**Front-End UI/UX Project**

**1. Title Page**

* **Project Title**: Fitness Tracker
* **Submitted By**: 
  + *Team Members-*

Blesson Baby (2460346)

Adi Sankar (2460308)

Chris Paul (2460350)

* + *College-E-mailid-*

[*blesson.baby@btech.christuniversity.in*](mailto:blesson.baby@btech.christuniversity.in)

[*adi.sankar@btech.christuniversity.in*](mailto:adi.sankar@btech.christuniversity.in)

[*chris.paul@btech.christuniversity.in*](mailto:chris.paul@btech.christuniversity.in)

* **Course**: *UI/UX Design Fundamentals*
* **Instructor Name**: *Dhiraj Alate.*
* **Institution**: *Christ University*
* **Date of Submission**: *26/09/2025*

**2. Abstract**

This project involved the development of a single-page, responsive, client-side **Fitness Tracker Dashboard** named "Fitter." The application allows users to log daily workouts and set personalized weekly and monthly fitness goals. The design focuses on a clean, dark-mode user interface, utilizing the Bootstrap framework for responsiveness and **Chart.js** to provide real-time data visualization of workout history and goal progress. All user data (workouts and goals) is persisted locally in the browser's storage. The project successfully delivered a functional, data-driven personal fitness management tool.

**3. Objectives**

* To design and implement a user interface that allows simple and efficient logging of fitness activities.
* To create a dedicated section for setting, tracking, and displaying weekly and monthly calorie and duration goals.
* To provide an intuitive dashboard using Chart.js for visual representation of user activity and goal attainment.
* To ensure the entire application is fully responsive across desktop, tablet, and mobile devices using Bootstrap 5.
* To utilize Local Storage for client-side data persistence, ensuring data is saved between sessions.

**4. Scope of the Project**

The project focuses on developing a fully functional front-end, single-page web application. The scope includes:

1. Data Input: Forms for logging workouts (date, type, duration, calories) and setting goals.
2. Dashboard View: Displaying total workouts, total minutes, and total calories burned.
3. Visualization: Bar charts showing progress against weekly and monthly goals.
4. Workout History: A table listing all recorded workouts with a delete function.
5. Data Management: Saving and retrieving user data using browser's Local Storage.

**5. Tools & Technologies Used**

|  |  |
| --- | --- |
| Tool/Technology | Purpose |
| HTML5 | Core structure and semantic markup. |
| CSS3 | Custom styling and dark-mode theme. |
| Bootstrap 5.3 | Responsive layout, grid system, and components. |
| Chart.js 4.4.3 | Dynamic and data-driven chart generation. |
| jQuery 3.7.1 | Efficient DOM manipulation and form handling. |
| Bootstrap Icons | Clear and concise iconography. |
| Bootstrap 5.3 | Responsive layout, grid system, and components |

**6. HTML Structure Overview**

The application is structured around a **single HTML file** (fitter.html) using semantic tags and Bootstrap's container classes.

* **Header/Navbar:** A simple fixed top navigation bar that contains the project title and brand.
* **Main Container:** A responsive container-fluid wraps the entire dashboard content.
* **Goal/Chart Section:** This includes the **Goal Setting Form** (within a Bootstrap Card) and the two **Chart Canvases** (one for Weekly, one for Monthly progress) arranged using the Bootstrap Grid.
* **Logging Section:** This holds the **Workout Submission Form** and the **Dashboard Metrics** (total calories, minutes, etc.).
* **Workout History:** A dedicated area for the workoutsTable that dynamically lists all stored workout entries.

**7. CSS Styling Strategy**

The primary styling approach leverages a **dark-mode, dashboard aesthetic** through CSS variables and utility classes.

* **CSS Variables (:root)**: Defined variables like --bg, --panel, and --border are used to manage the color palette, allowing for easy, centralized theme modification (e.g., swapping to light mode in the future).
* **Dark Theme:** The main background is set to a deep, dark color (--bg: #0f1320), while content panels use a slightly lighter shade (--panel: #12172a) to create depth and focus.
* **Bootstrap Customization:** Standard Bootstrap classes are used for layout (row, col-md-6), but custom CSS ensures the dark theme integrates seamlessly by styling the components (like cards and forms) with the defined panel colors and border styles.
* **Readability:** High contrast is maintained between the text (white/light grey) and the dark backgrounds to ensure excellent readability of all metrics and chart labels.

**8. Key Features**

* **Responsive Dashboard:** Utilizes the Bootstrap grid for optimal viewing on all screen sizes.
* **Workout Logging System:** Form for submitting workout type, duration, and calories.
* **Persistent Goal Setting:** Goals are saved via Local Storage and loaded automatically.
* **Data Visualization:** Bar charts update instantly to reflect goal progress.
* **Workout History Management:** Table lists all entries and allows individual deletion.
* **Full Data Reset:** Button to clear all saved data from Local Storage.

**9. Challenges Faced & Solutions**

|  |  |
| --- | --- |
| Challenge | Solution |
| Data Persistence | Used **Local Storage** to save and load all user data, ensuring data integrity across sessions. |
| Real-time Data Update | A single refreshUI() function updates all dashboard elements and charts after any data change. |
| Graphing Goal Progress | Integrated **Chart.js** with custom logic to calculate current totals and plot them against saved goals. |

**10. Outcome**

The Fitness Tracker Dashboard successfully demonstrates proficiency in building complex, data-driven front-end interfaces. By effectively combining Bootstrap, jQuery, and Chart.js, the project resulted in a functional, responsive, and visually appealing tool for personal fitness management. The use of client-side data persistence ensures the tool is immediately practical.

**11. Future Enhancements**

* User Authentication: Implement a full user login system to allow data to be stored on a server for multi-device access.
* Advanced Data Filtering: Add controls to filter the workout history and dashboard data by date range or workout type.
* Aesthetics: Integrate a light-mode/dark-mode toggle for user preference flexibility

**12. Code**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <title>Fitness Tracker Dashboard</title>

  <!-- Bootstrap 5 -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />

  <!-- Icons -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css" rel="stylesheet" />

  <!-- jQuery -->

  <script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

  <!-- Bootstrap JS -->

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

  <!-- Chart.js 4 -->

  <script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.3/dist/chart.umd.min.js"></script>

  <style>

    :root{

      --bg:#0f1320;

      --panel:#12172a;

      --glass:rgba(255,255,255,0.06);

      --border:rgba(255,255,255,0.12);

      --txt:#e8ecf5;

      --muted:#a3adc2;

      --pri:#7b5cff;

      --sec:#22d3ee;

      --pos:#22c55e;

      --neg:#ef4444;

    }

    body{

      background: radial-gradient(900px 600px at -10% -20%, #1c1045 0%, transparent 65%),

                  radial-gradient(900px 600px at 120% 0%, #083048 0%, transparent 65%),

                  var(--bg);

      color: var(--txt);

      min-height: 100vh;

    }

    .navbar{

      background: linear-gradient(135deg, #0c1020, #141a2c) !important;

      border-bottom: 1px solid var(--border);

    }

    .brand{

      background: linear-gradient(90deg, var(--pri), var(--sec));

      -webkit-background-clip: text;

      background-clip: text;

      color: transparent;

    }

    .panel{

      background: linear-gradient(180deg, rgba(255,255,255,0.04), rgba(255,255,255,0.02));

      border: 1px solid var(--border);

      border-radius: 16px;

      box-shadow: 0 8px 28px rgba(0,0,0,0.25);

    }

    .stat-card{

      background: var(--panel);

      border: 1px solid var(--border);

      border-radius: 16px;

      padding: 16px;

    }

    .muted{ color: var(--muted); }

    .btn-neo{

      background: linear-gradient(180deg, rgba(255,255,255,0.08), rgba(255,255,255,0.02));

      border: 1px solid var(--border);

      color: var(--txt);

    }

    .btn-neo:hover{ border-color: rgba(123,92,255,0.6); color: var(--txt); }

    .form-control, .form-select{

      background: rgba(255,255,255,0.06);

      border: 1px solid var(--border);

      color: var(--txt);

    }

    .form-control::placeholder{ color: var(--muted); }

    .table{

      color: var(--txt);

    }

    .progress{

      background-color: rgba(255,255,255,0.08);

      height: 10px;

    }

    .progress-bar{

      background-image: linear-gradient(90deg, var(--pri), var(--sec));

    }

    .footer{

      border-top: 1px solid var(--border);

      color: var(--muted);

    }

    @media (max-width: 991.98px){

      .sidebar{ position: static !important; height: auto !important; }

    }

  </style>

</head>

<body>

  <!-- Navbar -->

  <nav class="navbar navbar-expand-lg sticky-top">

    <div class="container py-2">

      <a class="navbar-brand d-flex align-items-center gap-2" href="#">

        <i class="bi bi-activity text-info fs-4"></i>

        <span class="fw-bold brand">Fitness Tracker</span>

      </a>

      <button class="navbar-toggler text-white" type="button" data-bs-toggle="collapse" data-bs-target="#nav">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div id="nav" class="collapse navbar-collapse">

        <div class="ms-auto d-flex align-items-center gap-2">

          <button class="btn btn-neo" data-bs-toggle="modal" data-bs-target="#aboutModal"><i class="bi bi-info-circle"></i></button>

        </div>

      </div>

    </div>

  </nav>

  <div class="container-fluid">

    <div class="row">

      <!-- Sidebar -->

      <aside class="col-lg-3 col-xl-2 p-3 sidebar">

        <div class="panel p-3">

          <h5 class="mb-3">Add Workout</h5>

          <form id="workoutForm" class="d-grid gap-2">

            <div>

              <label class="form-label">Date</label>

              <input type="date" id="wDate" class="form-control" required />

            </div>

            <div>

              <label class="form-label">Type</label>

              <select id="wType" class="form-select" required>

                <option value="">Select</option>

                <option>Running</option>

                <option>Cycling</option>

                <option>Walking</option>

                <option>Gym</option>

                <option>Yoga</option>

                <option>Swimming</option>

              </select>

            </div>

            <div>

              <label class="form-label">Duration (min)</label>

              <input type="number" id="wDuration" class="form-control" min="1" required placeholder="e.g., 30" />

            </div>

            <div>

              <label class="form-label">Calories</label>

              <input type="number" id="wCalories" class="form-control" min="0" required placeholder="e.g., 250" />

            </div>

            <button class="btn btn-primary" type="submit">

              <i class="bi bi-plus-circle"></i> Add Workout

            </button>

            <button class="btn btn-outline-light" type="button" id="resetAll">

              <i class="bi bi-trash3"></i> Clear All

            </button>

          </form>

        </div>

        <div class="panel p-3 mt-3">

          <h5 class="mb-3">Goals</h5>

          <form id="goalForm" class="d-grid gap-2">

            <div class="muted small">Weekly Goals</div>

            <div class="d-flex gap-2">

              <input type="number" id="gWeeklyMinutes" class="form-control" min="0" placeholder="Minutes" />

              <input type="number" id="gWeeklyCalories" class="form-control" min="0" placeholder="Calories" />

            </div>

            <div class="muted small mt-2">Monthly Goals</div>

            <div class="d-flex gap-2">

              <input type="number" id="gMonthlyMinutes" class="form-control" min="0" placeholder="Minutes" />

              <input type="number" id="gMonthlyCalories" class="form-control" min="0" placeholder="Calories" />

            </div>

            <button class="btn btn-neo" type="submit"><i class="bi bi-check2-circle"></i> Save Goals</button>

          </form>

          <hr />

          <div>

            <div class="d-flex justify-content-between">

              <span class="muted small">Weekly Minutes</span>

              <span id="wkMinLbl" class="small">0/0</span>

            </div>

            <div class="progress mb-2"><div id="wkMinBar" class="progress-bar" style="width:0%"></div></div>

            <div class="d-flex justify-content-between">

              <span class="muted small">Weekly Calories</span>

              <span id="wkCalLbl" class="small">0/0</span>

            </div>

            <div class="progress mb-2"><div id="wkCalBar" class="progress-bar" style="width:0%"></div></div>

            <div class="d-flex justify-content-between">

              <span class="muted small">Monthly Minutes</span>

              <span id="moMinLbl" class="small">0/0</span>

            </div>

            <div class="progress mb-2"><div id="moMinBar" class="progress-bar" style="width:0%"></div></div>

            <div class="d-flex justify-content-between">

              <span class="muted small">Monthly Calories</span>

              <span id="moCalLbl" class="small">0/0</span>

            </div>

            <div class="progress"><div id="moCalBar" class="progress-bar" style="width:0%"></div></div>

          </div>

        </div>

      </aside>

      <!-- Main -->

      <main class="col-lg-9 col-xl-10 p-3 p-lg-4">

        <!-- Top Stats -->

        <div class="row g-3">

          <div class="col-6 col-md-3">

            <div class="stat-card">

              <div class="muted small">This Week Minutes</div>

              <div class="fs-4 fw-bold" id="statWeekMin">0</div>

            </div>

          </div>

          <div class="col-6 col-md-3">

            <div class="stat-card">

              <div class="muted small">This Week Calories</div>

              <div class="fs-4 fw-bold" id="statWeekCal">0</div>

            </div>

          </div>

          <div class="col-6 col-md-3">

            <div class="stat-card">

              <div class="muted small">This Month Minutes</div>

              <div class="fs-4 fw-bold" id="statMonthMin">0</div>

            </div>

          </div>

          <div class="col-6 col-md-3">

            <div class="stat-card">

              <div class="muted small">This Month Calories</div>

              <div class="fs-4 fw-bold" id="statMonthCal">0</div>

            </div>

          </div>

        </div>

        <!-- Charts -->

        <div class="row g-3 mt-1">

          <div class="col-12 col-xl-7">

            <div class="panel p-3">

              <div class="d-flex justify-content-between align-items-center">

                <h5 class="mb-0">Weekly Duration</h5>

                <div class="muted small">Minutes/day</div>

              </div>

              <canvas id="weeklyLine" height="120"></canvas>

            </div>

          </div>

          <div class="col-12 col-xl-5">

            <div class="panel p-3">

              <div class="d-flex justify-content-between align-items-center">

                <h5 class="mb-0">Daily Calories</h5>

                <div class="muted small">This week</div>

              </div>

              <canvas id="caloriesBar" height="120"></canvas>

            </div>

          </div>

        </div>

        <div class="row g-3 mt-1">

          <div class="col-12">

            <div class="panel p-3">

              <div class="d-flex justify-content-between align-items-center">

                <h5 class="mb-0">Progress vs Goals</h5>

                <div class="muted small">Line = actual, Bars = goals</div>

              </div>

              <canvas id="comboChart" height="120"></canvas>

            </div>

          </div>

        </div>

        <!-- Workout Table -->

        <div class="panel p-3 mt-3">

          <div class="d-flex justify-content-between align-items-center mb-2">

            <h5 class="mb-0">Workout Log</h5>

            <div class="muted small" id="totalSummary">0 workouts • 0 min • 0 kcal</div>

          </div>

          <div class="table-responsive">

            <table class="table table-hover align-middle">

              <thead>

                <tr>

                  <th>Date</th>

                  <th>Type</th>

                  <th class="text-end">Duration (min)</th>

                  <th class="text-end">Calories</th>

                  <th class="text-end">Actions</th>

                </tr>

              </thead>

              <tbody id="workoutTbody"></tbody>

            </table>

          </div>

        </div>

      </main>

    </div>

  </div>

  <!-- About Modal -->

  <div class="modal fade" id="aboutModal" tabindex="-1">

    <div class="modal-dialog modal-dialog-centered">

      <div class="modal-content">

        <div class="modal-header border-0">

          <h5 class="modal-title">About</h5>

          <button class="btn-close btn-close-white" data-bs-dismiss="modal"></button>

        </div>

        <div class="modal-body">

          <p>This dashboard uses Bootstrap for the responsive layout, Chart.js for charts, and LocalStorage to persist workouts and goals across sessions.</p>

        </div>

      </div>

    </div>

  </div>

  <footer class="footer py-4 mt-4">

    <div class="container d-flex justify-content-between flex-wrap gap-2">

      <div>Built with Bootstrap, jQuery, and Chart.js for a modern, responsive fitness tracker experience.</div>

      <div class="muted">Data is stored locally in the browser.</div>

    </div>

  </footer>

  <script>

    // ----- Storage Keys -----

    const LS\_WORKOUTS\_KEY = "ft\_workouts\_v1";

    const LS\_GOALS\_KEY = "ft\_goals\_v1";

    // ----- State -----

    let workouts = []; // {id, date: 'YYYY-MM-DD', type, duration, calories}

    let goals = { weeklyMinutes: 150, weeklyCalories: 1500, monthlyMinutes: 600, monthlyCalories: 6000 };

    // ----- DOM -----

    const $tbody = $("#workoutTbody");

    const $summary = $("#totalSummary");

    const $wkMinLbl = $("#wkMinLbl");

    const $wkCalLbl = $("#wkCalLbl");

    const $moMinLbl = $("#moMinLbl");

    const $moCalLbl = $("#moCalLbl");

    const $wkMinBar = $("#wkMinBar");

    const $wkCalBar = $("#wkCalBar");

    const $moMinBar = $("#moMinBar");

    const $moCalBar = $("#moCalBar");

    const $statWeekMin = $("#statWeekMin");

    const $statWeekCal = $("#statWeekCal");

    const $statMonthMin = $("#statMonthMin");

    const $statMonthCal = $("#statMonthCal");

    // ----- Utils -----

    const uid = () => Math.random().toString(36).slice(2, 10);

    const todayStr = () => new Date().toISOString().slice(0,10);

    function startOfWeek(d){

      const dt = new Date(d);

      const day = dt.getDay(); // 0 Sun..6 Sat

      const diff = dt.getDate() - day + (day === 0 ? -6 : 1); // Monday as start

      return new Date(dt.setDate(diff));

    }

    function endOfWeek(d){

      const s = startOfWeek(d);

      return new Date(s.getFullYear(), s.getMonth(), s.getDate() + 6);

    }

    function startOfMonth(d){

      const dt = new Date(d);

      return new Date(dt.getFullYear(), dt.getMonth(), 1);

    }

    function endOfMonth(d){

      const dt = new Date(d);

      return new Date(dt.getFullYear(), dt.getMonth()+1, 0);

    }

    function fmt(n){ return Number(n||0).toLocaleString(); }

    // ----- Storage -----

    function saveAll(){

      localStorage.setItem(LS\_WORKOUTS\_KEY, JSON.stringify(workouts));

      localStorage.setItem(LS\_GOALS\_KEY, JSON.stringify(goals));

    }

    function loadAll(){

      try{

        const w = JSON.parse(localStorage.getItem(LS\_WORKOUTS\_KEY) || "[]");

        const g = JSON.parse(localStorage.getItem(LS\_GOALS\_KEY) || "null");

        workouts = Array.isArray(w) ? w : [];

        if (g && typeof g === "object"){

          goals = {...goals, ...g};

        }

      }catch(e){

        workouts = [];

      }

    }

    // ----- Rendering -----

    function renderTable(){

      $tbody.empty();

      const sorted = [...workouts].sort((a,b)=> a.date.localeCompare(b.date)); // ascending

      let totalDur = 0, totalCal = 0;

      sorted.forEach(w=>{

        totalDur += Number(w.duration)||0;

        totalCal += Number(w.calories)||0;

        const $tr = $(`

          <tr>

            <td>${w.date}</td>

            <td>${w.type}</td>

            <td class="text-end">${fmt(w.duration)}</td>

            <td class="text-end">${fmt(w.calories)}</td>

            <td class="text-end">

              <button class="btn btn-sm btn-outline-warning me-2" data-edit="${w.id}"><i class="bi bi-pencil"></i></button>

              <button class="btn btn-sm btn-outline-danger" data-del="${w.id}"><i class="bi bi-trash"></i></button>

            </td>

          </tr>

        `);

        $tbody.append($tr);

      });

      $("#totalSummary").text(`${fmt(sorted.length)} workouts • ${fmt(totalDur)} min • ${fmt(totalCal)} kcal`);

      bindRowActions();

    }

    function bindRowActions(){

      $tbody.find("[data-edit]").off("click").on("click", function(){

        const id = $(this).data("edit");

        const w = workouts.find(x=>x.id===id);

        if(!w) return;

        // simple inline edit with prompts (demo)

        const nd = prompt("Edit date (YYYY-MM-DD):", w.date) || w.date;

        const nt = prompt("Edit type:", w.type) || w.type;

        const ndur = Number(prompt("Edit duration (min):", w.duration)) || w.duration;

        const ncal = Number(prompt("Edit calories:", w.calories)) || w.calories;

        Object.assign(w, { date: nd, type: nt, duration: ndur, calories: ncal });

        saveAll();

        refreshUI();

      });

      $tbody.find("[data-del]").off("click").on("click", function(){

        const id = $(this).data("del");

        if (confirm("Delete this workout?")){

          workouts = workouts.filter(x=>x.id!==id);

          saveAll();

          refreshUI();

        }

      });

    }

    // ----- Aggregation -----

    function aggregateByDate(rangeStart, rangeEnd){

      const m = new Map(); // date => {minutes, calories}

      const s = new Date(rangeStart);

      const e = new Date(rangeEnd);

      for (let d = new Date(s); d <= e; d.setDate(d.getDate()+1)){

        const key = d.toISOString().slice(0,10);

        m.set(key, {minutes:0, calories:0});

      }

      workouts.forEach(w=>{

        const d = new Date(w.date);

        if (d >= s && d <= e){

          const key = w.date;

          const cur = m.get(key) || {minutes:0, calories:0};

          cur.minutes += Number(w.duration)||0;

          cur.calories += Number(w.calories)||0;

          m.set(key, cur);

        }

      });

      const labels = [...m.keys()].sort();

      const minutes = labels.map(k=> m.get(k).minutes);

      const calories = labels.map(k=> m.get(k).calories);

      return { labels, minutes, calories };

    }

    function totalsFor(rangeStart, rangeEnd){

      let minutes = 0, calories = 0;

      const s = new Date(rangeStart), e = new Date(rangeEnd);

      workouts.forEach(w=>{

        const d = new Date(w.date);

        if (d>=s && d<=e){

          minutes += Number(w.duration)||0;

          calories += Number(w.calories)||0;

        }

      });

      return { minutes, calories };

    }

    // ----- Charts -----

    let weeklyLineChart, caloriesBarChart, comboChart;

    function makeGrad(ctx, c1, c2){

      const g = ctx.createLinearGradient(0,0,0,180);

      g.addColorStop(0, c1);

      g.addColorStop(1, c2);

      return g;

    }

    function renderCharts(){

      const now = new Date();

      const ws = startOfWeek(now);

      const we = endOfWeek(now);

      const ms = startOfMonth(now);

      const me = endOfMonth(now);

      const weekAgg = aggregateByDate(ws, we);

      const weekLabels = weekAgg.labels.map(d => d.slice(5)); // MM-DD

      // Weekly line (minutes)

      const lineCtx = document.getElementById("weeklyLine").getContext("2d");

      weeklyLineChart && weeklyLineChart.destroy();

      weeklyLineChart = new Chart(lineCtx, {

        type: "line",

        data: {

          labels: weekLabels,

          datasets: [{

            label: "Minutes",

            data: weekAgg.minutes,

            borderColor: "#7b5cff",

            backgroundColor: makeGrad(lineCtx, "rgba(123,92,255,0.35)", "rgba(123,92,255,0.05)"),

            fill: true,

            tension: 0.35,

            pointRadius: 3

          }]

        },

        options: {

          responsive: true,

          plugins: { legend: { labels: { color: "#e8ecf5" } } },

          scales: {

            x: { ticks: { color: "#a3adc2" }, grid: { color: "rgba(255,255,255,0.06)" } },

            y: { ticks: { color: "#a3adc2" }, grid: { color: "rgba(255,255,255,0.06)" } }

          }

        }

      });

      // Calories bar (week)

      const barCtx = document.getElementById("caloriesBar").getContext("2d");

      caloriesBarChart && caloriesBarChart.destroy();

      caloriesBarChart = new Chart(barCtx, {

        type: "bar",

        data: {

          labels: weekLabels,

          datasets: [{

            label: "Calories",

            data: weekAgg.calories,

            backgroundColor: "#22d3ee"

          }]

        },

        options: {

          responsive: true,

          plugins: { legend: { labels: { color: "#e8ecf5" } } },

          scales: {

            x: { ticks: { color: "#a3adc2" }, grid: { color: "rgba(255,255,255,0.06)" } },

            y: { ticks: { color: "#a3adc2" }, grid: { color: "rgba(255,255,255,0.06)" } }

          }

        }

      });

      // Combo: monthly minutes vs goal

      const monthAgg = aggregateByDate(ms, me);

      const comboCtx = document.getElementById("comboChart").getContext("2d");

      comboChart && comboChart.destroy();

      const daysInMonth = monthAgg.labels.length;

      const dailyGoalMin = goals.monthlyMinutes > 0 ? (goals.monthlyMinutes / daysInMonth) : 0;

      const dailyGoalCal = goals.monthlyCalories > 0 ? (goals.monthlyCalories / daysInMonth) : 0;

      comboChart = new Chart(comboCtx, {

        data: {

          labels: monthAgg.labels.map(d=>d.slice(5)),

          datasets: [

            {

              type: "line",

              label: "Minutes (actual)",

              data: monthAgg.minutes,

              borderColor: "#7b5cff",

              backgroundColor: "rgba(123,92,255,0.15)",

              yAxisID: "y1",

              tension: 0.35

            },

            {

              type: "bar",

              label: "Minutes (goal/day)",

              data: Array(daysInMonth).fill(dailyGoalMin),

              backgroundColor: "rgba(34,211,238,0.35)",

              yAxisID: "y1"

            },

            {

              type: "bar",

              label: "Calories (goal/day)",

              data: Array(daysInMonth).fill(dailyGoalCal),

              backgroundColor: "rgba(34,197,94,0.35)",

              yAxisID: "y2"

            }

          ]

        },

        options: {

          responsive: true,

          plugins: { legend: { labels: { color: "#e8ecf5" } } },

          scales: {

            x: { ticks: { color: "#a3adc2" }, grid: { color: "rgba(255,255,255,0.06)" } },

            y1: { position: "left", ticks: { color: "#a3adc2" }, grid: { color: "rgba(255,255,255,0.06)" } },

            y2: { position: "right", ticks: { color: "#a3adc2" }, grid: { drawOnChartArea: false } }

          }

        }

      });

    }

    function renderStatsAndGoals(){

      const now = new Date();

      const ws = startOfWeek(now), we = endOfWeek(now);

      const ms = startOfMonth(now), me = endOfMonth(now);

      const wk = totalsFor(ws, we);

      const mo = totalsFor(ms, me);

      $statWeekMin.text(fmt(wk.minutes));

      $statWeekCal.text(fmt(wk.calories));

      $statMonthMin.text(fmt(mo.minutes));

      $statMonthCal.text(fmt(mo.calories));

      // Progress bars

      const wkMinPct = goals.weeklyMinutes ? Math.min(100, Math.round(100\*wk.minutes/goals.weeklyMinutes)) : 0;

      const wkCalPct = goals.weeklyCalories ? Math.min(100, Math.round(100\*wk.calories/goals.weeklyCalories)) : 0;

      const moMinPct = goals.monthlyMinutes ? Math.min(100, Math.round(100\*mo.minutes/goals.monthlyMinutes)) : 0;

      const moCalPct = goals.monthlyCalories ? Math.min(100, Math.round(100\*mo.calories/goals.monthlyCalories)) : 0;

      $wkMinLbl.text(`${fmt(wk.minutes)}/${fmt(goals.weeklyMinutes||0)}`);

      $wkCalLbl.text(`${fmt(wk.calories)}/${fmt(goals.weeklyCalories||0)}`);

      $moMinLbl.text(`${fmt(mo.minutes)}/${fmt(goals.monthlyMinutes||0)}`);

      $moCalLbl.text(`${fmt(mo.calories)}/${fmt(goals.monthlyCalories||0)}`);

      $wkMinBar.css("width", wkMinPct + "%");

      $wkCalBar.css("width", wkCalPct + "%");

      $moMinBar.css("width", moMinPct + "%");

      $moCalBar.css("width", moCalPct + "%");

    }

    function refreshUI(){

      renderTable();

      renderStatsAndGoals();

      renderCharts();

    }

    // ----- Events -----

    $(function(){

      // Init dates

      $("#wDate").val(todayStr());

      // Load persisted data

      loadAll();

      // Prefill goals UI

      $("#gWeeklyMinutes").val(goals.weeklyMinutes);

      $("#gWeeklyCalories").val(goals.weeklyCalories);

      $("#gMonthlyMinutes").val(goals.monthlyMinutes);

      $("#gMonthlyCalories").val(goals.monthlyCalories);

      // Submit workout

      $("#workoutForm").on("submit", function(e){

        e.preventDefault();

        const w = {

          id: uid(),

          date: $("#wDate").val(),

          type: $("#wType").val(),

          duration: Number($("#wDuration").val()),

          calories: Number($("#wCalories").val())

        };

        if(!w.date || !w.type || !w.duration || w.duration < 1){ return; }

        workouts.push(w);

        saveAll();

        this.reset();

        $("#wDate").val(todayStr());

        refreshUI();

      });

      // Clear All

      $("#resetAll").on("click", function(){

        if (confirm("Clear all workouts and goals?")){

          workouts = [];

          // keep goals but you can reset if desired

          saveAll();

          refreshUI();

        }

      });

      // Save goals

      $("#goalForm").on("submit", function(e){

        e.preventDefault();

        goals.weeklyMinutes = Number($("#gWeeklyMinutes").val()) || 0;

        goals.weeklyCalories = Number($("#gWeeklyCalories").val()) || 0;

        goals.monthlyMinutes = Number($("#gMonthlyMinutes").val()) || 0;

        goals.monthlyCalories = Number($("#gMonthlyCalories").val()) || 0;

        saveAll();

        refreshUI();

      });

      // First render

      refreshUI();

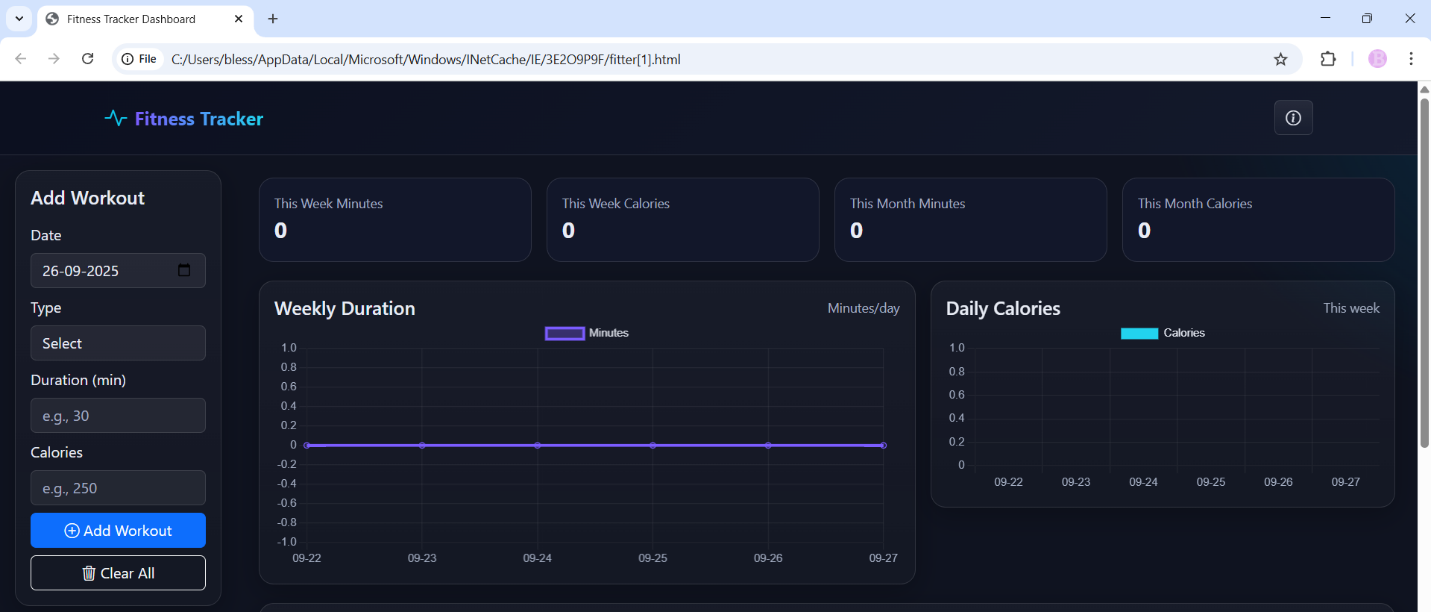
    });

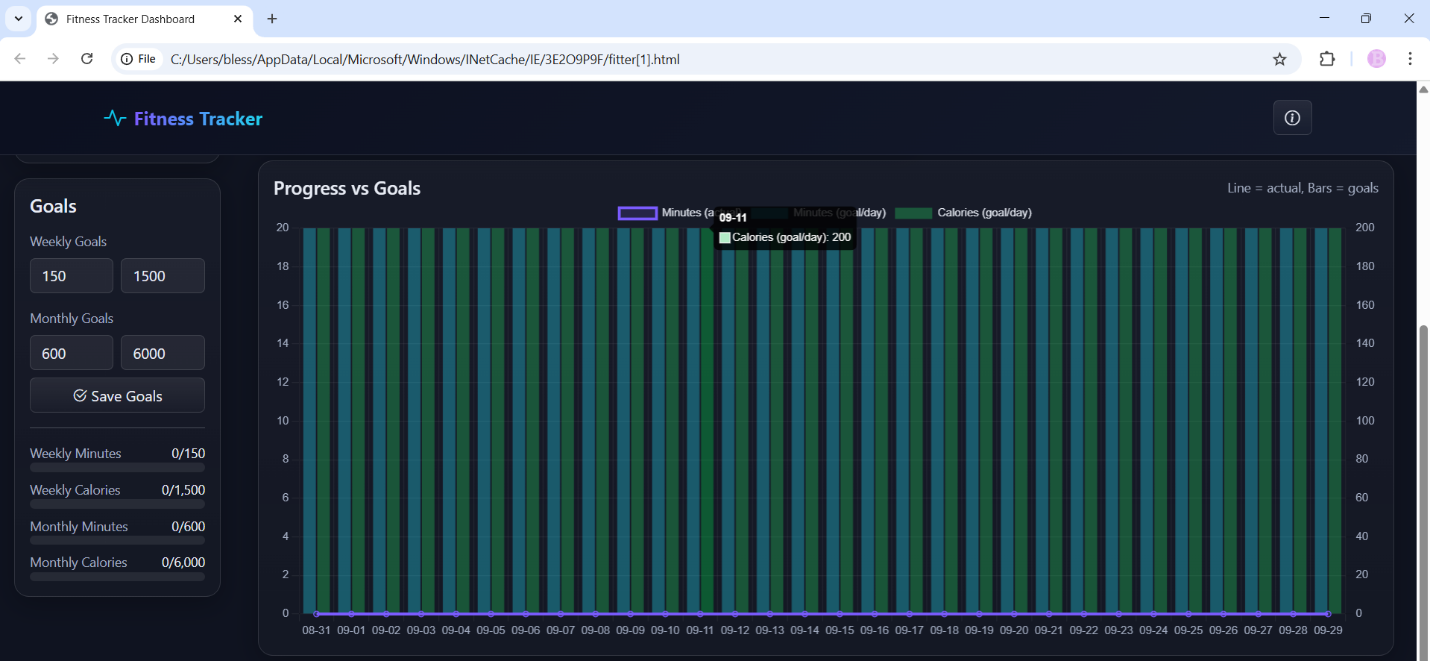
  </script>

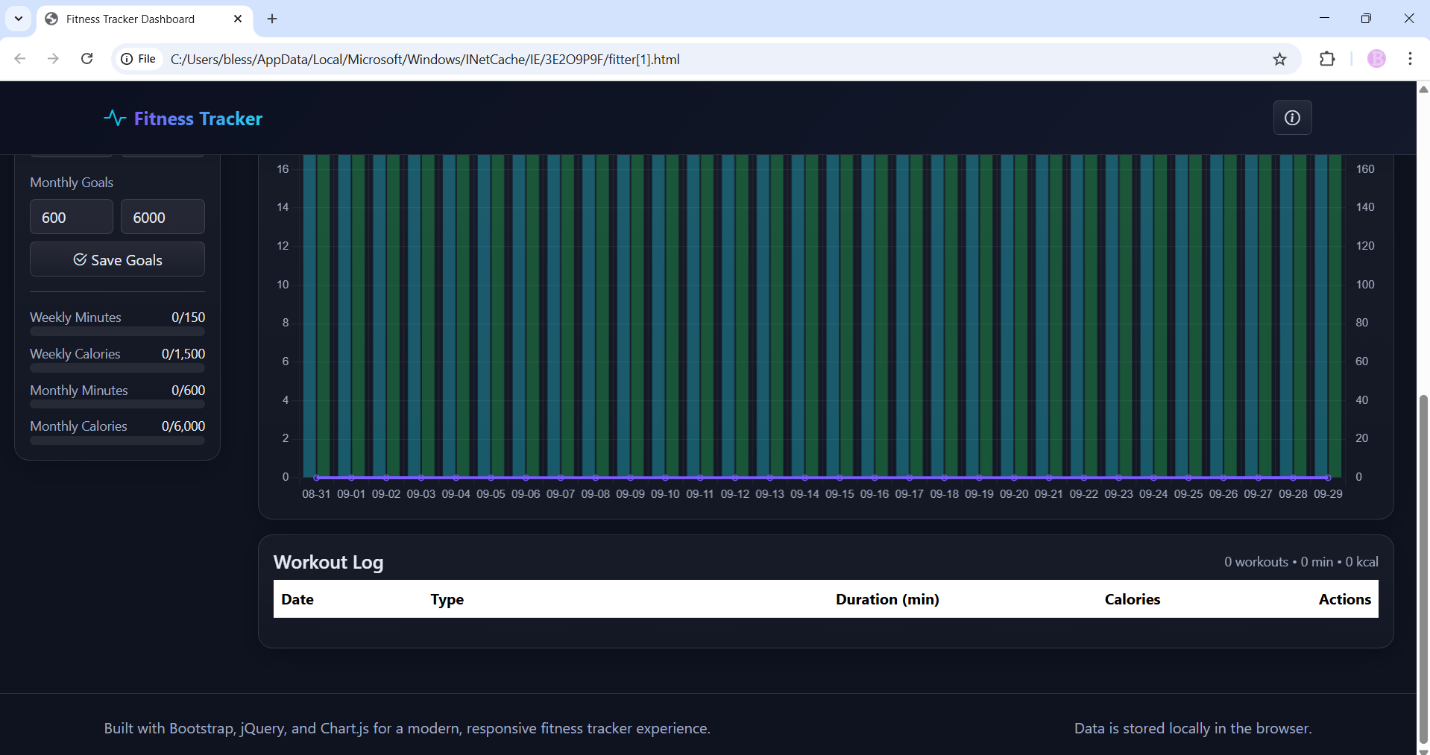
</body>

</html>

**13. Screenshots of Final Output**

****

****

****

**11. Conclusion**

The Fitness Tracker Dashboard project successfully demonstrates key principles of modern front-end development. The integration of **Bootstrap** for responsiveness, **jQuery** for DOM efficiency, and **Chart.js** for data visualization resulted in a highly functional and visually engaging user interface. By effectively managing user data client-side through **Local Storage**, the application fulfills its objective as a practical, stand-alone personal fitness tool. The project serves as a strong foundation for future full-stack integration and advanced feature development.

**12. References**

* L&T LMS : https://learn.lntedutech.com/Landing/MyCourse