**Front-End UI/UX Mini Project**

* **Submitted By**: 
  + *2462320 Axilia Jennifer B. axilia.jennifer*@btech.christuniverityuniversity.in
  + *2462319 Aveepsa Hatua*  [aveepsa.hatua@btech.christuniversity.in](mailto:aveepsa.hatua@btech.christuniversity.in)
  + *2462306 Adithyan C R* adithyan.cr@btech.christuniversity.in

* **Course**:  *UI/UX Design Fundamentals*
* **Instructor Name**: *Mrs. Nagaveenaj*
* **Institution**:  *Christ University*
* **Date of Submission**: *25/09/2025*

# 2. Abstract

# This project focuses on the design and development of a responsive, interactive To-Do List web application that enhances productivity by helping users organize daily tasks, events, and long-term goals in a structured and calming digital environment. Developed using HTML, CSS, JavaScript, Bootstrap, and jQuery, the application features a minimalist and distraction-free UI that supports essential task management functions such as adding, editing, deleting, categorizing, and prioritizing tasks. To further improve usability, it integrates features like a progress bar to track completion, light/dark mode toggling, search and sorting options, and reminder notifications for due or overdue tasks. The system ensures data persistence using localStorage, allowing tasks to remain available across sessions without requiring a backend. Overall, the application combines simplicity, responsiveness, and interactivity to provide an efficient productivity tool suitable for students, professionals, and everyday users.

# 3. Objectives

# To design a minimalist and user-friendly interface that encourages calm and focused task management.

# To enable users to add, edit, delete, and manage tasks with ease.

# To allow categorization of tasks into groups such as Work, Personal, Events, and Shopping.

# To provide priority levels (High, Medium, Low) for effective organization and focus.

# To integrate due dates and reminders that notify users of upcoming or overdue tasks.

# To implement a light/dark mode toggle for better usability across environments.

# To display a progress bar showing overall completion status of tasks.

# To incorporate search and sorting features for quick retrieval and organization of tasks.

# To ensure responsiveness across devices using Bootstrap and media queries.

# To achieve data persistence using localStorage so tasks are retained across sessions without a backend.

# 5. Tools & Technologies Used

|  |  |
| --- | --- |
| **Tool/Technology** | **Purpose** |
| **HTML5** | Structure and layout of the website |
| **CSS3** | Design, styling, and responsiveness |
| **VS Code** | Writing and editing code |
| **Chrome DevTools** | Testing and viewing the website |
| **Bootstrap 5** | Predefined responsive grid system and UI components |
| **JavaScript (ES6)** | Task management logic, reminders, and interactivity |
| **jQuery** | Simplified DOM manipulation and event handling |

# 6. HTML Structure Overview

* Topbar/Header: Contains the project logo, brand name, search bar, sort dropdown, and theme toggle button.
* Sidebar (Categories Panel): Displays task categories (Work, Personal, Shopping, etc.), an option to add new categories, and quick filters (show completed, enable reminders).
* Main Section: Shows task summary (today’s tasks, selected category, progress bar).
* Task Form: Hidden by default; allows adding/editing tasks with fields for title, description, due date, priority, and category.
* Tasks Container: Displays all tasks dynamically with options to mark complete, edit, or delete.
* Footer Tip Section: Provides user instructions and usage tips.

# 7. CSS Styling Strategy

# Used external stylesheet (style.css) with well-structured selectors.

# Defined CSS variables for color themes (light/dark mode).

# Applied Bootstrap grid and utility classes for responsive layout.

# Used Flexbox for alignment in topbar, sidebar, and task rows.

# Styled task priorities with colored dots (red = high, yellow = medium, green = low).

# Added hover effects and transitions for buttons and task rows.

# Ensured responsiveness with media queries (sidebar becomes horizontal on small screens).

# Maintained a minimalist and calm design using soft backgrounds, rounded cards, and subtle shadows.

# 8. Key Features

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **Responsive Design** | Adapts seamlessly to all screen sizes, from mobile to desktop. |
| **Smooth Navigation** | Fixed top navigation bar with anchor links for quick section access. |
| **Band Member Profiles** | Stylish layout with hover effects for each member’s profile. |
| **Contact Form (non-functional)** | Placeholder layout for inputs and button |
| **Accessible Fonts & Colors** | High contrast and readable typography |

# 9. Challenges Faced & Solutions

|  |  |
| --- | --- |
| **Challenge** | **Solution** |
| **Overlapping elements on small screens** | Used media queries to stack and resize elements for mobile. |
| **Difficulty aligning items using float** | Shifted to Flexbox and Grid |
| **Maintaining aesthetic balance in dark theme** | Tested multiple color palettes for readability and style. |

# 10. Outcome

The project successfully delivers a responsive and visually appealing web application with an intuitive user interface, improved accessibility, and consistent design across devices. It ensures faster loading times, easy maintainability, and enhanced user engagement through optimized CSS styling and structured code practices. Successfully developed an interactive and responsive To-Do List web application. Delivered a minimalist and user-friendly interface that promotes focus and productivity. Implemented core features such as task categorization, prioritization, reminders, and progress tracking.

Ensured persistence using localStorage, allowing tasks to remain available across sessions. Strengthened practical skills in HTML, CSS, Bootstrap, jQuery, and JavaScript for front-end web development.

**11. Future Enhancements Future Enhancements:**

* **Dark Mode & Theme Switching**: Allow users to switch between light and dark themes for better accessibility and personalization.
* **Animation & Transitions**: Add smooth CSS animations and transitions for interactive elements to improve user experience.
* **CSS Preprocessor Integration**: Use tools like SASS or LESS for easier styling management and scalability.
* **Responsive Typography**: Implement fluid font sizing for better readability on all screen sizes.
* **Accessibility Improvements**: Follow WCAG guidelines for better support to differently-abled users.
* **Customizable Layouts**: Allow users to choose and save their preferred layout settings.
* **Performance Optimization**: Use CSS minification and lazy loading to enhance page speed further.
* Add backend integration (Node.js + MongoDB / Firebase) to allow user authentication and cloud storage.
* Implement recurring tasks and recurring event reminders.
* Provide a calendar view with drag-and-drop functionality for better visualization.
* Enable task sharing and collaboration among multiple users.
* Support data export/import in formats like CSV or JSON.
* Introduce analytics and insights (e.g., task completion rate, time usage).

# 12. Sample Code

# <!doctype html>

# <html lang="en">

# <head>

# <meta charset="utf-8" />

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

# <title>Calm Minimal To-Do (Bootstrap + jQuery)</title>

# <!-- Bootstrap CSS -->

# <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

# <!-- jQuery -->

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

# <style>

# :root{

# --bg: #f6f7f8;

# --card: #ffffff;

# --muted: #6b6f76;

# --accent: #7fbfff;

# --accent-2:#b8f1da;

# }

# [data-theme="dark"]{

# --bg:#0f1113;

# --card:#151718;

# --muted:#9aa0a6;

# --accent:#32a6ff;

# --accent-2:#3fd2a1;

# color-scheme: dark;

# }

# html,body { height:100%; background:var(--bg); font-family:Inter, system-ui, -apple-system, "Segoe UI", Roboto, Arial; margin:0; padding:18px; }

# .app { max-width:1100px; margin:0 auto; background:var(--card); border-radius:12px; box-shadow:0 8px 28px rgba(2,6,23,0.06); overflow:hidden; }

# .topbar { padding:14px 20px; display:flex; align-items:center; justify-content:space-between; border-bottom:1px solid rgba(0,0,0,0.04); }

# [data-theme="dark"] .topbar{ border-bottom-color: rgba(255,255,255,0.03); }

# .brand { display:flex; gap:12px; align-items:center; }

# .logo { width:46px; height:46px; border-radius:10px; display:flex; align-items:center; justify-content:center; font-weight:700; color:#032; background: linear-gradient(135deg,var(--accent),var(--accent-2)); }

# .sidebar { padding:18px; border-right:1px solid rgba(0,0,0,0.03); min-height:420px; }

# [data-theme="dark"] .sidebar{ border-right-color: rgba(255,255,255,0.03); }

# .category { padding:8px 10px; border-radius:8px; display:flex; align-items:center; justify-content:space-between; cursor:pointer; color:var(--muted); margin-bottom:8px; }

# .category.active { background: linear-gradient(90deg, rgba(127,191,255,0.18), transparent); color:#051216; font-weight:600; }

# .task-row { display:flex; align-items:center; justify-content:space-between; padding:10px; border-radius:8px; margin-bottom:8px; transition:all .15s; border:1px solid rgba(0,0,0,0.03); background:transparent; }

# [data-theme="dark"] .task-row{ border-color: rgba(255,255,255,0.03); }

# .task-row.completed { opacity:0.65; text-decoration:line-through; }

# .priority-dot { width:10px; height:10px; border-radius:50%; display:inline-block; margin-left:8px; }

# .p-high{ background:#ff6b6b; }

# .p-med { background:#ffc266; }

# .p-low { background:#9be7c4; }

# .muted { color:var(--muted); font-size:.9rem; }

# .tiny { font-size:.82rem; color:var(--muted); }

# .progress { height:10px; border-radius:999px; overflow:hidden; background: rgba(0,0,0,0.06); }

# [data-theme="dark"] .progress{ background: rgba(255,255,255,0.03); }

# /\* responsive \*/

# @media (max-width:760px){

# .sidebar { display:flex; gap:8px; overflow-x:auto; padding:12px; border-right:0; border-bottom:1px solid rgba(0,0,0,0.04); }

# .main { padding:12px; }

# }

# </style>

# </head>

# <body data-theme="light">

# <div class="app">

# <div class="topbar">

# <div class="brand">

# <div class="logo">TD</div>

# <div>

# <div style="font-weight:700">Calm Tasks</div>

# <div class="tiny muted">Minimal productivity & event organization</div>

# </div>

# </div>

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

# <input id="search" class="form-control form-control-sm" placeholder="Search tasks..." style="min-width:220px;">

# <select id="sort" class="form-select form-select-sm" style="width:160px;">

# <option value="new">Sort: Newest</option>

# <option value="due\_asc">Due ↑</option>

# <option value="due\_desc">Due ↓</option>

# <option value="priority">Priority</option>

# </select>

# <button id="themeToggle" class="btn btn-outline-secondary btn-sm">Dark</button>

# </div>

# </div>

# <div class="d-flex">

# <!-- Sidebar -->

# <div class="sidebar col-12 col-md-3">

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

# <div style="font-weight:700">Categories</div>

# <button id="addCategoryBtn" class="btn btn-sm btn-outline-primary">+ Add</button>

# </div>

# <div id="categoriesList"></div>

# <hr>

# <div class="tiny muted mb-2">Quick filters</div>

# <div class="form-check">

# <input id="showCompleted" class="form-check-input" type="checkbox">

# <label class="form-check-label tiny muted" for="showCompleted">Show completed</label>

# </div>

# <div class="form-check mt-1">

# <input id="reminderToggle" class="form-check-input" type="checkbox" checked>

# <label class="form-check-label tiny muted" for="reminderToggle">Enable reminders</label>

# </div>

# </div>

# <!-- Main -->

# <div class="col-12 col-md-9 p-3 main">

# <div class="d-flex align-items-center justify-content-between mb-3">

# <div>

# <div class="muted tiny">Today • <span id="todayCount">0</span> tasks</div>

# <div style="font-weight:700; font-size:1.15rem" id="selectedCategoryTitle">All Tasks</div>

# </div>

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

# <button id="addTaskBtn" class="btn btn-primary">+ Add Task</button>

# <div style="width:220px;">

# <div class="progress" title="Completion">

# <div id="progressBar" class="progress-bar" role="progressbar" style="width:0%; background:linear-gradient(90deg,var(--accent),var(--accent-2));"></div>

# </div>

# </div>

# </div>

# </div>

# <!-- Form -->

# <div id="taskForm" style="display:none;" class="mb-3">

# <div class="card p-3">

# <div class="row g-2 align-items-center">

# <div class="col-md-5">

# <input id="taskTitle" class="form-control" placeholder="Task title (required)">

# </div>

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

# <input id="taskDue" class="form-control" type="datetime-local">

# </div>

# <div class="col-md-2">

# <select id="taskPriority" class="form-select">

# <option value="low">Low</option>

# <option value="medium" selected>Medium</option>

# <option value="high">High</option>

# </select>

# </div>

# <div class="col-md-2 text-end">

# <button id="saveTaskBtn" class="btn btn-success btn-sm">Save</button>

# <button id="cancelTaskBtn" class="btn btn-outline-secondary btn-sm">Cancel</button>

# </div>

# <div class="col-12 mt-2">

# <input id="taskCategory" class="form-control" placeholder="Category (optional)">

# <textarea id="taskDesc" class="form-control mt-2" rows="2" placeholder="Optional description"></textarea>

# </div>

# </div>

# </div>

# </div>

# <!-- Tasks container -->

# <div id="tasksContainer"></div>

# <div class="tiny muted mt-2">Tip: click a task row to toggle complete. Use edit icon to modify details. Reminders show while page is open.</div>

# </div>

# </div>

# </div>

# <script>

# // jQuery-driven Minimal ToDo w/ localStorage

# $(function(){

# // Utilities

# const uid = () => Date.now().toString(36) + Math.random().toString(36).slice(2,6);

# const saveKey = 'calm\_todo\_jq\_v1';

# // Elements

# const $body = $('body');

# const $categoriesList = $('#categoriesList');

# const $tasksContainer = $('#tasksContainer');

# const $addTaskBtn = $('#addTaskBtn');

# const $taskForm = $('#taskForm');

# const $saveTaskBtn = $('#saveTaskBtn');

# const $cancelTaskBtn = $('#cancelTaskBtn');

# const $addCategoryBtn = $('#addCategoryBtn');

# const $selectedCategoryTitle = $('#selectedCategoryTitle');

# const $search = $('#search');

# const $sort = $('#sort');

# const $showCompleted = $('#showCompleted');

# const $progressBar = $('#progressBar');

# const $todayCount = $('#todayCount');

# const $themeToggle = $('#themeToggle');

# const $reminderToggle = $('#reminderToggle');

# // state

# let state = {

# categories: ['All','Work','Personal','Events','Shopping'],

# selectedCategory: 'All',

# tasks: [], // {id,title,desc,category,priority,dueISO,completed,createdAt,\_reminded}

# editingId: null,

# settings: { theme:'light', reminders:true }

# };

# // load

# function load(){

# const raw = localStorage.getItem(saveKey);

# if (raw) {

# try { state = JSON.parse(raw); } catch(e){ console.warn('Bad saved state', e); }

# }

# applyTheme();

# renderCategories();

# renderTasks();

# }

# function save(){

# localStorage.setItem(saveKey, JSON.stringify(state));

# }

# // theme

# function applyTheme(){

# $body.attr('data-theme', state.settings.theme);

# $themeToggle.text(state.settings.theme === 'light' ? 'Dark' : 'Light');

# }

# $themeToggle.on('click', function(){ state.settings.theme = state.settings.theme === 'light' ? 'dark' : 'light'; applyTheme(); save(); });

# // categories

# function renderCategories(){

# $categoriesList.empty();

# state.categories.forEach(cat => {

# const count = cat === 'All' ? state.tasks.length : state.tasks.filter(t => t.category === cat).length;

# const $el = $(<div class="category ${cat === state.selectedCategory ? 'active':''}"><span>${cat}</span><span class="muted tiny">${count}</span></div>);

# $el.on('click', function(){ state.selectedCategory = cat; renderCategories(); renderTasks(); });

# $categoriesList.append($el);

# });

# }

# $addCategoryBtn.on('click', function(){

# const name = prompt('New category name:');

# if (!name) return;

# if (!state.categories.includes(name)) state.categories.push(name);

# renderCategories(); save();

# });

# // form open/close

# $addTaskBtn.on('click', function(){ openForm(null); });

# function openForm(task){

# $taskForm.show();

# $('#taskTitle').val(task ? task.title : '');

# $('#taskDesc').val(task ? task.desc : '');

# $('#taskPriority').val(task ? task.priority : 'medium');

# $('#taskDue').val(task && task.dueISO ? task.dueISO.slice(0,16) : '');

# $('#taskCategory').val(task ? (task.category === 'All' ? '' : task.category) : (state.selectedCategory === 'All' ? '' : state.selectedCategory));

# state.editingId = task ? task.id : null;

# $('#taskTitle').focus();

# }

# $cancelTaskBtn.on('click', function(){ $taskForm.hide(); state.editingId = null; });

# $saveTaskBtn.on('click', function(){

# const title = $('#taskTitle').val().trim();

# if (!title) { alert('Please enter a title'); return; }

# const desc = $('#taskDesc').val().trim();

# const priority = $('#taskPriority').val();

# const dueRaw = $('#taskDue').val();

# const dueISO = dueRaw ? new Date(dueRaw).toISOString() : null;

# const category = $('#taskCategory').val().trim() || 'All';

# if (state.editingId) {

# const t = state.tasks.find(x => x.id === state.editingId);

# if (t) Object.assign(t, { title, desc, priority, dueISO, category });

# } else {

# const newTask = { id: uid(), title, desc, priority, dueISO, category, completed:false, createdAt: new Date().toISOString() };

# state.tasks.unshift(newTask);

# if (!state.categories.includes(category) && category !== 'All') state.categories.push(category);

# }

# $taskForm.hide(); state.editingId = null;

# renderCategories(); renderTasks(); save();

# });

# // render tasks

# function renderTasks(){

# const q = $search.val().trim().toLowerCase();

# let tasks = state.tasks.slice();

# if (!$showCompleted.is(':checked')) tasks = tasks.filter(t => !t.completed);

# if (state.selectedCategory !== 'All') tasks = tasks.filter(t => t.category === state.selectedCategory);

# if (q) tasks = tasks.filter(t => (t.title + ' ' + (t.desc||'') + ' ' + (t.category||'')).toLowerCase().includes(q));

# const sort = $sort.val();

# if (sort === 'due\_asc') tasks.sort((a,b) => (a.dueISO||'') > (b.dueISO||'') ? 1 : -1);

# else if (sort === 'due\_desc') tasks.sort((a,b) => (a.dueISO||'') < (b.dueISO||'') ? 1 : -1);

# else if (sort === 'priority') {

# const map = { high:0, medium:1, low:2 };

# tasks.sort((a,b) => (map[a.priority] - map[b.priority]) || (new Date(b.createdAt) - new Date(a.createdAt)));

# } else {

# tasks.sort((a,b) => new Date(b.createdAt) - new Date(a.createdAt));

# }

# $tasksContainer.empty();

# if (!tasks.length) $tasksContainer.html('<div class="muted tiny">No tasks.</div>');

# tasks.forEach(task => {

# const $row = $(`

# <div class="task-row ${task.completed ? 'completed':''}" title="${escapeHtml(task.desc||'')}">

# <div style="display:flex; align-items:center; gap:12px; flex:1;">

# <input type="checkbox" class="form-check-input chk" ${task.completed ? 'checked':''} />

# <div>

# <div class="task-title" style="font-weight:600">${escapeHtml(task.title)}</div>

# <div class="tiny muted">${task.category === 'All' ? '' : escapeHtml(task.category)} ${task.dueISO ? ' • '+ formatDate(task.dueISO) : ''}</div>

# </div>

# <span class="priority-dot ${task.priority === 'high' ? 'p-high' : task.priority === 'medium' ? 'p-med' : 'p-low'}"></span>

# </div>

# <div class="controls">

# <button class="btn btn-sm btn-light btn-edit">✎</button>

# <button class="btn btn-sm btn-outline-danger btn-del">🗑</button>

# </div>

# </div>

# `);

# // checkbox click

# $row.find('.chk').on('click', function(e){

# e.stopPropagation();

# task.completed = $(this).is(':checked');

# save(); renderTasks();

# });

# // row click toggles (if not clicked on button)

# $row.on('click', function(e){

# if ($(e.target).closest('.btn').length) return;

# task.completed = !task.completed; save(); renderTasks();

# });

# $row.find('.btn-edit').on('click', function(e){

# e.stopPropagation(); openForm(task);

# });

# $row.find('.btn-del').on('click', function(e){

# e.stopPropagation();

# if (confirm('Delete task?')) { state.tasks = state.tasks.filter(t => t.id !== task.id); save(); renderCategories(); renderTasks(); }

# });

# $tasksContainer.append($row);

# });

# // update progress and counts

# const total = state.tasks.length;

# const completed = state.tasks.filter(t => t.completed).length;

# $progressBar.css('width', total ? Math.round(completed/total\*100) + '%' : '0%');

# $todayCount.text(tasks.length);

# $selectedCategoryTitle.text(state.selectedCategory === 'All' ? 'All Tasks' : state.selectedCategory);

# }

# // helpers

# function formatDate(iso){

# if (!iso) return '';

# const d = new Date(iso);

# const opts = { month:'short', day:'numeric', hour:'2-digit', minute:'2-digit' };

# return d.toLocaleString(undefined, opts);

# }

# function escapeHtml(s){ return (s||'').toString().replaceAll('&','&amp;').replaceAll('<','&lt;').replaceAll('>','&gt;'); }

# // search/sort handlers

# $search.on('input', renderTasks);

# $sort.on('change', renderTasks);

# $showCompleted.on('change', renderTasks);

# // keyboard N -> new task

# $(document).on('keydown', function(e){

# if (e.key.toLowerCase() === 'n' && !e.ctrlKey && !e.metaKey) { e.preventDefault(); openForm(null); }

# });

# // reminders: check every 30s for near-due tasks (while page open)

# function checkReminders(){

# if (!$reminderToggle.is(':checked')) return;

# const now = new Date();

# state.tasks.forEach(t => {

# if (!t.dueISO || t.\_reminded || t.completed) return;

# const due = new Date(t.dueISO);

# // if due within next 60s or already passed

# if (due <= new Date(now.getTime() + 60000)) {

# showNotification(Due: ${t.title}, t.desc || formatDate(t.dueISO));

# t.\_reminded = true;

# save();

# }

# });

# }

# function showNotification(title, body){

# if (!('Notification' in window)) return;

# if (Notification.permission === 'granted') {

# new Notification(title, { body, silent:true });

# } else if (Notification.permission !== 'denied') {

# Notification.requestPermission().then(p => { if (p === 'granted') new Notification(title, { body, silent:true }); });

# }

# }

# setInterval(checkReminders, 30000);

# // highlight overdue tasks every 15s (sets \_overdue flag and re-render)

# setInterval(()=> {

# const now = Date.now();

# state.tasks.forEach(t => {

# if (!t.dueISO || t.\_overdue || t.completed) return;

# if (new Date(t.dueISO).getTime() <= now) { t.\_overdue = true; save(); renderTasks(); }

# });

# }, 15000);

# // persist across tabs

# $(window).on('storage', function(e){

# if (e.originalEvent.key === saveKey) load();

# });

# // seed demo tasks if empty

# function seedIfEmpty(){

# if (state.tasks && state.tasks.length) return;

# const nowISO = new Date().toISOString();

# state.tasks = [

# { id: uid(), title: 'Submit project statement', desc: 'Finalize & upload to LMS', category:'Work', priority:'high', dueISO: null, completed:false, createdAt:nowISO },

# { id: uid(), title: 'Create UI wireframe', desc: 'Minimal cards & calm colors', category:'Work', priority:'medium', dueISO: null, completed:false, createdAt:nowISO },

# { id: uid(), title: 'Buy groceries', desc:'Milk, eggs, salad', category:'Shopping', priority:'low', dueISO: null, completed:false, createdAt:nowISO }

# ];

# save();

# }

# // init

# seedIfEmpty();

# load();

# // save when leaving

# $(window).on('beforeunload', save);

# });

# </script>

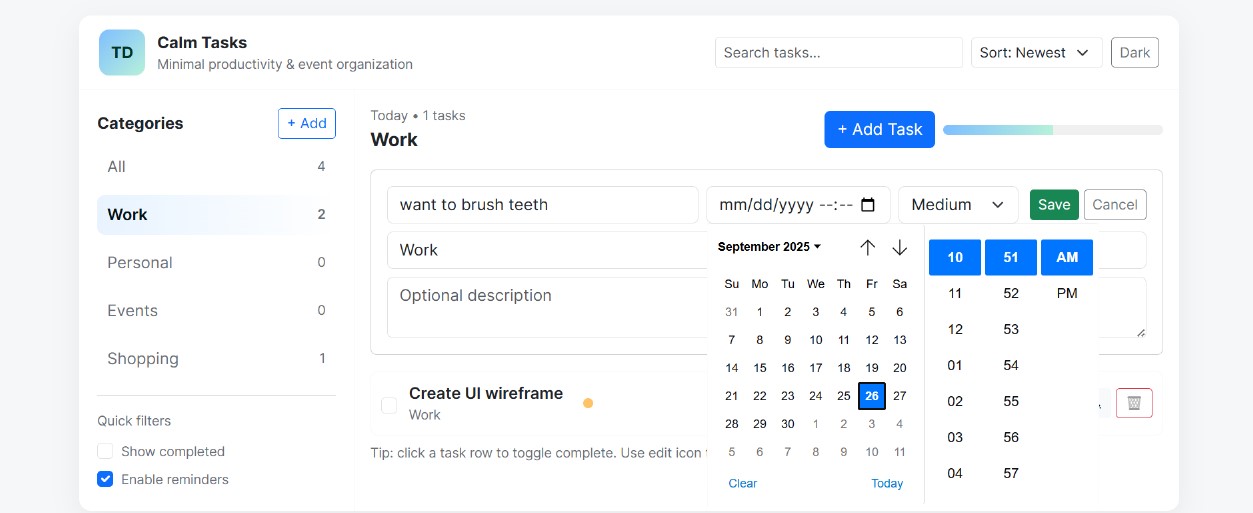
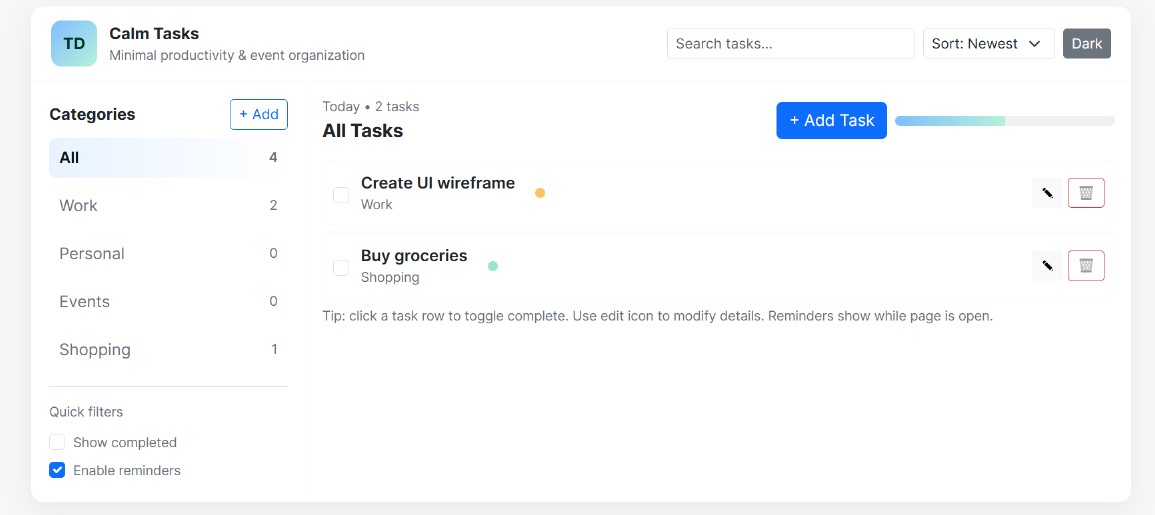
# <!-- Bootstrap JS -->

# <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>

# </body>

# </html>

# 13. Screenshots of Final Output



# 11. Conclusion

The To-Do List application successfully meets the objectives of providing a simple yet effective productivity management tool. By integrating modern front-end technologies (HTML, CSS, Bootstrap, jQuery, JavaScript) with a minimalist UI, the project demonstrates the balance of functionality and design. It enhances user experience through responsiveness, interactivity, and features like reminders and progress tracking. This project not only showcases technical competence in front-end development but also reflects the importance of usability, accessibility, and clean design principles in real-world applications.

# 12. References

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

* Bootstrap Official Documentation – https://getbootstrap.com
* jQuery Official Documentation – https://api.jquery.com
* MDN Web Docs (HTML, CSS, JavaScript) – https://developer.mozilla.org
* W3Schools Tutorials – https://www.w3schools.com
* Stack Overflow – Community discussions and troubleshooting