**Front-End UI/UX**

**Mini Project Report**

**1. Title Page**

**Project Title:** To-Do List Application  
**Submitted By:**

* Name: Rhea Tess Payapilly , Pavana M X , Jeevitha A
* Roll Number: 2462137 , 2462178 , 2462079
* College Email ID: [Rhea.tess@btech.christuniversity.in](mailto:Rhea.tess@btech.christuniversity.in) , pavana.mx@btech.christuniversity.in , jeevitha.a@btech.christuniversity.in

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

**2. Abstract**

This project involves designing and developing a To-Do List Web Application using HTML, CSS, JavaScript, and jQuery. The goal is to help users organize daily tasks efficiently with options to add, mark as complete, and delete tasks. The application focuses on a clean and responsive UI, simple navigation, and interactive functionality. Core technologies like DOM manipulation and event handling were used to provide a seamless experience. The final outcome is a fully functional task manager that enhances productivity and can be extended with advanced features in the future.

**3. Objectives**

* Design a simple, user-friendly task management interface.
* Enable adding, editing, and deleting tasks dynamically.
* Use JavaScript/jQuery for interactive functionalities.
* Ensure responsive layout across devices.
* Enhance usability with task completion indicators.

**4. Scope of the Project**

* Includes: Front-end design and interactive features using HTML, CSS, JavaScript, and jQuery.
* Excludes: Backend storage and user authentication.
* Works on desktop, tablet, and mobile devices.
* Uses open-source technologies without third-party frameworks (except jQuery).

**5. Tools & Technologies Used**

| **Tool/Technology** | **Purpose** |
| --- | --- |
| HTML5 | Structure of the task manager |
| CSS3 | Styling and layout |
| JavaScript | Logic for task addition, deletion, and completion |
| jQuery | Simplified DOM manipulation and event handling |
| VS Code | Code editor |
| Chrome DevTools | Testing and debugging |

**6. HTML Structure Overview**

* <header> with application title.
* <main> section containing task input field, add button, and task list container.
* <ul> for listing tasks dynamically.
* <footer> with credits or instructions.

**7. CSS Styling Strategy**

* External stylesheet (style.css) for maintainability.
* Flexbox for input field and button alignment.
* Hover and active states for buttons.
* Media queries for responsive design.
* Consistent theme colors for branding.

**8.** J**avaScript and jQuery in the To-Do List Project**

JavaScript was used to bring **interactivity and functionality** to the To-Do List application. With JavaScript, the application is able to:

* **Add Tasks:** Capture user input and dynamically create list items.
* **Mark Tasks as Completed:** Toggle a CSS class to strike through tasks when clicked.
* **Delete Tasks:** Remove specific list items from the DOM.
* **Event Handling:** Handle user actions like button clicks and list item selections.

**jQuery in the To-Do List Project**

jQuery was used to **simplify and shorten JavaScript code**. Instead of writing long DOM methods, jQuery provided **easy-to-use functions** for event handling and DOM manipulation. In the project, jQuery was particularly useful for:

* Adding tasks with a single line of code.
* Toggling task completion with .toggleClass().
* Deleting tasks with .remove().
* Ensuring cross-browser compatibility.

**9. Key Features**

| **Feature** | **Description** |
| --- | --- |
| Task Addition | Users can add new tasks easily |
| Task Completion | Checkbox to mark tasks as done |
| Task Deletion | Remove tasks dynamically |
| Responsive Design | Works seamlessly across devices |
| Interactive Feedback | Hover effects and strike-through completed tasks |

**10. Challenges Faced & Solutions**

| **Challenge** | **Solution** |
| --- | --- |
| Maintaining dynamic task updates | Used jQuery’s .append() and .remove() methods |
| Handling completed tasks visually | Applied CSS class toggling with JavaScript |
| Layout breaking on smaller screens | Used Flexbox and responsive units |

**11. Outcome**

* Successfully developed a functional and interactive To-Do List application.
* Learned how to combine HTML, CSS, JavaScript, and jQuery for dynamic UI/UX.
* Improved understanding of DOM manipulation and responsive design.

**12. Future Enhancements**

* Store tasks using local storage or database.
* Add task categories and priority levels.
* Include a due-date calendar for tasks.
* Add user authentication for personalized lists.

**13. Sample Code**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Colorful To-Do List</title>

    <!-- Bootstrap CSS -->

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

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

</head>

<body class="bg-gradient">

<div class="container my-5">

    <h1 class="text-center mb-4 heading">To-Do List</h1>

    <form id="todo-form" class="row g-3 justify-content-center mb-5">

        <div class="col-md-5">

            <input type="text" id="task-input" class="form-control form-control-lg" placeholder="Enter task" required>

        </div>

        <div class="col-md-3">

            <input type="date" id="due-date" class="form-control form-control-lg" required>

        </div>

        <div class="col-md-2 d-grid">

            <button type="submit" class="btn btn-gradient btn-lg">Add Task</button>

        </div>

    </form>

    <div id="todo-list" class="row g-4">

        <!-- Task cards will appear here -->

    </div>

</div>

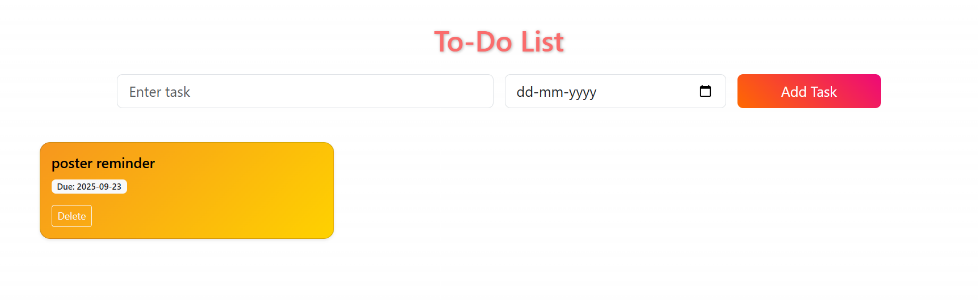
<!-- Script -->

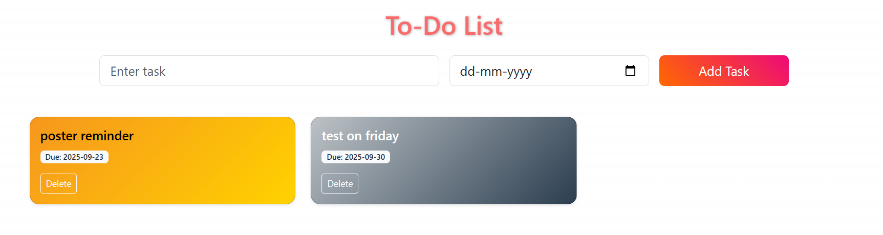
<script src="script.js"></script>

</body>

</html>

**13. Screenshots of Final Output**





**14. Conclusion**

The **To-Do List** project provided hands-on experience in building an interactive web application using front-end technologies. It enhanced my understanding of DOM manipulation, user-centered design, and responsive layouts. The project demonstrated how simple functionality can improve daily productivity while offering scope for future expansion.

**15. References**

* L&T EduTech LMS: <https://learn.lntedutech.com/Landing/MyCourse>
* W3Schools HTML & CSS Documentation: [https://www.w3schools.com](https://www.w3schools.com/)