

To Do List

Submitted By:

Aaron Jibin
2462303
aaron.jibin@btech.christuniversity.in

Anadhika Goswami
2462313
anadhika.goswami@btech.christuniversity.in

Anna Isson
2462316
anna.isson@btech.christuniversity.in

Course: UI/UX Design Fundamentals

Instructor Name: Ms. Nagaveena

Institution: CHRIST University

Date of Submission: 25/09/2025

Abstract

This project delivers a To-Do List Application designed for intuitive daily task management. The app features a simple form to add new tasks, checkboxes to mark completion, delete functions, and filtering options (All, Active, Completed). All task data is persisted with browser local storage, ensuring tasks remain across sessions. The application is styled with Bootstrap for a modern, clean, and responsive interface that works seamlessly on both desktop and mobile devices. Lightweight, fast, and free of external dependencies beyond Bootstrap and jQuery, the To-Do List App is ideal for users seeking straightforward productivity or as a foundation for more sophisticated task management solutions.

Objectives

- Design a user-friendly, visually clean interface using Bootstrap.
- Enable users to add, complete, delete, and filter tasks easily.
- Persist user data using browser local storage.
- Ensure the application is responsive and mobile-friendly.

Scope of the Project

- Focused on front-end implementation with Bootstrap and jQuery.
- No server-side or backend integration.
- Designed for both desktop and mobile users.
- Uses open-source tools and pure code, minimal dependencies.

Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Markup and content structure
CSS3/Bootstrap	Styling, layout management, UI
JavaScript/jQuery	Task functionality, DOM manipulation
VS Code	Code editor
Browser DevTools	Testing and debugging

HTML Structure Overview

- Semantic tags for structure: `<header>`, `<main>`, `<form>`, ``, ``.
- Main sections: Task input form, task list, filter controls.
- Navigation and filter UI built with Bootstrap classes.

CSS Styling Strategy

- Bootstrap for layout, spacing, and responsive design.
- Custom CSS for additional styling and overrides.
- Mobile-first design approach.
- Hover effects and smooth transitions for interactivity.

Key Features

Feature	Description
Add Tasks	Simple form to add new tasks
Complete Tasks	Mark tasks as completed with checkboxes
Delete Tasks	Remove finished tasks
Filter Tasks	View All, Active, or Completed tasks
Persistent Storage	Tasks saved in local storage
Responsive Design	Works on all screen sizes
Bootstrap UI	Modern, clean interface
Lightweight	Fast, minimal dependencies

Challenges Faced & Solutions

Challenge	Solution
Data persistence across sessions	Used browser local storage for tasks
Responsive UI on all devices	Leveraged Bootstrap's grid and utilities
Efficient DOM manipulation	Utilized jQuery for simple logic
Task filtering performance	Implemented efficient filter logic

Outcome

- Achieved a robust, user-friendly, and visually appealing task management app.
- All intended features work reliably across devices and browsers.
- Gained hands-on experience in Bootstrap, jQuery, and local storage.

Future Enhancements

- Add drag-and-drop task ordering.
- Integrate user authentication and cloud sync.
- Add task deadlines, reminders, and notifications.
- Provide theme toggling (light/dark mode).

Sample Code

Index.html

~~~

```
<form id="add-task-form">
  <input type="text" id="task-input" placeholder="Enter a new task..." required>
  <button type="submit">Add Task</button>
</form>
```

```
<button class="filter-btn" data-filter="all">All Tasks</button>
<button class="filter-btn" data-filter="active">Active Tasks</button>
<button class="filter-btn" data-filter="completed">Completed Tasks</button>
```

```
<ul id="task-list" class="list-group"></ul>
```

~~~

style.css

~~~

```
.task-item {
  display: flex;
  align-items: center;
}
```

```
.completed .task-text {
  text-decoration: line-through;
  color: #6c757d;
}
```

```
@media (max-width: 576px) {
  .task-item {
    flex-direction: column;
    align-items: flex-start;
  }
}
```

```
}
```

```
~~~
```

```
script.js
```

```
~~~
```

```
let tasks = JSON.parse(localStorage.getItem('tasks')) || [];
```

```
$addTaskForm.on('submit', function(e) {  
  e.preventDefault();  
  const taskText = $taskInput.val().trim();  
  if (taskText) {  
    tasks.push({  
      id: Date.now(),  
      text: taskText,  
      completed: false  
    });  
    saveTasks();  
    renderTasks();  
  }  
});
```

```
$taskList.on('change', '.task-checkbox', function() {  
  const taskId = $(this).data('id');  
  const task = tasks.find(t => t.id === taskId);  
  if (task) {  
    task.completed = !task.completed;  
    saveTasks();  
    renderTasks();  
  }  
});
```

```
$filterBtns.on('click', function() {  
  $filterBtns.removeClass('active');  
  $(this).addClass('active');  
  renderTasks();  
});
```

```
function saveTasks() {  
  localStorage.setItem('tasks', JSON.stringify(tasks));  
}
```

```

function renderTasks() {
  const filter = $('.filter-btn.active').data('filter') || 'all';
  let filteredTasks = tasks;

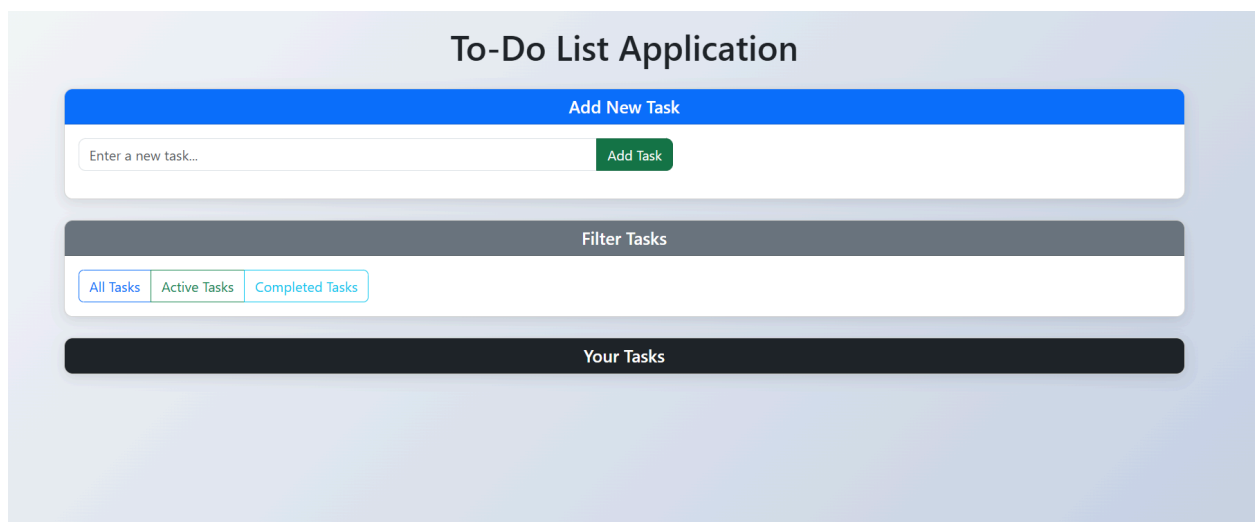
  if (filter === 'active') {
    filteredTasks = tasks.filter(task => !task.completed);
  } else if (filter === 'completed') {
    filteredTasks = tasks.filter(task => task.completed);
  }

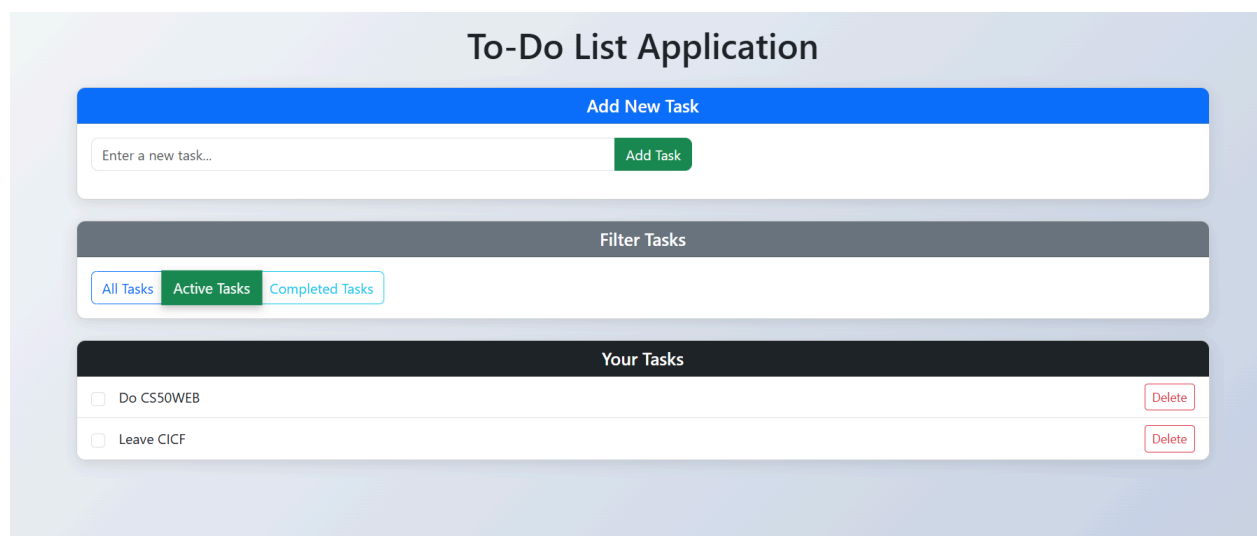
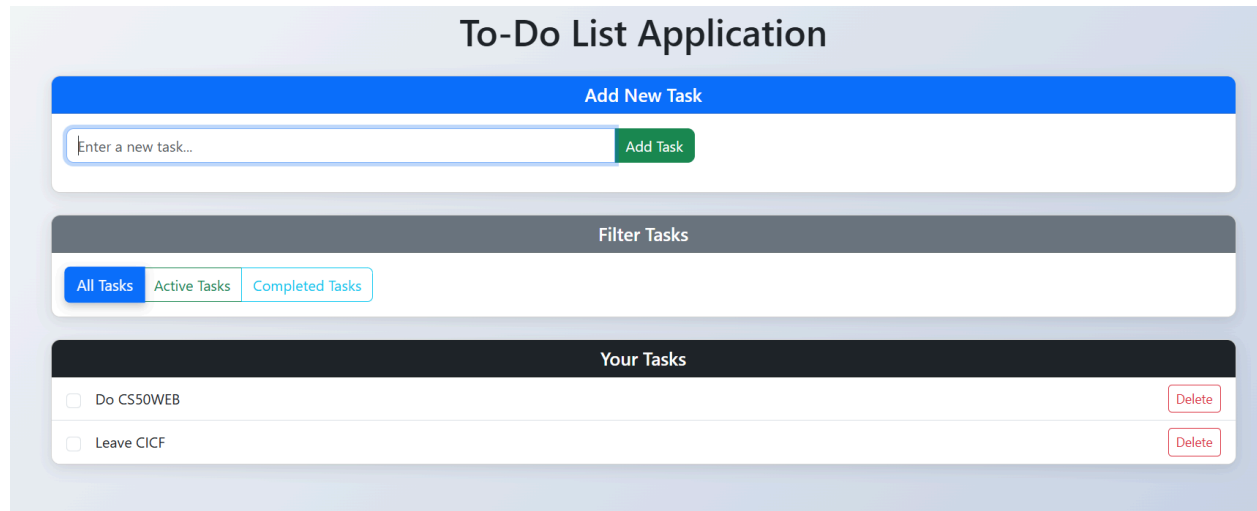
  $taskList.empty();
  filteredTasks.forEach(task => {
    const taskElement = `
      <li class="list-group-item d-flex align-items-center">
        <input type="checkbox" class="task-checkbox" ${task.completed ? 'checked' : ''}
data-id="${task.id}">
        <span class="${task.completed ? 'text-decoration-line-through' : ''}>${task.text}</span>
        <button class="delete-btn" data-id="${task.id}">Delete</button>
      </li>
    `;
    $taskList.append(taskElement);
  });
}

```

~~~

Screenshots of Final Output





Conclusion

The To-Do List Application project provided valuable insights into UI design, responsive layouts, and client-side storage. Building a clean, robust frontend with Bootstrap and jQuery enhanced proficiency in modern web development and reinforced best practices for user-centric, accessible design.

References

- GitHub Link of the Project: <https://github.com/Aarxn-Jibz/Frontend/tree/main/Project3>
- L&T LMS: <https://learn.Intedutech.com/Landing/MyCourse>