

Front-End UI/UX Development

Front-End UI/UX Development Mini Project

UI/UX DESIGN FUNDAMENTALS

Project Title: Daily Habit Tracker

Submitted By:

Name	Reg no.	E-mail address
Akash reddy	2462347	Pingali.akash@btech.christ university.in
Sandeep Siby	2462358	Sandeep.siby@btech.christ university.in
Vijay M	2462373	Vijay.M@btech.christuniver sity.in

- Course:**
- Instructor Name:** *Mr. Nagaveena*
- Institution:** christ university(Deemed to be University) kengeri campus
- Date of Submission:** 26/09/2025

Abstract

The **Daily Habit Tracker** is a lightweight web-based application designed to help users build and maintain positive routines. It allows individuals to create, edit, and manage custom habits such as *Exercise, Reading, or Drinking Water*, and provides a visual grid-based interface for marking daily completions. The system tracks user progress with streak counters (current and best), monthly progress indicators, and a color-coded daily tracker grid for intuitive visualization. Additional features include editing or deleting habits, month-to-month navigation, and reset/clear options for flexible usage. Data is stored locally in the browser using **localStorage**, ensuring persistence without requiring a backend. Developed using **HTML, CSS, JavaScript, Bootstrap, and jQuery**, the application emphasizes simplicity, responsiveness, and ease of interaction. This project demonstrates the integration of front-end technologies to build an interactive productivity tool that promotes consistency and self-improvement.

Objectives

- 1. Provide a simple platform for users to create, edit, and manage their daily habits.**
- 2. Enable daily progress tracking through an interactive grid with checkboxes or colored cells.**
- 3. Motivate users by showing streak counters (current and best) for each habit.**
- 4. Visualize progress using progress bars and indicators for each habit over a selected month.**
- 5. Offer flexibility with options to reset tracking or clear all habits.**
- 6. Ensure data persistence by storing user habits and progress in the browser using **localStorage**.**
- 7. Build a responsive and user-friendly interface using **HTML, CSS, Bootstrap, and jQuery**.**
- 8. Encourage consistency and self-improvement by helping users monitor their routines daily.**

Scope of the Project

- 1. Habit Management – Users can add, edit, and delete habits according to their personal goals.**

2. **Daily Tracking** – Each habit can be tracked on a daily basis using an interactive calendar grid.
3. **Streak Monitoring** – The system displays current and best streaks to motivate users toward consistency.
4. **Progress Visualization** – Monthly progress is shown through progress bars and visual indicators.
5. **Data Persistence** – User habits and progress are stored locally in the browser using `localStorage` for offline accessibility.
6. **Reset and Clear Options** – Provides flexibility to reset tracking data without deleting habits or to clear all habits entirely.
7. **Month Navigation** – Users can navigate between months to view or update past/future habit progress.
8. **Responsive Design** – The tracker is designed with Bootstrap for usability across desktops, tablets, and smartphones.
9. **Technology Demonstration** – Showcases practical implementation of HTML, CSS, JavaScript, Bootstrap, and jQuery in a real-world application.

Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Markup and content structure
CSS3	Styling and layout management
VS Code	Code editor
Chrome DevTools	Testing and debugging

HTML Structure Overview

1. **<!doctype html> & <html>**
 - Defines the document type and root HTML element.
2. **<head> Section**
 - Contains meta information (`charset`, `viewport`).
 - Title of the page (Daily Habit Tracker).
 - External stylesheets (Bootstrap CDN + custom CSS).

3. <body> Section

- **Main Container (<div class="container py-4">)** – Holds all application content.

Inside Main Container:

- **Header Section**
 - Page title (<h2>Daily Habit Tracker</h2>).
 - Buttons for *Reset All Tracking* and *Clear All Habits*.
- **Add Habit Form (<form id="add-habit-form">)**
 - Input field to enter a new habit.
 - Add button to submit the habit.
 - Small tip text for user guidance.
- **Month Navigation**
 - Buttons for *Previous Month* and *Next Month*.
 - Current month label (<h5 id="month-label">).
- **Habits List Section (<div id="habits-list">)**
 - Dynamically populated with habit cards using JavaScript.
 - **Each Habit Card includes:**
 - Habit name, Edit, and Delete buttons.
 - Streak counters (current and best).
 - Monthly progress bar.
 - Daily tracker grid (days of the current month with toggle checkboxes).
- **Footer Section**
 - Small text credit mentioning technologies used.

4. External Scripts (before </body>)

- **jQuery (for DOM manipulation).**
- **Bootstrap JS bundle.**
- **Custom JavaScript logic (handles habit creation, tracking, rendering, localStorage, etc.).**

Key techniques and strategies include:

1. Modular HTML Structure

- The page is organized into sections (header, form, habit list, footer) for clarity and scalability.

2. Responsive Design with Bootstrap

- Bootstrap grid system and utility classes ensure the tracker works smoothly on desktops, tablets, and mobiles.

3. Dynamic DOM Manipulation with jQuery

- Habits, streaks, and daily tracking grids are generated and updated dynamically without reloading the page.

4. LocalStorage for Data Persistence

- User data (habits, progress, streaks) is saved in the browser's localStorage, so it remains available across sessions without a backend.

5. Event-Driven Interactions

- Click events (for toggling days, editing, deleting habits) and form submissions drive the application's interactivity.

6. Progress Visualization

- Use of progress bars, colored cells, and streak counters to give users immediate feedback on their performance.

7. Month Navigation Strategy

- JavaScript date functions allow switching between months, enabling past/future progress tracking.

8. Data Validation and Safety

- Input fields check for empty values.
- Confirmations are used for critical actions (reset, delete).

9. Reusability and Maintainability

- Functions are written to handle adding, editing, deleting, and toggling habits systematically.

10. User Motivation Approach

- Visual indicators, streak counters, and progress percentages encourage habit consistency and long-term engagement.

Key Features

Feature	Description
Responsive Design	Adapts seamlessly to all screen sizes
Smooth Navigation	Fixed top nav with anchor links
Project Cards	Flex-based layout with hover effects
Contact Form (non-functional)	Placeholder layout for inputs and button
Accessible Fonts & Colors	High contrast and readable typography

Challenges Faced & Solutions

Challenge	Solution
Overlapping elements on small screens	Used media queries to stack elements
Difficulty aligning items using float	Shifted to Flexbox and Grid
Typography scaling issue	Used relative units (em/rem) instead of px

Outcome

The Daily Habit Tracker successfully provides users with a simple and interactive platform to monitor and maintain their daily routines. By allowing the creation, editing, and deletion of custom habits, the application ensures personalization to suit individual goals. The inclusion of daily tracking grids, streak counters, and progress indicators makes it easy for users to visualize their performance and stay motivated. Data persistence through browser localStorage enables users to retain their progress across sessions without requiring external databases. The responsive interface built with Bootstrap and jQuery ensures accessibility across devices, while reset and clear options offer flexibility in usage. Overall, the project demonstrates how front-end web technologies can be integrated effectively to create a practical productivity tool that encourages consistency, self-improvement, and habit formation.

Future Enhancements

- 1. User Authentication & Cloud Storage – Enable user login and sync habits across multiple devices using cloud databases.**
- 2. Reminders & Notifications – Add push notifications or email reminders to alert users about pending habits.**
- 3. Weekly & Yearly Views – Extend beyond monthly tracking to provide weekly summaries and yearly statistics.**
- 4. Data Analytics & Charts – Integrate visual charts (bar, line, or pie charts) to analyze long-term progress and patterns.**
- 5. Gamification – Introduce badges, rewards, or achievement levels to increase user motivation.**
- 6. Custom Themes & Dark Mode – Allow users to personalize the appearance of the tracker.**
- 7. Export/Import Options – Provide CSV/PDF export and import functionality for data backup and sharing.**
- 8. Integration with Health Apps – Connect with fitness trackers, calendars, or mobile health apps for seamless tracking.**
- 9. AI Suggestions – Recommend new habits or provide motivational tips based on user performance.**
- 10. Collaboration & Sharing – Allow users to share progress with friends or groups for accountability and social motivation.**

Sample Code:

Html code:

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Weekly Habit Tracker</title>

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link
        href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700&display=s
        wap" rel="stylesheet">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
        awesome/6.5.2/css/all.min.css">

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="app-container">

        <header class="app-header">

            <div class="header-text">

                <h1><i class="fa-solid fa-square-check"></i> DAILY HABIT
                TRACKER</h1>

                <p>Build your routine, track your progress.</p>

            </div>

            <div class="header-image-container">

                

                <div class="image-caption">Small Steps, Big Changes.</div>

            </div>

    
```

```
</header>

<main class="main-content">
  <aside class="habits-section">
    <h2>MY HABITS</h2>
    <ul id="habits-list">
      </ul>
    <div class="add-habit-form">
      <input type="text" id="new-habit-input" placeholder="Add a new habit...">
      <button id="add-habit-btn" title="Add Habit"><i class="fa-solid fa-plus"></i></button>
    </div>
  </aside>

  <section class="progress-section">
    <div class="week-navigation">
      <button id="prev-week-btn" title="Previous Week"><i class="fa-solid fa-chevron-left"></i></button>
      <h2 id="week-display"></h2>
      <button id="next-week-btn" title="Next Week"><i class="fa-solid fa-chevron-right"></i></button>
    </div>
    <div class="progress-grid-container">
      <table class="progress-grid">
        <thead id="grid-header">
          </thead>
        <tbody id="grid-body">
          </tbody>
      </table>
    </div>
  </section>
</main>
```

```
</div>

<div class="stats-and-controls">
  <div class="stats-container">
    <div id="streak-info">
      <p class="streak-title">🔥 Streak Counter</p>
      <p id="streak-placeholder">Select a habit to view its streak.</p>
      <div id="streak-display" class="hidden">
        <span>Current: <strong id="current-streak">0</strong>
        days</span>
        <span>Best: <strong id="best-streak">0</strong> days</span>
      </div>
    </div>
    </div>
    <div class="control-buttons">
      <button id="clear-all-btn">Clear All Data</button>
    </div>
  </div>
</section>
</main>

<footer>
  <p>Powered by HTML, CSS, JavaScript</p>
</footer>
</div>

<script src="script.js"></script>
</body>
</html>

Style.css code:
```

```
/* General Setup */

:root {
  --primary-bg: #f4f7fa;
  --secondary-bg: #ffffff;
  --header-bg: #2c3e50;
  --text-dark: #333;
  --text-light: #fdfdfd;
  --text-muted: #7f8c8d;
  --accent-color: #4CAF50;
  --border-color: #e0e0e0;
  --danger-color: #e74c3c;
  --shadow: 0 4px 12px rgba(0, 0, 0, 0.08);
  --selected-bg: #e8f5e9; /* Light green for selected items */
}

*
```

```
{

  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
```

```
body {
  font-family: 'Poppins', sans-serif;
  background-color: var(--primary-bg);
  color: var(--text-dark);
  line-height: 1.6;
}
```

```
/* Main App Container */
```

```
.app-container {  
    max-width: 1200px;  
    margin: 40px auto;  
    padding: 20px;  
    background-color: var(--secondary-bg);  
    border-radius: 12px;  
    box-shadow: var(--shadow);  
}  
  
/* Header */  
  
.app-header {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    background-color: var(--header-bg);  
    color: var(--text-light);  
    padding: 20px 30px;  
    border-radius: 8px;  
    margin-bottom: 30px;  
}  
  
.app-header h1 { font-size: 1.8rem; font-weight: 700; }  
.app-header i { color: var(--accent-color); margin-right: 10px; }  
.app-header p { font-size: 0.9rem; opacity: 0.8; }  
  
.header-image-container { position: relative; text-align: center; }  
  
.header-image-container img { width: 150px; height: 100px; object-fit: cover; border-radius: 8px; }  
  
.image-caption { position: absolute; bottom: 5px; left: 0; right: 0; font-size: 0.8rem; font-weight: 600; color: white; background: rgba(0, 0, 0, 0.4); padding: 2px 0; }  
  
/* Main Content Layout */
```

```
.main-content {
  display: flex;
  gap: 30px;
}

.habits-section { flex: 1; min-width: 250px; }

.progress-section { flex: 3; }

/* Habits List Section */
#habits-list { list-style: none; }

#habits-list li {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 12px 10px;
  border: 1px solid var(--border-color);
  border-radius: 6px;
  margin-bottom: 10px;
  font-weight: 600;
  cursor: pointer;
  transition: background-color 0.3s;
}

#habits-list li.selected-habit {
  background-color: var(--selected-bg);
  border-color: var(--accent-color);
}

#habits-list li:not(.selected-habit):hover {
  background-color: var(--primary-bg);
}

#habits-list .delete-btn { color: var(--text-muted); transition: color 0.3s; }
```

```
#habits-list .delete-btn:hover { color: var(--danger-color); }

.add-habit-form { display: flex; margin-top: 20px; }

#new-habit-input { flex-grow: 1; padding: 10px; border: 1px solid var(--border-color); border-radius: 6px 0 0 6px; font-family: 'Poppins', sans-serif; }

#add-habit-btn { padding: 10px 15px; border: none; background-color: var(--accent-color); color: white; cursor: pointer; border-radius: 0 6px 6px 0; font-size: 1.1rem; }

/* Progress Section */

.week-navigation {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-bottom: 20px;
}

.week-navigation h2 {
  font-size: 1.2rem;
  color: var(--header-bg);
  margin: 0;
  padding: 0;
  border: none;
  text-align: center;
}

.week-navigation button {
  background: none;
  border: 1px solid var(--border-color);
  border-radius: 50%;
  cursor: pointer;
  width: 35px;
  height: 35px;
  color: var(--text-muted);
}
```

```
font-size: 1rem;
transition: background-color 0.3s;
}

.week-navigation button:hover {
background-color: var(--primary-bg);
color: var(--text-dark);
}

.progress-grid { width: 100%; border-collapse: collapse; text-align: center; }
.progress-grid th {
padding-bottom: 15px;
font-size: 0.8rem;
color: var(--text-muted);
font-weight: normal;
}
.progress-grid th div:first-child { font-weight: 600; }
.progress-grid td { padding: 5px; vertical-align: middle; }
.progress-grid .habit-name { text-align: left; font-weight: 600; font-size: 0.9rem;
padding-left: 0; }

.status-circle { display: inline-block; width: 24px; height: 24px; border: 2px solid #ccc;
border-radius: 50%; cursor: pointer; transition: background-color 0.3s, border-color 0.3s; }

.status-circle.completed { background-color: var(--accent-color); border-color: var(--accent-color); }

.status-circle.future { background-color: #e0e0e0; border-color: #e0e0e0; cursor: not-allowed; }

/* Stats and Controls */

.stats-and-controls {
margin-top: 30px;
display: flex;
```

```
justify-content: space-between;
align-items: center;
background-color: var(--primary-bg);
padding: 15px;
border-radius: 8px;
}

.stats-container { flex-grow: 1; }

#streak-info .streak-title { font-weight: 600; margin-bottom: 5px; }
#streak-placeholder { color: var(--text-muted); font-size: 0.9rem; }
#streak-display { display: flex; gap: 20px; }
#streak-display span strong { color: var(--accent-color); font-size: 1.2rem; }
.hidden { display: none !important; }

.control-buttons button { padding: 10px 15px; border: none; border-radius: 6px; color: white; cursor: pointer; font-weight: 600; }
#clear-all-btn { background-color: var(--danger-color); }

/* Footer */

footer { text-align: center; margin-top: 30px; color: var(--text-muted); font-size: 0.8rem; }

/* Responsive Design */

@media (max-width: 992px) { .main-content { flex-direction: column; } }
@media (max-width: 768px) {
  .app-header { flex-direction: column; text-align: center; gap: 20px; }
  .stats-and-controls { flex-direction: column; gap: 20px; align-items: stretch; text-align: center; }
}
}

Js scripct code:

document.addEventListener('DOMContentLoaded', () => {
```

```
// DOM Elements

const habitsList = document.getElementById('habits-list');

const newHabitInput = document.getElementById('new-habit-input');

const addHabitBtn = document.getElementById('add-habit-btn');

const weekDisplay = document.getElementById('week-display');

const gridHeader = document.getElementById('grid-header');

const gridBody = document.getElementById('grid-body');

const prevWeekBtn = document.getElementById('prev-week-btn');

const nextWeekBtn = document.getElementById('next-week-btn');

const clearAllBtn = document.getElementById('clear-all-btn');

const streakPlaceholder = document.getElementById('streak-placeholder');

const streakDisplay = document.getElementById('streak-display');

const currentStreakEl = document.getElementById('current-streak');

const bestStreakEl = document.getElementById('best-streak');
```

// App State

```
let habits = [];

let currentDate = new Date();

let selectedHabitIndex = null;
```

// --- UTILITY FUNCTIONS ---

```
const formatDate = (date) => {

  const year = date.getFullYear();

  const month = String(date.getMonth() + 1).padStart(2, '0');

  const day = String(date.getDate()).padStart(2, '0');

  return `${year}-${month}-${day}`;

};
```

```
const getWeekRange = (date) => {
```

```
const start = new Date(date);

const dayOfWeek = start.getDay() // 0 for Sunday, 1 for Monday...

const diff = start.getDate() - dayOfWeek + (dayOfWeek === 0 ? -6 : 1); // Adjust to
make Monday the start

start.setDate(diff);

const end = new Date(start);
end.setDate(start.getDate() + 6);

const dates = [];
for (let i = 0; i < 7; i++) {
    const day = new Date(start);
    day.setDate(start.getDate() + i);
    dates.push(day);
}
return { start, end, dates };
};

// --- DATA HANDLING ---
const loadHabits = () => {
    const storedHabits = localStorage.getItem('habits-weekly');
    if (storedHabits) {
        habits = JSON.parse(storedHabits);
    } else {
        habits = [
            { name: 'Exercise', completed: [] },
            { name: 'Read (20 mins)', completed: [] },
            { name: 'Drink Water (2L)', completed: [] }
        ];
    }
};
```

```
        }
    };

const saveHabits = () => {
    localStorage.setItem('habits-weekly', JSON.stringify(habits));
};

// --- RENDERING FUNCTIONS ---
const renderApp = () => {
    renderHabitList();
    renderProgressGrid();
    updateStreakInfo();
};

const renderHabitList = () => {
    habitsList.innerHTML = '';
    habits.forEach((habit, index) => {
        const li = document.createElement('li');
        li.dataset.index = index;
        if (index === selectedHabitIndex) {
            li.classList.add('selected-habit');
        }
        li.innerHTML = `
            <span>${habit.name}</span>
            <i class="fa-solid fa-trash-can delete-btn" data-index="${index}" title="Delete Habit"></i>
        `;
        habitsList.appendChild(li);
    });
};
```

```
};
```

```
const renderProgressGrid = () => {
  const { start, end, dates } = getWeekRange(currentDate);
  const monthNames = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul",
  "Aug", "Sep", "Oct", "Nov", "Dec"];
  weekDisplay.textContent = `${monthNames[start.getMonth()]} ${start.getDate()} - ${monthNames[end.getMonth()]} ${end.getDate()}, ${end.getFullYear()};
```

```
gridHeader.innerHTML = "";
```

```
gridBody.innerHTML = "";
```

```
const headerRow = document.createElement('tr');
headerRow.innerHTML = '<th>HABIT</th>';
dates.forEach(date => {
  const dayName = date.toLocaleDateString('en-US', { weekday: 'short' });
  headerRow.innerHTML +=
<th><div>${dayName}</div><div>${date.getDate()}</div></th>;
});
gridHeader.appendChild(headerRow);
```

```
const today = new Date();
```

```
today.setHours(0, 0, 0, 0);
```

```
habits.forEach(habit, habitIndex) => {
  const row = document.createElement('tr');
  row.innerHTML = <td class="habit-name">${habit.name}</td>;
  dates.forEach(date => {
    const dateString = formatDate(date);
    const isCompleted = habit.completed.includes(dateString);
```

```
const isFuture = date > today;

let circleClass = 'status-circle';
if (isCompleted) circleClass += ' completed';
if (isFuture) circleClass += ' future';

row.innerHTML += <td><span class="${circleClass}" data-habit-index="${habitIndex}" data-date="${dateString}"></span></td>;
});

gridBody.appendChild(row);
});

};

const updateStreakInfo = () => {
if (selectedHabitIndex === null || habits.length === 0) {
streakPlaceholder.classList.remove('hidden');
streakDisplay.classList.add('hidden');
return;
}

streakPlaceholder.classList.add('hidden');
streakDisplay.classList.remove('hidden');

const habit = habits[selectedHabitIndex];
const completedDates = habit.completed.map(d => new Date(d + "T00:00:00")).sort((a, b) => a - b);

if (completedDates.length === 0) {
currentStreakEl.textContent = '0';
bestStreakEl.textContent = '0';
}
```

```
    return;
}

let bestStreak = 0;
let currentStreak = 0;
let tempStreak = 1;

for (let i = 0; i < completedDates.length; i++) {
    if (i > 0) {
        const diff = (completedDates[i] - completedDates[i-1]) / (1000 * 60 * 60 * 24);
        if (diff === 1) {
            tempStreak++;
        } else {
            bestStreak = Math.max(bestStreak, tempStreak);
            tempStreak = 1;
        }
    }
    bestStreak = Math.max(bestStreak, tempStreak);

// Calculate current streak
const today = new Date();
today.setHours(0,0,0,0);
const yesterday = new Date(today);
yesterday.setDate(yesterday.getDate() - 1);

const lastCompletedDate = completedDates[completedDates.length - 1];
if (lastCompletedDate.getTime() === today.getTime() ||
lastCompletedDate.getTime() === yesterday.getTime()) {
```

```
let streak = 0;

let dateToCheck = new Date(lastCompletedDate);

let dateSet = new Set(habit.completed);

while(dateSet.has(formatDate(dateToCheck))){  
    streak++;  
    dateToCheck.setDate(dateToCheck.getDate() - 1);  
}  
  
currentStreakEl.textContent = currentStreak;  
bestStreakEl.textContent = bestStreak;  
};  
  
// --- EVENT HANDLERS ---  
  
const handleAddHabit = () => {  
    const habitName = newHabitInput.value.trim();  
    if (habitName) {  
        habits.push({ name: habitName, completed: [] });  
        newHabitInput.value = '';  
        saveHabits();  
        renderApp();  
    }  
};  
  
const handleHabitListClick = (e) => {  
    const targetLi = e.target.closest('li');
```

```
if (!targetLi) return;

if (e.target.classList.contains('delete-btn')) {
    const index = e.target.dataset.index;
    if (confirm(Are you sure you want to delete "${habits[index].name}"?)) {
        habits.splice(index, 1);
        if (selectedHabitIndex === index) selectedHabitIndex = null;
        else if (selectedHabitIndex > index) selectedHabitIndex--;
        saveHabits();
        renderApp();
    }
} else {
    selectedHabitIndex = parseInt(targetLi.dataset.index);
    renderHabitList(); // Re-render to show selection
    updateStreakInfo();
}
};

const handleGridClick = (e) => {
    if (e.target.classList.contains('status-circle') &&
!e.target.classList.contains('future')) {
        const habitIndex = e.target.dataset.habitIndex;
        const dateString = e.target.dataset.date;

        const habit = habits[habitIndex];
        const completedIndex = habit.completed.indexOf(dateString);

        if (completedIndex > -1) {
            habit.completed.splice(completedIndex, 1);
        }
    }
};
```

```
    } else {
        habit.completed.push(dateString);
    }

    saveHabits();
    renderProgressGrid(); // Only re-render grid for performance
    if (parseInt(habitIndex) === selectedHabitIndex) {
        updateStreakInfo();
    }
};

const handleClearAll = () => {
    if(confirm('WARNING: This will delete ALL habits and progress. Are you sure?'))
    {
        localStorage.removeItem('habits-weekly');
        habits = [];
        selectedHabitIndex = null;
        renderApp();
    }
};

// --- EVENT LISTENERS ---
addHabitBtn.addEventListener('click', handleAddHabit);
newHabitInput.addEventListener('keydown', (e) => e.key === 'Enter' &&
handleAddHabit());
habitsList.addEventListener('click', handleHabitListClick);
gridBody.addEventListener('click', handleGridClick);
clearAllBtn.addEventListener('click', handleClearAll);
```

```
prevWeekBtn.addEventListener('click', () => {
    currentDate.setDate(currentDate.getDate() - 7);
    renderProgressGrid();
});

nextWeekBtn.addEventListener('click', () => {
    currentDate.setDate(currentDate.getDate() + 7);
    renderProgressGrid();
});

// --- INITIALIZATION ---
loadHabits();
renderApp();
});
```

Output:

DAILY HABIT TRACKER
Build your routine, track your progress.

MY HABITS

Exercise	
Read (20 mins)	
Drink Water (2L)	

Add a new habit...

Sep 22 - Sep 28, 2025

	Mon 22	Tue 23	Wed 24	Thu 25	Fri 26	Sat 27	Sun 28
HABIT	Exercise						
Exercise							
Read (20 mins)							
Drink Water (2L)							

Streak Counter
Current: 1 days Best: 1 days

Powered by HTML, CSS, JavaScript

Conclusion

The Daily Habit Tracker project demonstrates how front-end web technologies can be effectively utilized to build a practical and user-friendly productivity tool. By combining HTML, CSS, Bootstrap, JavaScript, and jQuery, the system provides an interactive platform for users to create, monitor, and maintain daily habits with ease. Features such as streak counters, progress indicators, and visual trackers enhance motivation and encourage consistency, while localStorage ensures data persistence without the need for a backend. The project not only helps users in developing discipline and self-improvement but also highlights the importance of intuitive design and responsive interfaces in web applications. Overall, the Habit Tracker stands as a simple yet impactful solution for routine management and lays the groundwork for future enhancements like notifications, cloud sync, and advanced analytics.

References

- L&T LMS: <https://learn.lntedutech.com/Landing/MyCourse>