Major Individual Project(IINET312A)

FItFusion Assistant: Personalized Fitness Generator

Final Project Report

Submitted By: Ashitosh Dhungana

Date: 11-10-2024

Abstract:

Fitfusion is a personalised fitness recommender that seeks to develop a fitness module with detailed exercise regimens that integrates a Flask-based web interface, machine learning models along with MySQL database in order to provide a cutting-edge fitness plan. The key concepts of this project would include User-centered design, fitness goal optimisation, and personal health profiling . The project's integration of several machine learning models for customised diet and exercise recommendations helps the project to place a strong emphasis on high responsiveness, user-friendly design, and data security, all of which are necessary for a satisfying user experience.

A screenshot of a computer

Description automatically generated

Fig: Diagram of how the project operates

Introduction:

FitFusion Assistant is a project that is created in response to the growing need for customised fitness apps. FitFusion employs machine learning to provide customers with personalised advice based on their physical parameters and exercise goals in a digital age when health and fitness are valued highly.

This project outlines the purpose, scope and methodology with developmental milestones, challenges encountered and all the risk mitigation strategies that appear to be associated with the project development.

Additionally, the report will summarize project proposals, literature findings, and projected impacts.

Project purpose and goals:

The primary purpose of FitFusion is to provide the ultimate solution that closes the knowledge gap for young population who want to live healthier lifestyle but lack the direction and self-assurance to do so. Even in this day and age, when the value of physical fitness is widely acknowledged, a sizeable segment of the populace still finds it difficult to start or maintain a fitness journey because they don't know enough about safe exercise practices, healthy eating habits, and typical workout mistakes. FitFusion is justified by the growing demand for accessible, expert-guided fitness advice tailored to individual needs. Conventional fitness methods sometimes overlook the individual needs of each participant, which results in routines that are unproductive and eventually the abandoning of fitness objectives

Meanwhile, the primary objective of the FitFusion Assistant is to provide users with a comprehensive, personalized fitness and wellness experience. To achieve this, following goals were focused:

Personalised Workout and Diet Recommendation:

The objective of this goal is to give consumers the option to get customised nutrition and exercise regimens depending on their age, BMI, fitness objectives, and target muscle groups.

Numerous fitness applications provide general plans that might not suit each user's requirements or objectives. In order to solve this, FitFusion uses machine learning to customise recommendations based on the particular needs of each individual. For instance, FitFusion will give priority to strength-training activities if a user wants to gain muscle. The app will suggest calorie-conscious food choices and aerobic exercises if weight loss is the aim.

With this approach, this project aims to improve user engagement, adherence to all the fitness plans and overall health outcomes.

Progress Tracking:

This project also possesses the facility to track the user’s fitness progress over time, including changes in BMI, weight, and workout performance, strength and provide insights into improvement trends.

As progress tracking is extremely crucial for user satisfaction and adjusting the workout plans, this project uses regression analysis to predict changes in the user’s BMI and weight-based performance. Additionally, this method will help users see quantifiable results and encourage them to stay consistent.

This will allow the users’ motivation to increase as they can observe their path in quantifiable terms . Additionally, this offers input that might help modify diet or exercise regimens to better suit the user's changing requirements.

Common Workout Mistakes and Prevention:

The objective of this goal is to help user avoid common mistake that will potentially lead to injuries, especially for newcomers that will hinder the progress.

Users can learn about common workout errors in this section, such as improper form, overtraining, and omitting warm-ups or cool-downs. To help customers recognise and fix these errors and improve the quality of their workouts, FitFusion offers guidance and visual aids.

Furthermore, the scope of the project would be to provide a comprehensive fitness application that offers users individualised exercise regimens and dietary advice, with a special emphasis on young people looking for customised fitness solutions. The application will collect and process user- specific data which contains Body Mass Index and other fitness related goals which will be utilized to generate workout plans according to each user’s unique physical characteristics.

A user-centric interface will be launched so that users can easily enter their information, browse and choose workout plans, as well as appropriate diet plans and mental health advice for their satisfaction. A sophisticated machine learning model will be used to generate the routine based on user input.

Research Question and Purpose:

The following project seeks to bridge the gap in automated fitness by offering real-time and data driven recommendations for fitness enthusiasts.

Statement of the problem:

The lack of easily accessible, automated technologies that provide genuinely individualised fitness help is the issue our project attempts to solve. While some apps generate fitness recommendations based on scant user data, few deploy extensive machine learning models to modify recommendations over time in response to user progress. Due to a lack of information and the particular gym anxiety that arises from inadequate guidance, new members of the gym must endure a great deal of hardship, this project aims in building a fitness software that continuously modifies its initial recommendations in addition to personalising them so that users obtain the most pertinent advice as they edge closer to their fitness objectives.

Purpose:

The primary purpose of this study is to develop an intelligent fitness app that leverages machine learning to transform personalisatsion in fitness and related activity.

This project intends to show how technology can offer a more efficient and interesting fitness experience by combining safe data management with predictive algorithms. FitFusion aims to assist customers on their fitness journey by offering data-driven, adaptable insights while respecting ethical AI use and data protection norms.

The use of this approach will be addressing both the ethical and technical aspects that is essential for deploying a reliable fitness solution which will eventually contribute to the advancement of knowledge in terms of health and fitness.

In addition to addressing the important issues of data privacy and user-centered design, the goal of this study is to determine how well machine learning models can improve the precision and applicability of fitness regimens. The following enquiries are the focus of the study:

Research Questions:

How does machine learning improve accuracy when it comes to personalizing then fitness plans?

The potential of machine learning algorithms especially, classification and regression models in order to customise diet and fitness advice based on user-specific information such as BMI, fitness objectives, and progress trends is the subject of this enquiry. The study assesses how well these models can forecast appropriate activities, monitor user progress, and modify suggestions to successfully suit specific needs.

How can user information be managed safely and used to make tailored recommendations?

Provided the sensitivity of health-related data, the following question investigates secure data handling procedures during the development of this application. The study highlights how crucial it is to protect user identification and health information by anonymizing user data, keeping it safe, and making sure privacy regulations are followed. The study also considers how to analyze data in a way that preserves confidentiality while producing useful insights.

Project background in a conceptual framework:

This project embraces the development of a user-centric design emphasizing the intersection of health monitoring through machine learning. With the integration of physical body metrics, progress data and fitness goals, the project implements machine learning technique(Random Forest Classifier) to create an adaptive fitness recommender. The framework prioritizes the mentioned activities:

Personlisation:

Deploying machine learning models to provide dynamic and changing recommendations by monitoring and reacting to changes in user progress.

Data Privacy:

The following project will be putting in place strong procedures for managing data securely, guaranteeing that user data is kept safe at all times.

Monitoring and Data Adjustment: The use of machine learning models to provide dynamic and changing recommendations by monitoring and reacting to changes in user progress is a key consideration that this project focuses on.

Literature Review:

When analysing the literature review, it has been evident that it discloses significant progress in terms of the development of personalised fitness application and it has largely attributed to advancements in machine learning.

Numerous studies demonstrate how data-driven methods may revolutionise the prediction and optimisation of fitness regimens, emphasizing the vital role that tailored, flexible recommendations play in attaining successful fitness results.

Current, Balanced, Relevant Research:

The current developments in Personalised Fitness Application depict that the accuracy of fitness recommendations can be greatly increased by including user-specific information, such as age, BMI, and exercise history. Research has indicated that machine learning algorithms, specifically Random Forest Model enhance the precision of exercise regimen and diet predictions, customising them to meet the individual requirements of each user. This finding also aligns with the statement which outlines the message that (AlSlaity et al., 2022) proposed which says that methods from various fields, including healthcare, sports science, and artificial intelligence, have shaped the design of personalized fitness applications using Machine Learning.

Studies and research have also proved that applications like Fitbit, MyFitnessPal use machine learning algorithms to evaluate user input and predict the guidance. With the depiction of(Veeraiah et al., 2023), it has been analyzed that by adapting over time, these applications keep users motivated and support their fitness journeys with increasingly accurate recommendations similar to FitFusion.

Multidisciplinary and Extensive Approach:

Personalised Monitoring for user Engagement:

Research indicates that more user engagement and adherence to exercise regimens are two essential components for long-term health benefits and are also associated with customised fitness plans. Personalised suggestions not only support users' fitness objectives but also make the experience more interesting and inspiring, which improves consistency and retention (Gay and Leijdekkers, 2012). Therefore, Fitfusion is aiming to leverage these insights to deliver optimum satisfaction towards the user by adapting all the recommendations based on the user’s feedback.

The interdisciplinary approaches on health monitoring applications suggest that development of effective health monitoring draws from multiple disciplines including artificial intelligence. These multidisciplinary ideas has guided FitFusion's design, which combines AI-derived machine learning algorithms, psychology-derived motivational strategies, and healthcare-derived health monitoring standards which will be acting as a complete solution for fitness fanatics because of its interdisciplinary approach, which guarantees that it not only offers correct recommendations but also promotes long-term exercise habits.

Comprehensive and clear connection to the proposed research:

Key findings from the literature:

The literature unveils that fitness recommenders are a great source of user engagement and motivation. Research continuously demonstrates that consumers are more likely to stick to fitness regimens that are customised to their individual objectives, tastes, and physical circumstances. This is mostly because fitness becomes more approachable, attainable, and compatible with personal lifestyles when customised strategies are used. Users can advance at their own pace thanks to the tips' adaptable nature, which helps them avoid the typical drawbacks of one-size-fits-all strategies. Furthermore, (Grundy, Wang, and Bero, 2016), depict that long-term user commitment has been demonstrated to increase using adaptive learning, in which a system continuously changes in response to user behavior and feedback. Exercise intensity, frequency, and kind can be dynamically changed by fitness software that applies adaptive learning in response to user feedback and progress. The machine learning-based exercise suggestions made by FitFusion are also based on research findings. Similar to adaptive learning strategies described in research, algorithms are used to generate customised exercise regimens, guaranteeing that each user obtains a training routine appropriate for their unique fitness level, objectives, and physical parameters. This feature helps FitFusion achieve its goal of providing a customised fitness assistant that changes with the user.

FitFusion's design decisions are purposefully in line with the advantages mentioned in the literature. FitFusion seeks to improve user happiness and sustained engagement by integrating personalised recommendations and adaptive learning methods. The literature's demand for data-driven, individualised fitness solutions is directly addressed by the emphasis on machine learning suggestions and classsifictaion based progress tracking(Miah et al., 2022).

Project Methodology:

Drawing comparisons with brands like Fitbit and MyFitnessPal, it can be contextualized that this project highlights the innovations of strength and fitness to offer a unique approach, and this is how it has been planned.

Research Design :Justification and Appropriateness

The research design of Fitfusion comes from a data driven, user-centric approach aimed at designing a personalised fitness solution. This architecture is built on AI and machine learning techniques and allows for dynamic diet and fitness suggestions based on user data.

Provided the goal of personalised fitness recommendations, a machine learning methodology is well-suited for this project. Random Forest for workout prediction and diet plan prediction is a machine learning technique that enable suggestions to be continuously improved based on user progress. This makes it possible for FitFusion to adapt to each user's particular fitness path and offer timely and pertinent recommendations.

When mentioning the strength, this machine learning method has a high scale provision of adaptability, scalability and efficiency. Real-time generation of personalised recommendations can increase user motivation and engagement.

Similarly, data dependency can be considered as a potential limitation on this project. FitFusion’s recommendations are dependent on the quantity of the user data that are collected. Other important factors are protecting user privacy and managing user data in an ethical manner.

Intended Population and Selection Criteria:

The ones who are targeted for the use of FitFusion are individuals(especially newcomers) in the gym seeking a guided and a personalised fitness experience. On the other hand, it can be used by experienced individuals as well who aim to improve their fitness standards.

The sampling of the users is classified below:

Users are sampled based on their fitness goals (e.g., weight loss, muscle building, maintenance), age, BMI, and initial fitness level. This information is collected during onboarding to tailor the initial set of recommendations.

Users that consistently engage over a specified time period with a specified workout are given preference for the initial data gathering and to train and fine-tune the algorithms. This guarantees that the training data is thorough and records how user behavior and progress evolve over time.

Data collection and Processing Tools:

FitFusion’s user interface serves as the primary instrument for data collection. It includes input fields for initial parameters where users provide baseline metrics of their body such as age, height, weight, fitness goal along with the muscle group to target with a progress tracking dashboard.

The database used will be MySQL to store user information, including workout and diet data. Also, tables are structured to capture initial user details and exercise performance metrics.

Similarly, the Machine Learning Model used for the recommendations of exercise and diet will be Random Forest Classifier and Decision Tree.

Data Collection:Tools and Measurement Methods

Age, BMI, muscle group, fitness objective, workout progress, weight change, and BMI change are the main variables. These variables are used to monitor user progress and customise the recommendations.

The tools used are as follows:

Flask framework for the web app and real-time user interaction.

MySQL database for storing structured user data.

Scikit-learn for implementing machine learning models.

The progress measurement is carried out via the user’s engagement level(consistency and frequency of workouts)along with the changes in BMI and adherence to the workout and diet recommendations.

Reasons for choosing these tools and data collection techniques:

Machine Learning Methods:

Because of its resilience and adaptability to different fitness objectives and user types, the Random Forest Classifier was chosen. Similarly, due to its interpretability, which enables people to comprehend the rationale behind the recommendations for particular diets, the Decision Tree was selected for dietary recommendations.

For the purpose of database storage, real-time data retrieval and storage are made possible by MySQL's structured data storage and simple Flask integration, therefore it was chosen.

Data collection techniques: Feedback forms and direct user participation guarantee that the data is current and pertinent. Continuous data from user progress tracking is essential for enhancing recommendations.

Data Analysis:

Data Preprocessing: The data is cleaned, normalized and transformed according to the need of the user and then converted into numerical data for its appropriate use in the machine learning algorithm.

Training:

Initial users' history data is used to train and evaluate the Random Forest and Decision Tree models. Cross-validation is used to prevent overfitting and guarantee model accuracy.

Performance Metrics: The models are evaluated using metrics such as accuracy for classification tasks (workout and diet recommendations).

Potential threats to reliability:

Data Quality:

The accuracy of the model may be impacted by missing data resulting from inconsistent user interaction. This is addressed by using only users who consistently enter data for validation and training.

User Variation:

Fitness levels, plan adherence, and user motivation can all affect how reliable the progress-tracking model is. Reliability is increased as the system adjusts to a variety of user behaviours by regularly adding new data to the models.

Data Privacy:

Data privacy is a key concern with personalised applications. To protect user information, measures like access restriction and data encryption are put in place.

Planning and Analysis:

Fitfusion is designed to address the knowledge gap when it comes to the field of fitness applications by introducing machine learning mechanisms for workout and diet recommendations.

While traditional fitness applications provide general information, they frequently lack real-time, adaptive counsel based on user success. By investigating how tailored, data-driven insights might enhance user engagement, adherence, and overall fitness outcomes, this study seeks to advance the field of fitness technology.

The advance contributions provided by this application are:

Adaptive fitness planning: Unlike conventional, static fitness regimens, FitFusion adapts fitness suggestions depending on individual metrics through ongoing learning from user data.

Holistic approach: By incorporating workout plans, diet suggestions, progress tracking, common errors that people make in the gym along with the methods to rectify them, FitFusion offers a comprehensive solution that supports users in various aspects of their fitness journey.

Display of machine learning and its application: A framework for future fitness applications may be provided by the study's successful application of Random Forest and Decision Tree models for individualised fitness planning.

Timelines and Planning:

The following project was divided into multiple phases, with certain milestones to ensure appropriate progression.

Phase 1: Literature review and requirements collection (2 weeks)

Phase 2: Three Weeks of Data Collection and Preprocessing

Phase 3: Training and Model Selection (4 weeks)

Random Forest for workout recommendations and Decision Tree for diet recommendations.

Phase 4: Three Weeks of Integration with the Flask Web Application.

Phase 5: Two Weeks of Testing and Iteration.

User testing was part of this to improve suggestions in response to comments.

Phase 6: Two Weeks of Documentation, Final Presentation, Implementation and Report Writing.

The statistical data analysis used in this project can be classified as follows:

Random Forest and Decision Tree Algorithms: These methods were used for workout recommendations, ensuring that each user receives a tailored workout plan and applied to diet recommendations, allowing transparent decision-making based on user inputs.

Cross-Validation: Several data subsets were used to validate the results, which guaranteed model reliability and decreased overfitting.

Therefore, the following research shows how machine learning may revolutionise fitness applications by offering a strong basis for individualised exercise assistance. FitFusion fills an important gap in user-centered health technology by accommodating individual requirements and growth. This model's scalability may encourage more advancements in digital health, opening up more access to individualised wellness assistance.

Ethical Considerations for the Project:

Recruitment and Consent of Participants:

FitFusion is a user-centered fitness app, thus all users are given explicit information about the type of data that is gathered and why.

Upon registration, consumers are informed of how their data will be used to generate customised fitness suggestions, and their consent is requested.

In order to respect their autonomy and control over their personal information, users are granted the option to opt out or erase their data.

Data Confidentiality:

Considering the gathering of personal health data (e.g., age, BMI, and exercise progress), data privacy is a crucial factor. Thus, to stop unwanted access or misuse of user data, the app employs stringent privacy safeguards, such as data anonymization and encryption.

Transparency:

With a transparent data usage policy available within the app, users are well-informed about data usage. FitFusion uses data only to enhance user experience and no information is shared with exterior services without the express approval of the user.

Documentation and Compliance:

The project manager of this project is responsible and aware and complies with established norms thorough documentation of all ethical and safety measures.  
To keep the project to be at its best, the execution of most recent ethical software development and data protection practices are chosen with regular training sessions being held.

Conclusion:

Thus, FitFusion uses machine learning to offer a customised exercise guide that adjusts to each user's particular fitness and health objectives. FitFusion is a novel solution that meets the increasing demand for easily accessible and customised fitness advice by fusing developments in AI, user-centered design, and data security procedures. In addition to addressing a need for automated fitness support, this project places a high value on privacy, safety, and ethical issues, making it a complete and responsible tool for people looking to enhance their health and wellbeing. To conclude, FitFusion sets a bar for future advancements in the fitness and health monitoring sector by showcasing the potential of technology to improve individual fitness journeys.

References:

Gay, V. and Leijdekkers, P., 2012. Personalised mobile health and fitness apps: lessons learned from myFitnessCompanion®. In pHealth 2012 (pp. 248-253). IOS Press.

AlSlaity, A., Suruliraj, B., Oyebode, O., Fowles, J., Steeves, D. and Orji, R., 2022. Mobile applications for health and wellness: a systematic review. Proceedings of the ACM on Human-Computer Interaction, 6(EICS), pp.1-29.

Miah, J., Mamun, M., Rahman, M.M., Mahmud, M.I., Ahmad, S. and Nasir, M.H.B., 2022, December. Mhfit: Mobile health data for predicting athletics fitness using machine learning models. In 2022 2nd International seminar on machine learning, Optimization, and Data Science (ISMODE) (pp. 584-589). IEEE.

Veeraiah, V., Ramesh, J.V.N., Koujalagi, A., Talukdar, V., Namdev, A. and Gupta, A., 2023, August. Health Fitness Tracker System Using Machine Learning Based on Data Analytics. In International Conference on Mobile Radio Communications & 5G Networks (pp. 765-775). Singapore: Springer Nature Singapore.

Grundy, Q.H., Wang, Z. and Bero, L.A., 2016. Challenges in assessing mobile health app quality: a systematic review of prevalent and innovative methods. American journal of preventive medicine, 51(6), pp.1051-1059.

Kelleher, J.D., Mac Namee, B. and D'arcy, A., 2020. Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT press.

Project Code:

from flask import Flask, render\_template, request, redirect, url\_for, flash, make\_response  
from flask\_sqlalchemy import SQLAlchemy  
import pandas as pd  
import numpy as np  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.preprocessing import LabelEncoder  
import pdfkit  
  
app = Flask(\_\_name\_\_)  
app.secret\_key = 'supersecretkey'  
  
# Configure the MySQL database connection  
app.config['SQLALCHEMY\_DATABASE\_URI'] = 'mysql+pymysql://root:%40pplE%21375@localhost/fitdata'  
app.config['SQLALCHEMY\_TRACK\_MODIFICATIONS'] = False  
  
# Initialize the database  
db = SQLAlchemy(app)  
  
# Define the WorkoutData model to map to the fitness\_recommender table  
class WorkoutData(db.Model):  
 \_\_tablename\_\_ = 'fitness\_recommender'  
  
 id = db.Column(db.Integer, primary\_key=True, autoincrement=True)  
 age = db.Column(db.Integer, nullable=False)  
 bmi = db.Column(db.Float, nullable=False)  
 muscle\_group = db.Column(db.String(50), nullable=False)  
 specific\_muscle = db.Column(db.String(50), nullable=False)  
 fitness\_goal = db.Column(db.String(50), nullable=False)  
 workout\_plan = db.Column(db.String(50), nullable=False)  
 exercises = db.Column(db.Text, nullable=False)  
 exercise\_image\_1 = db.Column(db.String(255), nullable=True)  
 exercise\_image\_2 = db.Column(db.String(255), nullable=True)  
 exercise\_image\_3 = db.Column(db.String(255), nullable=True)  
 diet\_recommendation = db.Column(db.String(50), nullable=False)  
 daily\_calorie\_intake = db.Column(db.Integer, nullable=False)  
 diet\_details = db.Column(db.Text, nullable=True)  
 initial\_weight = db.Column(db.Float, default=0)  
 mistakes\_performed = db.Column(db.Text, nullable=True)  
 video\_link = db.Column(db.String(255), nullable=True)  
 strength\_level = db.Column(db.Integer, default=50)  
 weight\_change = db.Column(db.Float, default=0)  
 body\_fat\_percentage\_change = db.Column(db.Float, default=0)  
 training\_duration = db.Column(db.Integer, default=1)  
  
@app.route('/')  
def home():  
 return render\_template('index.html')  
  
@app.route('/calculate\_bmi', methods=['POST'])  
def calculate\_bmi():  
 try:  
 # Get user input from form  
 height = float(request.form['height'])  
 weight = float(request.form['weight']) # Get current weight from form  
 age = int(request.form['age'])  
 fitness\_goal = request.form['fitness\_goal'].lower()  
 muscle\_group = request.form['muscle\_group'].lower()  
 specific\_muscle = request.form['specific\_muscle'].lower()  
  
 # Calculate BMI  
 bmi = weight / (height \* height)  
  
 # Fetch the user's workout data from the database  
 workout\_data = WorkoutData.query.all()  
  
 # Prepare the data for training the Random Forest model  
 df = pd.DataFrame([(d.age, d.bmi, d.muscle\_group, d.specific\_muscle, d.fitness\_goal, d.workout\_plan,  
 d.diet\_recommendation, d.daily\_calorie\_intake)  
 for d in workout\_data],  
 columns=['age', 'bmi', 'muscle\_group', 'specific\_muscle', 'fitness\_goal',  
 'workout\_plan', 'diet\_recommendation', 'daily\_calorie\_intake'])  
  
 # Encode categorical columns  
 le\_goal = LabelEncoder()  
 le\_muscle = LabelEncoder()  
  
 df['muscle\_group'] = le\_muscle.fit\_transform(df['muscle\_group'])  
 df['specific\_muscle'] = df['specific\_muscle'].astype('category').cat.codes  
 df['fitness\_goal'] = le\_goal.fit\_transform(df['fitness\_goal'])  
 df['workout\_plan'] = df['workout\_plan'].astype('category')  
  
 # Features and target for workout plan prediction  
 X = df[['age', 'bmi', 'muscle\_group', 'specific\_muscle', 'fitness\_goal']]  
 y = df['workout\_plan'].cat.codes  
  
 # Train the Random Forest Classifier model for workout plans  
 rf\_model = RandomForestClassifier(n\_estimators=100, random\_state=42)  
 rf\_model.fit(X, y)  
  
 # Encode the user input for prediction  
 fitness\_goal\_encoded = le\_goal.transform([fitness\_goal])[0]  
 muscle\_group\_encoded = le\_muscle.transform([muscle\_group])[0]  
 specific\_muscle\_encoded = pd.Series([specific\_muscle]).astype('category').cat.codes[0]  
  
 # Input data for prediction  
 input\_data = np.array([[age, bmi, muscle\_group\_encoded, specific\_muscle\_encoded, fitness\_goal\_encoded]])  
  
 # Predict workout plan  
 predicted\_plan\_encoded = rf\_model.predict(input\_data)[0]  
 predicted\_plan = df['workout\_plan'].cat.categories[predicted\_plan\_encoded]  
  
 # Fetch specific exercises and images based on predicted plan  
 recommended\_exercises = WorkoutData.query.filter\_by(workout\_plan=predicted\_plan,  
 specific\_muscle=specific\_muscle).first()  
  
 if recommended\_exercises:  
 exercises = recommended\_exercises.exercises  
 image\_1 = recommended\_exercises.exercise\_image\_1  
 image\_2 = recommended\_exercises.exercise\_image\_2  
 image\_3 = recommended\_exercises.exercise\_image\_3  
 diet\_recommendation = recommended\_exercises.diet\_recommendation  
 daily\_calorie\_intake = recommended\_exercises.daily\_calorie\_intake  
 diet\_details = recommended\_exercises.diet\_details  
  
 # Fetch mistakes and video link for the exercises  
 mistakes = recommended\_exercises.mistakes\_performed  
 video\_link = recommended\_exercises.video\_link  
 else:  
 exercises = "No exercises found."  
 image\_1 = image\_2 = image\_3 = "images/default.jpg"  
 diet\_recommendation = "No diet recommendation found."  
 daily\_calorie\_intake = "No calorie intake data available."  
 diet\_details = "No detailed diet plan available."  
 mistakes = "No workout mistakes available."  
 video\_link = "#"  
  
 # Check if initial\_weight is 0 and set it to the current weight  
 if recommended\_exercises and recommended\_exercises.initial\_weight == 0:  
 recommended\_exercises.initial\_weight = weight  
 db.session.commit() # Save the updated initial weight to the database  
  
 # Render the results.html template, passing all the data  
 return render\_template('results.html',  
 bmi=bmi,  
 plan=predicted\_plan,  
 exercises=exercises,  
 image\_1=image\_1,  
 image\_2=image\_2,  
 image\_3=image\_3,  
 diet=diet\_recommendation,  
 calories=daily\_calorie\_intake,  
 diet\_details=diet\_details,  
 mistakes=mistakes,  
 video\_link=video\_link)  
  
 except ValueError:  
 flash('Please enter valid numbers for height, weight, and age.', 'error')  
 return redirect(url\_for('home'))  
  
  
@app.route('/update\_progress', methods=['POST'])  
def update\_progress():  
 try:  
 # Get progress tracking data from form  
 current\_weight = float(request.form['current\_weight'])  
 strength\_lifted = float(request.form['strength\_lifted'])  
 body\_fat\_percentage = request.form.get('body\_fat\_percentage', None)  
  
 if body\_fat\_percentage:  
 body\_fat\_percentage = float(body\_fat\_percentage)  
  
 # Fetch the user's data from the database  
 user\_data = WorkoutData.query.first()  
  
 # Calculate weight change based on initial\_weight (already stored)  
 initial\_weight = user\_data.initial\_weight  
 weight\_change = current\_weight - initial\_weight  
  
 # Calculate strength level increase  
 initial\_strength = 50 # Placeholder for initial strength  
 strength\_increase = (strength\_lifted - initial\_strength) / initial\_strength \* 100  
  
 # Optional: Calculate body fat percentage change  
 if body\_fat\_percentage:  
 initial\_body\_fat = 20 # Placeholder, adjust as needed  
 body\_fat\_change = body\_fat\_percentage - initial\_body\_fat  
 else:  
 body\_fat\_change = None  
  
 # Update the user's progress in the database  
 user\_data.weight\_change = weight\_change  
 user\_data.strength\_level = strength\_increase  
 user\_data.training\_duration += 1 # Increment training duration  
 if body\_fat\_change is not None:  
 user\_data.body\_fat\_percentage\_change = body\_fat\_change  
  
 db.session.commit() # Save the updated values  
  
 # Fetch mistakes and video link again  
 mistakes = user\_data.mistakes\_performed  
 video\_link = user\_data.video\_link  
  
 flash('Progress updated successfully!', 'success')  
  
 # Render the results page again with updated progress data and workout mistakes  
 return render\_template('results.html',  
 bmi=user\_data.bmi,  
 plan=user\_data.workout\_plan,  
 exercises=user\_data.exercises,  
 image\_1=user\_data.exercise\_image\_1,  
 image\_2=user\_data.exercise\_image\_2,  
 image\_3=user\_data.exercise\_image\_3,  
 diet=user\_data.diet\_recommendation,  
 calories=user\_data.daily\_calorie\_intake,  
 diet\_details=user\_data.diet\_details,  
 weight\_change=weight\_change,  
 strength\_level=strength\_increase,  
 body\_fat\_percentage\_change=body\_fat\_change,  
 training\_duration=user\_data.training\_duration,  
 mistakes=mistakes,  
 video\_link=video\_link)  
  
 except Exception as e:  
 print(f"Error: {e}")  
 flash('There was an error updating your progress. Please try again.', 'danger')  
 return redirect(url\_for('home'))  
  
  
# PDF Generation Route  
config = pdfkit.configuration(wkhtmltopdf=r'C:\Program Files\wkhtmltopdf\bin\wkhtmltopdf.exe')  
  
@app.route('/generate\_pdf', methods=['POST'])  
def generate\_pdf():  
 diet = request.form.get('diet', 'Balanced Diet') # Default to Balanced Diet  
 calories = 2200  
 breakfast = "Oats with fresh fruits and almonds."  
 lunch = "Grilled chicken with quinoa and mixed vegetables."  
 dinner = "Salmon with steamed broccoli and sweet potato."  
 snacks = "Greek yogurt, a handful of nuts."  
  
 # Determine the image based on the diet  
 if diet == "Low Carb Diet":  
 diet\_image = "images/low\_carb.png"  
 breakfast = "Scrambled eggs with avocado."  
 lunch = "Grilled salmon with a side of spinach."  
 dinner = "Zucchini noodles with a tomato sauce."  
 elif diet == "High Protein Diet":  
 diet\_image = "images/high\_protein.png"  
 breakfast = "Egg white omelette with spinach."  
 lunch = "Grilled chicken breast with quinoa and vegetables."  
 dinner = "Turkey meatballs with brown rice."  
 else:  
 diet\_image = "images/balanced\_diet.png"  
  
 rendered = render\_template('diet\_pdf.html',  
 diet=diet,  
 calories=calories,  
 breakfast=breakfast,  
 lunch=lunch,  
 dinner=dinner,  
 snacks=snacks,  
 diet\_image=diet\_image)  
  
 pdf = pdfkit.from\_string(rendered, False, configuration=config)  
 response = make\_response(pdf)  
 response.headers['Content-Type'] = 'application/pdf'  
 response.headers['Content-Disposition'] = 'attachment; filename=diet\_plan.pdf'  
 return response  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 with app.app\_context():  
 db.create\_all()  
  
 app.run(debug=True)