Project Report: Fitness Tracker Dashboard

# 1. Overview

The Fitness Tracker Dashboard is a browser-based application that allows users to log their workouts, track calories burned, and monitor progress against weekly fitness goals. The app is fully client-side, storing all data in the browser’s LocalStorage, which makes it lightweight, private, and portable (no backend needed).

# 2. Technologies Used

* HTML5 – Structure of the web application.
* CSS3 / Bootstrap 5 – Styling and responsive design.
* JavaScript (ES6) – Application logic, DOM manipulation, and data handling.
* Chart.js – Visualization of workout progress over time.
* LocalStorage API – Persistent storage for workout logs and user goals.

# 3. Key Features

## a) Workout Logging

Users can input:  
- Date  
- Workout type (e.g., Run, Yoga, Gym)  
- Duration (minutes)  
- Calories burned  
All data is saved to LocalStorage for persistence.

## b) Workout Log Table

Displays workouts in a table format, sorted by date (most recent first). If no workouts are logged, a message is displayed instead of an empty table.

## c) Weekly Goal Tracking

Users can set a weekly calorie-burn target. The app displays the saved goal, allows updates, and summarizes progress for the current week.

## d) Progress Visualization

A line chart powered by Chart.js shows the calories burned over the last 14 days. This helps users easily visualize workout patterns and progress.

## e) Data Management

The application provides a 'Clear All' button to reset the entire workout log. A confirmation dialog prevents accidental deletion.

# 4. System Design

The application follows a modular structure with clear separation of concerns:  
- HTML: Defines the layout and semantic structure of the dashboard.  
- CSS: Provides styling, theme consistency, and responsiveness.  
- JavaScript: Handles application logic, rendering, chart updates, and event handling.  
- LocalStorage: Ensures persistence of user data without requiring a server.

# 5. Advantages

* Lightweight and fully client-side (no backend required).
* Data privacy – workouts are stored only in the user’s browser.
* Responsive design works across desktop and mobile.
* Easy to extend with additional features (e.g., categories, export options).

# 6. Limitations

* Data is stored only in the browser – clearing cache or switching devices will lose data.
* No authentication or cloud sync available.
* Limited analytics (only calories tracking, no advanced statistics).

# 7. Future Enhancements

* Add authentication and cloud sync to access workouts across devices.
* Include more workout parameters (distance, heart rate, intensity levels).
* Export data to CSV or Excel for offline use.
* Add mobile app version using frameworks like React Native or Flutter.