Exercise and Diet by AI

A Comprehensive AI-Powered Health & Fitness Platform

HELWAN UNIVERSITY

Faculty of Computing and Artificial Intelligence

Information Systems Department

Graduation Project Team

- Omar Zakaria Mohy ALdeen (20210603)
- Seif Eldin Ahmed Mohamed (20210434)
- Ali Ahmed Abdelmawgoud (20210573)
- Omar Khaled Abdo Hussien (20210599)
- Ziad Mohamed Abdallah Hassan (20210374)
- Shehab Tamer Salama (20210458)

Supervised by: Dr. Mostafa Adel

June 2025

Abstract & Overview

Project Abstract

This project details the design, development, and implementation of a comprehensive **Diet and Exercise Plans Website**. The platform provides a robust solution for individuals seeking personalized fitness and nutrition guidance, connecting them with professional coaches and leveraging **artificial intelligence** for tailored diet recommendations.

The system supports two primary user roles: **Trainees**, who receive customized plans and track their progress, and **Coaches**, who manage their subscribed trainees and create bespoke exercise programs.

Goal: Empowering users to achieve health and fitness goals through structured plans, expert guidance, and peer interaction.

Key Features

- **AI-Powered Diet Plans:** Personalized nutrition recommendations based on individual metrics and preferences
- **Professional Coaching:** Direct connection with certified coaches for customized exercise programs
- **Progress Tracking:** Comprehensive monitoring of fitness journey and goal achievement
- **Community Platform:** Vibrant community for sharing experiences, motivation, and peer support
- **Multi-Platform Access:** Responsive web design for desktop and mobile accessibility

Problem Statement

Challenges in Traditional Fitness & Nutrition Approaches

Lack of Personalization

Generic diet plans and exercise routines fail to account for individual variations in body type, dietary preferences, health conditions, fitness levels, and lifestyle constraints. This one-size-fits-all approach leads to plans that are difficult to adhere to and ultimately ineffective.

Limited Professional Access

Many individuals lack affordable and convenient access to qualified fitness coaches and nutritionists. Traditional coaching models are expensive, geographically restrictive, and time-consuming, creating barriers for those who need expert guidance most.

Inefficient Progress Tracking

Users struggle to effectively track their progress, understand the impact of their efforts, and receive timely, actionable feedback. This lack of clear progress visibility leads to demotivation and inability to make necessary plan adjustments.

Absence of Community Support

The fitness journey can be isolating. Traditional methods don't provide platforms for users to connect with peers, share experiences, seek motivation, or engage in supportive environments, which is crucial for long-term adherence and success.

Overall Impact

These limitations result in **suboptimal outcomes**, **widespread user disengagement**, and **failure to achieve sustainable health and fitness goals**. Static, non-responsive recommendations that don't adapt to changing needs further hinder continuous improvement.

Scope and Objectives

Project Scope

- **Dual User Roles:** Distinct functionalities for Trainees and Coaches with specialized interfaces and capabilities
- **User Authentication:** Secure user registration, login, and comprehensive profile management system
- **AI-Powered Diet Plans:** Intelligent diet plan generation based on user metrics and preferences
- **Exercise Management:** Coach-assigned exercise plan creation, modification, and progress tracking
- **Community Platform:** Interactive forum for user engagement, experience sharing, and peer support

Key Objectives

- **Personalized Health Solutions:** Deliver customized fitness and nutrition guidance tailored to individual needs
- **Professional Connection:** Bridge the gap between users and qualified fitness professionals
- **Progress Empowerment:** Enable effective tracking and monitoring of fitness journey milestones
- **Community Building:** Foster a supportive environment for motivation and peer interaction
- **Accessibility:** Provide a robust, scalable, and user-friendly application across multiple platforms

Future Enhancements

Advanced analytics, real-time chat functionalities, and integration with wearable devices are considered for subsequent development phases to further enhance user experience and platform capabilities.

Functional Requirements

User Management

- User registration and account creation for **Trainees and Coaches**
- Secure login system with **credential authentication**
- Profile management including **name, email, and photo**
- Support for **distinct user roles** with role-based access

AI Diet Management

- AI-driven **diet plan generation** based on user metrics
- Input collection for **height, weight, preferences**
- Personalized recommendations using **machine learning algorithms**
- Plan viewing and **adjustment requests** functionality

Exercise Management

- Trainee **subscription to coaches** for personalized plans
- Coach dashboard to **view subscribed trainees**
- Exercise plan **creation and editing** capabilities
- Progress tracking with **weight and repetition updates**

Community Features

- Community page accessible to **all users**
- Post upload and **content sharing** capabilities
- View posts from **other community members**
- Content moderation for **appropriate community standards**
- Coach notifications for **plan updates**

Non-Functional Requirements

Performance

- Login response: **≤ 2 seconds**
- Plan generation: **≤ 5 seconds**
- Support **concurrent users** without degradation

Security

- Secure **authentication & authorization**
- Data encryption **in transit & at rest**
- Protection against **SQL injection, XSS**

Usability

- **Intuitive interface** for both user roles
- Clear **feedback** on user actions
- **Responsive design** for all devices

Reliability

- System availability: **99.9%**
- Robust **error handling** mechanisms
- Maintain **data integrity** through validation

Scalability

- Support **future expansion** in users & features
- Integration capability for **new AI models**
- Accommodate **growing data volume**

Maintainability

- **Well-documented** and modular codebase
- Adherence to **coding standards**
- Compatible with **modern browsers**

System Architecture & Design

System Actors

User (General)

Base actor representing any individual interacting with the system. Both Trainees and Coaches are specialized types of Users.

Trainee

Specialized user seeking fitness guidance and programs. Interacts with coaches, follows training programs, and participates in community engagement.

Coach

Specialized user providing training programs and managing trainees. Creates exercise plans and contributes to the community platform.

Key Use Cases

- **Login:** User authentication and system access
- **View Home Page:** Access to personalized dashboard
- **View Community:** Browse community forum and posts
- **Generate Diet Plan:** AI-powered nutrition recommendations
- **Manage Exercise Plans:** Create and track workout routines
- **Track Progress:** Monitor fitness journey and achievements

Technology Stack

FastAPI	Spring Boot	Thymeleaf	MySQL/PostgreSQL
Backend Framework	Java Framework	Template Engine	Database Systems

Recommendation System Architecture

User Input

Nutritional requirements (9 features) and optional ingredient preferences

Data Preprocessing

StandardScaler normalization and feature extraction from dataset

Nearest Neighbors

Cosine similarity with brute-force algorithm for recipe matching

Ingredient Filtering

Optional regex-based filtering for specific ingredient requirements

Recipe Output

Top N recommended recipes with nutritional information and instructions

Algorithm Components

Similarity Metric:
Cosine similarity for measuring recipe similarity based on nutritional profiles

Search Algorithm:
Brute-force approach optimized for small to medium datasets

Normalization:
StandardScaler for feature scaling and improved similarity calculations

Performance Metrics

Fast computation for datasets up to 231K+ recipes with sub-second response times

System Features

Configurable Parameters:
Adjustable number of neighbors and distance return options

Ingredient Filtering:
Regex-based filtering for dietary restrictions and preferences

Pipeline Architecture:
Scikit-learn pipeline for streamlined preprocessing and prediction

API Integration:
FastAPI implementation with Pydantic models for type safety

Dataset Overview

Recipe Dataset

231K+
Recipes

20+
Features

9
Nutrition Values

CSV
Format

Key Columns:

- RecipeId, Name, AuthorName
- CookTime, PrepTime, TotalTime
- RecipeIngredientParts
- Nutritional Content (9 features)
- RecipeInstructions

The dataset contains comprehensive recipe information including detailed nutritional breakdowns, cooking instructions, and ingredient lists for machine learning analysis.

Nutritional Features

Calories:
Total energy content per serving

Fat Content:
Total and saturated fat amounts

Cholesterol:
Cholesterol content in milligrams

Sodium:
Salt content for dietary restrictions

Carbohydrates:
Total carbs, fiber, and sugar content

Protein:
Protein content for muscle building

Data Processing Workflow

Data Loading
CSV file with 231K+ recipes

→

Preprocessing
Extract nutrition features

→

Scaling
StandardScaler normalization

→

Similarity
Cosine similarity matching

Nearest Neighbors Algorithm

Algorithm Theory

The **Nearest Neighbors algorithm** is an unsupervised learning method that finds the most similar items in a dataset based on feature similarity. For diet recommendations, it identifies recipes with nutritional profiles closest to user requirements.

Cosine Similarity Formula

$$\cos(\theta) = (A \cdot B) / (||A|| \times ||B||)$$

Measures the cosine of the angle between two vectors A and B

```
# Sklearn Implementation from sklearn.neighbors import
NearestNeighbors neigh = NearestNeighbors( metric='cosine',
algorithm='brute' ) neigh.fit(prep_data)
```

Implementation Steps

1. **Data Preprocessing**
   Extract nutritional features (columns 6-15) and apply StandardScaler normalization
2. **Model Training**
   Fit NearestNeighbors model with cosine metric and brute-force algorithm
3. **Pipeline Creation**
   Build sklearn Pipeline combining scaler and nearest neighbors transformer
4. **Similarity Search**
   Find k-nearest neighbors based on user's nutritional input vector
5. **Result Filtering**
   Apply optional ingredient-based filtering using regex patterns

Algorithm Advantages

Fast Computation
Brute-force approach optimized for small datasets

High Accuracy
Cosine similarity provides precise nutritional matching

No Training Required
Instance-based learning with immediate deployment

ML Pipeline & Data Processing

Data Extraction

Extract nutritional features from columns 6-15 of the recipe dataset

Scaling

Apply StandardScaler normalization to ensure feature equality

Model Fitting

Train NearestNeighbors model with cosine similarity metric

Prediction

Find k-nearest neighbors and return recommended recipes

Data Preprocessing

```
def scaling(dataframe): scaler = StandardScaler() # Extract nutritional columns 6-15 prep_data = scaler.fit_transform( dataframe.iloc[:,6:15].to_numpy() ) return prep_data, scaler
```

- Calories (energy content)
- Fat Content (total fats)
- Saturated Fat Content
- Cholesterol Content
- Sodium Content
- Carbohydrate Content
- Fiber Content
- Sugar Content
- Protein Content

Pipeline Architecture

```
def build_pipeline(neigh, scaler, params): transformer = FunctionTransformer( neigh.kneighbors, kw_args=params ) pipeline = Pipeline([ ('std_scaler', scaler), ('NN', transformer) ]) return pipeline
```

```
def apply_pipeline(pipeline, _input, data): _input = np.array(_input).reshape(1, -1) indices = pipeline.transform(_input)[0] return data.iloc[indices]
```

Pipeline Benefits

- **Modularity:** Separate preprocessing and prediction steps
- **Reusability:** Consistent scaling for new predictions
- **Efficiency:** Streamlined data transformation workflow
- **Maintainability:** Easy to modify individual components

API Implementation

FastAPI Structure

```
from fastapi import FastAPI from pydantic import BaseModel, conlist import pandas as pd from model import recommend app = FastAPI() dataset = pd.read_csv('../Data/dataset.csv')
```

GET `/`

Health check endpoint to verify API status

POST `/predict/`

Main prediction endpoint for recipe recommendations

```
@app.post("/predict/") def update_item(prediction_input: PredictionIn): recommendation_df = recommend( dataset, prediction_input.nutrition_input, prediction_input.ingredients, prediction_input.params.dict() ) output = output_recommended_recipes(recommendation_df) return {"output": output}
```

Pydantic Models

PredictionIn

Input model for recipe recommendation requests

```
nutrition_input: conlist(float, min_items=9, max_items=9) ingredients: list[str] = [] params: Optional[params]
```

Recipe

Output model representing a single recipe recommendation

```
Name: str CookTime: str, PrepTime: str, TotalTime: str RecipeIngredientParts: list[str] Nutritional values: float (9 fields) RecipeInstructions: list[str]
```

PredictionOut

Response model containing list of recommended recipes

```
output: Optional[List[Recipe]] = None
```

params

Configuration parameters for the recommendation algorithm

```
n_neighbors: int = 5 return_distance: bool = False
```

API Features

Type Safety

Pydantic models ensure data validation and type checking

High Performance

FastAPI provides async support and automatic documentation

Configurable

Adjustable parameters for recommendation customization

Implementation Details

Technology Stack

Backend Technologies

FastAPI
High-performance Python web framework for building APIs

Spring Boot
Java-based framework for microservice development

Frontend Technologies

Thymeleaf
Modern server-side Java template engine

Responsive Design
Mobile-first approach for cross-platform compatibility

Database Systems

MySQL
Relational database for structured data storage

PostgreSQL
Advanced object-relational database system

API Endpoints

POST `/api/auth/login`
User authentication and session management

POST `/api/diet/generate`
AI-powered diet plan generation based on user metrics

GET `/api/exercise/plans`
Retrieve exercise plans assigned by coaches

PUT `/api/progress/update`
Update trainee progress and workout metrics

GET `/api/community/posts`
Fetch community posts and user interactions

Deployment Strategy

Security
HTTPS, Authentication, Data Encryption

Scalability
Microservices, Load Balancing

Performance
Caching, Database Optimization

Accessibility
Cross-platform, Responsive Design

Conclusion & Future Work

Project Achievements

Comprehensive Platform Development

Successfully developed a robust, scalable, and user-friendly web application addressing core health and fitness needs

AI Integration

Implemented intelligent diet recommendation system using Nearest Neighbors algorithm with cosine similarity for personalized nutrition plans

Dynamic Coach-Trainee Model

Created an innovative interaction system connecting professional coaches with trainees for personalized exercise planning

Community Platform

Established a vibrant community environment for user engagement, motivation, and peer support

Cross-Platform Accessibility

Delivered responsive design ensuring seamless experience across desktop and mobile devices

Future Enhancements

Advanced Analytics

Implement comprehensive data analytics for deeper insights into user progress, trends, and optimization opportunities

Real-Time Communication

Develop live chat functionalities for instant communication between coaches and trainees

Wearable Device Integration

Connect with fitness trackers and smartwatches for automated data collection and real-time monitoring

Enhanced AI Models

Integrate more sophisticated machine learning algorithms for improved recommendation accuracy and personalization

Mobile Application

Develop native mobile applications for iOS and Android platforms with offline capabilities

Project Impact

This project successfully addresses the critical shortcomings of traditional fitness and nutrition approaches by delivering an **innovative platform** that integrates **AI-driven personalization**, **professional coaching**, and **community support**. The result is a comprehensive solution that empowers users to achieve sustainable health and fitness outcomes.

Thank You for Your Attention