## 25. Fitness Tracker

Creating a Fitness Tracker app using the MEAN stack (MongoDB, Express.js, Angular, Node.js) is a great way to help users monitor their fitness progress. Below is a high-level overview of how to build such an app, along with some code snippets to guide you:

### Project Setup and Structure

Set up a new project folder and structure for your Fitness Tracker app. Install the required Node.js packages and create a basic Angular application.

```
# Create a new Angular application
ng new fitness-tracker-app
```

## - Backend (Node.js & Express.js)

Create the backend of your Fitness Tracker app using Node.js and Express.js.

### Installation of Packages

Install the necessary packages for Express.js, Mongoose (for MongoDB), and other dependencies.

```
npm install express mongoose cors
```

**Setting up Express.js**

Create your Express.js server, set up middleware, and handle routes.

- **javascript**

```
// server.js

const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');

const app = express();

// Middleware
app.use(express.json());
app.use(cors());

// Database connection
mongoose.connect('mongodb://localhost/fitness-tracker-app', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  useCreateIndex: true,
});

// Define Mongoose models for User, Workout, and Goal data
const User = mongoose.model('User', {
  username: String,
  password: String, // Use hashing for security
  // Add more user-related fields as needed
});

const Workout = mongoose.model('Workout', {
  userId: mongoose.Schema.Types.ObjectId,
  date: Date,
  exercise: String,
  sets: Number,
  reps: Number,
  weight: Number,
  // Add more workout-related fields as needed
});

const Goal = mongoose.model('Goal', {
  userId: mongoose.Schema.Types.ObjectId,
  target: String,
  targetValue: Number,
  // Add more goal-related fields as needed
});

// Routes for managing users, workouts, and goals
app.post('/api/register', async (req, res) => {
  // Register a new user
  // Store hashed password in the database
});

app.post('/api/login', async (req, res) => {
  // Authenticate user and generate a JWT token
});

app.post('/api/workouts', async (req, res) => {
  // Create a new workout entry
  // Save the workout to the database
});

app.get('/api/workouts', async (req, res) => {
  // Retrieve a list of workouts
});
```

```
// Create similar routes for managing goals, user profiles, and progress tracking
```

- **Frontend (Angular)**

Create the frontend of your Fitness Tracker app using Angular. Design the user interface for tracking workouts, setting goals, and user accounts.

### Design and UI

Design the user interface for your Fitness Tracker app using Angular components, templates, and styles.

### Workout Tracking

Create components and forms for users to record their workouts, including exercise, sets, reps, and weights.

### Goal Setting

Design components for users to set fitness goals, specifying the target and target value.

### User Authentication

Implement user registration and login functionality.

- **typescript**

```
// workout-tracking.component.ts
import { Component } from '@angular/core';
import { WorkoutService } from './workout.service';

@Component({
  selector: 'app-workout-tracking',
  templateUrl: './workout-tracking.component.html',
})
export class WorkoutTrackingComponent {
  exercise: string;
  sets: number;
  reps: number;
  weight: number;

  constructor(private workoutService: WorkoutService) {}

  addWorkout() {
    this.workoutService.addWorkout(this.exercise, this.sets, this.reps, this.weight);
  }
}
```

## MongoDB

Create a MongoDB database to store user profiles, workout data, goal information, and user progress.

## Putting It All Together

Integrate the frontend and backend by making API requests from Angular components to Node.js routes. Ensure that you handle workout tracking, goal setting, user authentication, and progress tracking properly.

Building a Fitness Tracker app is a valuable project for fitness enthusiasts. You can expand it with features like workout statistics, workout history, exercise libraries, workout templates, and performance analytics for a more comprehensive fitness tracking experience. It's a versatile app that can be customized to your specific fitness goals.