# Workout Planner App - System Specifications

## 1. Introduction

The Workout Planner App is a web-based application designed to generate personalized workout and diet plans for users based on their fitness goals. The system provides a user-friendly platform for beginners, fitness enthusiasts, and busy professionals to simplify fitness planning.

## 2. System Objectives

- Provide tailored plans based on user inputs.  
- Maintain a simple and intuitive interface.  
- Ensure the app is mobile-friendly and works on all modern browsers.  
- Deliver results quickly, even with a high number of users.  
- Protect sensitive user data with robust security measures.

## 3. Functional Specifications

### 3.1. User Management

- Users can register, log in, log out, and reset passwords.  
- Secure authentication for all user accounts.

### 3.2. Profile Management

- Users can input and update personal details: age, gender, weight, height, activity level, and fitness goals.

### 3.3. Workout Plan Generation

- The app generates workout plans tailored to fitness level, user goals, and available days per week.

### 3.4. Diet Recommendations

- Calorie calculation based on BMR and user goals (maintenance, surplus, deficit).  
- Macronutrient breakdown (protein, fats, carbohydrates).

### 3.5. Admin Features

- Manage user accounts.  
- Monitor system logs for performance and security.

## 4. Non-Functional Specifications

### 4.1. Performance

- Generate workout and diet plans within 10 seconds.

### 4.2. Scalability

- Handle up to 1000 concurrent users effectively.

### 4.3. Security

- Encrypt user passwords using PBKDF2 or bcrypt.  
- Prevent unauthorized data access with secure authentication mechanisms.

### 4.4. Compatibility

- Ensure responsiveness for mobile devices (smartphones, tablets).  
- Support major modern browsers: Chrome, Firefox, Safari, Edge.

### 4.5. Usability

- Maintain an intuitive design with clear instructions.

### 4.6. Maintainability

- Use modular and well-documented code.

## 5. System Architecture

The system consists of client-side components (frontend), server-side components (backend), and a database for storing user data and plans.

### 5.1. Client-Side

- Frontend Framework: HTML, CSS, JavaScript.  
- Responsive Design: Adapt layout for mobile and desktop views.

### 5.2. Server-Side

- Backend Framework: Django (Python).  
- Authentication: Django's built-in authentication system.  
- APIs: RESTful endpoints for managing user data, workout plans, and diet recommendations.

### 5.3. Database

- Database Type: PostgreSQL or MySQL (preferred for scalability).  
- Tables:  
 - Users: Stores user information (e.g., email, hashed password).  
 - Profiles: Stores user inputs (e.g., height, weight, goals).  
 - Plans: Stores generated workout and diet plans.

## 6. Use Case Examples

### 6.1. User Registration

Actors: User, System  
Steps:  
1. User enters email and password.  
2. System validates input and creates a new account.  
3. User receives a confirmation email.

### 6.2. Generate Workout Plan

Actors: User, System  
Steps:  
1. User selects a goal (e.g., weight loss).  
2. User inputs details (e.g., weight, height, activity level).  
3. System generates a workout plan tailored to the user.

### 6.3. Generate Diet Recommendation

Actors: User, System  
Steps:  
1. User selects a goal (e.g., calorie deficit).  
2. System calculates calorie needs and macro distribution.  
3. User receives a detailed diet recommendation.