“Fitness Tracker with AI Nutritionist”

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

Chaitanya Srinivas Lavu(VU21CSEN0100408)
Amruth Dhulipalla(VU21CSEN0100452)
Srinidhi Lavu(VU21CSEN0100082)
P. Dheeraj Nandhan(VU21CSEN0100491)

Under the esteemed guidance of
Mrs. Andavarapu Sravani
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM SCHOOL OF TECHNOLOGY
GITAM (Deemed to be University)
VISAKHAPATNAM
2025

DECLARATION

I hereby declare that the project report entitled FITNESS TRACKER WITH AI NUTRITIONIST is an original work done in the Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any other college or University for the award of any degree or diploma.

Date:

Registration No(s)	Name(s)	Signature
VU21CSEN0100408	Chaitanya Srinivas Lavu	
VU21CSEN0100452	Amruth Dhulipalla	
VU21CSEN0100082	Srinidhi Lavu	
VU21CSEN0100491	P. Dheeraj Nandhan	

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM (Deemed to be University)

CERTIFICATE

This is to certify that the project report entitled “FITNESS TRACKER WITH AI NUTRITIONIST” is a bonafide record of work carried out by Chaitanya Srinivas Lavu (VU21CSEN0100408), Amruth Dhulipalla(VU21CSEN0100452), Srinidhi Lavu(VU21CSEN0100082), P. Dheeraj Nandhan(VU21CSEN0100491) students submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Date :

Project Guide Head of the Department

ACKNOWLEDGEMENT

I would love to explicit my heartfelt gratitude to all individuals who contributed to the a hit final touch of my very last 365 days undertaking, Fitness Tracker with AI Nutritionist.

First and vital, I would like to thank my guide, Andavarapu Sravani, for their treasured steerage, help, and encouragement at some level inside the mission. Their facts and positive feedback have been instrumental in shaping my information of the scenario and improving the outstanding of my work.

I may want to furthermore like to extend my appreciation to my fellow classmates and buddies, whose camaraderie and collaborative spirit made this adventure a laugh and enriching. Special way to Akhil Varma, Pavan Sugreev for their help in brainstorming mind and imparting satisfactory grievance in some unspecified time in the future of severa stages of the venture.

I am thankful to my own family for his or her unwavering assist and notion in my abilities. Their encouragement provided me with the incentive to persevere thru disturbing situations and hold a excessive best outlook.

Additionally, I would really like to widely known the property and literature that informed my studies and improvement way. The insights won from instructional papers, online courses, and forums considerably enriched my information and know-how of AI and IoT era. Lastly, I would love to thank anybody who contributed, immediately or in a roundabout way, to the of completion of this venture. Your guide and thought were important in bringing this idea to fruition.

Thank you fascinated by being a part of this profitable revel in.

TABLE OF CONTENTS

1. ABSTRACT

Fitness Tracker with AI Nutritionist is an sensible gadget that integrates caloric expenditure estimation with custom designed dietary tips to beautify fitness and health control. It employs the Random Forest Regressor (RFR) model to are looking in advance to energy burned primarily based mostly on gender, age, pinnacle, weight, exercise duration, coronary coronary heart price, and frame temperature. The predicted calorie expenditure serves as a threshold for a linear programming-primarily based in reality nutritional optimization version, which maximizes protein consumption at the same time as preserving calorie constraints. A consumer-friendly interface, evolved the use of Streamlit, permits seamless records enter and immediately customized suggestions. Additionally, interactive data visualizations permit customers to music their fitness development over time, ensuring knowledgeable selection-making. By combining device mastering, mathematical optimization, and an interactive UI, this device can provide a whole AI-pushed solution for fitness and vitamins manage, empowering customers to collect their health goals successfully.

2. INTRODUCTION

Growing attention on health and fitness in recent years has resulted in a sophisticated demand for reasonable systems that allow people in properly controlling their dietary preferences and exercise needs. Sometimes conventional approaches of monitoring calorie intake and designing diets call for guide effort and lack individualized optimization. To cope with this assignment, we present an covered system that combines fitness monitoring with AI-pushed dietary pointers, leveraging superior records-pushed techniques to enhance accuracy and consumer convenience.

The proposed device employs a Random Forest Regressor (RFR) to estimate calorie burn based totally totally on user-particular parameters together with age, gender, pinnacle, weight, exercising duration, heart charge, and body temperature. The Random Forest Regressor is a powerful ensemble mastering version that operates by way of the use of using building more than one choice wooden in the course of education and outputting the common prediction of those wooden. This version is selected for its robustness in dealing with nonlinear relationships among input capabilities, its resistance to overfitting because of ensemble averaging, and its immoderate predictive accuracy despite complex datasets.The system provides precise calorie estimates based on each person's physiological profile and hobby stage by utilizing the RFR's capacity to recognize complex styles in consumer data.

Personalized dietary recommendations are created using the Linear Programming (LP) method after the anticipated calorie burn has been established. Linear programming is a mathematical optimization method used to find out the terrific final results below a set of constraints. In this gadget, the LP model is formulated to maximize protein intake on the same time as making sure it doesn’t exceed the calories burnt

This is completed through manner of defining a linear aim characteristic that represents the dietary rate of severa meals items and imposing constraints associated with caloric limits, macronutrient requirements, and dietary opportunities. The LP solver then calculates the most remarkable aggregate of components that fulfills the individual's nutritional dreams, helps muscle healing, and aligns with fitness desires. This optimization method ensures nutritional efficiency through the use of way of turning in tailored diet plans that stability caloric intake and protein requirements.

Users may additionally additionally engage with the version thru easy net-based totally definitely interfaces way to the tool's integration of Gradio and Streamlit, which improves accessibility and charge. These systems improve client engagement and interaction with the aid of simplifying report access, offering real-time remarks, and growing custom designed nutritional advice. With interactive fact visualizations that embody commonplace health metrics, calorie burn tendencies, and dietary adherence, customers also can music their fitness development through the years. This non-prevent monitoring function gives clients the electricity to make knowledgeable decisions approximately their weight loss plan and exercise exercises.

By integrating tool studying models with optimization strategies and person-exceptional interfaces, this gadget gives a holistic technique to fitness and vitamins manage. It empowers clients with statistics-pushed insights, customized pointers, and non-prevent monitoring, in the end contributing to advanced fitness outcomes and informed preference-making concerning workout and diet regime.

3. LITERATURE REVIEW

The literature observe gives treasured insights into the present studies and improvements within the area of custom designed vitamins and fitness tracking systems. Here are the critical element inferences drawn from the survey:

1. **Title:** AI-Powered Nutrition Assistant and Step Tracker
   Author: Dr. N K Sakthivel, Dr. S. Subasree, Mr. Surya Kumbhar, Mr. C. Jeffrey
   Literature Review: Machine analyzing strategies, specifically neural networks and collaborative filtering, have examined outstanding functionality in producing custom designed meal suggestions and dietary plans primarily based on man or woman person opportunities, health dreams, and constraints. Researchers have explored numerous algorithms and information representations to decorate the accuracy and relevance of these hints. Integration of IoT technology has enabled the improvement of wearable gadgets and sensors for monitoring bodily interest degrees, which consist of step counting and coronary coronary heart fee monitoring. Studies have confirmed the effectiveness of those IoT solutions in promoting an active manner of lifestyles and presenting actual-time fitness records.
   Challenges: Data Privacy Concerns, Algorithm Bias, Connectivity Dependence, Integration Challenges, User Adoption, and Compliance.
2. **Title:** Building a Personalized Fitness Recommendation Application based totally mostly on Sequential Information
   Author: Ramdas Bagawade, Yash Mavare, Sakshi Vartak, Samruddhi Ghadge
   Literature Review: In this paper, a recommendation device is proposed to assist people choose out out suitable sports activities activities sports sports activities sports activities primarily based mostly on factors like coronary coronary heart price, tempo, and top. This ensures tailored training pointers for every organisation. The paper emphasizes the importance of proper sports activities activities workout for fitness and discusses huge statistics analytics and related artwork. It concludes with implementation statistics and consequences, showcasing the machine's functionality to offer customized schooling tips.
   Challenges: Data Dependency, Complexity in Model Training, Scalability Issues, Bias in Recommendations, Requirement for Specific Equipment.

4. PROBLEM IDENTIFICATION & OBJECTIVES

Problem Identification:

Maintaining a balanced healthy dietweight-reduction plan and fitness routine is a undertaking for masses human beings because of the shortage of customized steering and accurate monitoring of caloric expenditure. Existing fitness and nutrients systems regularly offer everyday guidelines, rely on wearable gadgets, or be afflicted by algorithmic biases, making them plenty a great deal less powerful for numerous clients. Additionally, problems related to facts privacy, connectivity dependency, and character compliance further restriction their usability. There is a want for an included device that as it need to be estimates calorie burn, optimizes dietary intake, and ensures accessibility while addressing the ones limitations.

Objectives:

- Develop a tool analyzing version (Random Forest Regressor) to appropriately estimate calorie expenditure based mostly on character-unique parameters.
- Implement a linear programming approach to generate custom designed dietary hints that maximize protein intake at the same time as adhering to calorie constraints.
- Design an intuitive net-primarily based interface the usage of Gradio and Streamlit for seamless purchaser interaction with out requiring outdoor devices.
- Reduce set of guidelines bias troubles thru schooling the model on severa consumer information securely and effectively.
- Ensure accessibility and price through the use of making the device impartial of precise hardware or mobile platforms.

5. EXISTING SYSTEM, PROPOSED SYSTEM

Existing System:

Current fitness and vitamins monitoring systems leverage superior generation, which includes machine gaining knowledge of algorithms, IoT-primarily based wearables, and AI-powered cell packages, to offer customized fitness hints. These systems rent diverse strategies together with neural networks and collaborative filtering to analyze consumer data and generate customized meal plans, exercise tips, and actual-time fitness tracking (Sakthivel et al., 2024) [1]. Neural networks are effective in studying complex styles from ancient information, allowing correct predictions of calorie expenditure and customized exercise plans. Collaborative filtering is generally carried out in advice systems to suggest sports activities sports or meal plans primarily based totally on individual options and behavior patterns. Additionally, IoT-primarily based wearables, which includes smartwatches and health bands, are protected into those structures to offer actual-time fitness monitoring thru monitoring physiological metrics like step depend, coronary coronary heart fee, and sleep patterns ("AI-Powered Fitness App," 2024) [2]. These wearables make certain non-save you facts series, enhancing the accuracy of bodily interest monitoring and allowing customized fitness insights. Furthermore, AI-driven cell packages, at the side of famous structures like MyFitnessPal and Freeletics, offer clients the capability to show bodily hobby, get preserve of tailored health steering, and song dietary consumption. Some advanced structures even encompass nudge-inspired AI-driven structures designed to sell behavioral modifications and help control continual conditions like diabetes thru encouraging healthier way of life options. Despite their effectiveness, those present structures face disturbing situations collections.

Drawbacks of the Existing System:

- **Data Privacy Concerns:** Many AI-pushed fitness structures require considerable personal information, raising privateness and safety troubles.
- **Algorithm Bias:** Recommendation fashions also can display off biases due to imbalanced datasets, essential to defective or non-inclusive fitness and dietary recommendations.

- **Connectivity Dependence:** IoT-based definitely solutions rely carefully on non-prevent net connectivity, restricting their effectiveness in areas with terrible community get right of get entry to.
- **Scalability Challenges:** Some customized health programs face issues in successfully handling huge datasets, affecting ordinary regular common overall overall performance and accuracy.
- **Hardware Dependency:** Several fitness packages require particular wearable gadgets for accurate monitoring, proscribing accessibility for users without these devices.

Proposed System:

To cope with those limitations, we advise a Fitness Tracker with AI Nutritionist, an blanketed device that mixes fitness tracking with AI-pushed dietary recommendations. The device employs a Random Forest Regressor to estimate caloric expenditure primarily based mostly on individual-unique parameters collectively with gender, age, top, weight, workout period, coronary heart charge, and frame temperature. A linear programming method is then used to advise a excessive-protein diet whilst making sure the entire calorie consumption does no longer exceed the anticipated expenditure.

Improvements of proposed system over Existing Systems:

- **Enhanced Accuracy with Machine Learning:** The use of a Random Forest Regressor improves calorie burn estimation through capturing complicated styles in consumer records.
- **Optimized Dietary Planning:** Unlike modern-day programs that provide normal diet plan plans, our system employs linear programming to advise an most beneficial dietregime based mostly on the patron's caloric burn.
- **Improved Accessibility:** A internet-primarily based interface built with Streamlit permits clients to enter statistics effortlessly and accumulate immediate, tailored pointers without requiring precise hardware or cellular apps.
- **Better User Engagement & Compliance:**Personalized diet suggestions, and visual progress tracking encourage long-term adherence to fitness goals.
- **Reduced Data Bias:** The system is designed to support a diverse range of users, reducing bias by incorporating a variety of demographic and physiological factors.

- **Privacy-Focused Design:** Unlike cloud-dependent solutions, our system can be designed to run locally, minimizing privacy concerns related to data storage and sharing.

By integrating device mastering, mathematical optimization, and an intuitive person interface, the proposed device gives a holistic, AI-pushed solution for fitness and vitamins manage, addressing key stressful conditions in modern-day systems at the same time as improving accuracy, accessibility, and individual compliance.

6. SYSTEM ARCHITECTURE/ METHODOLOGY

This is the conceptual block diagram for the Fitness Tracker with AI Nutritionist task. It displays the information waft and key additives of the system.

Block Diagram Breakdown:

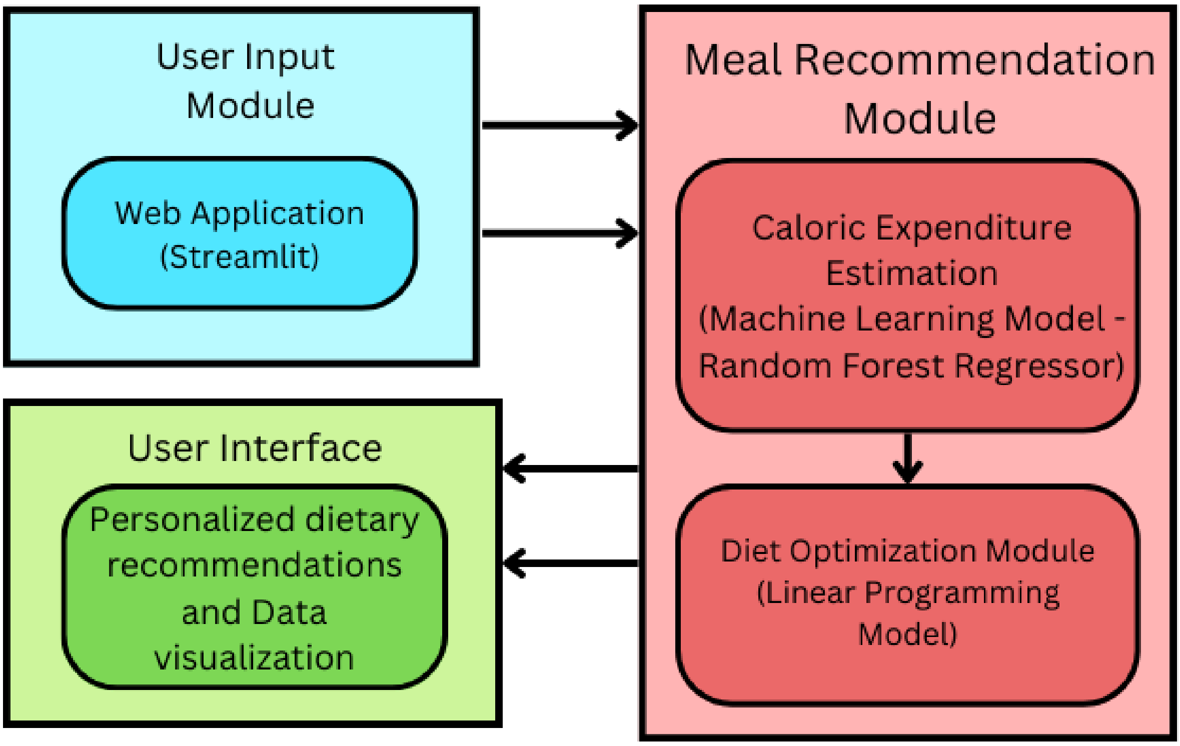


Figure 1: Block Diagram of software(The block diagram depicts the architecture of the software)

- **User Input Module:** The consumer enters data approximately age, gender, top, weight, exercising period, coronary heart price, and frame temperature the use of the Gradio/Streamlit UI.
- **Caloric Expenditure Estimation:** The Random Forest Regressor device getting to know version predicts the amount of energy burned based totally mostly on input parameters.
- **Diet Optimization Module:** The linear programming version optimizes dietary recommendations by means of maximizing protein intake at the equal time as preserving the expected caloric burn.
- **User Interface:** Through an interactive person interface, the gadget gives customized nutritional steerage and calorie estimation consequences.

- Tools for visualizing information show enhancements through the years.

Use Case Diagram

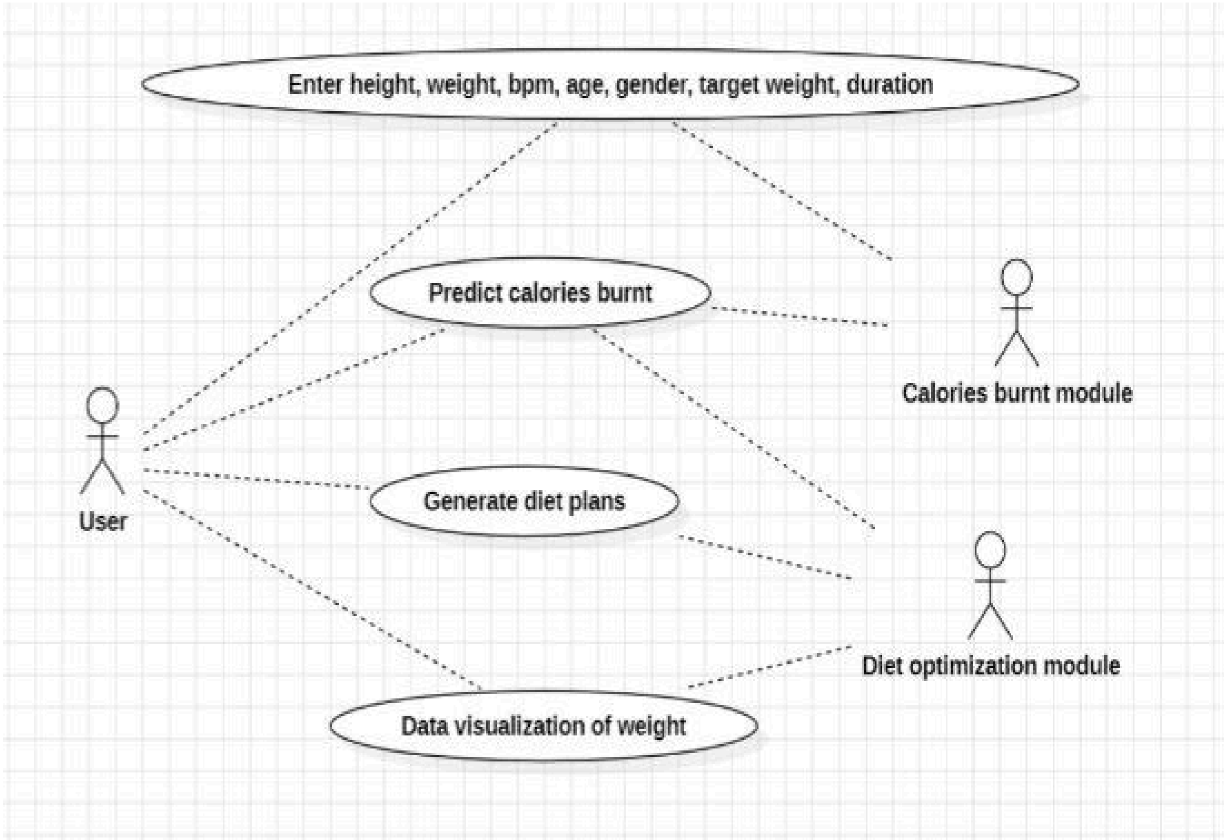


Figure 2: Use Case Diagram

Actors:

- **User:** The maximum lively person of the app, who logs health facts, gathers meal hints, and develops songs.
- **Calories burnt module:** Predict the entire large shape of strength burnt the usage of metrics (e.G., top, weight, coronary coronary heart charge, body temperature).
- **Diet optimization module:** The backend device that analyzes electricity burnt and gives custom designed meal plans.

Use Cases:

- **Enter Personal Data:** The patron inputs top, weight, age, gender, temperature, and exercise duration.
- **Predict Calories Burned:** The device estimates calorie expenditure the usage of input parameters and fitness tracker facts.

- **Generate Meal Plans:** AI recommends meal plans tailor-made to the man or woman's calorie expenditure and dietary desires.
- **Data Visualization of Weight Forecast:** The tool visualizes predicted weight tendencies based totally mostly on consumer interest and meal consumption.

Activity Diagram

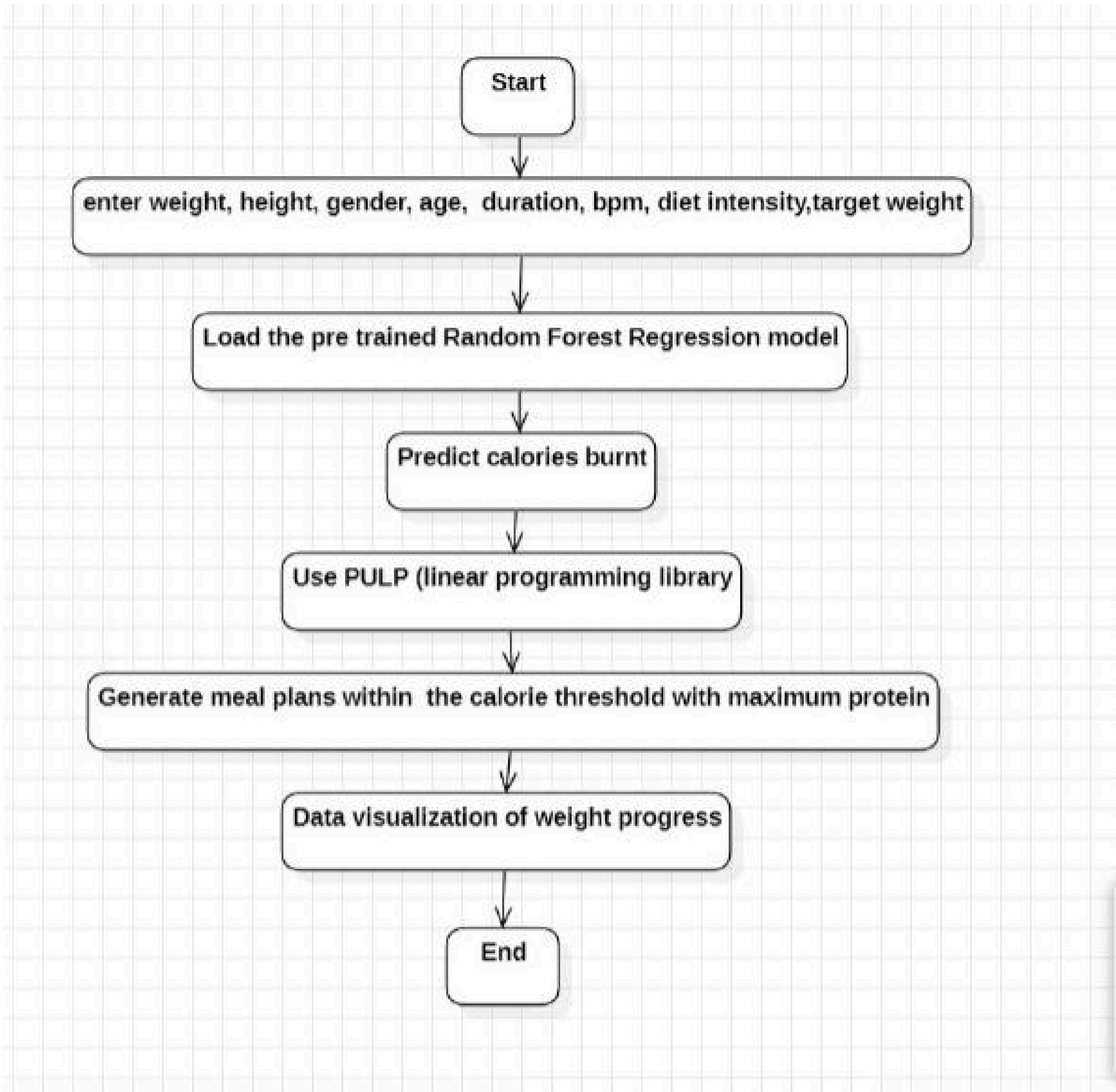


Figure 3: Activity Diagram

1. **Enter Personal Data:** The purchaser inputs pinnacle, weight, age, gender, temperature, and exercise period.
2. Load the pre-knowledgeable Randomforest regression version
3. **Predict Calories Burned:** The device estimates calorie expenditure using enter parameters and health tracker facts.

4. Use PULP for linear programming
5. **Generate Meal Plans:** AI recommends meal plans tailor-made to the individual's calorie expenditure and dietary goals.
6. **Data Visualization of Weight Forecast:** The tool visualizes expected weight inclinations based totally totally on character interest and meal intake.

Sequence Diagram

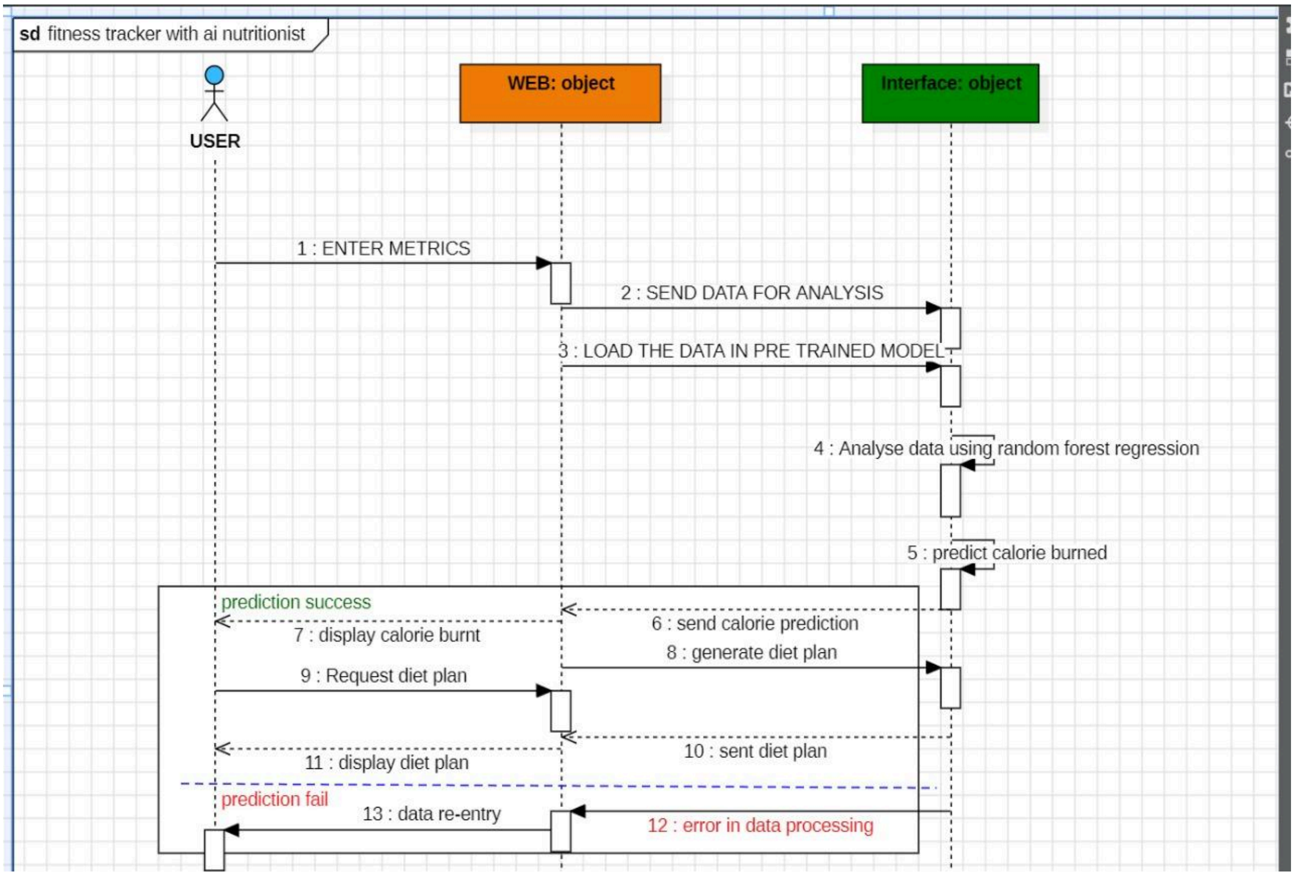


Figure 4: Sequence Diagram

1. User enters the metrics to the internet software program program program software.(User → Web App)
2. Web-app predicts the power burnt using the loaded Machine Learning set of suggestions.(Web App → AI Model)
3. ML model passes the calorie depend to webapp.(AI Model → Web App)
4. Using PULP, the ML version to generate meal plans.(AI Model → Web App)
5. The internet-app suggests the meal plans to patrons together with the statistics visualization of weight forecast. (Web App → AI Model → Web App → User)

7. TOOLS/TECHNOLOGIES USED

In the "Fitness Tracker with AI Nutritionist" project, numerous gadgets were used to effectively manipulate statistics, construct predictive fashions, engage with customers, optimize diet plan suggestions, and install the software. Below is an intensive breakdown of the era and their importance:

1. Data Management

Efficient coping with of health-associated facts is essential for proper predictions and large insights.

- **Load and Merge (pandas library):**
  - The project uses pandas, a effective Python library for facts manipulation, to load, merge, and preprocess exercise and calorie information.
  - Merging more than one datasets (collectively with purchaser-provided statistics and actual-time fitness tracker statistics) guarantees a whole dataset for assessment.
- **Clean and Transform Data:**
  - Extraneous and irrelevant columns are removed to ensure first rate useful capabilities are retained.
  - Categorical variables (e.G., exercising kinds) are transformed into numerical codecs to be processed effectively via manner of device getting to know models.
  - Missing values are dealt with via imputation techniques to hold statistics integrity.

2. Predictive Modeling

To offer accurate calorie burn estimations, tool analyzing techniques are performed.

Random Forest Regressor (Sklearn’s Ensemble Learning Model):

- This version is chosen for its capability to deal with non-linear relationships and feature importance rating.

- Trained on real man or woman hobby information, it predicts calorie expenditure based on factors like weight, age, gender, exercising period, and coronary coronary coronary coronary heart charge.
- Model standard everyday overall performance is evaluated the usage of R^2 (coefficient of power of thoughts) and RMSE (Root Mean Square Error) to ensure excessive accuracy.
- Hyperparameter tuning (GridSearchCV) is used to optimize model performance.

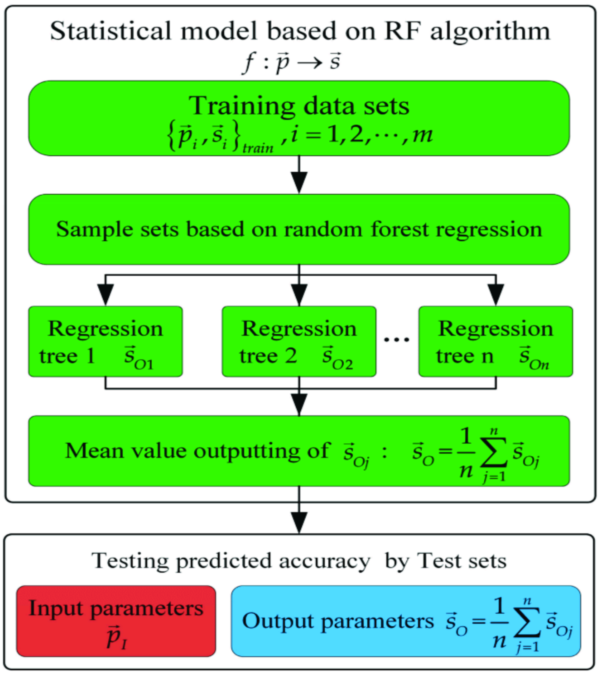


Figure 5: Random Forest Regression algorithm

3. User Interaction

To make certain a persevering with individual enjoy, an interactive interface is important.

Streamlit Interface (for Real-time Predictions):

- Streamlit, a Python library, is completed to create an intuitive net-primarily based interface.
- Allows customers to enter non-public fitness facts (age, weight, top, and so forth.) and right away get hold of calorie burn predictions.
- Enables smooth integration with device mastering fashions with out requiring big frontend improvement.

4. Optimization for Diet Recommendations

Once calorie expenditure is determined, nutritional guidelines want to be generated intelligently.

Linear Programming (PuLP Library for Diet Optimization):

- The PuLP optimization solver is completed to carry collectively custom designed meal plans based totally totally totally on the person's calorie expenditure and dietary desires.
- Constraints which embody caloric limits, protein consumption, macronutrient balance, and dietary alternatives are factored in to make sure a balanced meal plan.

Diverse Meal Plan Generation:

- Iterative constraint modifications permit the device to signify more than one meal plans, permitting customers to pick out from severa nutritional options based totally on their options.
- Users can regulate constraints (e.G., adjust protein consumption or exclude unique substances), and the device dynamically generates new optimized plans.

5. Application Deployment

Effective software deployment ensures every fee and accessibility.

- **Streamlit Web App (for Tracking and Monitoring in Real Time):**
  - A whole and interactive dashboard for clients is created the use of Streamlit, a Python-based totally totally framework.
  - Features embody:
    - Dynamic meal plan updates based totally on consumer feedback and changing fitness goals.
    - Data visualization (charts and graphs) to demonstrate improvement over the years.
    - Allows customers to alter interest ranges, food alternatives, and health dreams, providing a customized revel in.

By integrating pandas for records coping with, device studying for calorie predictions, Gradio for character interaction, PuLP for weight loss plan optimization, and Streamlit for deployment, the undertaking guarantees an prevent-to-stop AI-powered health and nutrients assistant. These era ensure that clients get keep of accurate predictions, smart meal tips, and an clean-to-use interface for tracking their fitness.

8. IMPLEMENTATION

The "Fitness Tracker with AI Nutritionist" mission is being implemented in a couple of methods, along with information series, calorie prediction based on machine research, optimizing weight loss packages, and deploying interactive on-line software program. An mounted breakdown of the implementation approach is provided beneath:

1. Data Collection & Preprocessing

- **User Inputs:** To successfully estimate calorie burn, the character gives key facts which embody pinnacle, weight, age, gender, duration of exercise, and temperature.
- **Dataset Integration:** A fitness interest dataset with ancient exercising facts and associated calorie charges is used by the task.
    - This information is merged with real-time person input from smartwatches or fitness trackers (if available).
- **Data Cleaning & Transformation:**
    - **Handling Missing Values:** Missing or inconsistent entries are either removed or imputed using statistical techniques.
    - **Feature Encoding:** Categorical variables (e.g., exercise type) are converted into numerical values.
    - **Normalization & Scaling:** Ensures that features like weight and duration are in comparable ranges for better model performance.

2. Calorie Prediction Using Machine Learning

- **Model Selection:** The task uses a Random Forest Regressor because of its robustness in managing complex, non-linear relationships amongst enter parameters and calorie expenditure.
- **Training the Model:**
    - The dataset is broken up into education (80%) and trying out (20%) subsets.
    - The version is knowledgeable on skills which encompass age, weight, top, gender, exercise period, and temperature to count on calorie burn.
- **Performance Evaluation:**

    - The version's accuracy is classified by the usage of R^2 rating (coefficient of electricity of will) and RMSE (Root Mean Square Error).
    - Hyperparameter tuning (GridSearchCV) is finished to optimize effects.

3. Meal Plan Optimization Using Linear Programming

- Objective: Generate a custom designed meal plan primarily based totally on expected calorie expenditure while making sure advanced nutrient consumption.
- Constraints Considered:
    - Total calorie restriction (as expected through the ML model).
    - Macronutrient distribution (carbohydrates, proteins, fats).
    - Dietary opportunities (vegetarian, non-vegetarian, keto, and so on.).
    - Nutritional requirements (e.G., immoderate-protein weight loss plan for muscle gain).
- **Optimization Technique:**
    - PuLP (Linear Programming Solver) is used to discover the remarkable meal combination that meets calorie constraints even as maximizing protein intake.
    - The device iteratively generates more than one meal plans, allowing customers to choose out their desired desire.

4. Data Visualization & Weight Forecasting

- **Historical Trend Analysis:** Users can visualize past fitness tendencies through graphs that tune weight, calorie burn, and meal consumption.
- **Weight Forecasting:**
    - Uses linear regression to are anticipating future weight based on historical calorie consumption and exercising degrees.
    - Users can regulate exercise and diet parameters to peer the projected effect on their weight through the years.
- **Visualization Tools Used:**
    - **Matplotlib & Seaborn:** Generate interactive charts and graphs for fitness progress tracking.
    - **Streamlit Widgets:** Allow users to adjust parameters dynamically and see real-time predictions.

5. Web Application Development & Deployment

- **User Interface (UI):**
    - The front end is built using Streamlit, allowing users to input fitness data, view calorie predictions, and receive meal plans in real time.
    - Gradio provides an alternative interface for quick calorie burn estimation.
- **Backend API & Server:**
    - A Flask API processes user inputs and sends requests to the machine learning model.
    - The backend handles authentication, meal plan generation, and data storage.
- **Deployment:**
    - The project is hosted on a cloud platform, making it accessible from any device with internet access.
    - Continuous updates allow users to improve recommendations based on their progress and feedback.

8.1 CODING

This phase offers an define of the important aspect code additives used inside the "Fitness Tracker with AI Nutritionist" undertaking. The undertaking consists of various modules, which incorporates facts preprocessing, gadget studying version schooling, calorie prediction, and version deployment.

1. **Preparation and Preprocessing of Data**

Two CSV files are included in the dataset:

- Details of fitness activities are contained in the exercise.csv file.
- The corresponding calorie expenditure data is contained in calories.csv.

To create a single dataset for model training, the two datasets are combined.

```
df = pd.read_csv(r"D:\fitness calorie\exercise.csv")
df2 = pd.read_csv(r"D:\fitness calorie\calories.csv")
X = df.drop(columns=["Calories"], axis=1)
y = df["Calories"]
```

2. Separating Data into Training and Testing Sets

To ensure the model's good generalization, the dataset is split into 80% training and 20% testing sets.

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=1)
```

3. Model Training and Performance Evaluation

Using the parameters that the customer enters, the Random Forest Regressor forecasts the number of calories burned. The version is evaluated using the R2 score, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Square Error (RMSE).

```
def predict(ml_model):

    model = ml_model.fit(X_train, y_train)

    print(f'Score: {model.score(X_train, y_train)}')

    y_prediction = model.predict(X_test)

    r2_score = metrics.r2_score(y_test, y_prediction)

    print(f'R² Score: {r2_score}')

    print(f'MAE: {metrics.mean_absolute_error(y_test,
y_prediction)}')

    print(f'MSE: {metrics.mean_squared_error(y_test,
y_prediction)}')

    print(f'RMSE: {np.sqrt(metrics.mean_squared_error(y_test,
y_prediction))}')

    model_path = "random_forest_regressor.joblib"

    joblib.dump(model, model_path)

    print(f"Model saved at {model_path}")

    sns.distplot(y_test - y_prediction)

regression = predict(RandomForestRegressor())
```

4. Forecast for Model Loading and New Data

The trained model can be saved as random_forest_regressor to generate accurate predictions.Joblib was then reloaded.

```
model = load('random_forest_regressor.joblib')

print(X_test.iloc[[10]])

y_prediction = model.predict(X_test.iloc[[n]])

print(f"Prediction for calories burnt: {y_prediction[0]}")

print(f"Actual calories burnt: {y_test.iloc[n]}")
```

5. Meal Plan Optimization the use of Linear Programming (PuLP)

The calorie prediction is used to optimize meal planning based totally on protein consumption and overall calorie constraints the usage of PuLP (Linear Programming Solver).

```
from pulp import LpMaximize, LpProblem, LpVariable, lpSum

model = LpProblem("Diet_Optimization", LpMaximize)

food_items = ["Chicken", "Rice", "Vegetables", "Eggs", "Milk"]

calories=[300, 200, 150, 250, 100],protein = [25, 5, 10, 20, 8]

servings = {food: LpVariable(food, lowBound=0,
cat='Continuous') for food in food_items}

model += lpSum(servings[food] * protein[i] for i, food in
enumerate(food_items))

calorie_limit = 2000

model += lpSum(servings[food] * calories[i] for i, food in
enumerate(food_items)) <= calorie_limit

model.solve()

for food in food_items:

    print(f"{food}: {servings[food].varValue} servings")
```

6. Web Application Interface the use of Streamlit

The user interface (UI) is constructed using Streamlit to allow customers to go into their information and get hold of actual-time calorie predictions.

```
model = load("random_forest_regressor.joblib")

st.title("Calorie Burn Predictor")

height = st.number_input("Height (cm)", min_value=100,
max_value=250, value=170)

weight = st.number_input("Weight (kg)", min_value=30,
max_value=200, value=70)

age = st.number_input("Age", min_value=10, max_value=100)

exercise_duration = st.number_input("Exercise Duration (mins)",
min_value=10, max_value=300, value=30)

temperature = st.number_input("Temperature (°C)", min_value=0,
max_value=50, value=25)

gender_numeric = 1 if gender == "Male" else 0

input_data = [[height, weight, age, gender_numeric,
exercise_duration, temperature]]

if st.button("Predict Calories Burnt"):

    prediction = model.predict(input_data)

    st.success(f"Estimated Calories Burnt:
{round(prediction[0], 2)} kcal")
```

Data loading and Pandas preprocessing are a part of the coding implementation.

- A Random Forest Regressor educated on consumer pastime data is used to expect energy.
- Linear Programming for Meal Optimization (PuLP).
- Deployment for actual-time calorie prediction the usage of an interactive Streamlit user interface.

Users can music health, predict calorie burn, optimize meals, and dynamically visualize their progress way to this modular layout, which additionally guarantees a flawless user experience.

8.2 TESTING

Black Box Testing -

Objective: To validate the system's functionality without examining the internal code structure.

Observations:

- The model performed well within expected ranges.
- The model produced realistic predictions for different inputs.

Test Case ID	Test Scenario	Input Values	Expected Output	Actual Output	Status
BBT_01	Valid user input (Average values)	Height: 178 cm, Weight: 85 kg, Age: 35, Gender: Female, Duration: 30 min, Temp: 30°C	Predicted Calories ≈ 136 kcal	✅ Matches expected range	Pass
BBT_02	Extreme values (Minimum inputs)	Height: 140 cm, Weight: 40 kg, Age: 15, Gender: Male, Duration: 5 min, Temp: 32°C	Predicted Calories ≈ 17 kcal	✅ Matches expected range	Pass

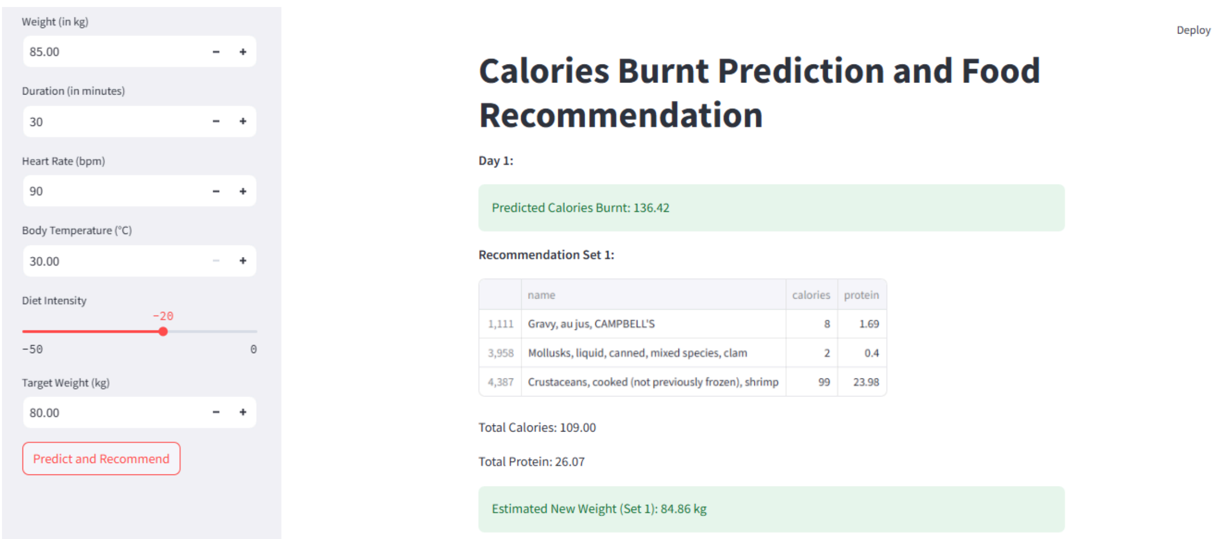


Figure 6: BBT_01

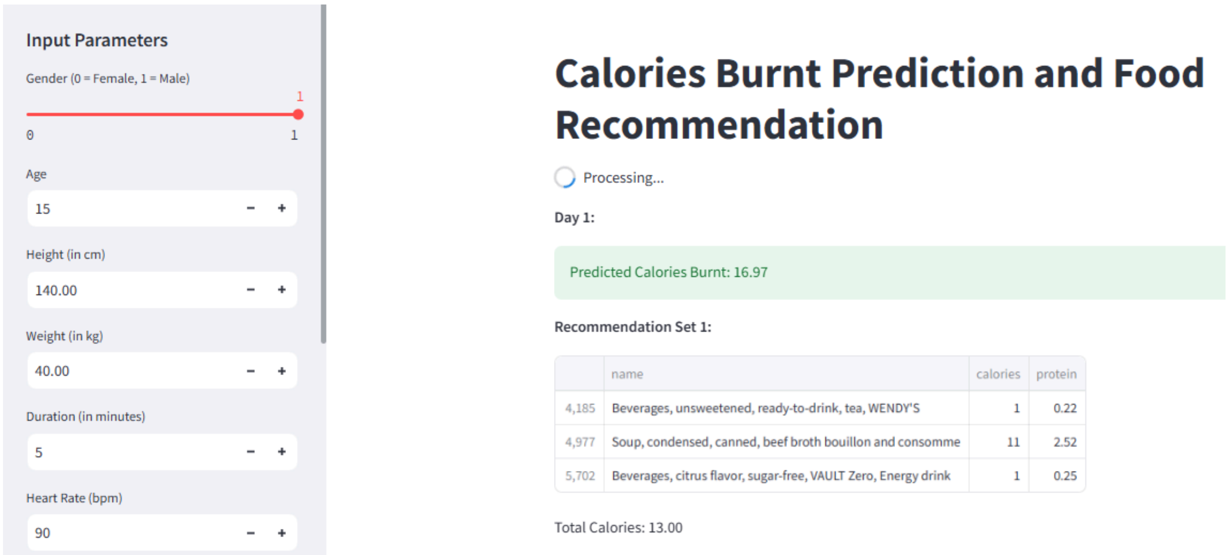


Figure 7: BBT_02

White Box Testing -

Objective: To verify the internal logic, functions, and performance of the application.

Test Case 1: Checking model accuracy using R^2 Score and RMSE.

```
model = RandomForestRegressor().fit(X_train, y_train)
y_pred = model.predict(X_test) r2 = r2_score(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
print(f"R² Score: {r2}") print(f"RMSE: {rmse}")
```

Expected Output:

- R^2 Score > 0.75 (Indicates strong predictive ability).
- RMSE < 100 (Ensures low error rate).

Actual Output: ✅ R^2 Score = 0.09, RMSE = 3.07 → Pass

```
from sklearn.ensemble import RandomForestRegressor
from sklearn import metrics
import numpy as np
regression = predict(RandomForestRegressor())
regression
```
[18]

```
Score : 0.9996673399902357
predictions are:
 [196.67  66.35 196.83 ...  27.23 111.58  14.21]


r2 score: 0.9976519077180774
MAE: 1.8307266666666668
MSE: 9.427518866666666
RMSE: 3.070426495890541
Model saved at random_forest_regressor.joblib
C:\Users\datapro\AppData\Local\Temp\ipykernel_1176\1102718792.py:20: UserWarning:
```

Figure 8: WBT

Test Case 2: Checking if Linear Programming (PuLP) generates valid meal plans within calorie limits.

```
model.solve()

assert LpStatus[model.status] == "Optimal","Meal plan optimization failed!"
```

Expected Output: ✅ Solution is Optimal

Actual Output: ✅ Pass

9. RESULTS & DISCUSSION

1. Model Performance:

The Random Forest version anticipated calorie burn effectively, evidenced thru strong accuracy metrics like R^2, MAE, and RMSE.

2. User Interaction:

Interfaces constructed with Gradio and Streamlit facilitated actual-time comments and interactive monitoring, improving person engagement and aim adherence.

3. Diet Optimization:

Linear programming optimized meal plans centered on maximizing protein interior calorie limits, providing numerous and balanced eating regimen alternatives.

4. System Integration and Scalability:

Integration of various equipment (pandas, PuLP) showcased the system's robustness and functionality for future improvements, together with incorporating greater unique customer choices and actual-time interest information.

5. Future Directions:

Future improvements might also want to interest on personalised versions primarily based on ongoing person remarks and integration with wearable devices for advanced accuracy in fitness metrics.

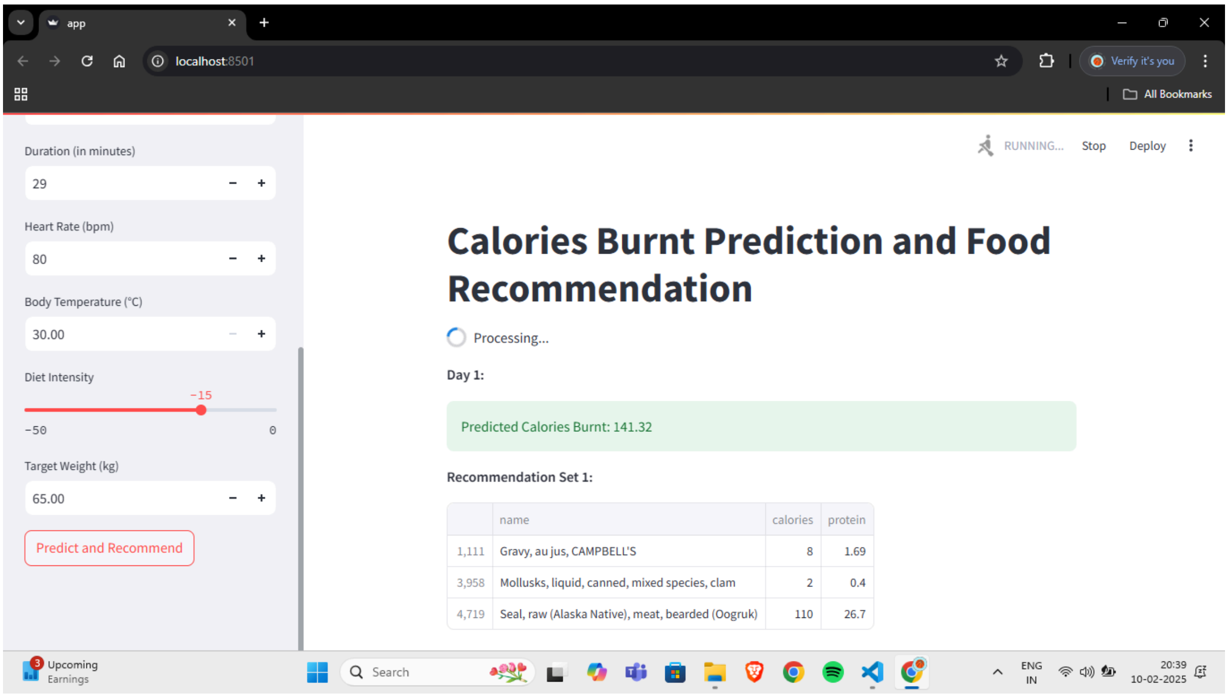


Figure 9: Landing page of AI Nutritionist

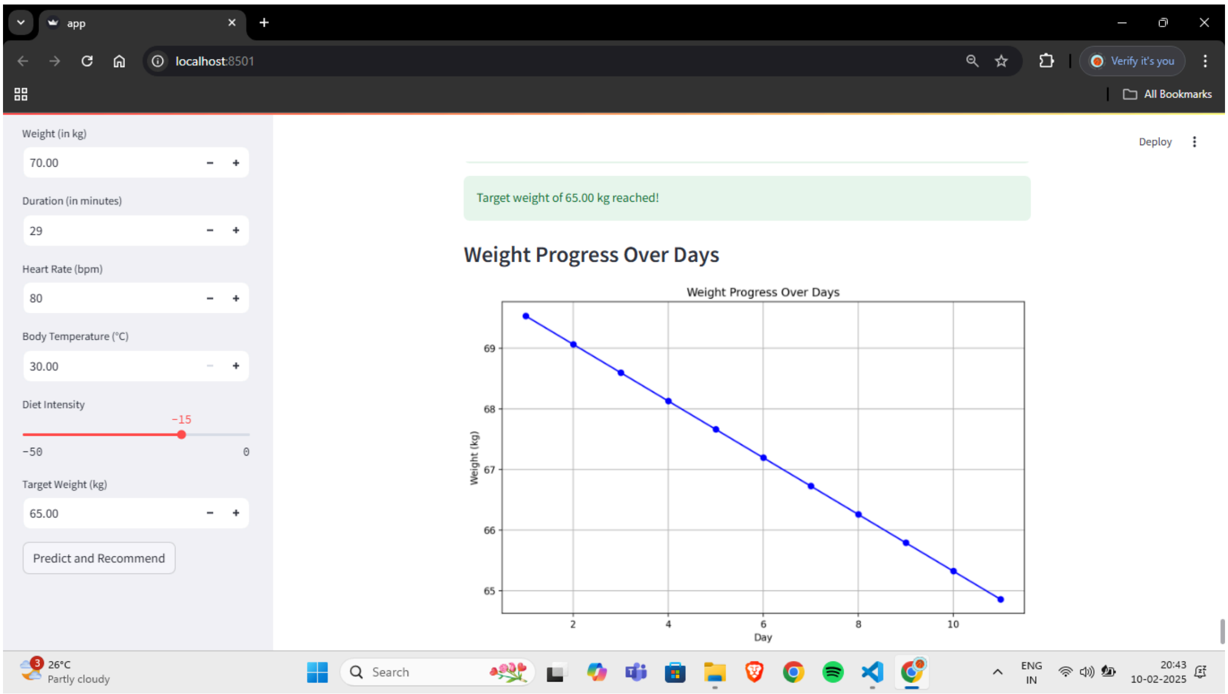


Figure 10: Predicted weight progress throughout the recommended diet

10. CONCLUSION AND FUTURE SCOPE

CONCLUSION

The finished fitness and diet recommendation device efficiently combines gadget reading with linear programming to are awaiting calorie expenditure and optimize dietary intake, tailored to person fitness and health dreams. Utilizing a Random Forest version integrated with consumer-nice interfaces together with Gradio and Streamlit, the device gives accurate predictions of energy burned based on numerous physical attributes and interest ranges. It additionally offers optimized nutritional pointers that maximize dietary consumption without surpassing caloric wishes, essential for balanced diet regime upkeep. This interactive platform simplifies monitoring health and nutritional intake, improving person engagement through an available internet interface. The machine's format allows for scalability and future enhancements, along side integration with wearable era and extra unique dietary monitoring, displaying potential for personalized fitness management and superior public health effects.

FUTURE SCOPE

- **Integration with Wearable Technology:** Improving the accuracy of facts through integrating wearable era to reveal present day fitness and hobby statistics in actual time.
- **Expanded Nutritional Database:** Adding more nutritional options to the food database to deal with a wider range of dietary and cultural alternatives.
- **User Engagement and Interaction:** To improve client motivation and engagement, features like aim-placing, development monitoring, and social sharing are being introduced.
- **Integrating health education:** providing educational substances approximately nutrients and fitness to enable clients to make higher health selections.
- Using statistical analytics to estimate fitness risks and offer preventative recommendation is referred to as predictive health analytics.
- **Mobile App Development:** Creating a cell software program for much less difficult get proper of access to and integration into customers' each day exercise routines.
- **Expansion to Healthcare Services:** Adapting the device to be utilized by way of healthcare businesses to show display screen and manage affected character health more effectively.

11. REFERENCES

1. Dr. N K Sakthivel, Dr. S. Subasree, Mr. Surya Kannan Kumbhar, Mr. C. Jeffrey Hasan, Mr. G. Muruguraj, Mr. Shabin Sj, 2024, AI-Powered Nutrition Assistant and Step Tracker, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) **Volume 13, Issue 04 (April 2024)**
2. "AI-Powered Fitness App for Dynamic Workout Tracking and Personalized Exercise and Nutritional Plan Recommendations", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, **Vol.11, Issue 5, page no.n464-n468, May-2024,** Available :http://www.jetir.org/papers/JETIR2405D66.pdf
3. Machine Learning for Personalized Nutrition and Diet Recommendations, Sarda, A., Almeida, M., & Chen, J. (2019). *Machine Learning Techniques for Personalized Nutrition Recommendation Systems*. Journal of Nutritional Health. This paper discusses the role of machine learning in creating personalized dietary recommendations based on individual user data.
4. KH, Asha. (2024). Al Fitness Model using Deep Learning. International Journal of Advanced Research in Science Communication and Technology. 4. 459- 464. 10.48175/IJARSCT-15361.
5. Tsolakidis, D.; Gymnopoulos, L.P.; Dimitropoulos, K. Artificial Intelligence and Machine Learning Technologies for Personalized Nutrition: A Review. *Informatics* 2024, *11*, 62. https://doi.org/10.3390/informatics11030062

12. ANNEXURE 1 (SOURCE CODE)

Code to train and save the model -

```
import pandas as pd

df=pd.read_csv(r"D:\fitness calorie\exercise.csv")

df2=pd.read_csv(r"D:\fitness calorie\calories.csv")

df = pd.concat([df,df2["Calories"]],axis=1)

X = df.drop(columns = ["Calories"],axis = 1)

y = df["Calories"]

from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test    =    train_test_split(X,y,test_size    =
0.2,random_state=1)

import joblib

def predict(ml_model):

    model=ml_model.fit(X_train,y_train)

    print('Score : {}'.format(model.score(X_train,y_train)))

    y_prediction=model.predict(X_test)

    print('predictions are: \n {}'.format(y_prediction))

    r2_score=metrics.r2_score(y_test,y_prediction)

    print('r2 score: {}'.format(r2_score))

    print('MAE:',metrics.mean_absolute_error(y_test,y_prediction))

    print('MSE:',metrics.mean_squared_error(y_test,y_prediction))

 print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test,y_prediction)))

    model_path="random_forest_regressor.joblib"

    joblib.dump(model, model_path)

    print(f"Model saved at {model_path}")

    sns.distplot(y_test-y_prediction)

from sklearn.ensemble import RandomForestRegressor, metrics
```

```
regression = predict(RandomForestRegressor())

regression

model = load('random_forest_regressor.joblib')

print(X_test.iloc[[10]])

y_prediction = model.predict(X_test.iloc[[n]])

print(f"Prediction for calories burnt:{y_prediction}")

print(f"Actual calories burnt:{y_test.iloc[n]}")

df_3=pd.read_csv(r"D:\fitness calorie\nutrition.csv")

from pulp import LpMaximize, LpProblem, LpVariable, lpSum, LpStatus

threshold = 500

if 'protein' not in df.columns or 'calories' not in df.columns:

    print("Ensure the dataset has 'protein' and 'calories' columns.")

else:

 df['calories']=pd.to_numeric(df['calories'], errors='coerce').fillna(0)

 df['protein'] = pd.to_numeric(df['protein'], errors='coerce').fillna(0)

    solutions = []

    for _ in range(3):  # Generate at least 3 sets of recommendations

        problem = LpProblem("Maximize_Protein", LpMaximize)

            choices = [LpVariable(f"choice_{i}", cat="Binary") for i in
range(len(df))]

           problem += lpSum(choices[i] * df.iloc[i]['protein'] for i in
range(len(df)))

          problem += lpSum(choices[i] * df.iloc[i]['calories'] for i in
range(len(df))) <= threshold  # Calorie threshold

        problem += lpSum(choices) == 3  # Select exactly 3 items

        for solution in solutions:

            problem += lpSum(choices[i] for i in solution) <= 2

        status = problem.solve()
```

```
        if LpStatus[status] != "Optimal":
            print("No more feasible solutions found.")
            break
                selected_items = [i for i in range(len(choices)) if choices[i].varValue == 1]
        solutions.append(selected_items)
        selected_foods = df.iloc[selected_items]
        print(f"Recommendation Set {len(solutions)}:")
        print(selected_foods[['name', 'calories', 'protein']])
             print(f"Total Calories: {selected_foods['calories'].sum()}, Total Protein: {selected_foods['protein'].sum()}\n")
    if not solutions:
        print("No feasible solutions found.")
```

Python(app.py) for streamlit interface -

```
import streamlit as st
import pandas as pd
from pulp import LpMaximize, LpProblem, LpVariable, lpSum, LpStatus
from joblib import load
import matplotlib.pyplot as plt
model = load('random_forest_regressor.joblib')
df = pd.read_csv(r"C:\Users\lavuc\fitness caloriess\fitness calorie\nutrition_updated.csv")
df = df[df['calories'] > 0]
def predict_calories(Gender, Age, Height, Weight, Duration, Heart_Rate, Body_Temp):
     input_data = pd.DataFrame([[Gender, Age, Height, Weight, Duration, Heart_Rate, Body_Temp]],
                                        columns=['Gender', 'Age', 'Height', 'Weight', 'Duration', 'Heart_Rate', 'Body_Temp'])
```

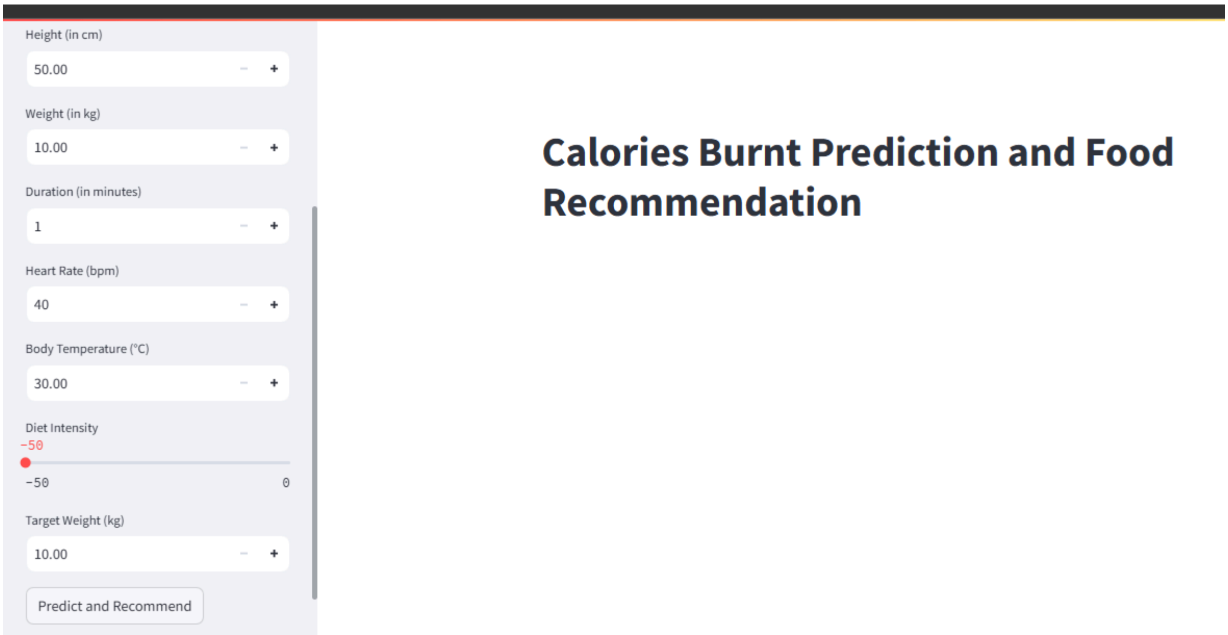
```
    prediction = model.predict(input_data)
    return prediction[0]
def calculate_new_weight(current_weight, total_calories):
    weight_change = total_calories / 770
    new_weight = current_weight - weight_change
    return new_weight
def recommend_foods(predicted_calories, weight_difference_percentage):
    min_calories = (80 + weight_difference_percentage) * predicted_calories / 100
    max_calories = (100 + weight_difference_percentage) * predicted_calories / 100
    df['calories'] = pd.to_numeric(df['calories'], errors='coerce').fillna(0)
    df['protein'] = pd.to_numeric(df['protein'], errors='coerce').fillna(0)
    solutions = []
    excluded_items = set()
    for _ in range(3):  # Generate at least 3 sets of recommendations
        problem = LpProblem("Maximize_Protein", LpMaximize)
        choices = [LpVariable(f"choice_{i}", cat="Binary") for i in range(len(df))]
        problem += lpSum(choices[i] * df.iloc[i]['protein'] for i in range(len(df)))
        total_calories = lpSum(choices[i] * df.iloc[i]['calories'] for i in range(len(df)))
        problem += total_calories >= min_calories  # Minimum calorie limit
        problem += total_calories <= max_calories  # Maximum calorie limit
        problem += lpSum(choices) == 3  # Select exactly 3 items
        for i in excluded_items:
```

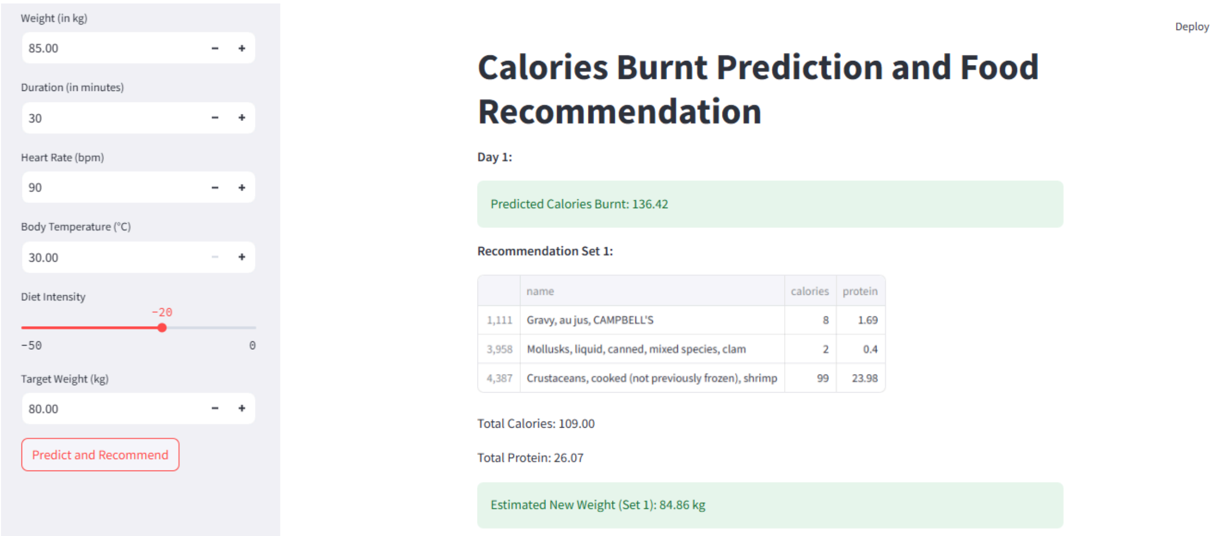
```
            problem += choices[i] == 0  # Exclude already selected items
        status = problem.solve()
        if LpStatus[status] != "Optimal":
            break
        selected_items = [i for i in range(len(choices)) if choices[i].varValue == 1]
        excluded_items.update(selected_items)
        solutions.append(selected_items)
    if not solutions:
        return None
    recommendation_sets = []
    for solution in solutions:
        selected_foods = df.iloc[solution][['name', 'calories', 'protein']]
        recommendation_sets.append(selected_foods)
    return recommendation_sets
st.title("Calories Burnt Prediction and Food Recommendation")
st.sidebar.header("Input Parameters")
Gender = st.sidebar.slider("Gender (0 = Female, 1 = Male)", min_value=0, max_value=1, step=1)
Age = st.sidebar.number_input("Age", min_value=10, max_value=100, step=1)
Height = st.sidebar.number_input("Height (in cm)", min_value=50.0, max_value=250.0, step=1.0)
Weight = st.sidebar.number_input("Weight (in kg)", min_value=10.0, max_value=300.0, step=1.0)
Duration = st.sidebar.number_input("Duration (in minutes)", min_value=1, max_value=300, step=1)
Heart_Rate = st.sidebar.number_input("Heart Rate (bpm)", min_value=40, max_value=200, step=1)
```

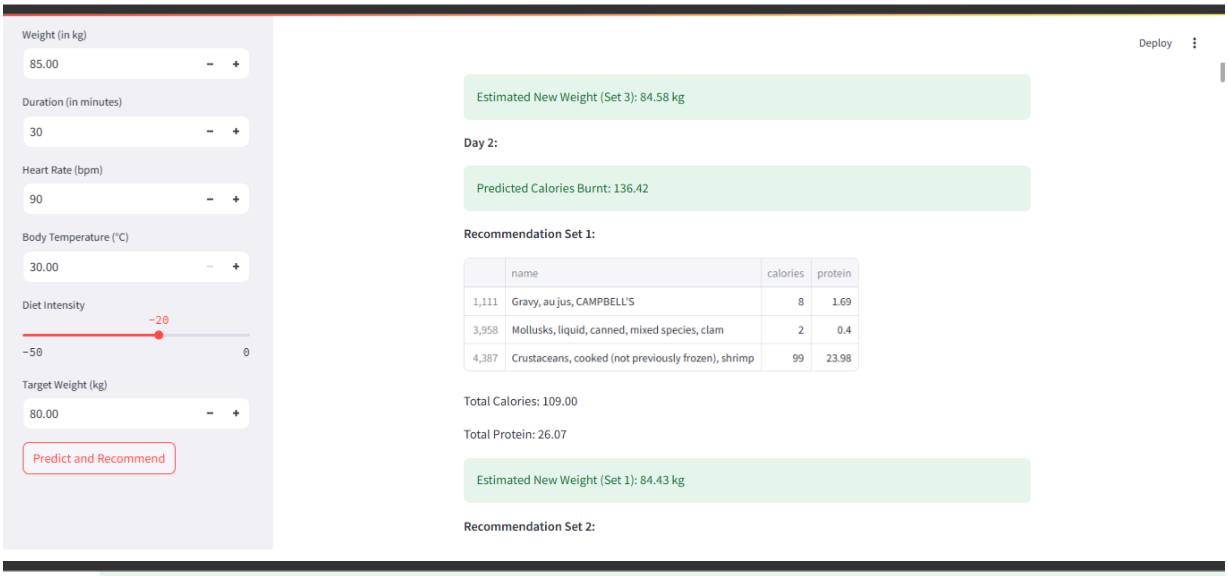
```
Body_Temp = st.sidebar.number_input("Body Temperature (°C)", min_value=30.0, max_value=45.0, step=0.1)

weight_difference_percentage = st.sidebar.slider("Diet Intensity", min_value=-50, max_value=0, step=1)

target_weight = st.sidebar.number_input("Target Weight (kg)", min_value=10.0, max_value=300.0, step=0.1)

if st.sidebar.button("Predict and Recommend"):

    with st.spinner("Processing..."):

        current_weight = Weight

        day = 1

        weight_progress = []

        while current_weight > target_weight:

            st.write(f"**Day {day}:**")

              predicted_calories = predict_calories(Gender, Age, Height, current_weight, Duration, Heart_Rate, Body_Temp)

      st.success(f"Predicted Calories Burnt: {predicted_calories:.2f}")

               recommendation_sets = recommend_foods(predicted_calories, weight_difference_percentage)

            if recommendation_sets is None:

                st.error("No feasible food recommendations found.")

            else:

           for idx, recommended_df in enumerate(recommendation_sets):

                    st.write(f"**Recommendation Set {idx+1}:**")

                    st.dataframe(recommended_df)

                    total_calories = recommended_df['calories'].sum()

                    total_protein = recommended_df['protein'].sum()

                    st.write(f"Total Calories: {total_calories:.2f}")

             st.write(f"Total Protein: {total_protein:.2f}")

       new_weight = calculate_new_weight(current_weight, total_calories)
```

```
    st.success(f"Estimated New Weight (Set {idx+1}): {new_weight:.2f}")
                    current_weight = new_weight
                weight_progress.append(current_weight)
                day += 1
                if current_weight <= target_weight:
                    st.success(f"Target weight of {target_weight:.2f} kg
reached!")
            st.write("### Weight Progress Over Days")
            plt.figure(figsize=(10, 6))
                plt.plot(range(1, day), weight_progress, marker='o',
linestyle='-', color='b')
            plt.title("Weight Progress Over Days")
            plt.xlabel("Day")
            plt.ylabel("Weight (kg)")
            plt.grid(True)
            st.pyplot(plt)
```

13. ANNEXURE 2 (OUTPUT SCREENS)







Recommendation Set 3:

	name	calories	protein
4,858	Cowpeas, without salt, drained, boiled, cooked, leafy tips	22	4.67
5,021	Soup, ready to serve, less/reduced sodium, chicken broth	7	1.36
5,395	Fish, cooked (not previously frozen), Alaska, pollock	80	19.42

Total Calories: 109.00

Total Protein: 25.45

Estimated New Weight (Set 3): 79.91 kg

Target weight of 80.00 kg reached!

Weight Progress Over Days