Project Title: Smart Fitness Routine & Meal Planner

Objective

Develop a full-stack web application that provides personalized fitness routines and meal plans based on user goals using Angular, Node.js (TypeScript), and MySQL.

Functional Requirements

1. User Roles

- **User**: Can register, set fitness goals, view workouts and meal plans, and track progress.

- **Trainer/Admin** (Optional): Can manage workout templates, meal plans, and provide analytics.

---

2. Goal & Profile Management

- Users can create a profile with:

    - Name, age, gender, height, weight

    - Fitness goal (weight loss, muscle gain, maintenance)

- Users can update profile and goals anytime.

UI Suggestions:

- Profile Page: Input and update personal info and goals.

- Goal Selection Page: Choose or update fitness goal.

---

3. Fitness Routine Management

- Users get **auto-generated weekly workout routines** based on goals.

- Users can:

    - View exercises per day

    - Mark exercises as completed

    - Track progress over time

UI Suggestions:

- Workout Routine Page: Shows daily exercises with instructions.

- Progress Page: Visual progress tracking (charts/graphs).

---

4. Meal Planner Management

- Users get **meal plans with calorie tracking** based on goals.

- Users can:

    - View daily meals (breakfast, lunch, dinner, snacks)

    - Mark meals as consumed

    - Adjust meals if needed

UI Suggestions:

- Meal Planner Page: Shows meals for each day.

- Nutrition Tracker Page: Tracks calories consumed vs target.

---

5. Frontend Requirements (Angular 16)

- Use Angular Material for UI components.

- Structured folder/component hierarchy.

Suggested Components:

1. ProfileComponent → Manage user profile & goals.

2. WorkoutRoutineComponent → Display weekly workouts.

3. MealPlannerComponent → Display meal plan & calories.

4. ProgressTrackerComponent → Show graphs of fitness progress.

Suggested Routes:

Path	Component	Description
/profile	ProfileComponent	View and update profile & goals
/workouts	WorkoutRoutineComponent	View weekly workout routine
/meals	MealPlannerComponent	View daily meal plan & calorie tracker
/progress	ProgressTrackerComponent	Track exercise & diet progress

6. Backend Requirements (Node.js + TypeScript + Express)

- Use MySQL as the database.

- Implement REST APIs for user profiles, workouts, and meal plans.

- Include validation (required fields, goal-based logic) and error handling.

7. Database Structure (MySQL)

Users Table

- id

- name

- age

- gender

- height

- weight

- goal (weight loss / muscle gain / maintenance)

- created_at

WorkoutMealPlans Table

- id

- user_id (relates to Users table)

- day (Monday-Sunday)

- exercises (JSON, list of exercises)

- meals (JSON, meal plan with calories)

- completed_status (JSON, tracks completed exercises/meals)

Maximum 2 tables; simple relation between Users and WorkoutMealPlans.

8. Validation Outline

- Profile fields (name, age, height, weight) must not be empty.

- Goal must be one of the allowed options (weight loss, muscle gain, maintenance).

- Workout and meal plan entries cannot be empty.

● Include all necessary validation to ensure proper user workflow and data integrity.

---

9. Exception Handling

● Handle exceptions and errors gracefully in frontend and backend.

● Display meaningful error messages in the UI.

● Log errors on the server side for debugging and maintenance.

---

10. Role-Based Access (Using Angular Guards)

● Use Angular route guards to restrict access to components based on user roles:

  ○ **User** → Can access all user-related components (ProfileComponent, WorkoutRoutineComponent, MealPlannerComponent, ProgressTrackerComponent).

  ○ **Trainer/Admin** → Can access management components (if implemented).

● Unauthorized users attempting to access restricted routes should be redirected to a login page or "Not Authorized" page.

● Guards ensure secure, role-appropriate display of components while maintaining proper workflow.