# **Project Report: TaskFlow - Task Manager**

**Date:** September 26, 2025

## **1. Introduction**

This report provides a detailed overview of the **TaskFlow** single-page web application, a project management tool designed as a Kanban board. The project utilizes a foundational technology stack (HTML, CSS, Bootstrap, jQuery) and incorporates the jQuery UI library to enable its core drag-and-drop functionality. Its column-based layout is fundamentally different from typical grid-based sites.

## **2. Technology Stack**

* **Structure:** HTML5
* **Styling:** CSS3, Bootstrap 5.3
* **Interactivity:** JavaScript (ES6), jQuery 3.7, **jQuery UI 1.13** (for drag-and-drop)
* **Components:** Bootstrap (Navbar, Buttons, Modal)

## **3. Site Breakdown**

The taskflow.html file is built around the Kanban concept:

* **Navbar:** A simple header for the application title.
* **Main Board Header:** Contains a title and an "Add New Task" button that triggers a modal.
* **Kanban Columns:** The primary layout element, a single row containing three columns, each representing a task status ("To Do," "In Progress," "Done").
* **Task Lists:** Inside each column, a <ul> element serves as a container for the draggable task cards.
* **Add Task Modal:** A hidden form for creating new tasks.

## **4. Key Code Snippets**

### **4.1. Initializing Drag-and-Drop (from taskflow.js)**

This snippet is the heart of the application. It uses the .sortable() method from **jQuery UI** to make the task lists interactive. The connectWith option allows cards to be moved between the lists, and the receive event handler updates the task's status in the JavaScript data array when a drop is completed.

// --- Make lists sortable and connected (Drag-and-Drop) ---

$('.task-list').sortable({

connectWith: ".task-list", // Allows dragging between lists

placeholder: "ui-sortable-placeholder", // A class for the drop area highlight

receive: function(event, ui) {

const taskId = ui.item.data('id');

const newStatus = $(this).parent().attr('id'); // e.g., 'todo', 'inprogress'

// Find the task in the data array and update its status

const task = tasks.find(t => t.id === taskId);

if (task) {

task.status = newStatus;

}

renderTasks(); // Re-render to apply correct styling

}

}).disableSelection();

### **4.2. Adding a New Task (from taskflow.js)**

This function handles the submission of the "Add Task" form. It creates a new task object, adds it to the main tasks array, re-renders the entire board, and then programmatically hides the Bootstrap modal.

// --- Handle New Task Form Submission ---

$('#addTaskForm').on('submit', function(event) {

event.preventDefault();

const newTask = {

id: Date.now(), // Use timestamp for a unique ID

title: $('#taskTitle').val(),

description: $('#taskDescription').val(),

status: 'todo' // New tasks always start in 'To Do'

};

tasks.push(newTask);

renderTasks();

// Hide the modal and reset the form

const addTaskModal = bootstrap.Modal.getInstance(document.getElementById('addTaskModal'));

addTaskModal.hide();

$(this).trigger('reset');

});