

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

### **TO-DO LIST APPLICATION- “ *TO-DO LIST* ”**

#### **SUBMITTED BY:**

1. Sandra Shaju  
2462143  
[sandra.shaju@btech.christuniversity.in](mailto:sandra.shaju@btech.christuniversity.in)
2. Rebecca Lenin Koshy  
2462134  
[rebecca.lenin@btech.christuniversity.in](mailto:rebecca.lenin@btech.christuniversity.in)
3. Nandhitha Sri V  
2462118  
[nandhitha.sri@btech.christuniversity.in](mailto:nandhitha.sri@btech.christuniversity.in)

**COURSE:** UI/UX Design Fundamentals

**INSTRUCTOR NAME:** Nagaveena

**INSTITUTION:** Christ University, Kengeri

**DATE OF SUBMISSION:** 25 SEPTEMBER 2025

# ABSTRACT

## Project Overview:

A To-Do List web application allowing users to manage tasks with categories and due dates, featuring a cute, pastel-themed interface.

## Key Goal:

Create an interactive, user-friendly task management tool with task categorization, due date tracking, and local storage persistence.

## Core Technologies:

- **HTML:** Structures the app with a form for task input, tabs for category navigation, and a list for task display.
- **CSS:** Styles the app with a pastel bunny-themed background, custom fonts, and responsive design for mobile compatibility.
- **JavaScript (jQuery):** Handles dynamic task addition, deletion, completion, and filtering, with localStorage for data persistence.
- **Bootstrap:** Provides responsive layout components (e.g., input groups, buttons, tabs) and consistent styling across devices.

## Basic interactivity:

Users can add tasks with categories and due dates, toggle task completion with confetti effects, filter tasks by category, and clear all tasks.

## Final Outcomes and Usefulness:

Delivers a functional, visually appealing task manager that helps users organize tasks efficiently, with persistent storage and mobile responsiveness.

## OBJECTIVES

1. **Enhance Task Management:** Provide a simple, intuitive tool for users to organize, categorize, and track tasks with due dates.
2. **Improve User Engagement:** Create a visually appealing, cute-themed interface with interactive elements like confetti to make task completion enjoyable.
3. **Ensure Data Persistence:** Implement `localStorage` to save tasks, allowing users to retain their data across sessions.
4. **Support Cross-Device Usability:** Design a responsive app using Bootstrap and custom CSS for seamless use on desktop and mobile devices.

## SCOPE

### 1. Website Structure:

- A single-page web application with a form for task input, category tabs, and a task list display.
- Utilizes HTML for structured layout, including input fields, buttons, and navigation tabs.
- Integrates Bootstrap for responsive grid and component organization.

### 2. Design:

- Features a pastel-themed UI with a bunny-patterned background and custom fonts (Kalam, Patrick Hand).
- Employs CSS for styling, animations (e.g., hover effects), and responsive adjustments for mobile devices.
- Uses Bootstrap for consistent button and tab styling, enhanced with custom colors.

### 3. Exclusions:

- No backend server or database; relies on `localStorage` for data persistence.
- Excludes user authentication, multi-user support, or cloud synchronization.
- No advanced task features like recurring tasks or priority levels.

4. Deliverables:

- A functional single-page To-Do List app with task management (add, delete, complete, filter).
- Responsive HTML/CSS/JavaScript code with Bootstrap and jQuery integration.
- Persistent task storage using localStorage and a celebratory confetti effect for task completion.

TOOLS AND TECHNOLOGIES USED

TOOL/TECHNOLOGY	PURPOSE
HTML	Structures the app's layout, including task input form, category tabs, and task list.
CSS	Styles the interface with a pastel theme, custom fonts, and responsive design.
VS CODE	Code editor
JavaScript (jQuery)	Manages task operations (add, delete, complete, filter) and localStorage.
Bootstrap	Provides responsive components (grids, buttons, tabs) for consistent styling.
Chrome DevTools	Testing and debugging

# HTML Structure Overview

## Main Structure

- `<!DOCTYPE html>`: Declares HTML5 document type.
- `<html lang="en">`: Root element with English language attribute.
- `<head>`: Contains metadata and external resources.
  - `<meta charset="UTF-8">`: Sets character encoding to UTF-8.
  - `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Ensures responsive design.
  - `<title>`: Sets page title to "To-Do List App".
  - `<link>`: Imports Bootstrap CSS, Google Fonts (Kalam, Patrick Hand), and custom style.css.
  - `<script>`: Includes jQuery for JavaScript functionality.
- `<body>`: Main content container.
  - `<main>`: Wraps primary content for semantic structure.
  - `<div class="todo-container">`: Centers and styles the main app container.

## Main Content

- `<h2>`: Displays "To-Do List" heading with centered styling.
- `<form id="taskForm">`: Form for adding new tasks.
  - `<div class="input-group mb-3">`: Bootstrap input group for task input, category, and date.
    - `<label for="taskInput" class="visually-hidden">`: Hidden label for accessibility.
    - `<input type="text" id="taskInput">`: Text input for task description.
    - `<select id="taskCategory">`: Dropdown for task categories (Personal, Work, Entertainment).

- `<input type="date" id="taskDueDate">`: Date picker for task due date.
- `<button id="addTaskBtn">`: Button to submit new task.
- `<button id="clearTasksBtn">`: Button to clear all tasks.
- `<ul class="nav nav-tabs" id="categoryTabs">`: Bootstrap navigation tabs for filtering tasks.
  - `<li class="nav-item">`: List items for each category tab (All, Personal, Work, Entertainment).
  - `<a class="nav-link">`: Links for category filtering with data-category attributes.
- `<ul class="list-group mt-3" id="taskList">`: Container for dynamic task list items.
- `<div class="alert alert-info" id="noTasksMessage">`: Hidden alert for empty task lists.
- `<script>`: Imports Bootstrap JS, confetti library, and custom script.js.

## CSS Styling Strategy

- **Theme and Aesthetics:** Applies a pastel, bunny-themed background with custom fonts (Kalam, Patrick Hand) for a cute, engaging look.
- **Responsive Design:** Uses media queries and fluid layouts to ensure mobile compatibility, adjusting padding and element sizes.
- **Interactive Effects:** Implements hover effects (e.g., scale, shadow) and transitions for smooth visual feedback on task items and buttons.
- **Custom Overrides:** Enhances Bootstrap components with vibrant pink/cyan colors and rounded borders for consistency.

## jQuery Strategy

- **Dynamic Interactions:** Handles task addition, deletion, completion, and category filtering via event listeners (e.g., form submit, click).
- **Data Management:** Uses localStorage for task persistence and filters out invalid inputs (e.g., "hello").
- **UI Updates:** Dynamically renders task lists based on category selection and triggers confetti for completed tasks.

## Bootstrap Strategy

- **Layout Components:** Utilizes input groups, nav tabs, and list groups for structured, responsive task input and display.
- **Styling Consistency:** Leverages Bootstrap’s CSS for buttons and alerts, customized with pastel colors to match the theme.
- **Responsive Utilities:** Ensures cross-device compatibility with Bootstrap’s grid system and responsive classes.

## KEY FEATURES

FEATURES	DESCRIPTION
Task Management:	Add, delete, and mark tasks as complete with category selection (Personal, Work, Entertainment) and due date tracking.
Category Filtering:	Navigate tasks via tabs to view all tasks or filter by specific categories.

<b>Local Storage:</b>	Persists tasks across sessions using browser localStorage for data retention.
<b>Interactive UI:</b>	Features confetti animation on task completion, hover effects, and a responsive, pastel-themed design.

## CHALLENGES FACED AND SOLUTIONS:

CHALLENGE	SOLUTION
Ensuring tasks persist across sessions without a backend.	Used localStorage to save and retrieve tasks as JSON, ensuring data retention.
Creating a responsive design for mobile and desktop.	Leveraged Bootstrap's grid and responsive utilities, with custom CSS media queries for adjustments.
Preventing invalid inputs (e.g., "hello").	Added input validation in jQuery to filter out specific words and empty submissions.
Enhancing user engagement with interactive elements.	Integrated confetti animations for task completion and hover effects for visual feedback.



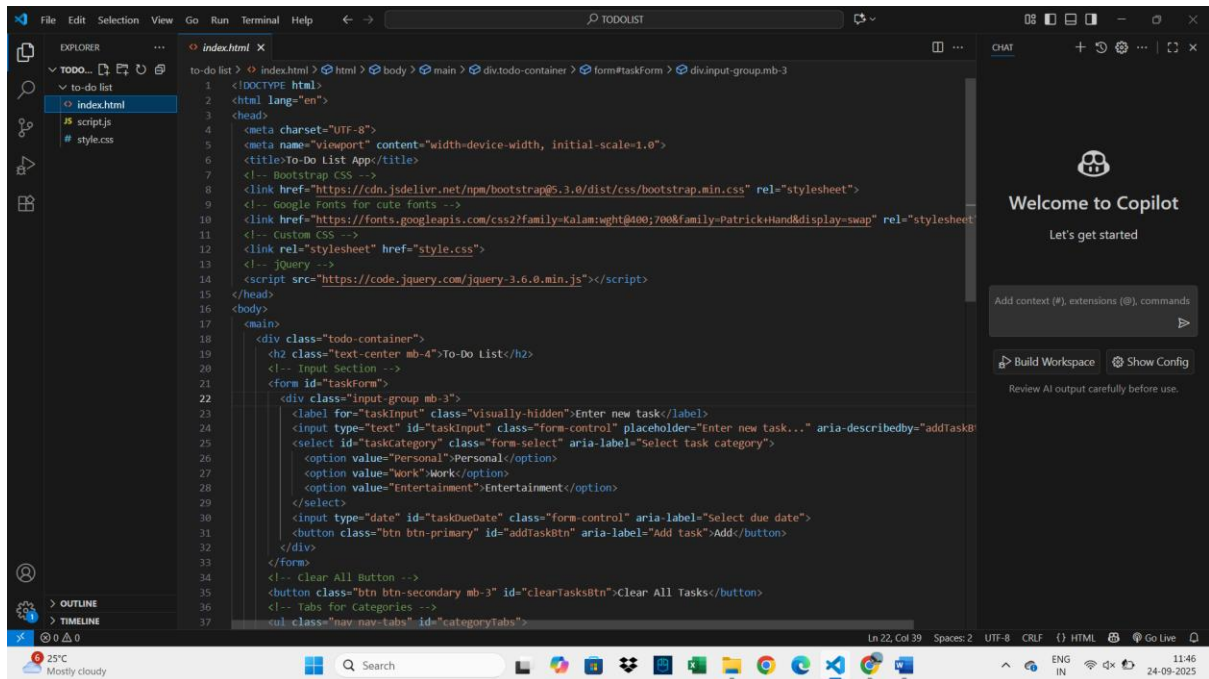
## OUTCOMES:

- **Functional Task Management:** Users can efficiently add, categorize, complete, and delete tasks with due date tracking.
- **Engaging User Experience:** Pastel-themed interface with confetti animations and hover effects enhances user interaction.
- **Cross-Device Compatibility:** Responsive design ensures seamless use on desktop and mobile devices.
- **Data Persistence:** LocalStorage with JSON serialization ensures tasks are saved and accessible across sessions.

## FUTURE ENHANCEMENTS:

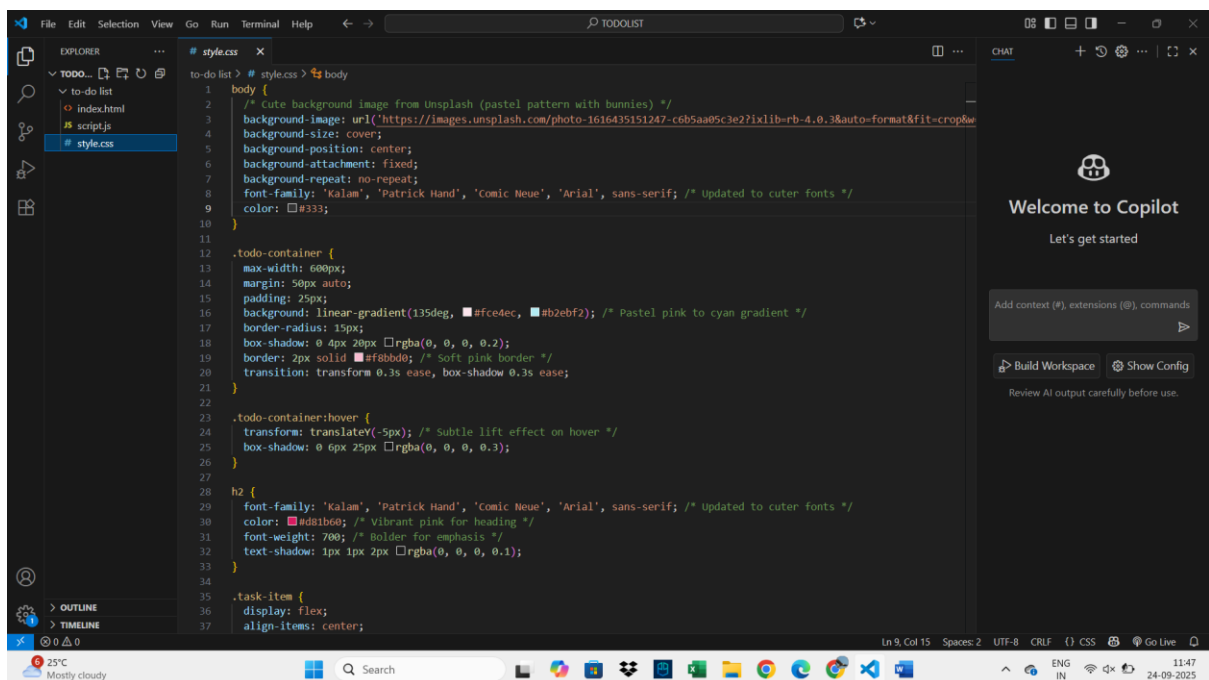
- **Task Prioritization:** Add priority levels (e.g., high, medium, low) to sort and highlight tasks.
- **Reminders/Notifications:** Implement browser notifications for upcoming or overdue task due dates.
- **Cloud Sync:** Integrate a backend API to sync tasks across devices using a database.
- **Task Editing:** Allow users to edit existing tasks directly within the list.

## SAMPLE CODE:



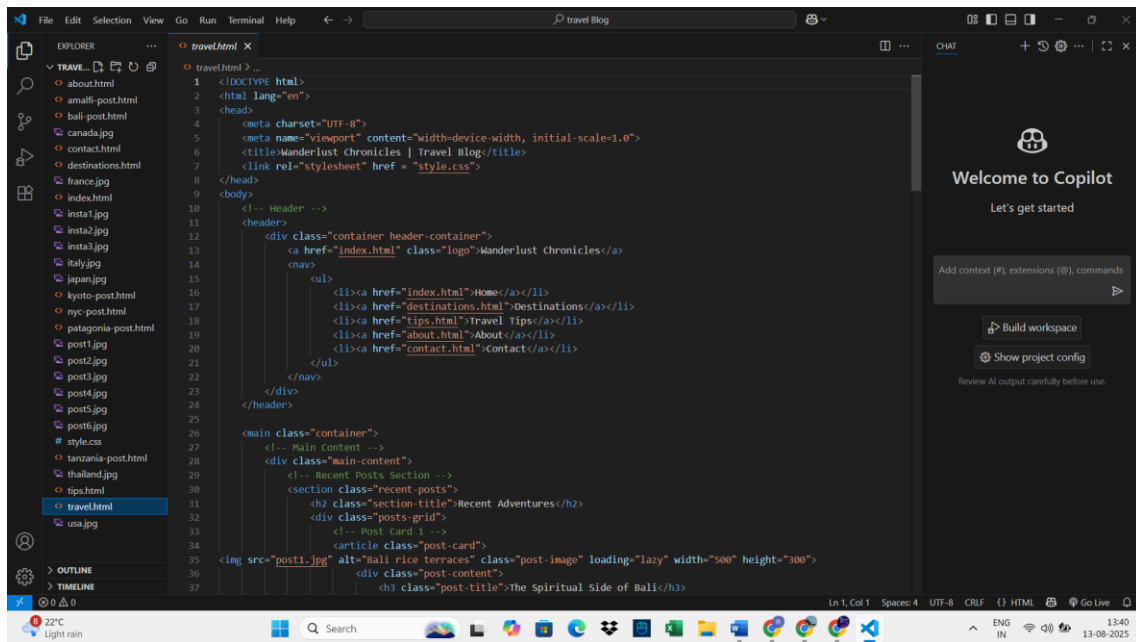
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>To-Do List App</title>
7 <!-- Bootstrap CSS -->
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
9 <!-- Google Fonts for cute fonts -->
10 <link href="https://fonts.googleapis.com/css2?family=Kalam:wght@400;700&family=PatrickHand&display=swap" rel="stylesheet">
11 <!-- Custom CSS -->
12 <link rel="stylesheet" href="style.css">
13 <!-- jQuery -->
14 <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
15 </head>
16 <body>
17 <!-- Main Content -->
18 <div class="todo-container">
19 <h2 class="text-center mb-4">To-Do List</h2>
20 <!-- Input Section -->
21 <form id="taskForm">
22 <div class="input-group mb-3">
23 <input type="text" id="taskInput" class="form-control" placeholder="Enter new task..." aria-describedby="addTaskBtn">
24 <input type="text" id="taskCategory" class="form-select" placeholder="Select task category">
25 <input type="text" id="taskDueDate" class="form-control" placeholder="Select due date">
26 <button class="btn btn-primary" id="addTaskBtn" aria-label="Add task">Add</button>
27 </div>
28 <div class="form">
29 <!-- Clear All Button -->
30 <button class="btn btn-secondary mb-3" id="clearTasksBtn">Clear All Tasks</button>
31 <!-- Tabs for Categories -->
32 <div class="nav nav-tabs" id="categoryTabs">
```

## SNIPPET FROM MAIN CODE



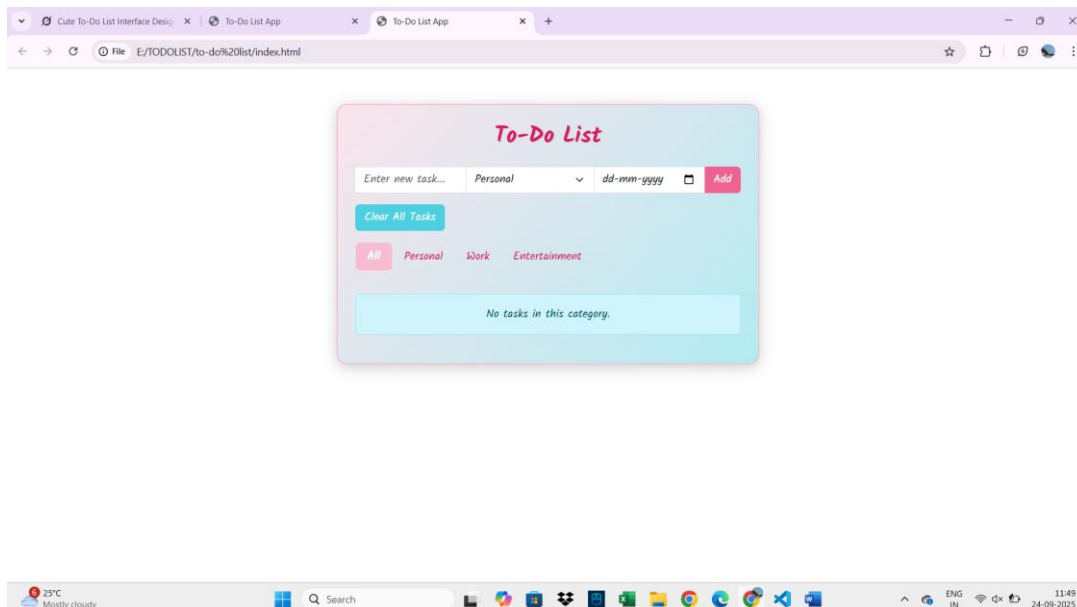
```
1 body {
2 /* Cute background image from Unsplash (pastel pattern with bunnies) */
3 background-image: url('https://images.unsplash.com/photo-1616435151247-c6b5aa05c3e2?ixlib=rb-4.0.3&auto=format&fit=crop&w=1200&q=80');
4 background-size: cover;
5 background-position: center;
6 background-attachment: fixed;
7 background-repeat: no-repeat;
8 font-family: 'Kalam', 'Patrick Hand', 'Comic Neue', 'Arial', sans-serif; /* Updated to cuter fonts */
9 color: #333;
10 }
11
12 .todo-container {
13 max-width: 600px;
14 margin: 50px auto;
15 padding: 25px;
16 background: linear-gradient(135deg, #fce4ec, #b2ebf2); /* Pastel pink to cyan gradient */
17 border-radius: 15px;
18 box-shadow: 0 4px 20px 0 rgba(0, 0, 0, 0.2);
19 border: 2px solid #f8bbd0; /* Soft pink border */
20 transition: transform 0.3s ease, box-shadow 0.3s ease;
21 }
22
23 .todo-container: hover {
24 transform: translateY(-5px); /* Subtle lift effect on hover */
25 box-shadow: 0 6px 25px 0 rgba(0, 0, 0, 0.3);
26 }
27
28 h2 {
29 font-family: 'Kalam', 'Patrick Hand', 'Comic Neue', 'Arial', sans-serif; /* Updated to cuter fonts */
30 color: #d81b60; /* Vibrant pink for heading */
31 font-weight: 700; /* Bolder for emphasis */
32 text-shadow: 1px 1px 2px 0 rgba(0, 0, 0, 0.1);
33 }
34
35 .task-item {
36 display: flex;
37 align-items: center;
```

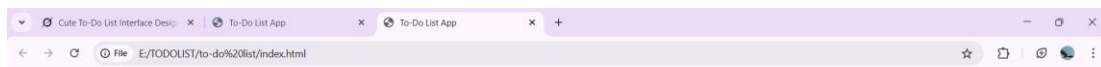
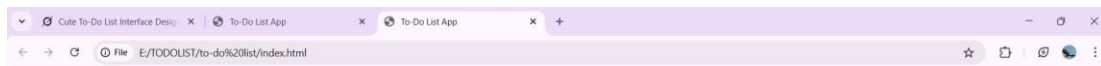
## SNIPPET OF THE CSS STYLESHEET



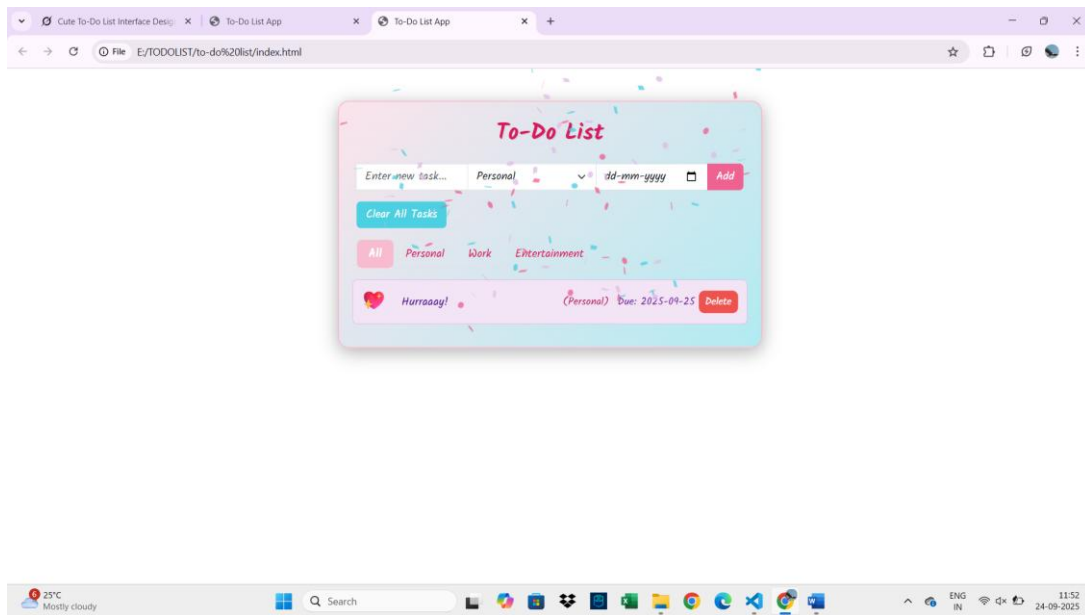
SNIPPET OF THE jQUERY CODE

OUTPUT:

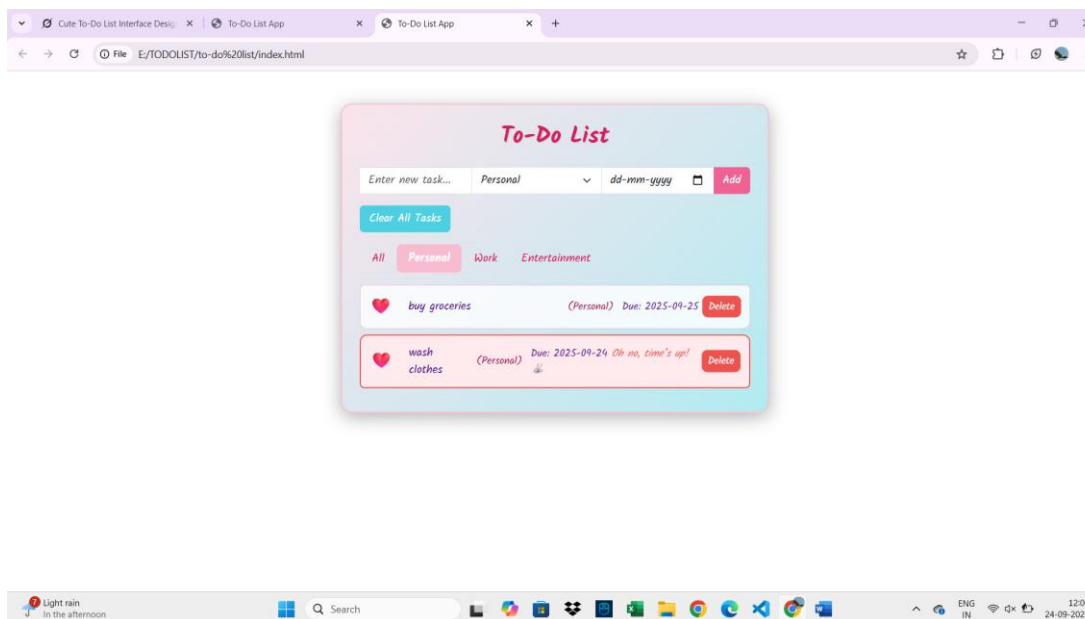




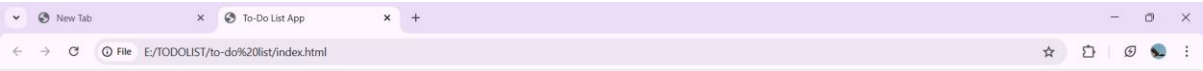
If the task is done then click on the heart and the following appears:



If the due date is today the:



If due date was yesterday:



### To-Do List

Personal dd-mm-yyyy Add

Clear All Tasks

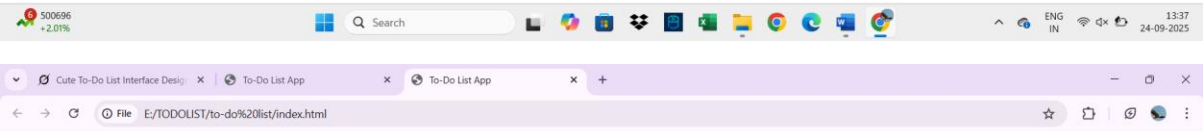
All Personal Work Entertainment

read books

(Personal)

Due: 2025-09-23 Oh no, time's up!

Delete



### To-Do List

Personal dd-mm-yyyy Add

Clear All Tasks

All Personal Work Entertainment

buy groceries

(Personal)

Due: 2025-09-25

Delete

wash clothes

(Personal)

Due: 2025-09-24 Oh no, time's up!

Delete

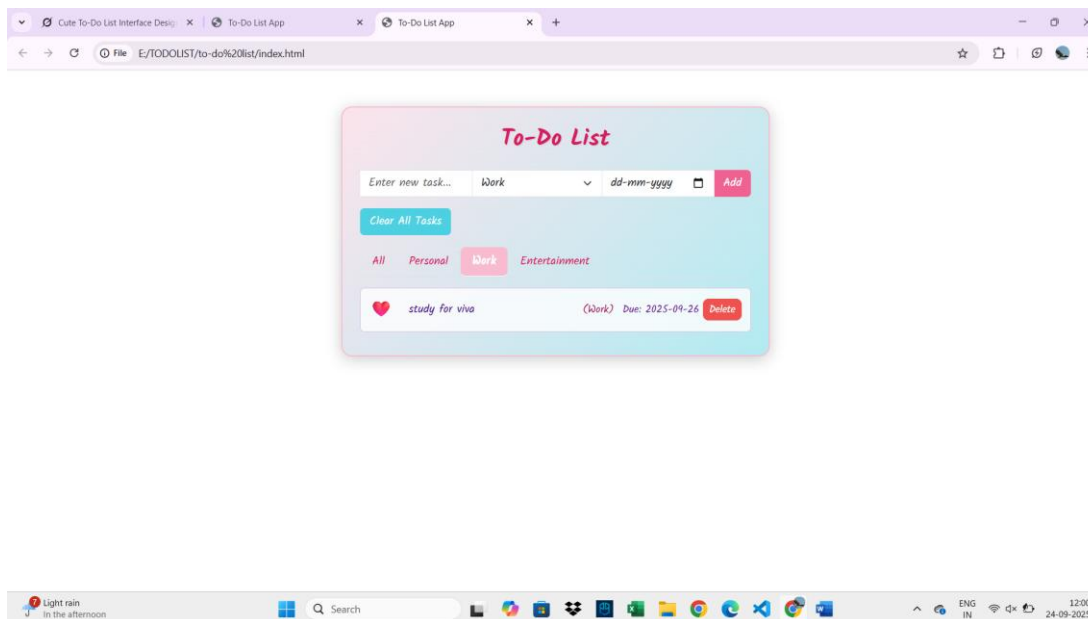
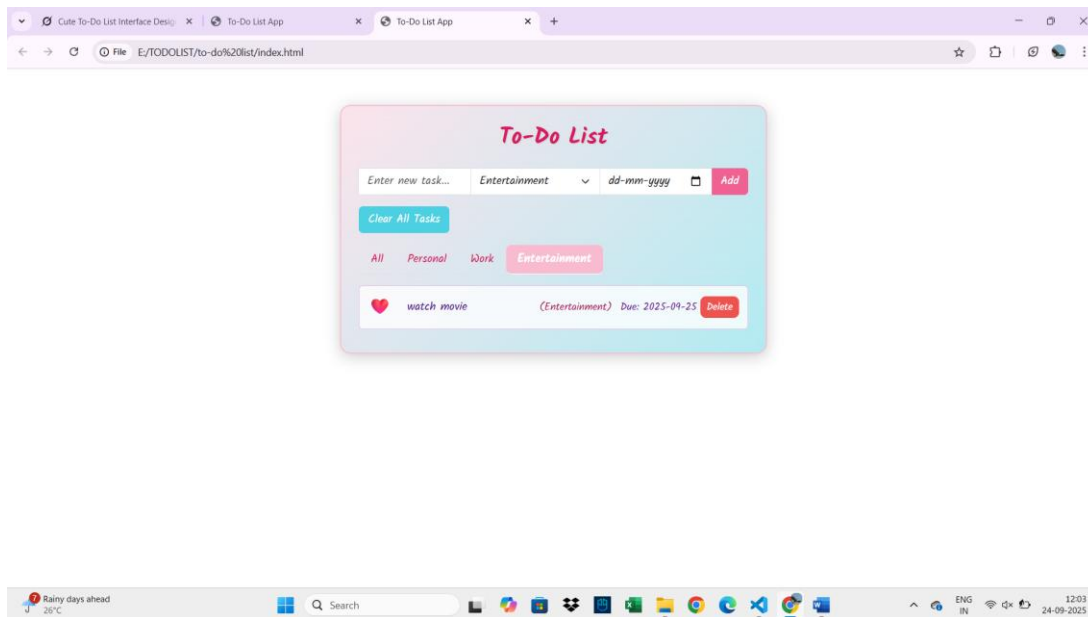
clean laptop

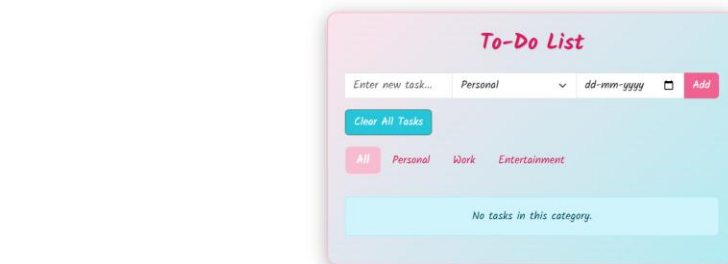
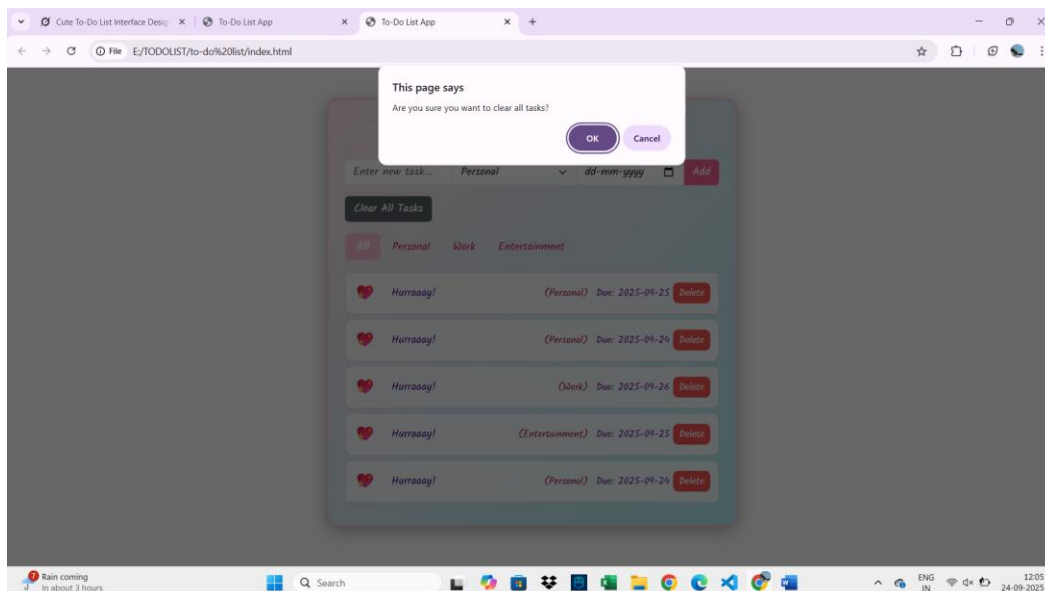
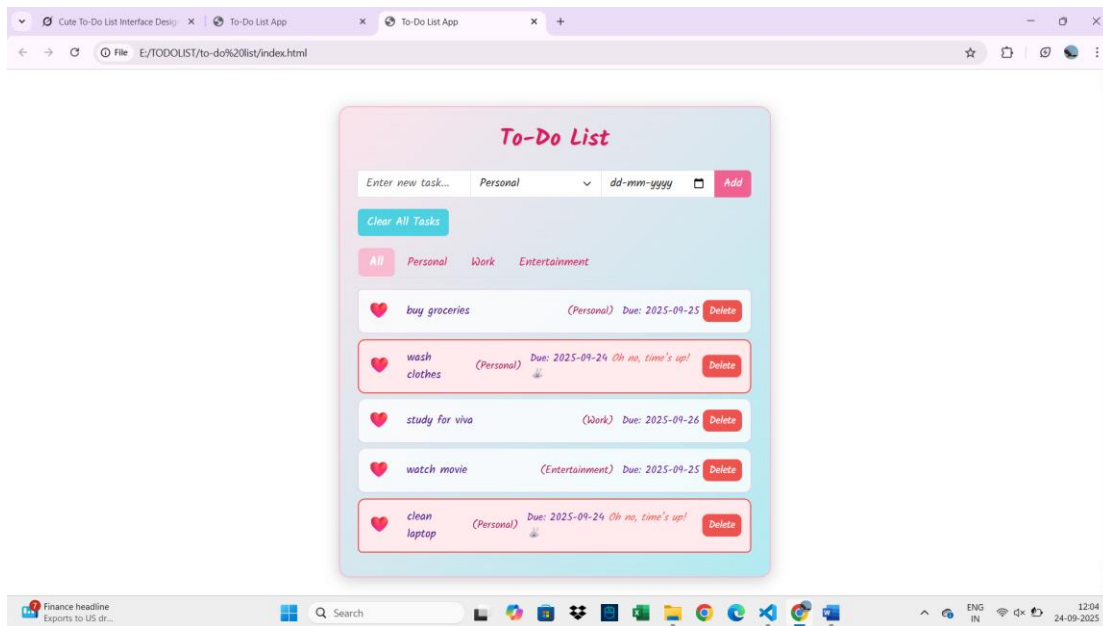
(Personal)

Due: 2025-09-24 Oh no, time's up!

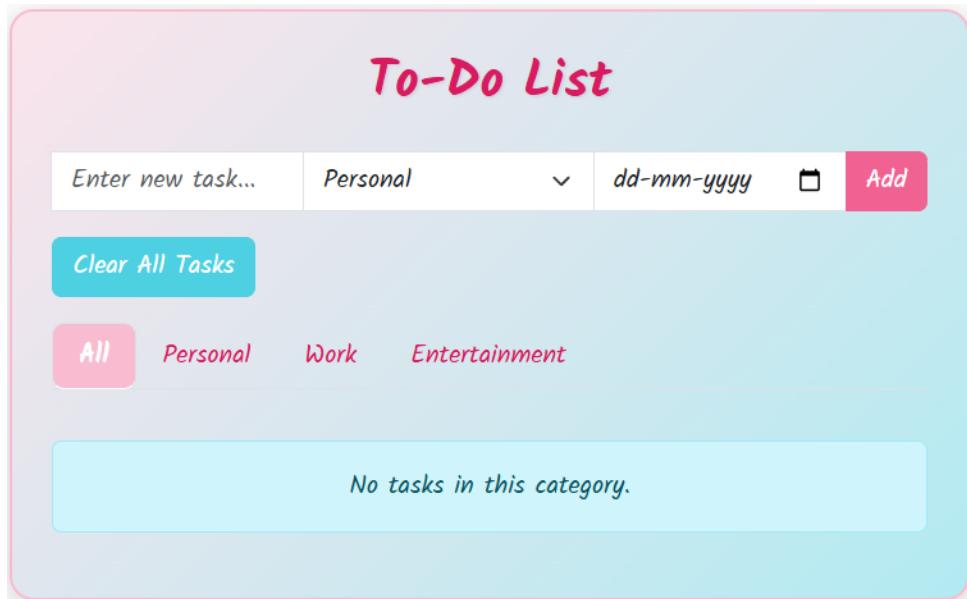
Delete











## CONCLUSION

The To-Do List app successfully delivers a user-friendly, visually appealing task management tool with a pastel-themed interface. Built with HTML, CSS, jQuery, and Bootstrap, it offers task categorization, due date tracking, and localStorage persistence, ensuring functionality and engagement across devices. Future enhancements like task prioritization and cloud sync could further elevate its utility.

## REFERENCE:

- HTML: [MDN Web Docs - HTML](#) - Covers HTML structure and semantics.
- CSS: [MDN Web Docs - CSS](#) - Guides on styling, responsive design, and animations.
- JavaScript/jQuery: [jQuery Documentation](#) - Official reference for jQuery methods and event handling.
- Bootstrap: [Bootstrap 5 Documentation](#) - Details on components like nav tabs and input groups.
- localStorage/JSON: [MDN Web Docs - Web Storage API](#) - Explains localStorage and JSON usage.
- Confetti Library: [Canvas Confetti GitHub](#) - Documentation for the confetti animation library.