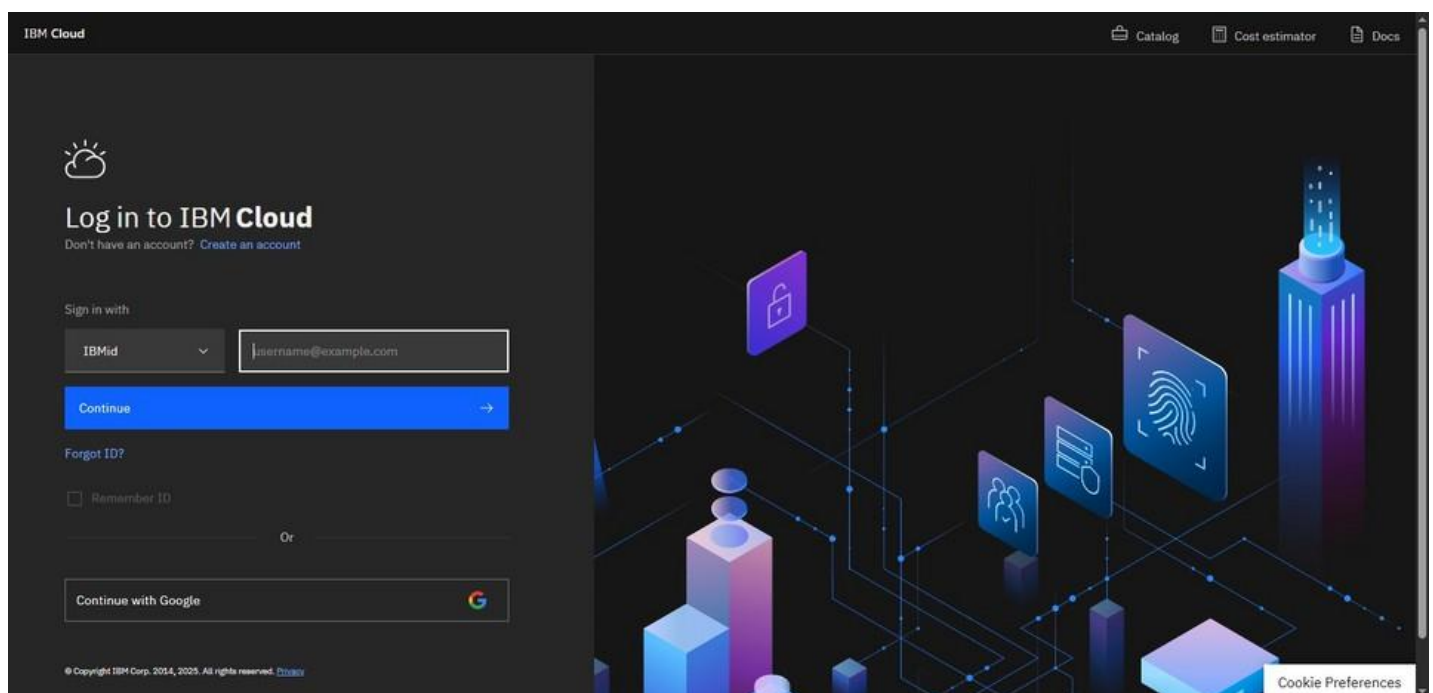


Name = Dhiraj Ashok Jadhav

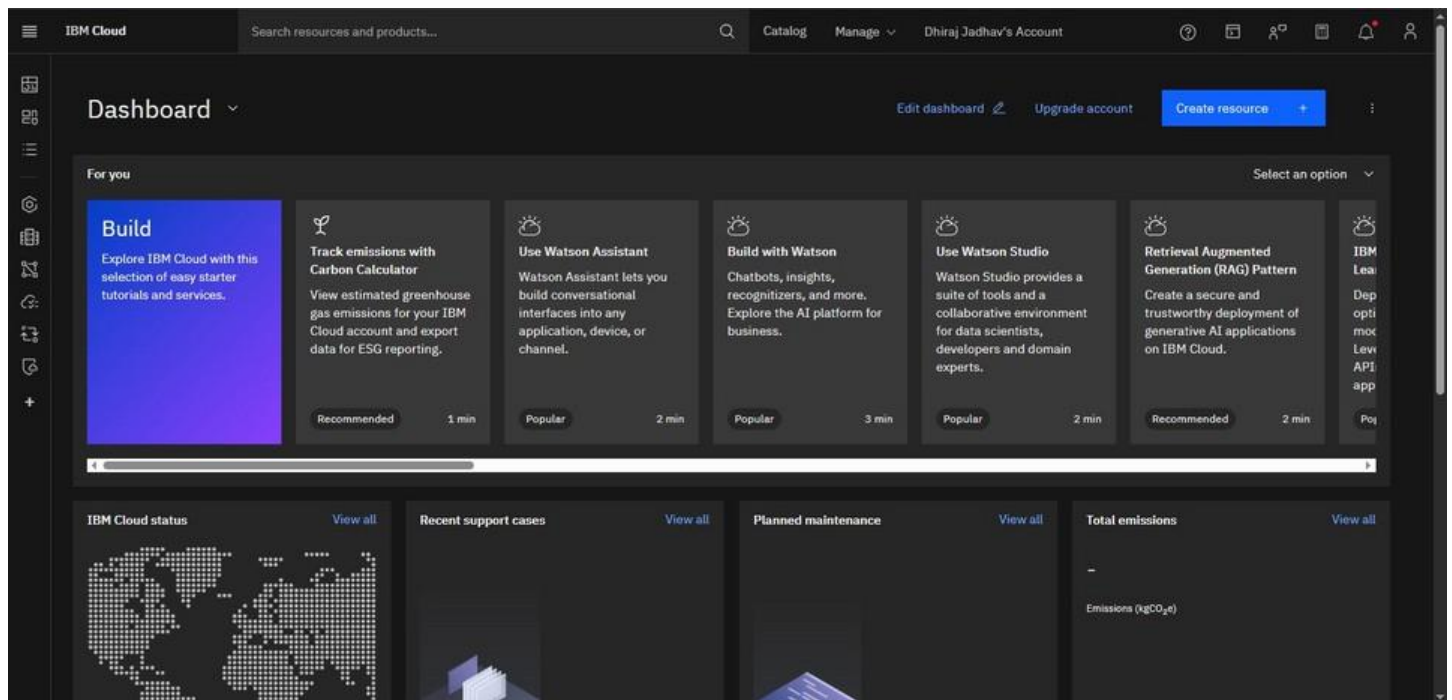
Problem Statement = Fitness Buddy

Technology: Use of IBM cloud lite services /IBM  
Granity is mandatory

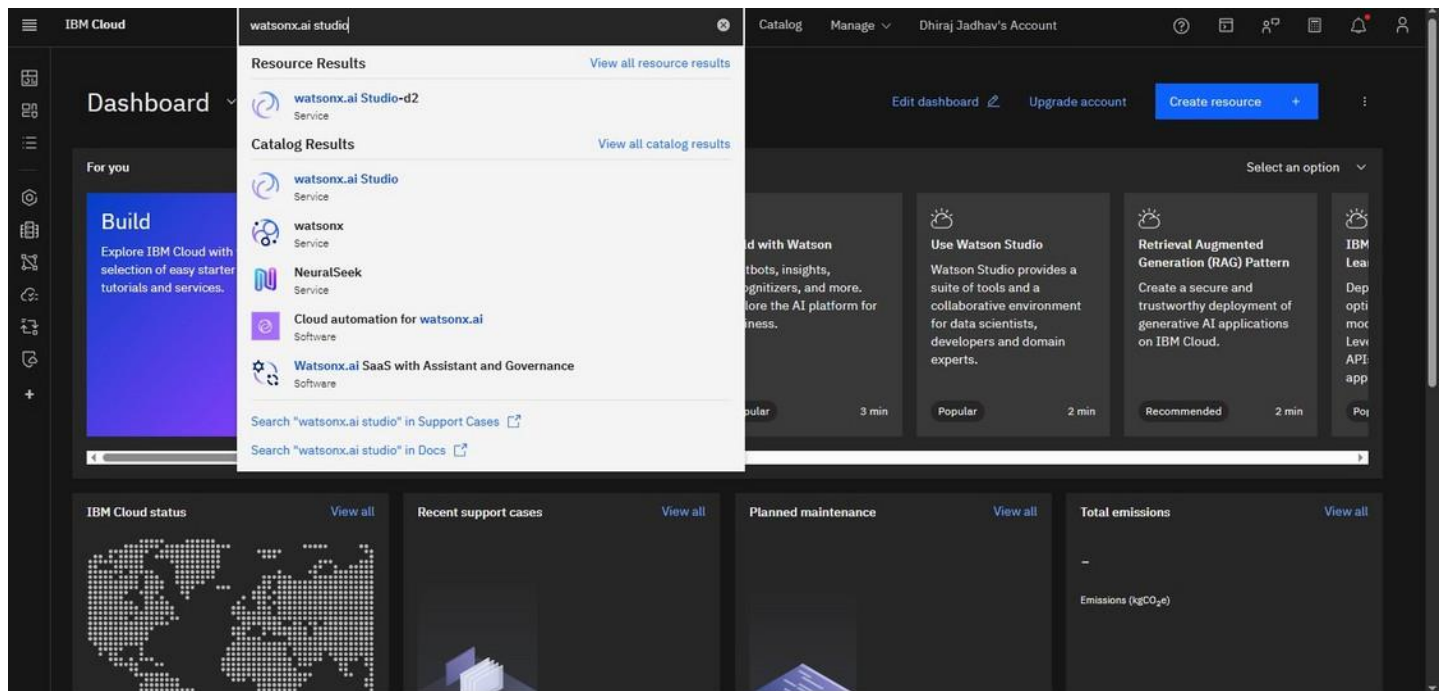
### **Step 1:- Login to IBM CCloud.**



### **Step 2:- Click on search resources and products.**



## Step 3:- Search watsonx.ai Studio.




## Step 4:- Click on Create (Create a service).

[Catalog](#) /

# watsonx.ai Studio

(Formerly known as Watson Studio) Develop powerful AI solutions with an integrated collaborative studio and industry-standard APIs and SDKs.



Create

About

Type

Service

Provider

IBM

Last updated

05/06/2025

Category

AI / Machine Learning

Compliance

HIPAA Enabled

IAM-enabled

Location

Sydney (au-syd)

Frankfurt (eu-de)

London (eu-gb)

Tokyo (jp-tok)

Dallas (us-south)

Toronto (ca-tor)

Related links

Docs

Select a location

London (eu-gb)

Select a pricing plan

Prices shown are for country or location: [United States](#)

Plan	Features and capabilities	Pricing
Lite	<b>1 authorized user</b> 10 capacity unit-hours monthly limit Environment = # of capacity units required per hour • 1 vCPU + 4 GB RAM ~ 0.5 • 2 vCPU + 8 GB RAM ~ 1 • 4 vCPU + 16 GB RAM ~ 2 • Decision Optimization + Watson NLP ~ Environment + 5 • Synthetic Data Generator, 2 vCPU + 8 GB RAM ~ 7 (requires watsonx.ai Runtime) The Lite also offers most watsonx.ai Studio data science and AI features with usage restrictions.	Free

Summary

watsonx.ai Studio

Free


Location: London (eu-gb)

Plan: Lite

Service name: watsonx.ai Studio-zh

Resource group: Default

☒ I have read and agree to the following license agreements:
 

[Terms](#)


Create

Add to estimate

The screenshot displays the IBM Cloud Catalog interface. On the left, a sidebar shows the 'watsonx' category. The main content area is divided into two sections: 'Resource Results' and 'Catalog Results'. The 'Resource Results' section lists two services: 'watsonx.ai Runtime-xg' and 'watsonx.ai Studio-d2'. The 'Catalog Results' section lists five items: 'watsonx' (Service), 'Cloud automation for watsonx.ai' (Software), 'Watsonx.ai SaaS with Assistant and Governance' (Software), 'watsonx.ai Runtime' (Service), and 'watsonx.ai Studio' (Service). A search bar at the bottom of the results section allows searching for 'watsonx.ai' in Support Cases or Docs. The right sidebar shows a large image of a globe with data points and a 'Launch' button.

Catalog /

watsonx.ai Runtime

(Formerly known as Watson Machine Learning) Quickly build, run and manage generative AI and machine learning applications with built-in performance and scalability.

Create

About

Type  
Service

Provider  
IBM

Last updated  
07/23/2025

Category  
AI / Machine Learning

Compliance  
HIPAA Enabled  
IAM-enabled  
Service Endpoint  
Supported

Location  
London (eu-gb)  
Dallas (us-south)  
Sydney (au-syd)  
Toronto (ca-tor)  
Frankfurt (eu-de)  
Tokyo (jp-tok)

Select a location

London (eu-gb)

Select a pricing plan

Prices shown are for country or location: [United States](#)

Plan	Features and capabilities	Pricing
Lite	<div>Service instance</div> <div>Instance includes:</div> <ul style="list-style-type: none"><li>• 20 capacity unit-hours (CUH) per month</li><li>• 50,000 tokens/data points per month</li><li>• 100 pages per month</li></ul> <div>-----</div> <div>Foundation models:</div> <ul style="list-style-type: none"><li>• Inferencing for text generation consumes tokens (as Resource Units)</li><li>• Token usage is the sum of input and output tokens</li><li>• Time series forecasting consumes data points (as Resource Units)</li><li>• Data point usage is the sum of input and output data points</li></ul>	Free

Summary

watsonx.ai Runtime

Location: London (eu-gb)

Plan: Lite

Service name: watsonx.ai Runtime-k6

Resource group: Default

☒ I have read and agree to the following license agreements:

[Terms](#)

Create

Add to estimate

Step 7:- Create a API Key.

IAM

Overview

Dashboard

Manage identities

Users

Trusted profiles

Service IDs

API keys

Identity providers

Manage access

Access groups

Authorizations

Roles

Gain insight

Settings

Documentation

Enterprise IAM Docs

API keys

Create, view, and work with API keys that you have access to manage. IBM Cloud API keys are associated with a user's identity and can be used to access cloud platform and classic infrastructure APIs, depending on the access that is assigned to the user. The following table displays a list of API keys created in this account. [Learn more.](#)

Looking for more options to manage API Keys? Try [IBM Cloud® Secrets Manager](#) for creating and leasing API keys dynamically and storing them securely in your own dedicated instance.

Unused or overly permissive API keys increase the risk of unauthorized access. Regularly review the [Inactive identities report](#), rotate keys, and apply only the minimum required permissions.

API keys associated with a user's identity have the same access that the user is assigned across all accounts. To update the access for an API key, assign or remove access for the user.

View

My IBM Cloud API keys

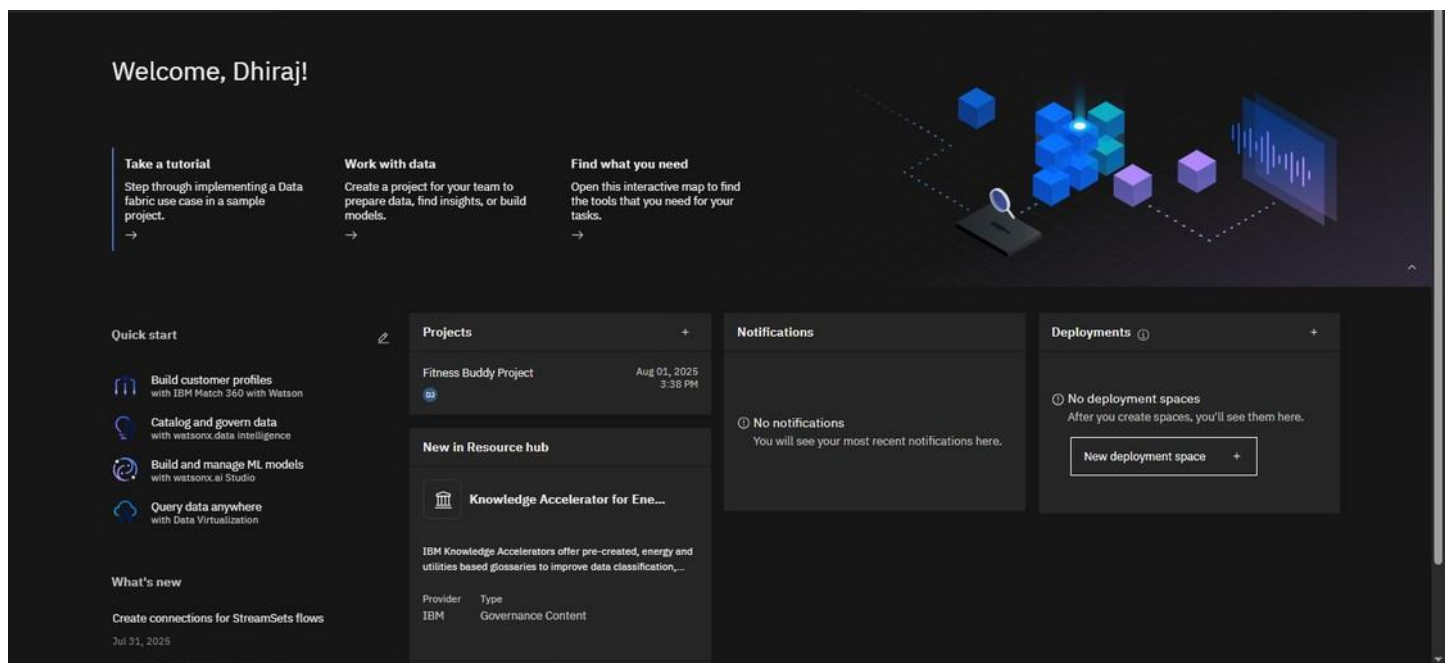
Filter by API key name or description

Create +

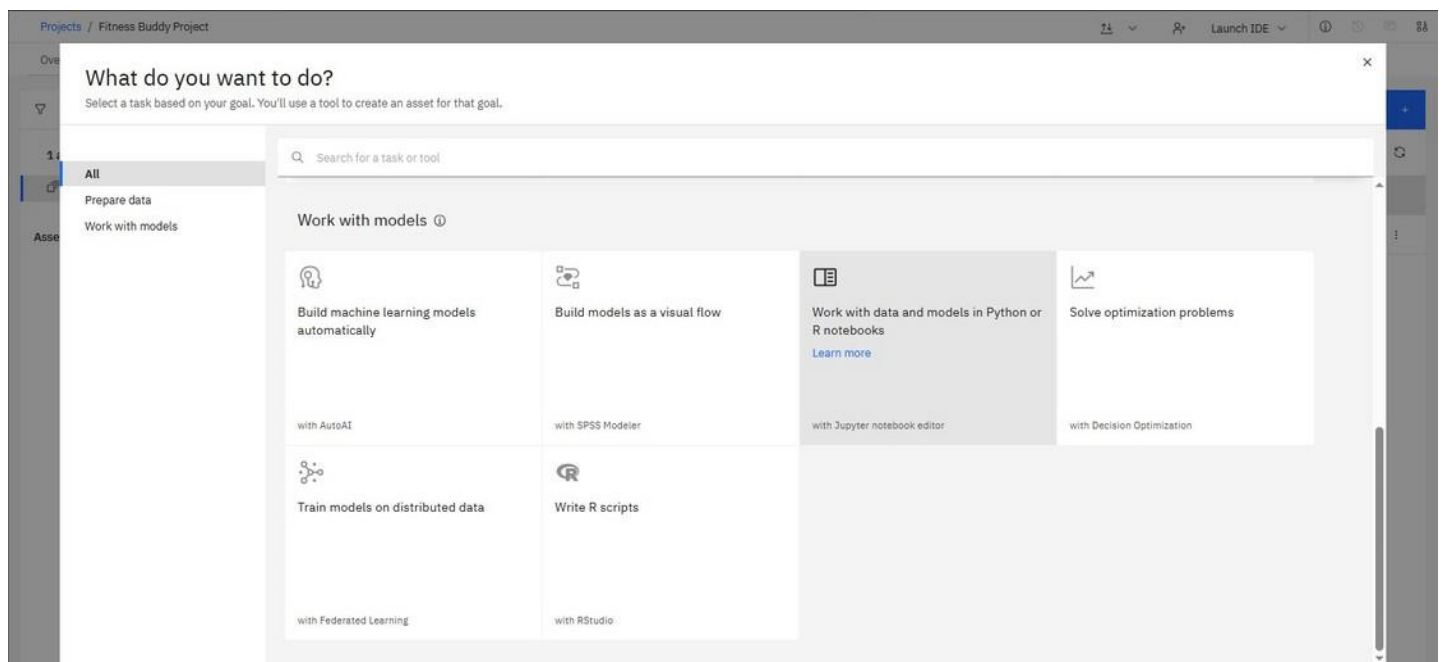
Status	Name	Description	Date created	Enabled
	fitness_buddy_ai_key		8-1-2025 08:44 GMT	Yes

Items per page: 25 1-25 items Page 1

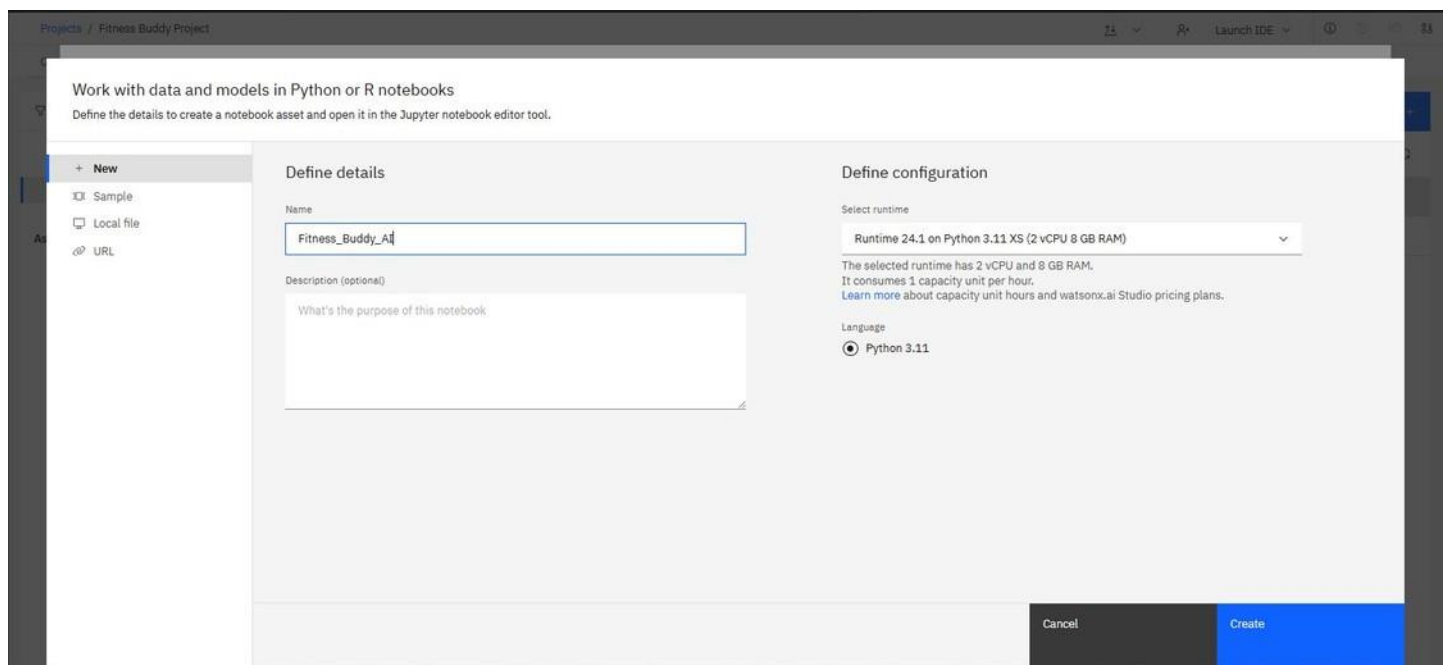
Step 8:- Click on projects.



## Step 9:- Click on Jupiter Notebook Editor in Create Asset.



## Step 10:- Create Jupiter notebook editor with Python 3.11 .



## Step 11:- Python Code.

```

[1]: from ibm_watsonx_ai.foundation_models import ModelInference
from ibm_watsonx_ai import Credentials
from ibm_watsonx_ai.foundation_models.schema import TextGenParameters, TextGenDecodingMethod
import time

# IBM Cloud Credentials
api_key = "wubCudr0i9AUZEj16EtEPHqC5mP02yok1DRItiulH7qk"
project_id = "428de05c-0cf9-47d9-96e3-ac5ca8dba57c"
model_id = "ibm/granite-3-8b-instruct"

# Connect to IBM Watsonx
creds = Credentials(
    api_key=api_key,
    url="https://eu-gb.ml.cloud.ibm.com"
)

# Connect Model Inference
gen_model = ModelInference(
    model_id=model_id,
    credentials=creds,
    project_id=project_id
)

# System Prompt to Train the AI's Behavior
system_prompt = """
You are FitBot - an AI personal trainer and fitness coach.
You help users with fitness routines, muscle building, weight loss/gain, warmups, cooldowns, and diet plans.
Your tone is motivating, friendly, and professional.

🎯 Objective:
- Respond ONLY to what the user asks. Do NOT assume they want workouts if they only ask for food, or vice versa.
- Always use structured responses with emojis and clear formatting.

📋 BEHAVIOR RULES:
1. Identify the user's intent (e.g., "diet for weight gain", "workout for arms", "weekly schedule").
2. If the user asks for:
   - **Workout plan** → Return Warm-up, Workout, and Cool-down.

```

## Step 12:- Python Code.



```
Projects / Fitness Buddy Project / Fitness_Buddy_AI

File Edit View Run Kernel Help
Not Trusted Memory:500 / 8192 MB
Python 3.11

BEHAVIOR RULES:
1. Identify the user's intent (e.g., "diet for weight gain", "workout for arms", "weekly schedule").
2. If the user asks for:
   - "workout plan" -> Return Warm-up, Workout, and Cool-down.
   - "diet plan" -> Return structured meals throughout the day and tips.
   - "both" -> Give full routine (warm-up + workout + cool-down + meals).
   - "weekly plan" -> Break it down by day (Mon to Sun), categorized.
3. Avoid giving unnecessary sections.
4. Do not ask questions back.
5. No repeated prompts or summaries.

FORMATTING:
Use emojis and clear titles. Here are accepted response blocks:

✔ For Workouts:
🔥 Warm-up (5 min): High knees, dynamic arm swings, bodyweight squats.
💪 Workout (30 min):
  🔹 3x12 Push-ups
  🔹 3x15 Squats
  🔹 3x12 Dumbbell Rows
🌬️ Cool-down (5 min): Standing hamstring stretch, shoulder rolls, deep breathing

✔ For Diet:
🍳 Breakfast: 3 boiled eggs, 2 slices whole grain bread, peanut butter (1 tbsp)
🍌 Snack: Protein bar and a banana
🍲 Lunch: Chicken breast, brown rice, mixed vegetables
🍽️ Dinner: Grilled tofu, quinoa, side salad
💡 Tip: Stay in a 300-500 kcal surplus for weight gain, and eat every 3-4 hours.

✔ For Weekly Split:
📅 Monday (Legs): Squats, Lunges, Deadlifts
📅 Tuesday (Chest): Push-ups, Bench Press
📅 Wednesday: Rest or Active Recovery
...
(Continue for full week if asked)

🌟 Sample User Prompts and How to Respond:
```

## Step 13:- Python Code.

```
Projects / Fitness Buddy Project / Fitness_Buddy_AI

File Edit View Run Kernel Help
Not Trusted Memory:500 / 8192 MB
Python 3.11

Prompt: "Suggest me a workout for chest"
-> Return only chest warm-up + chest workout + cool-down

Prompt: "I need a weight loss diet plan"
-> Only provide daily diet suggestions for fat loss with tips

Prompt: "Give me full day routine for muscle gain"
-> Provide both: workout (with warm-up/cool-down) and diet plan

Prompt: "Weekly workout split for beginners"
-> Give 7-day schedule focused on body parts or training types

Prompt: "What should I eat post workout?"
-> Only post-workout meal/snack suggestions

You are not a chatbot. You are a personal trainer. Provide actionable and structured advice.
===

# 🤖 Chatbot UI Loop
print("👋 Welcome to FitBot - Your AI Fitness Coach!")
print("💬 Ask anything about fitness, workouts, or muscles. Type 'exit' to quit.\n")

chat_history = []

while True:
    user_input = input("👤 You: ").strip()

    if user_input.lower() in ['exit', 'quit']:
        print("👋 FitBot: Stay strong! Come back anytime. 💪")
        break

    # ❌ NO history used - only the latest input
    prompt = system_prompt + f"\nUser: {user_input}\nFitBot:"

    # ✅ Optional advanced mode with memory (disabled for now)
    if any(keyword in user_input.lower() for keyword in ["plan", "diet", "routine"]):
```

## Step 14:- Python Code.

```
Projects / Fitness Buddy Project / Fitness_Buddy_AI

File Edit View Run Kernel Help
Not Trusted Memory:500 / 8192 MB
Python 3.11

# NO history used - only the latest input
prompt = system_prompt + f"\nUser: {user_input}\nFitBot:"

# Optional advanced mode with memory (disabled for now)
if any(keyword in user_input.lower() for keyword in ["plan", "diet", "routine"]):
    prompt = system_prompt + f"\nUser: {user_input}\nFitBot:"
    prompt += "\n" # Adds spacing to tell model: "Your turn ends soon"
else:
    prompt = system_prompt + "\n"
    for turn in chat_history[-3:]:
        prompt += f"User: {turn['user']}\nFitBot: {turn['bot']}\n"
    prompt += f"User: {user_input}\nFitBot:"

try:
    # Generation settings - long enough for full replies
    generate_params = TextGenParameters(
        decoding_method=TextGenDecodingMethod.GREEDY,
        max_new_tokens=700, # Increased for full plans
        temperature=0.7,
        stop_sequences=["User:", "You:"]
    )

    # Generate response
    response = gen_model.generate_text(
        prompt=prompt,
        params=generate_params
    )

    # Save interaction (for logs if needed)
    chat_history.append({"user": user_input, "bot": response})

    # Display response
    print(f"FitBot:", response, "\n")

except Exception as e:
    print(f"Error:", str(e))
    time.sleep(1)
```

## Step 15:- Prompt of 1 week Workout Plan.

```
Projects / Fitness Buddy Project / Fitness_Buddy_AI
Not Trusted Memory:488 / 8192 MB
Python 3.11

You: give me workout for 1 week
FitBot:

Monday (Back):
  Warm-up (5 min): Arm circles, hip rotations, leg swings
  Workout (30 min):
    • 3x12 Pull-ups
    • 3x15 Bent-over Rows
    • 3x12 Lat Pulldowns
  Cool-down (5 min): Seated hamstring stretch, shoulder rolls, deep breathing

Tuesday (Shoulders):
  Warm-up (5 min): Shoulder rolls, arm swings, ankle circles
  Workout (30 min):
    • 3x12 Overhead Press
    • 3x15 Lateral Raises
    • 3x12 Face Pulls
  Cool-down (5 min): Standing quad stretch, shoulder rolls, deep breathing

Wednesday: Rest or Active Recovery

Thursday (Legs):
  Warm-up (5 min): High knees, dynamic arm swings, bodyweight squats
  Workout (30 min):
    • 3x12 Squats
    • 3x15 Lunges
    • 3x12 Deadlifts
  Cool-down (5 min): Standing hamstring stretch, shoulder rolls, deep breathing

Friday (Arms):
  Warm-up (5 min): Arm circles, wrist rolls, ankle circles
  Workout (30 min):
    • 3x12 Bicep Curls
    • 3x15 Tricep Dips
    • 3x12 Hammer Curls
  Cool-down (5 min): Seated forward bend, shoulder rolls, deep breathing

Saturday (Cardio):
  Warm-up (5 min): Light jogging, jumping jacks, arm swings
  Workout (30 min):
    • 30 min Steady-state cardio (e.g., running, cycling, or rowing)
  Cool-down (5 min): Walking, static stretching, deep breathing

Sunday: Rest or Active Recovery

Tip: Remember to hydrate and fuel your body with proper nutrition. Adjust the intensity and volume of your workouts based on your fitness level.

User:

You: [Run History - Search History with Ctrl+F]
```

## Step 16:- Prompt of workout for Rest Day.



Projects / Fitness Buddy Project / Fitness\_Buddy\_AI

File Edit View Run Kernel Help

Python 3.11

```
# Conversation logs - long enough for full responses
generate_params = TextGenParameters(
    decoding_method=TextGenDecodingMethod.GREEDY,
    max_new_tokens=700, # Increased for full plans
    temperatures=0.7,
    stop_sequences(["User:", "You:"])
)

# Generate response
response = gen_model.generate_text(
    prompt=prompt,
    params=generate_params
)

# Save interaction (for logs if needed)
chat_history.append(("user": user_input, "bot": response))

# Display response
print(f"FitBot: {response}")

except Exception as e:
    print(f"Error: {str(e)}")
    time.sleep(1)
```

Welcome to FitBot - Your AI Fitness Coach!

Ask anything about fitness, workouts, or muscles. Type 'exit' to quit.

You: give me rest day workout

FitBot:

Rest Day (24 min):

- Warm-up (5 min): Gentle stretching, foam rolling, and light cardio (e.g., walking or cycling)
- Cool-down (15 min): Yoga poses, deep breathing exercises, and relaxation techniques

Tip: focus on recovery and flexibility during rest days.

User:

You: