Front-End UI/UX Mini Project

---

1. Title Page

Project Title: Fitness Tracker

Submitted By:

- **Team Members:**
    1. M SANJAY 02461015   m.sanjay@btech.christuniversity.in
    2. ATTI CHANDRAKANTH 2461005 atti.chandrakanth@btech.christuniversity.in
    3. SOORAJ S 2461028 sooraj.s@btech.christuniversity.in

Course: LnT Front End UI/UX

Instructor Name: Mr.Dhiraj Atale

Institution: Christ University

Date of Submission: 26/09/2025

---

2. Abstract

This report details the development of a Fitness Tracker Dashboard, a web application designed to help users monitor their fitness journey. The project provides a user-friendly interface for logging daily workouts, visualizing progress through interactive charts, and setting personal fitness goals. Built using HTML, CSS, JavaScript, and Bootstrap, the dashboard features a responsive layout to ensure a seamless experience across both desktop and mobile devices. Key functionalities include a Workout Log for detailed data entry and a Progress Chart for a visual representation of user activity over time.

---

3. Objectives

The primary objective of this project is to develop a functional and user-friendly Fitness Tracker Dashboard to assist users in managing their personal fitness goals. This will be achieved through the following specific objectives:

- Design and Implement a User Interface: Create a clean and intuitive user interface using HTML and CSS that provides easy navigation and a positive user experience.
- Develop a Workout Logging System: Build a feature using JavaScript and jQuery that allows users to record and store details of their daily workouts, including type, duration, and calories burned.
- Visualize User Progress: Integrate a charting library, such as Chart.js, to generate dynamic and interactive charts (e.g., bar and line charts) that visually represent a user's workout data over time.
- Enable Goal Setting and Tracking: Implement a functionality that allows users to set specific weekly or monthly fitness goals and track their progress against these goals.
- Ensure Responsive Design: Utilize Bootstrap to ensure the dashboard's layout is fully responsive and provides a seamless viewing and interaction experience on various devices, including desktops and mobile phones.
- Integrate Technologies: Successfully combine and utilize the specified technologies—HTML, CSS, JavaScript, Bootstrap, jQuery, and a charting library—to create a cohesive and robust web application.

---

4. Scope of the Project

- Front-End Development: The project will be a front-end application utilizing HTML, CSS, JavaScript, and jQuery for its functionality.
- Responsive Design: The layout will be fully optimized for a consistent user experience across desktops, tablets, and mobile devices using the Bootstrap framework.
- Interactive Features: The application will include interactive elements for logging data, displaying charts, and setting goals, all powered by JavaScript and jQuery.
- External Libraries: The project will incorporate external libraries, specifically Bootstrap for responsive design and jQuery for simplified scripting.
- Client-Side Functionality: All features and data management will be handled on the client-side; no server-side or backend development is included in this scope.

---

5. Tools & Technologies Used

Tool/Technology	**Purpose**
HTML5	Markup and structure
CSS3	Styling and layout management
Javascript	Client-side scripting and interactive functionality.

Jquery	A fast, small, and feature-rich JavaScript library for simplified scripting.
Bootstrap	A CSS framework for responsive design and UI components.
VS Code	Code editor
Chrome DevTools	Testing and debugging

---

6. HTML Structure Overview

- Semantic HTML: The project will use modern semantic HTML tags, including <header>, <section>, and <footer>, to structure the page content logically and improve accessibility.

---

7. CSS Styling Strategy

- Custom Properties: The :root selector is used to define global CSS variables for colors, such as --primary-blue and --text-muted, which promotes a consistent and easily maintainable color scheme.

- Goal Badges: Different visual styles are applied to goal badges based on their status using class selectors like .goal-badge.weekly, .goal-badge.monthly, and .goal-badge.completed. This uses a combination of background color, text color, and gradients to indicate the goal type and completion status.

- Empty State: A dedicated styling block for the .empty-state class provides a consistent look for sections with no data. It uses centered text, a large icon, and muted colors to clearly communicate that there is nothing to display yet.

- Form Focus States: The .form-control:focus and .form-select:focus selectors are used to customize the appearance of form inputs and select menus when they are active.

They apply a custom blue border and a subtle box shadow to provide clear visual feedback to the user.

- Responsive Adjustments: The @media (max-width: 768px) media query is used to adjust the layout and font sizes for smaller screens (tablets and mobile devices). This includes reducing the font size of the app title and stats values to ensure they fit well on the screen.

- Chart and Utility Styles: A specific style is defined for the chart container (#progressChart) to set its maximum height, ensuring it doesn't take up too much vertical space. Additionally, utility classes like .text-primary-custom are created to easily apply specific colors defined by the CSS variables.

---

8. Key Features

Feature	Description
Responsive Updates	The entire dashboard refreshes instantly after an upload without page reload
Responsive Layout	Works on desktop, tablet, and mobile.
Workout Logging	Users can log new workouts with details like type, duration, calories, and date, which are stored locally for instant dashboard updates.
Dynamic Stats	The dashboard calculates and displays real-time weekly statistics for total workouts, calories, minutes, and average calories per workout.
Progress Visualization	An interactive bar chart dynamically visualizes weekly calories and minutes, offering a clear view of fitness trends.

---

9. Challenges Faced & Solutions

Challenge	Solution
Creating a clean, readable, and accessible HTML structure.	We used semantic HTML5 tags (<header>, <main>, <section>) to organize the content, making it meaningful for both developers and assistive technologies like screen readers.
Ensuring a consistent and responsive design across all devices.	The Bootstrap framework was used to build a flexible grid system. We also applied custom CSS and media queries to fine-tune the layout and styling, ensuring the dashboard looks and works well on desktops, tablets, and mobile phones.
Managing workout data and updating the UI without a backend server.	All data is stored in a client-side JavaScript array. We created functions to handle data manipulation and a core rendering function that refreshes the entire dashboard. This ensures the UI is always synchronized with the data, providing a smooth, real-time user experience.

10. Outcome

The final outcome of this project is a fully functional, client-side fitness tracker dashboard. It successfully delivers a user-friendly interface that allows users to log workouts, visualize their progress through dynamic charts, and track their fitness goals. The application is built with HTML, CSS, JavaScript, and Bootstrap, making it fully responsive and accessible on a wide range of devices.

11. Future Enhancements

- User Accounts and Authentication

- Data Visualization Improvements
- Personalized Recommendations
- Social Sharing and Community Features

---

12. Sample Code

```
1.<section class="mb-5">
   <div class="row g-4" id="statsCards">
      <div class="col-md-6 col-xl-3">
         <div class="stats-card">
            <div class="d-flex justify-content-between align-items-center">
               <div class="flex-grow-1">
                  <div class="stats-label">Workouts This Week</div>
                  <div class="stats-value">7</div>
               </div>
               <div class="stats-icon primary">
                  <i class="bi bi-activity"></i>
               </div>
            </div>
         </div>
      </div>
   </div>
</section>

<div class="modal fade" id="workoutModal" tabindex="-1">
   <div class="modal-dialog">
      <div class="modal-content">
         <form id="workoutForm">
            <div class="modal-header">
               <h5 class="modal-title">Add New Workout</h5>
            </div>
            <div class="modal-body">
               <div class="mb-3">
                  <label for="workoutType" class="form-label">Workout Type</label>
                  <input type="text" class="form-control" id="workoutType" required>
               </div>
               <div class="mb-3">
                  <label for="duration" class="form-label">Duration (minutes)</label>
                  <input type="number" class="form-control" id="duration" min="1" required>
               </div>
            </div>
            <div class="modal-footer">
```

```
                <button type="submit" class="btn btn-primary">Add Workout</button>
            </div>
        </form>
    </div>
</div>
</div>
```

2./* Custom properties for color and styling consistency */

```
:root {
    --primary-blue: #3b82f6;
    --success-green: #059669;
    --text-muted: #6b7280;
}

/* Styles for a card displaying statistics */
.stats-card {
    background-color: #fff;
    border-radius: 0.5rem;
    padding: 1.5rem;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

/* Styles for goal badges to indicate their status or type */
.goal-badge.weekly {
    background-color: var(--primary-blue);
    color: white;
}

.goal-badge.completed {
    background: linear-gradient(45deg, var(--success-green), #22c55e);
    color: white;
}

/* Styles for the empty state message when a section has no content */
.empty-state {
    text-align: center;
    padding: 3rem 1rem;
    color: var(--text-muted);
}

/* Responsive adjustments for smaller screens */
@media (max-width: 768px) {
    .app-title {
        font-size: 1.5rem;
    }
}
```

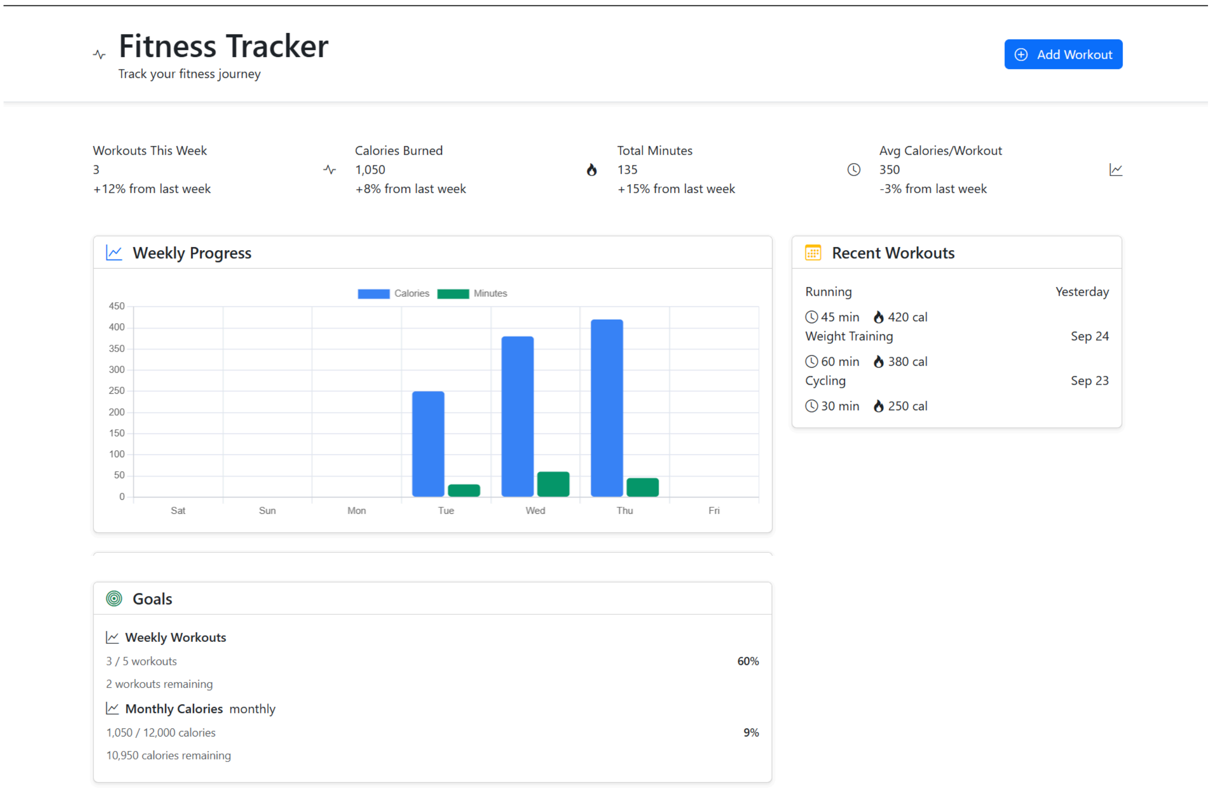
3..album {

```
    background: #1a1a1a;
    border-radius: 10px;
    width: 300px;
    box-shadow: 0 2px 12px rgba(0,0,0,0.7);
    overflow: hidden;
    padding-bottom: 15px;
}
.album img {
    width: 100%;
    height: 300px;
    object-fit: cover;
    display: block;
}
.album-title {
    padding: 12px;
    font-weight: bold;
    color: #fdd835;
    font-size: 1.3em;
    text-align: center;
    border-bottom: 2px solid #e52d27;
}
```

```
3.$(document).ready(function() {
    let workouts = []; // Array to store workout objects

    // Event handler for the workout form submission
    $('#workoutForm').on('submit', function(e) {
        e.preventDefault(); // Prevents the default form submission behavior

        // Create a new workout object from the form's input values
        const newWorkout = {
            type: $('#workoutType').val(),
            duration: parseInt($('#duration').val()),
            calories: parseInt($('#calories').val()),
            date: new Date().toISOString().split('T')[0]
        };

        // Add the new workout to the beginning of the array
        workouts.unshift(newWorkout);

        // Update all parts of the UI
        renderStatsCards();
        renderWorkoutsList();

        // Reset the form and close the modal
        $('#workoutForm')[0].reset();
        $('#workoutModal').modal('hide');
    });
    // Function to render the recent workouts list
    function renderWorkoutsList() {
        const workoutsHtml = workouts.slice(0, 5).map(workout => {
            return `
                <div class="workout-item">
                    <span>${workout.type}</span>
                    <span>${workout.duration} min</span>
                </div>
            `;
        }).join('');

        $('#workoutsList').html(workoutsHtml);
    }

    // Initial call to render the UI on page load
    renderWorkoutsList();
});
```

---

13. Screenshots of Final Output



Fitness Tracker
Track your fitness journey

Add Workout

Workouts This Week
3
+12% from last week

Calories Burned
1,050
+8% from last week

Total Minutes
135
+15% from last week

Avg Calories/Workout
350
-3% from last week

Weekly Progress

Recent Workouts

Running	Yesterday
45 min 420 cal	
Weight Training	Sep 24
60 min 380 cal	
Cycling	Sep 23
30 min 250 cal	

Goals

Weekly Workouts
3 / 5 workouts 60%
2 workouts remaining

Monthly Calories monthly
1,050 / 12,000 calories 9%
10,950 calories remaining

14. Conclusion

The Fitness Tracker Dashboard project successfully delivered a functional, client-side web application for personal fitness management. By leveraging HTML, CSS, JavaScript, and Bootstrap, the project provides a responsive interface for logging workouts, visualizing progress with charts, and tracking goals. This project showcases a strong understanding of front-end development principles and serves as a solid foundation for future enhancements, such as user authentication and data persistence.

15. References

- W3Schools HTML & CSS & Javascript & Jquery & Bootstrap
- L&T LMS : https://learn.lntedutech.com/Landing/MyCourse