

# CODE:

```
# Name- Sonam Yadav, Date-10/11/2025 ,Title- DAILY CALORIE TRACKER
"""
DAILY CALORIE TRACKER

Author: Sonam Yadav
Date: 10/11/2025

This Python-based tool allows users to monitor daily calorie intake in a
quick and simple way.
It prompts users to input meal names and calorie amounts, calculates
total and average calorie
consumption per meal, and validates user responses for interactive
prompts.

Functions:
    - do_you_want_permission: Requests and validates yes/no input from
the user.
    - calc_avg_calorie: Computes the average calorie intake per meal,
rounded to two decimal places.

Usage:
    Run the script and follow the interactive prompts to track and
analyze your daily meal calorie
data.

"""

print("Welcome, This Python tool helps you quickly track and analyze your
daily calorie intake")

#ALL FUNCTIONS HERE
def do_you_want_permission(question_prompt):
    """
    Prompts the user with a yes/no question and validates the input.

    Args:
        question_prompt (str): The question to display to the user.

    Returns:
        bool: True if the user answers 'yes', False if the user answers
    'no'.
    """

    while True:
        user_input = input(f"{question_prompt} (yes/no)? -")
        .lower().strip()
        if user_input == "yes":
            return True
        if user_input == "no":
```

```

        return False
    print("Please enter a valid answer (Yes or No)")

def calc_avg_calorie():
    """
    Calculates the average calories per meal.

    Returns:
        float: The average calories per meal, rounded to two decimal
    places.
    """
    avg_calorie = CALORIE_SUM / num_MEALS
    rounded_avg_calorie = round(avg_calorie,2)
    return rounded_avg_calorie

#main code stars here
Meal = []
Calories = []
num_MEALS = int(input("Enter the number of meals you want to add - "))
for i in range(1,num_MEALS + 1):
    Meal_name = input("Enter the name of your meal - ")
    Meal.append(Meal_name)
    calorie_amt = float(input("Enter the amount of your calorie intake - "))
    Calories.append(calorie_amt)

CALORIE_SUM = 0
for p in Calories:
    CALORIE_SUM += p
print("Total amount of your calorie intake is - ", float(CALORIE_SUM))

if do_you_want_permission("Do you want to see the average calories per
meal?"):
    print("Calculating your data...")
    print(calc_avg_calorie())           #calculating and printing the
average
else:
    print("Action cancelled")

#daily calorie limit according to google it was 3000 for an avg healthy
human so i set it to this
DAILY_LIMIT = 3000
if do_you_want_permission("Do you want to set your own daily calorie
limit?"):
    DAILY_LIMIT = float(input("Enter your daily calorie limit: "))
    print("Daily calorie limit Set to - ",DAILY_LIMIT)
else:
    print("Using default daily calorie limit of 3000.")

if CALORIE_SUM > DAILY_LIMIT :

```

```

        print("Your calorie intake for today exceeded your daily limit!!!")
else:
    print("Your calorie intake for today is in your daily limit.")

print("Meal name\tCalories")
print("-----")
print(f"{Meal[0]}\t{Calories[0]}")
for j in range(1,num_MEALS):
    print(f"{Meal[j]}\t\t{Calories[j]}")
print(f"Total calorie\t{CALORIE_SUM}")
print(f"Average calorie\t{calc_avg_calorie()}\n")

# process of saving the daily log starts here
if do_you_want_permission("Do you want to save todays report"):
    print("Proceeding with action")
    filename = input("Enter the name as of the file to be created! -\n").strip()
    Date = input("Enter todays date in any format - ").strip()
    with open(filename, "a", encoding="UTF-8") as report:
        report.write(f"\n")
        report.write("Meal name\tCalories\n")
        report.write("-----\n")
        report.write(f"{Meal[0]}\t{Calories[0]}\n")
        for j in range(1, num_MEALS):
            report.write(f"\t\t{Calories[j]}\n")
        report.write(f"Total calorie\t{CALORIE_SUM}\n")
        report.write(f"Average calorie\t{calc_avg_calorie()}\n")
else:
    print("Action Cancelled!")

```

## INTRODUCTION:

A **Daily Calorie Tracker** is a command-line based application designed to help users monitor and manage their everyday calorie intake. In this project, users can log the foods they eat, record the calories consumed, and track their daily progress towards their health or fitness goals. The tracker allows users to add entries, view their total calories for the day.

## Objective:

The primary objective of the **Daily Calorie Tracker** is to provide users with a simple and effective tool to record and monitor their daily calorie intake. The project aims to help users develop healthy eating habits by tracking the

calories consumed throughout the day and comparing them with their dietary goals.

## **WORKING DESCRIPTION:**

The **Daily Calorie Tracker** operates through a menu-driven command-line interface that allows the user to interact with the system step by step. When the program starts, it displays a list of options such as adding a food item, viewing total calories, updating an entry, deleting an entry, and exiting the application. Users can input the name of the food and its calorie value, which the program stores in a JSON file for future reference. Every time the user adds or modifies data, the file gets updated, ensuring that the calorie log is saved permanently. The tracker calculates the total calories consumed by summing up all recorded entries. Users can also check previously stored records, make corrections, or remove unnecessary items.

## **SCREENSHOT:**

The screenshot shows a GitHub repository interface for a project named "python-assignment-1-krmu". The repository has a main branch and several other branches like "Assignment2", "assignment 1", and "sample 1". The "sample 1" branch is currently selected. Inside "sample 1", there is a file named "sample1.py" which contains the following code:

```
1 10/11/2025
2 Meal name Calories
3 -----
4 Rice 400.0
5 Rajma 500.0
6 Total calorie 900.0
7 Average calorie 450.0
```

## CONCLUSION:

The Daily Calorie Tracker provides an efficient and user-friendly way to record and monitor everyday calorie consumption. Through its menu-driven interface and simple structure, users can easily add, view, update, or delete food entries while keeping track of their total daily intake. The project successfully demonstrates key programming concepts such as loops, functions, conditional logic, exception handling, and JSON file storage.