1. \*\*Set up your Angular project:\*\*

- Use Angular CLI to create a new project:

```bash

ng new fitness-tracker

cd fitness-tracker

```

2. \*\*Create Components:\*\*

- Create components for the main fitness tracker dashboard, workout log, and any other necessary components:

```bash

ng generate component fitness-dashboard

ng generate component workout-log

```

3. \*\*Create a Service:\*\*

- Generate a service to handle workout-related functionalities. Use Angular CLI:

```bash

ng generate service workout

```

4. \*\*Update the Workout Service:\*\*

- Update the `workout.service.ts` file to manage workout-related operations. Here's a simplified example:

```typescript

// workout.service.ts

import { Injectable } from '@angular/core';

import { BehaviorSubject } from 'rxjs';

@Injectable({

providedIn: 'root',

})

export class WorkoutService {

private workoutsSubject = new BehaviorSubject<Workout[]>([]);

workouts$ = this.workoutsSubject.asObservable();

addWorkout(workout: Workout): void {

const currentWorkouts = this.workoutsSubject.value;

const newWorkouts = [...currentWorkouts, workout];

this.workoutsSubject.next(newWorkouts);

}

getWorkouts(): Workout[] {

return this.workoutsSubject.value;

}

}

export interface Workout {

date: Date;

type: string;

duration: number; // in minutes

calories: number;

}

```

5. \*\*Update the Workout Log Component:\*\*

- Update the `workout-log.component.ts` file to use the Workout Service and display the workout log:

```typescript

// workout-log.component.ts

import { Component, OnInit } from '@angular/core';

import { WorkoutService, Workout } from '../services/workout.service';

@Component({

selector: 'app-workout-log',

templateUrl: './workout-log.component.html',

styleUrls: ['./workout-log.component.css'],

})

export class WorkoutLogComponent implements OnInit {

workouts: Workout[] = [];

constructor(private workoutService: WorkoutService) {}

ngOnInit() {

this.workouts = this.workoutService.getWorkouts();

this.workoutService.workouts$.subscribe((workouts) => {

this.workouts = workouts;

});

}

}

```

6. \*\*Update the Workout Log Component HTML:\*\*

- Update the `workout-log.component.html` file to display the workout log:

```html

<!-- workout-log.component.html -->

<div>

<h2>Workout Log</h2>

<ul>

<li \*ngFor="let workout of workouts">

<p>Date: {{ workout.date | date }}</p>

<p>Type: {{ workout.type }}</p>

<p>Duration: {{ workout.duration }} minutes</p>

<p>Calories: {{ workout.calories }}</p>

</li>

</ul>

</div>

```

7. \*\*Use the Components in the Fitness Dashboard:\*\*

- Update the `fitness-dashboard.component.html` file to use the Workout Log component and add functionality for adding workouts:

```html

<!-- fitness-dashboard.component.html -->

<div>

<app-workout-log></app-workout-log>

<div>

<label for="workoutType">Enter Workout Type:</label>

<input type="text" id="workoutType" [(ngModel)]="newWorkoutType" />

<label for="workoutDuration">Enter Workout Duration (minutes):</label>

<input type="number" id="workoutDuration" [(ngModel)]="newWorkoutDuration" />

<label for="workoutCalories">Enter Calories Burned:</label>

<input type="number" id="workoutCalories" [(ngModel)]="newWorkoutCalories" />

<button (click)="addWorkout()">Add Workout</button>

</div>

</div>

```

- Update the `fitness-dashboard.component.ts` file:

```typescript

// fitness-dashboard.component.ts

import { Component } from '@angular/core';

import { WorkoutService, Workout } from '../services/workout.service';

@Component({

selector: 'app-fitness-dashboard',

templateUrl: './fitness-dashboard.component.html',

styleUrls: ['./fitness-dashboard.component.css'],

})

export class FitnessDashboardComponent {

newWorkoutType: string;

newWorkoutDuration: number;

newWorkoutCalories: number;

constructor(private workoutService: WorkoutService) {}

addWorkout(): void {

if (this.newWorkoutType && this.newWorkoutDuration && this.newWorkoutCalories) {

const newWorkout: Workout = {

date: new Date(),

type: this.newWorkoutType,

duration: this.newWorkoutDuration,

calories: this.newWorkoutCalories,

};

this.workoutService.addWorkout(newWorkout);

this.newWorkoutType = '';

this.newWorkoutDuration = null;

this.newWorkoutCalories = null;

}

}

}

```

Make sure to add `FormsModule` to your `app.module.ts` for the `ngModel` to work. Add the following import statements and update the imports array:

```typescript

// app.module.ts

import { FormsModule } from '@angular/forms';

@NgModule({

// ...

imports: [

// ...

FormsModule,

],

// ...

})

```