

# Todo List Report

**Project Title:** To-Do List Web Application

**Submitted By:** Denil Biju-2460358

Diya Ann Dennis-2460361

Deon Benny-2460359

**Course:** UI/UX Design Fundamentals

**Instructor:** Dhiraj Alate

**Institution:** Christ University

**Date of Submission:** 26/08/2025

## Abstract

This project involves the design and development of a simple To-Do List web application using HTML, CSS (Bootstrap), jQuery, and JavaScript. The application allows users to add, edit, delete, and mark tasks as completed. It also includes filtering options to view all, active, or completed tasks. The main objective was to practice building an interactive front-end project with clean UI/UX principles, while learning how to manipulate the DOM dynamically using JavaScript and jQuery. The outcome is a responsive, lightweight, and

user-friendly task manager that can be used for daily productivity.

## Objectives

- Design a clean and user-friendly task management interface.
- Implement dynamic task handling: add, edit, delete, complete.
- Use filtering to manage active and completed tasks.
- Apply responsive UI principles with Bootstrap.
- Enhance interactivity with jQuery and JavaScript.

## Scope of the Project

- Focused on front-end functionality only.
- No database or backend integration.
- Works on desktops, tablets, and mobile devices.
- Uses open-source libraries (Bootstrap, jQuery).
- Built for productivity and UI/UX practice.

# Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Markup and structure
CSS3 / Bootstrap 5	Styling and layout
JavaScript / jQuery	Interactivity and DOM manipulation
VS Code	Code editor
Chrome DevTools	Testing and debugging

## HTML Structure Overview

- Semantic elements for layout.
- `<div>` containers styled with Bootstrap classes.
- Input field and button for task creation.
- `<ul>` list dynamically updated with tasks.

## CSS Styling Strategy

- Bootstrap 5 for responsive grid/layout.
- Custom CSS for completed-task styling (line-through).

- Consistent spacing and alignment using Bootstrap utilities.
- 

## Key Features

Feature	Description
Task Management	Add, edit, delete tasks easily
Mark Complete	Checkbox toggles task completion
Filters	Show all, active, or completed tasks
Responsive Design	Adapts to different screen sizes
Interactive UI	Smooth DOM manipulation with jQuery

## Challenges Faced & Solutions

Challenge	Solution
Handling dynamic task states	Used data-status attributes
UI alignment issues	Leveraged Bootstrap flex utilities

## Challenge

Preventing empty task entries

## Solution

Added input validation

Editing task inline

Used JavaScript `prompt()` for updates

# Outcome

The To-Do List project successfully demonstrates interactive front-end design. It provides essential task management functionality while maintaining a clean and responsive interface. The project improved my skills in jQuery, DOM manipulation, and Bootstrap integration.

# Future Enhancements


- Persistent task storage using `LocalStorage` or backend database.
- User authentication for personalized task lists.
- Drag-and-drop task reordering.
- Theme toggler (light/dark mode).

# Sample Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>To-Do List</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/di
st/css/bootstrap.min.css">
  <style>
    body {
      background-color: #f8f9fa;
    }
    .container {
      max-width: 600px;
    }
    .list-group-item {
      display: flex;
```

```
        justify-content: space-between;
        align-items: center;
    }
    .completed-task {
        text-decoration: line-through;
        color: #6c757d;
    }
</style>
</head>
<body>

    <div class="container mt-5">

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

        <div class="input-group mb-3">

            <input type="text" id="taskInput" class="form-control" placeholder="Add a new task..." aria-label="Add a new task">

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

        </div>
```

```
<div class="btn-group mb-3" role="group" aria-label="Filter tasks">
```

```
<button type="button" class="btn btn-secondary active" id="showAll">All</button>
```

```
<button type="button" class="btn btn-secondary" id="showActive">Active</button>
```

```
<button type="button" class="btn btn-secondary" id="showCompleted">Completed</button>
```

```
</div>
```

```
<ul class="list-group" id="taskList">
```

```
</ul>
```

```
</div>
```

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

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

```
<script>
```

```
$(document).ready(function() {
```



```

// Add a new task

$('#addTaskBtn').on('click', function() {
    let taskText = $('#taskInput').val().trim();
    if (taskText !== '') {
        let newTask = `<li class="list-group-item d-
flex justify-content-between align-items-center" data-
status="active">
            <div class="form-check">
                <input class="form-check-
input" type="checkbox">
                <span class="task-
text">${taskText}</span>
            </div>
            <div>
                <button class="btn btn-sm
btn-outline-warning edit-btn">✎</button>
                <button class="btn btn-sm
btn-outline-danger delete-btn">✖ </button>
            </div>
        </li>`;
        $('#taskList').append(newTask);
    }

```

```
    $('#taskInput').val('');  
  }  
});
```

```
// Mark task as completed  
$('#taskList').on('change',  
'input[type="checkbox"]', function() {  
  let listItem = $(this).closest('li');  
  if ($(this).is(':checked')) {  
    listItem.find('.task-  
text').addClass('completed-task');  
    listItem.attr('data-status', 'completed');  
  } else {  
    listItem.find('.task-  
text').removeClass('completed-task');  
    listItem.attr('data-status', 'active');  
  }  
});
```

```
// Delete a task
```

```
$('#taskList').on('click', '.delete-btn', function() {
```

```
$(this).closest('li').remove();  
});
```

```
// Edit a task
```

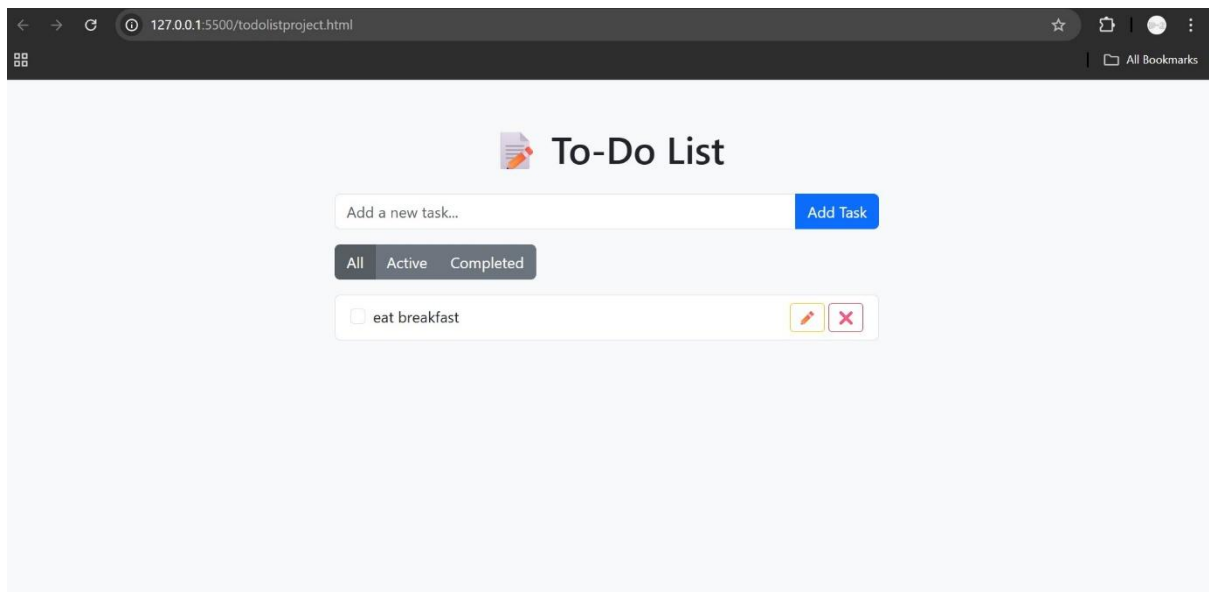
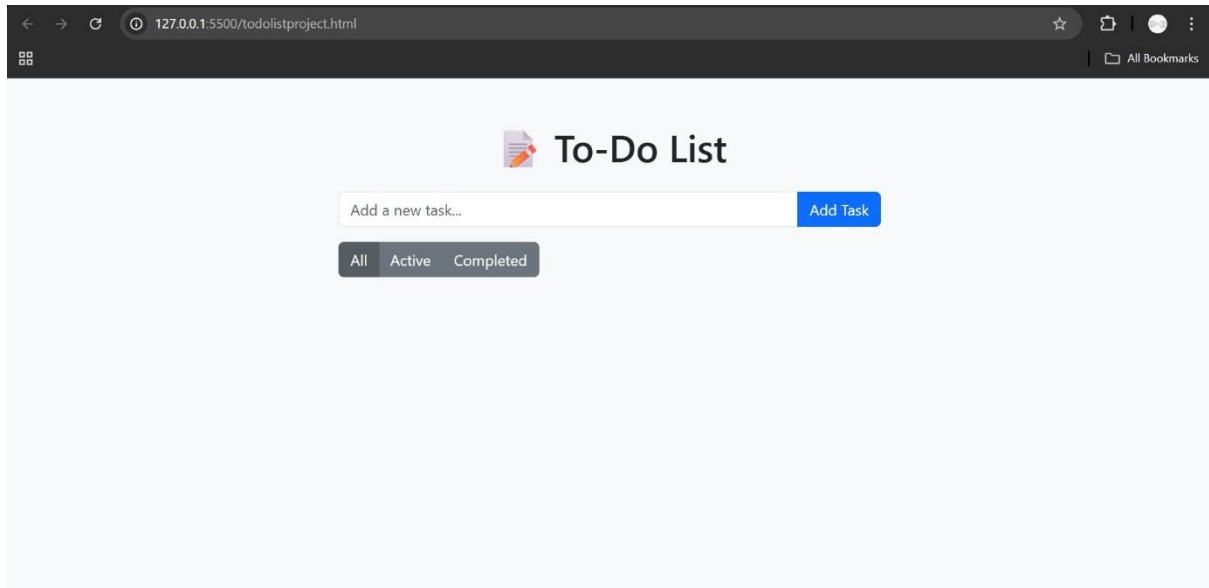
```
$('#taskList').on('click', '.edit-btn', function() {  
    let listItem = $(this).closest('li');  
    let taskTextSpan = listItem.find('.task-text');  
    let currentText = taskTextSpan.text();  
    let newText = prompt('Edit your task:',  
currentText);  
    if (newText !== null && newText.trim() !== '') {  
        taskTextSpan.text(newText);  
    }  
});
```

```
// Filter tasks
```

```
$('.btn-group').on('click', 'button', function() {  
    $('.btn-group button').removeClass('active');  
    $(this).addClass('active');
```

```
    let filter = $(this).attr('id');
    $('#taskList li').each(function() {
        let status = $(this).attr('data-status');
        if (filter === 'showAll') {
            $(this).show();
        } else if (filter === 'showActive' && status
=== 'active') {
            $(this).show();
        } else if (filter === 'showCompleted' &&
status === 'completed') {
            $(this).show();
        } else {
            $(this).hide();
        }
    });
});
});
</script>
</body>
</html>
```

# Screenshots of Final Output



## Conclusion

The To-Do List application demonstrates the application of HTML, CSS (Bootstrap), JavaScript, and jQuery to create a practical front-end project. This project

enhanced my understanding of UI/UX principles, responsiveness, and interactivity. It also highlighted the importance of clean design and functionality for productivity applications.

## References

- jQuery Documentation: <https://api.jquery.com/>
- Bootstrap Docs: <https://getbootstrap.com/>
- L&T LMS :  
<https://learn.intedutech.com/Landing/MyCourse>