



# Final Project Requirement Documentation

---

**formClient:** TaskFlow Solutions

**Project:** Interactive Task Manager

## 1. Project Overview

TaskFlow Solutions, a company specializing in productivity tools, has hired you to develop a **Task Manager Web Application** to help users organize and track their tasks efficiently. This application will allow users to add, edit, delete, and mark tasks as important or completed. The application must be built using **HTML, CSS, and JavaScript** and should focus on **functionality, user experience, and clean code implementation**.

Your role as the developer is to ensure that the application meets all functional requirements while maintaining best coding practices and usability.

## 2. Work Statement

You are responsible for developing a fully functional **Task Manager** web application that meets the following criteria:

- The application must allow users to create, modify, and delete tasks dynamically.
- Tasks must be displayed in a dedicated section of the webpage using JavaScript.
- Tasks should be styled to indicate priority, importance, and completion status.
- The application must log task updates to the browser console using **JSON.stringify()**.
- The final product must be **visually appealing, user-friendly, and well-documented**.

At the end of the development process, you will submit a **GitHub repository** containing your project files and a **link to the deployed project on the Wake Tech server**.

## 3. Functional Requirements

### 3.1 User Interface (UI) Requirements

The Task Manager must include the following elements:

- An **input field** where users can enter a task name.
- A **dropdown menu** to select task priority (**High, Medium, Low**).
- A **checkbox** for marking a task as important
- A **checkbox** for marking a task as completed (or done).
- A **submit button** to add the task to the list.
- A **delete button** to remove the task to the list.
- A **date** to show when the task was added
- A div with an **id of "taskmanager"** where tasks will be displayed dynamically.

Tasks should have the following visual styles using **JavaScript .style property**:

- Important tasks must be highlighted in red
- Completed tasks must have a strikethrough

### 3.2 JavaScript Functionality Requirements

- **Event Handling & DOM Manipulation**
  - Users should be able to submit tasks via the form.
  - Tasks must be displayed dynamically in the #taskmanager div using .innerHTML.
  - Users must be able to delete tasks.
  - Users must be able to toggle task completion.
  - Task must display the date task was added using the Date object
- **Task Data Structure**
  - Tasks must be stored in an array of objects with the following properties:

```
{  
  id: 1,  
  name: "Finish JavaScript project",  
  priority: "High",  
  isImportant: true,  
  isCompleted: false,  
  date: today  
}
```

- **Conditional Logic & Styling**
  - Apply styling based on priority.
  - Highlight important tasks in **red**.
  - Apply a **strikethrough** to completed tasks.
- **Console Logging**
  - Every time a task is added, updated, or deleted, the full task list must be logged in the console using:

```
console.log(JSON.stringify(tasks));
```

## 4. Non-Functional Requirements

- The application must be coded using **vanilla JavaScript** (no external libraries or frameworks).
- The code must follow **clean coding practices**, including proper indentation, meaningful variable names, and comments.
- The UI should be **responsive and visually intuitive**.
- The application must **prevent invalid input**, such as empty task names.

## 5. Deliverables & Submission

### 5.1 Deliverables

The final submission must include:

1. **A clickable link to your GitHub repository** containing all project files.
2. **A brief reflection document (200 words)** describing your approach and any challenges faced.
3. **A clickable link to your project hosted on the Wake Tech server.**

## 5.2 Grading Rubric

Criteria	Points
Form & Input Handling	20
Task Display in DOM	20
Use of JavaScript Objects & Arrays	20
Conditional Styling (Important/Completed)	5
Event Handling	10
Console Logging of Tasks (JSON.stringify)	10
Code Organization & Readability	5
Creativity & User Experience	5
Submission Requirements	5

**\*see rubric in submission link for details**

## 6. Acceptance Criteria

The project will be considered **successfully completed** when:

- The Task Manager meets all functional requirements.
- The UI is intuitive and responsive.
- Tasks are correctly displayed, updated, and styled.
- Console logging accurately reflects task changes.
- The code is well-structured and documented.

**Watch the sample Task Manager program video for inspiration.**

Your program should be **user-friendly** and **visually engaging**—feel free to get creative!

⚠ *Please note: The video has no sound.*

## Final Project WEB115

