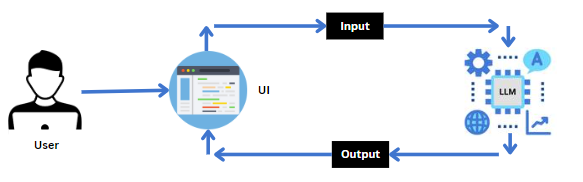
**Scenario 2: Personalized Coaching for Fitness Enthusiasts**

For those who are more serious about their fitness goals, FitSync AI offers personalized coaching that enhances their performance and results. Users input their workout data, and the AI provides tailored advice on optimizing their routines. Whether it's suggesting new exercises, adjusting intensity, or offering recovery tips, FitSync AI ensures that users are always on track to achieve their fitness goals. This personalized guidance helps fitness enthusiasts refine their techniques and reach new heights in their fitness journey.

**Scenario 3: Customized Fitness Content for Trainers and Influencers**

FitSync AI can also be a valuable tool for fitness trainers and influencers by generating customized content for their audience. The AI can create workout plans, fitness tips, and motivational content based on the latest trends and user preferences. By integrating FitSync AI into their platforms, trainers and influencers can offer personalized content that resonates with their audience, helping them stay relevant and maintain engagement. This automation allows fitness professionals to focus on building their brand and expanding their reach, while still providing high-quality content to their followers.

**Architecture:**



**Project Flow:**

* Users enter their desired inputs like fitness goals , workout types etc to stay into the Streamlit UI. Additional preferences or interests can also be specified if needed.
* The input details are sent to the FitnessAI backend, which utilizes the generative AI model to process the information.
* The AI model processes the user’s input to generate a detailed and personalized fitness plan based on the specifications given by the user.
* The AI autonomously creates a well-structured and engaging fitness guide, including tips and diet to followed by the user.
* The generated sheet is sent back to the frontend of the Streamlit app for display to the user.
* Users can review the generated itinerary, make additional customizations if desired, and either export or copy the content for their fitness planning.

To accomplish this, we have to complete all the activities listed below,

* Initialize Gemini Pro LLM:
  + Generate Gemini Pro API
  + Initialize the pre-trained model
* Interfacing with Pre-trained Model
  + Fitnessplan Generation
* Model Deployment
  + Deploy the application using Streamlit

**Prior Knowledge:**

You must have prior knowledge of the following topics to complete this project.

* LLM & Gemini Pro:

A **large language model** is a type of artificial intelligence algorithm that applies neural network techniques with lots of parameters to process and understand human languages or text using self-supervised learning techniques. Tasks like text generation, machine translation, summary writing, image generation from texts, machine coding, chat-bots, or Conversational AI are applications of the Large Language Model. Examples of such LLM models are Chat GPT by open AI, BERT (Bidirectional Encoder Representations from Transformers) by Google, etc.

<https://llama.meta.com/docs/model-cards-and-prompt-formats/llama3_1>

<https://cloud.google.com/vertex-ai/docs/generative-ai/learn-resources>

* Streamlit:

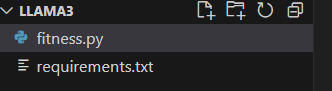
Basic knowledge of building interactive web applications using Streamlit.

Understanding of Streamlit’s UI components and how to integrate them with backend logic.

<https://www.datacamp.com/tutorial/streamlit>

**Project Structure:**

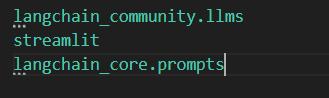
Create the Project folder which contains application file as shown below



**Milestone 1: Requirements Specification**

Specifying the required libraries in the requirements.txt file ensures seamless setup and reproducibility of the project environment, making it easier for others to replicate the development environment.

**Activity 1: Create a requirements.txt file to list the required libraries.**

****

**Activity 2: Install the required libraries.**

****

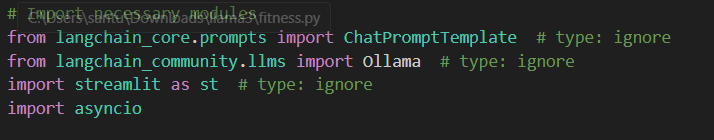
**Milestone 2: Initializing the Model**

For initializing the model we need to generate PALM API.



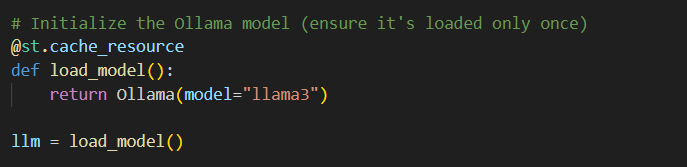
**Activity 1: Initialize the pre-trained model**

**Activity 1.1: Import necessary files**

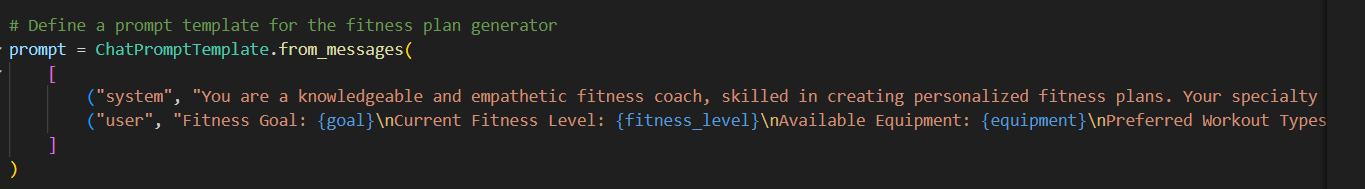


* Streamlit, a popular Python library, is imported as **st**, enabling the creation of user interfaces directly within the Python script.

**Activity 1.2: Define the model to be used**

****

* **Generation Settings:** Configured parameters such as temperature, top\_p, top\_k, max\_output\_tokens, and response\_mime\_type to control the output characteristics.

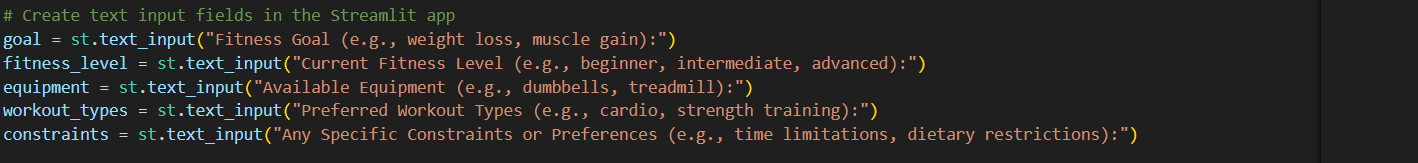


* **Created an instance load\_model with the function named load\_model".**

**Milestone 3: Interfacing with Pre-trained Model**

In this milestone, we will build a prompt template to generate feedback based on the project details entered by the user.

**Activity 1: Create a text input to generate fitness guide**

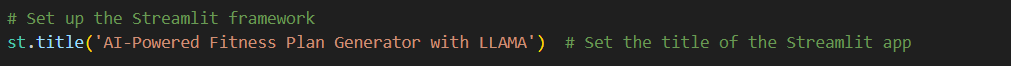
****

* The generate\_itinerary function is designed to create a fitness plan based on the user's specified fitness goal,fitness level , equipment and workout types.
* The function takes one parameters: goal,fitness level, equipment, workout types and constraints.
* It starts a chat session with the AI model using the provided details, initializing the session with a user message that specifies the request for a fitness guide.
* The function sends a detailed message to the chat session to generate the fitness plan, then retrieves and returns the generated content as text.

**Milestone 4: Model Deployment**

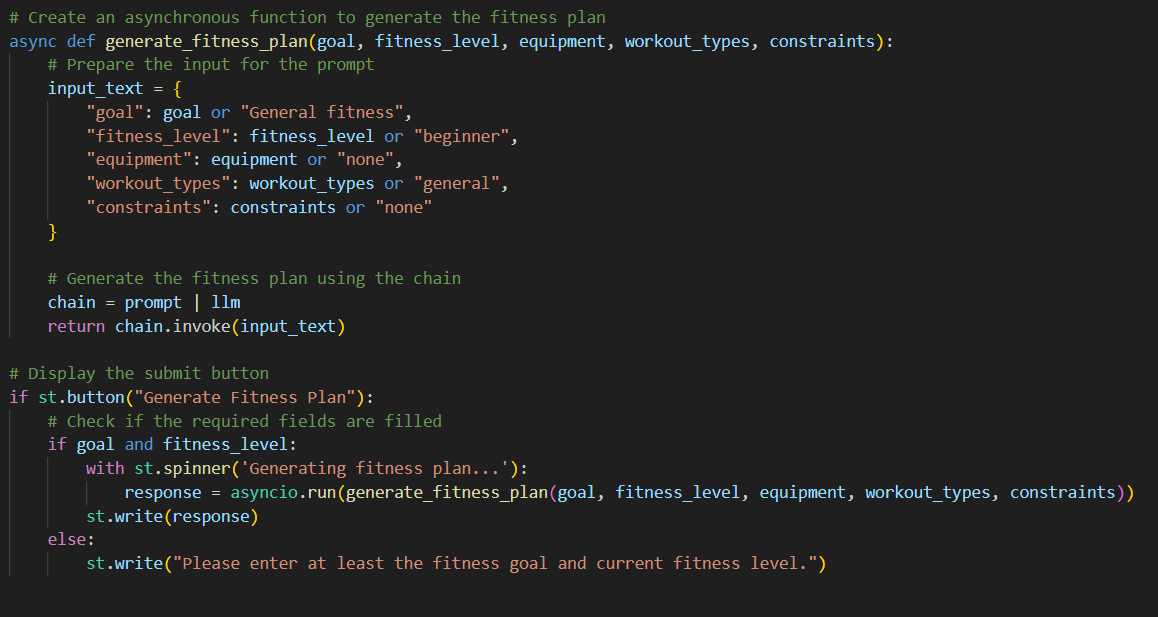
In this milestone, we are deploying the created model using streamlit. Model deployment using Streamlit involves creating a user-friendly web interface, enabling users to interact with the model through a browser. Streamlit provides easy-to-use tools for developing and deploying data-driven applications, allowing for seamless integration of models into web-based applications.

**Activity 1: Give the project title**

****

* The main function to start the Streamlit application, allowing users to interact with the web app.
* st.title("AI-powered fitness plan generator with llama")
* Sets the title of the Streamlit app to "fitness Generator." This title is prominently displayed at the top of the web page, providing users with a clear understanding of the app’s purpose and functionality.

**Activity 2: Create fields for user to input data for generating blog**

****

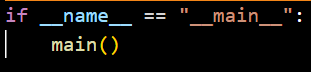
* goal = st.text\_input("Enter your fitness goal:")  
  Creates a text input field labeled "Enter your fitness goal :" where users can type in their goal. The default value is an empty string.
* Fitness\_level: st.text\_input(“enter your level”)

Create a text input field labeled “enter your level” where users can type their level.

* Equipment: st.text\_input(“enter your equipment type”)

Create a text input field labeled “enter your equipment “ where users can type their goal.

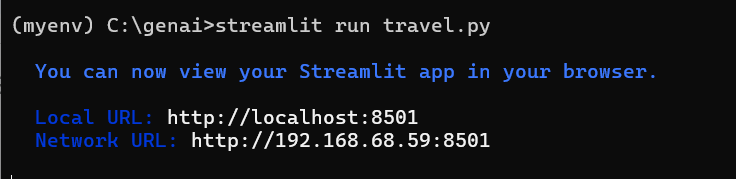
* else:  
  Executes if any input field is empty or invalid.
* st.error("Please enter at least the fitness goal and current fitness level.")  
  Shows an error message asking the user to ensure that all inputs are provided and valid if any field is empty or invalid.



* Finally, the main() function is called to execute the Streamlit app.

**Activity 5: Run the web application**

* Open the anaconda prompt from the start menu
* Navigate to the folder where your Python script is.
* Now type “streamlit run app.py” command
* Navigate to the localhost where you can view your web page

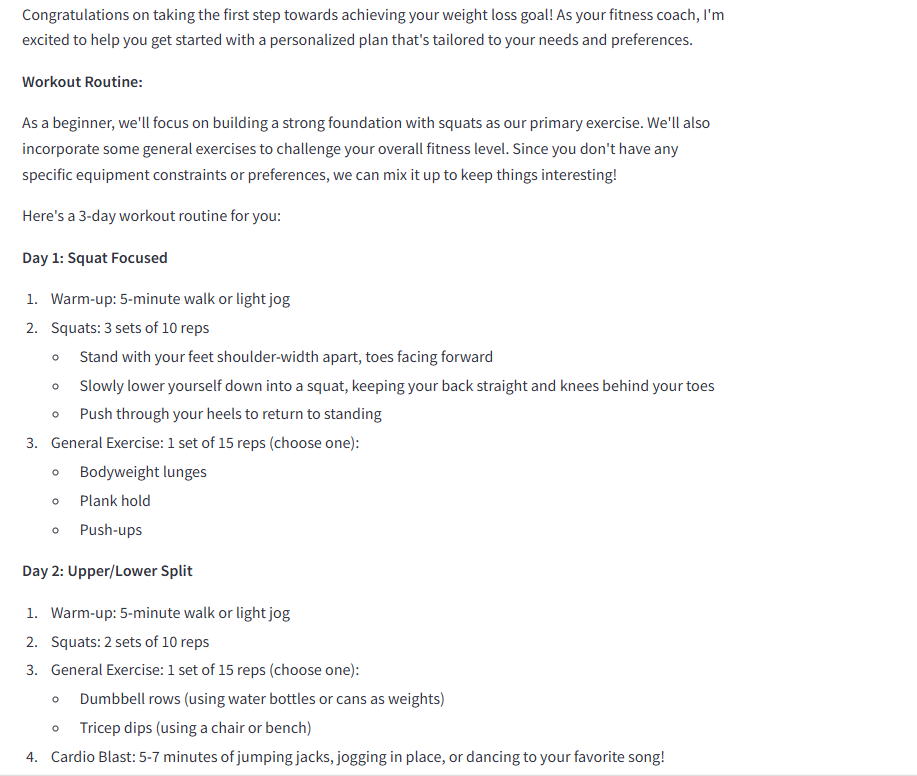


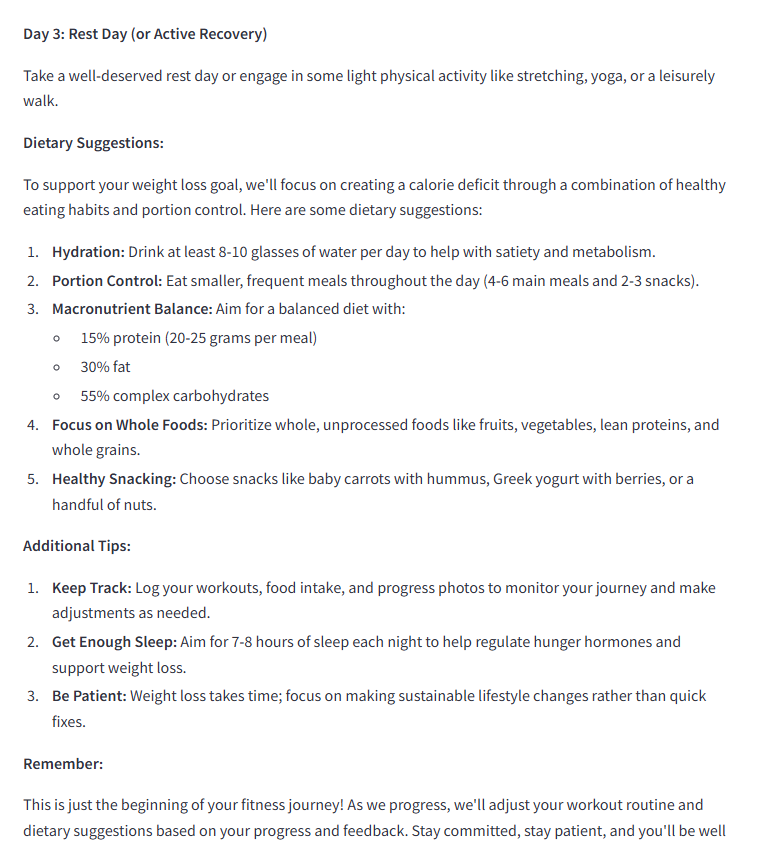
Now, the application will open in the web browser,

After giving the input:



The Output generated:





**Conclusion:**

This project is an innovative web application designed to elevate the board game experience through advanced AI technology. By integrating Streamlit with a sophisticated AI model, the application allows users to engage with intelligent board game opponents and receive strategic guidance tailored to their gameplay. The system ensures an interactive and immersive experience by adapting to player skill levels, offering real-time strategy advice, and generating dynamic game scenarios. This project highlights the transformative potential of AI in enhancing board game enjoyment and strategy, providing both casual players and enthusiasts with a seamless and personalized gaming experience.