



# To-Do List Application

## Phase 5

**COLLEGE CODE** : 9506  
**COLLEGE NAME** : Einstein College of Engineering  
**DEPARTMENT** : B.E COMPUTER SCIENCE  
**STUDENT NM-ID** : 569499d7d464f30a31f49877e45750c8  
**ROLL NUMBER** : 9506231040104  
**DATE** : 10/10/2025

**Submitted By,**

**NAME** : M.THIRUMALAIKUMAR  
**MOBILE NUMBER** : 9361350749

# Phase 5

## Project Demonstration & Documentation

### Final Demo Walkthrough

#### Objective:

Showcase the complete working application with all features implemented in Phase 4.

#### Demo Steps:

Open the application in a browser.

- Double-click on index.html.
- The app will open in your **default browser** (Chrome, Firefox, Edge, etc.).

Demonstrate **adding tasks** with date, time, and priority.

### Welcome Buddy !!

### Welcome Buddy !!

Show **editing tasks inline** using double-click.

☐

2025-10-09 19:53 • Medium

Mark tasks as **completed**, and use **filters** (All, Active, Completed).

AllActiveCompleted

☒ To remain to night shit work at 10:00 pm

📅 2025-10-09 ⌚ 21:45 • Medium 🗑

2 active • 1 complete • 3 all • Clear completed

Drag & drop tasks to reorder.

Add a task...

☐ hello everyone

☐ add

☐ add

Show **search functionality** in action.

To

🌙

Welcome Buddy !!

Add a task...

dd-mm-yyyy 📅

--:-- -- ⌚

Medium ▾

Add

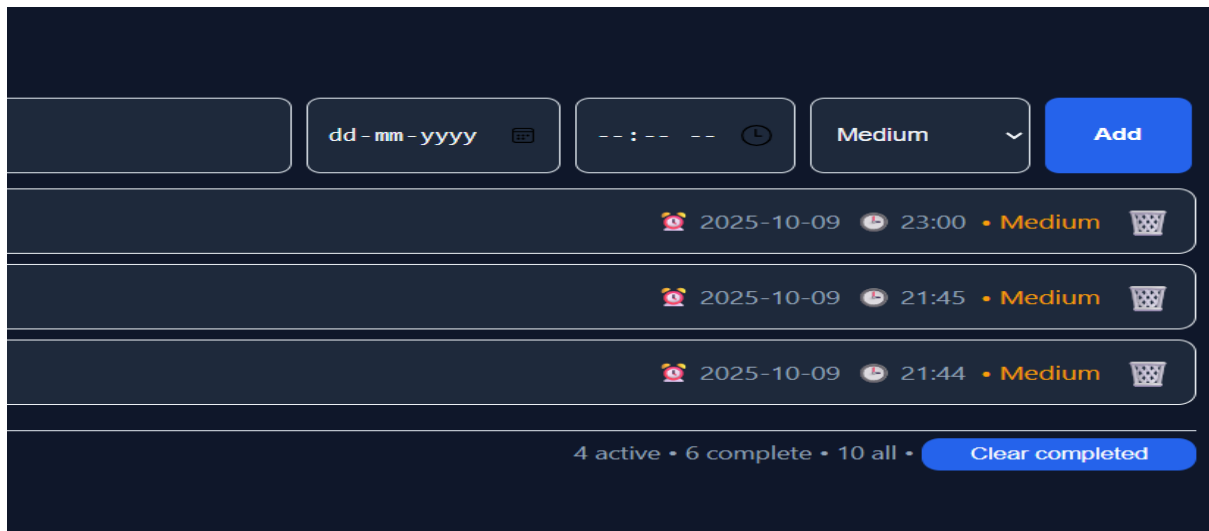
☐ Today night 11:00 meeting for manager

📅 2025-10-09 ⌚ 23:00 • Medium 🗑

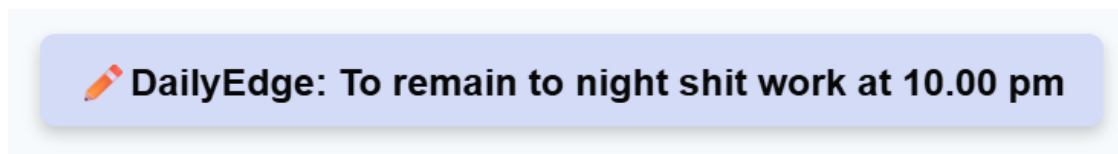
☐ To remain to night shit work at 10.00 pm

📅 2025-10-09 ⌚ 21:45 • Medium 🗑

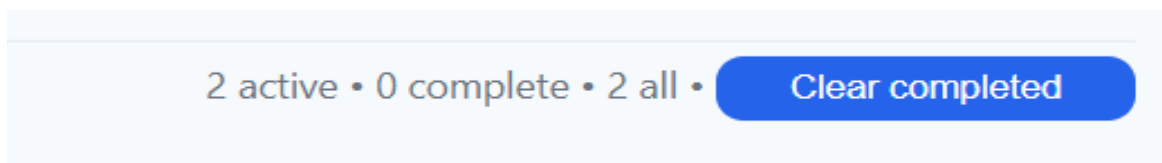
Toggle **dark mode**.



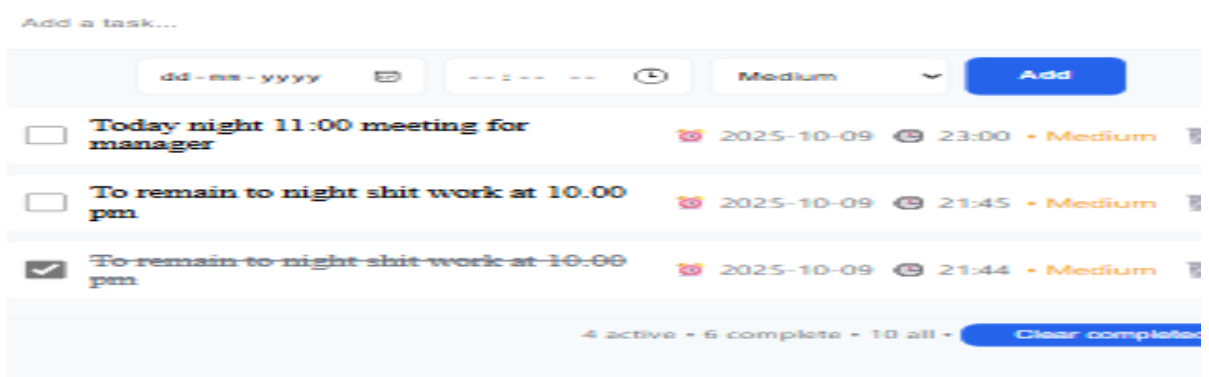
Demonstrate **reminder notifications** with sound alerts.



Clear completed tasks and observe count updates.



Showcase responsiveness on **desktop, tablet, and mobile**.



# Project Report

## Introduction:

The **DailyEdge To-Do List** application is designed to help users efficiently manage their daily tasks and stay organized. Unlike a simple checklist, DailyEdge offers advanced features such as task prioritization, due dates, reminders with sound alerts, inline editing, and drag-and-drop task reordering.

The primary purpose of the application is to provide a **user-friendly, interactive, and responsive platform** where users can:

- **Add, edit, and delete tasks** effortlessly.
- **Prioritize tasks** based on urgency (High, Medium, Low).
- **Set due dates and times** and receive timely reminders.
- **Search and filter tasks** for better task management.
- **Maintain task order and completion status**, even after closing the browser, through localStorage persistence.
- **Toggle between light and dark themes** for better visual comfort.

The application emphasizes **usability, accessibility, and performance**, making it suitable for both personal and professional task management. DailyEdge aims to enhance productivity while providing a smooth and engaging user experience.

- **Phase 4 Enhancements:** Additional features, UI/UX improvements, performance & security checks.
- **Implementation Details:** HTML, CSS, and JavaScript structure; localStorage usage.
- **Conclusion:** Summary of features and future scope.

# Implementation Details:

## HTML

```
<body>
  <main class="app-container">
    <header class="app-header">
      <div class="header-left">
        <h1 class="app-title"> DailyEdge</h1>
      </div>
      <div class="header-center">
        <input id="searchBox" class="search-input" placeholder="Search tasks..." aria-label="Search tasks" />
      </div>
      <div class="header-right">
        <button id="themeToggle" class="btn-theme" aria-label="Toggle Dark Mode">🌙</button>
      </div>
    </header>

    <div class="main-content">
      <nav class="sidebar">
        <button class="filter-btn" data-filter="all" id="filterAll">All
        <i class="fa-solid fa-bars" style="margin-left: 258px; color: black;"></i>
      </button>
        <button class="filter-btn" data-filter="active" id="filterActive">Active
        <i class="fa-solid fa-eye" style="margin-left: 220px;"></i></button>
        <button class="filter-btn" data-filter="completed" id="filterCompleted">Completed
        <i class="fa-solid fa-circle-check" style="margin-left: 178px;"></i></button>
      </nav>
    </div>

    <div class="content">
      <div class="welcome-container">
        <h1 class="welcome">Welcome Buddy !!</h1>
      </div>

      <div class="main-content2">

        <section class="task-area">
          <section class="task-input-area" id="taskInputArea">
```

# CSS

```
201 .task-input {
202   padding: 10px;
203   border-radius: 8px;
204   border: 1px solid #e6edf3;
205   min-width: 120px;
206   font-size: 14px;
207   color: inherit;
208   background: var(--card);
209   width: 630px;
210   height: 60px;
211
212 }
213 .task-date,
214 .task-time,
215 .task-priority {
216   padding: 10px;
217   border-radius: 8px;
218   border: 1px solid #e6edf3;
219   min-width: 120px;
220   font-size: 14px;
221   color: inherit;
222   background: var(--card);
223 }
224
225 .task-time {
226   max-width: 130px;
227 }
228
229 .btn-add {
230   padding: 10px 16px;
231   border-radius: 8px;
232   border: none;
233   background: var(--accent);
234   color: #fff;
235   cursor: pointer;
236   transition: background-color 0.3s ease;
237   min-width: 80px;
```

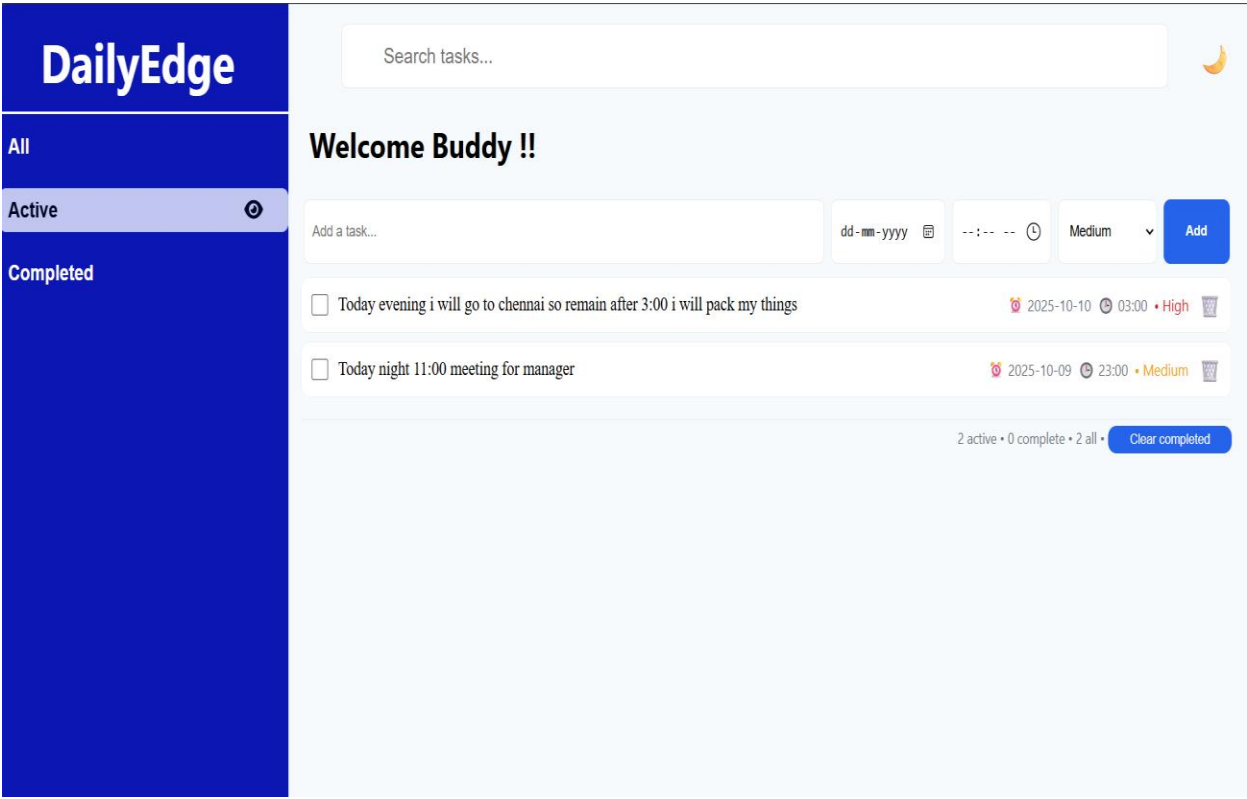
# JS

```
JS script2.js > ...
114 function renderTasks() {
131   shown.forEach(task => {
132     const li = document.createElement('li');
133     li.className = 'task-item';
134     li.draggable = true;
135     li.dataset.id = task.id;
136
137     // Checkbox
138     const cb = document.createElement('input');
139     cb.type = 'checkbox';
140     cb.checked = task.completed;
141     cb.className = 'task-checkbox';
142     cb.setAttribute('aria-label', `Mark task "${task.text}" completed`);
143     cb.addEventListener('click', () => toggleTaskComplete(task.id));
144
145     // Task Text
146     const span = document.createElement('div');
147     span.className = 'task-text' + (task.completed ? ' completed' : '');
148     span.textContent = task.text;
149
150     // Inline Edit
151     span.addEventListener('dblclick', () => {
152       const input = document.createElement('input');
153       input.type = 'text';
154       input.value = task.text;
155       input.className = 'task-text';
156       input.setAttribute('aria-label', 'Edit task text');
157
158       input.addEventListener('keydown', e => {
159         if (e.key === 'Enter') {
160           if (input.value.trim()) editTask(task.id, input.value);
161         } else if (e.key === 'Escape') {
162           renderTasks();
163         }
164       });
165     });
```

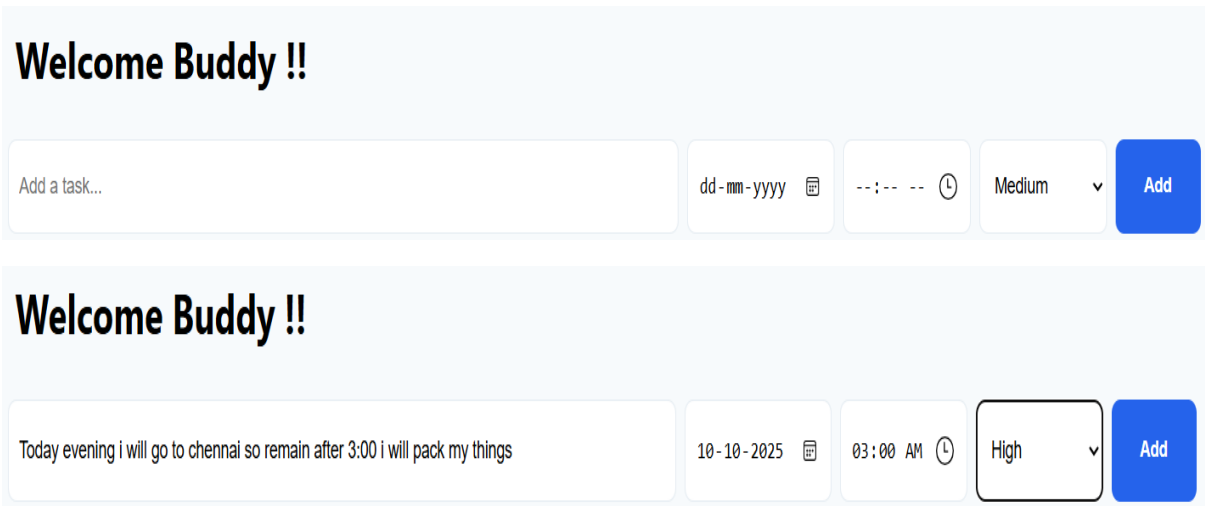


# Screenshots / API Documentation

## Welcome Screen / Dashboard




## Adding a Task






## Task Completed & Filtered View

All

Active

Completed 

☒ To remain to night shift work at 10.00 pm

 2025-10-09  21:45 • Medium 

2 active • 1 complete • 3 all • Clear completed

## Inline Editing

☐

to remain tomorrow nan Mudhalvan session


xxxxxxxxxx

 2025-10-09  19:53 • Medium 


## Dark Mode View

DailyEdge

All


Active 


Completed




Welcome Buddy !!

Add a task...




dd-mm-yyyy 

--:-- -- 




Medium 

Add

☐ Today evening i will go to chennai so remain after 3:00 i will pack my things

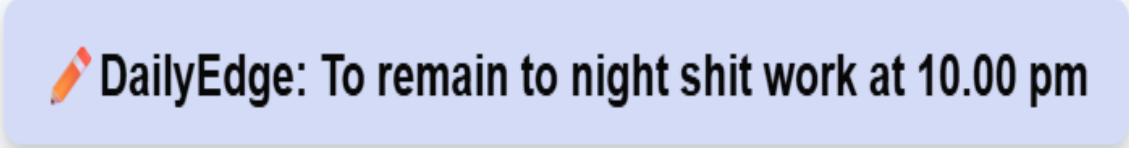
 2025-10-10  03:00 • High 


☐ Today night 11:00 meeting for manager

 2025-10-09  23:00 • Medium 

2 active • 0 complete • 2 all • Clear completed

## Reminder Alert Popup

A light blue rounded rectangular popup with a subtle drop shadow. On the left is a small orange pencil icon. To its right, the text "DailyEdge: To remain to night shit work at 10.00 pm" is written in a bold, black, sans-serif font.

 **DailyEdge: To remain to night shit work at 10.00 pm**

## API Documentation:

*If API is used in future enhancement, currently localStorage functions as “pseudo-API”*

### addTask()

```
function addTask() {  
  if (!taskInput.value.trim()) return;  
  tasks.unshift(createTask(taskInput.value, taskDate.value, taskTime.value, taskPriority.value));  
  taskInput.value = '';  
  taskDate.value = '';  
  taskTime.value = '';  
  taskPriority.value = 'medium';  
  saveTasks();  
  renderTasks();  
}
```

### editTask()

```
function editTask(id, newText) {  
  tasks = tasks.map(t => t.id === id ? { ...t, text: newText } : t);  
  saveTasks();  
  renderTasks();  
}
```

## deleteTask()

```
function deleteTask(id) {  
  tasks = tasks.filter(t => t.id !== id);  
  saveTasks();  
  renderTasks();  
}
```

# Challenges & Solutions

## Challenges:

- Implementing drag & drop correctly with task reordering.
- Managing reminders for tasks with date & time.
- Maintaining localStorage persistence with live updates.
- Ensuring responsive UI for various screen sizes.

## Solutions:

- Used dragstart, dragover, drop events for drag & drop.
- Periodic check with setInterval for reminders.
- Tasks array stored and updated in localStorage.
- Flexbox and CSS media queries for responsive design.

# GitHub README & Setup Guide

## Case-1

### Clone and Open Locally

1. Clone the repository:
2. `git clone <your-repo-link>`
3. Navigate to the project folder.
4. Open `index.html` in your preferred browser (Chrome, Firefox, Edge, etc.).

## Case-2

### Using VS Code Live Server

1. Open the project folder in VS Code.
2. Right-click `index.html` → Open with Live Server.
3. The app will open in a browser with live reload support.

## Case-3

### Deploy Online (Optional)

1. Deploy on Netlify or Vercel for easy access.
2. Drag & drop the project folder (or connect GitHub repo) to deploy.
3. Share the live link in your README.

# Final Submission

The final submission ensures that all project files are organized, functional, and accessible for evaluation. This includes both the **source code repository** and the **deployed live application**.

## GitHub Repository

### Contents:

- **index.html** → Main HTML file for the application.
- **styles.css** → All styles including responsive layouts, dark mode, and animations.
- **script.js** → All JavaScript logic including task management, inline editing, drag & drop, reminders, search & filter, and theme toggle.
- **README.md** → Detailed project documentation, installation instructions, usage guide, screenshots, and contributor info.

## Deployment Link

### Netlify or Vercel:

- The project is deployed on a cloud platform for easy access
- Users can interact with the **live application** without downloading files.
- Ensures all features (adding/editing/deleting tasks, reminders, drag & drop, filters, dark mode) are fully functional.

## Expected Behavior:

- Opening the deployed link should display the **DailyEdge Dashboard** immediately.
- All buttons, task interactions, and notifications should work exactly as in the local version.
- Responsive design ensures proper display on desktop, tablet, and mobile devices.

## Verification Before Submission

### Check that:

- All tasks can be added, edited, deleted, and filtered correctly.
- Drag & drop reordering is functional.
- Reminders appear with **custom notifications and sound**.
- Dark mode toggle works without affecting usability.
- LocalStorage properly saves tasks, so refreshing the page retains data.
- Search bar filters tasks accurately in real time.
- Ensures that the submission meets the project requirements and **provides a seamless user experience**.

## Submission Checklist

- GitHub repository is complete with all project files.
- README file contains installation, usage instructions, and screenshots.
- Application deployed on **Netlify/Vercel**.
- All features tested and working correctly.
- Screenshots included in the report or README for reference.