Project Name: Personalized Health and Fitness Management System

Problem Statement

Traditional fitness applications often provide generic recommendations that fail to consider individual health metrics, goals, and preferences. This one-size-fits-all approach leads to less effective fitness and dietary outcomes, making it difficult for users to achieve their health objectives. Moreover, existing platforms may lack motivational features, making it harder for users to stay consistent with their fitness routines.

Objective

Develop a web-based health and fitness tracker that provides personalized workout and dietary recommendations based on user inputs such as BMI, fitness goals, and activity logs. The application will engage users with progress tracking, motivational badges, and interactive features to encourage consistency.

Functional and Non-functional Requirements

Functional Requirements:

User Features:

1. User Registration and Profile Setup:
    - Users can create accounts and set up health profiles with details like age, weight, height, fitness goals, and dietary preferences.
2. Personalized Recommendations:
    - Generate custom workout and meal plans based on the user's health profile.
3. Activity Logging:
    - Allow users to log daily workouts and meals.
4. Progress Tracking:
    - Display progress in visual formats such as graphs and charts.
5. Motivational Features:
    - Award badges for milestones such as completing workouts for a week or reaching a weight goal.

Admin Features:

1. User Management:
    - View, update, or deactivate user accounts.
    - Monitor user progress and activity logs.
2. Content Management:
    - Add, edit, or delete workout and dietary recommendations.
    - Update nutritional API configurations.
3. Achievement Management:
    - Customize motivational badges and milestone criteria.
4. Analytics and Reports:
    - Generate reports on user activity and overall platform performance.
    - Monitor system usage trends.
5. System Monitoring:
    - Address security issues and ensure smooth performance.

Non-functional Requirements:

1. **Scalability:**
    - Support multiple users simultaneously without performance degradation.
2. **Performance:**
    - Generate recommendations in under 2 seconds.
3. **Security:**
    - Use secure authentication methods to protect user data.
4. **Usability:**
    - Ensure an intuitive and user-friendly interface.
5. **Availability:**
    - The system should have 99% uptime for a smooth user experience.

Use Case Diagram:

Actors:

- **Users**
- **Admins**

Use Cases:

- **User:**
  - Register/Login
  - Set up Health Profile
  - Log Workouts and Meals
  - View Progress
  - Get Recommendations
  - Earn Badges
  - Update Profile
- **Admin:**
  - Manage Users
  - Manage Content
  - Manage Achievements
  - View Reports
  - Monitor System

Database Diagram

The database schema includes the following tables:

1. **Users:**
   - `UserID` (Primary Key)
   - `Name`
   - `Email`
   - `Password`
   - `DateOfBirth`
2. **HealthProfiles:**
   - `ProfileID` (Primary Key)
   - `UserID` (Foreign Key)
   - `Height`
   - `Weight`
   - `BMI`
   - `FitnessGoal` (e.g., Weight Loss, Muscle Gain)
   - `DietPreference`
3. **Workouts:**
   - `WorkoutID` (Primary Key)
   - `UserID` (Foreign Key)
   - `Date`
   - `WorkoutDetails`
   - `CaloriesBurned`

4. **Meals:**
    - `MealID` (Primary Key)
    - `UserID` (Foreign Key)
    - `Date`
    - `MealDetails`
    - `CaloriesConsumed`
5. **Achievements:**
    - `AchievementID` (Primary Key)
    - `UserID` (Foreign Key)
    - `AchievementName`
    - `DateEarned`
6. **Progress:**
    - `ProgressID` (Primary Key)
    - `UserID` (Foreign Key)
    - `Date`
    - `Weight`
    - `BMI`
7. **`Admins Table:`**
    - `AdminID (Primary Key)`
    - `Name`
    - `Email`
    - `Password`
    - `Role`
8. **`AuditLogs Table:`**
    - `LogID (Primary Key)`
    - `AdminID (Foreign Key)`
    - `ActionType (e.g., User Suspension, Data Modification)`
    - `ActionDetails`
    - `Timestamp`

Project Timeline

Day 1-2: Requirement analysis and application design.

Day 3-6: Development of core functionalities (user registration, health profile, and activity logging).

Day 7: Integration of APIs and motivational features.

Day 8: Functional and non-functional testing.

Day 9-10: Deployment and final adjustments.