



SF Dev Workshop – 2025

Assignment – 2 | JAVASCRIPT

Deadline – 10 July 2025(EOD)

TO-DO APP

Introduction

Are you a beginner web developer looking to enhance your skills and take on a rewarding challenge? This project will guide you through building a dynamic **To-Do App** using HTML, CSS, and JavaScript. By working on this project, you will strengthen your understanding of DOM manipulation, event handling, and JavaScript data structures, while also practicing responsive design and creative styling.

PROJECT OBJECTIVES

- Create a responsive web page structure using HTML and CSS to display the To-Do list.
- Use JavaScript to manage tasks, including adding, completing, and deleting tasks.
- Incorporate features like persistent storage (localStorage), task filtering, and dynamic task rendering.
- Ensure a visually appealing and user-friendly interface with CSS styling.

PROJECT TASKS

1. SETUP THE PROJECT

- Create an HTML file for the To-Do App.
- Create a basic CSS file to style the app.
- Link the HTML and CSS files together.
- Include a JavaScript file and link it to the HTML.

2. DESIGN THE TO-DO INTERFACE

- Create a header for the app title.
- Design an input field and button to add new tasks.
- Add a section to display the list of tasks.

- Include a way to mark tasks as completed (e.g., a checkbox or button).
- Add a delete option for each task.

3. DEFINE TASK DATA STRUCTURE

- Use an array or object in JavaScript to store tasks.
- Each task should have properties like description, completed (true/false), and a unique id.
- Optionally, implement localStorage to persist tasks across page reloads.

4. DISPLAY TASKS

- Use JavaScript to render the list of tasks dynamically on the page.
- Show all tasks in a list format.
- Style completed tasks differently (e.g., strikethrough or faded color).
- Optionally, move completed tasks to a separate section.

5. HANDLE USER INPUT

- Capture user input when adding a new task.
- Validate that the input is not empty before adding.
- Allow users to mark tasks as completed or uncompleted.
- Enable users to delete tasks from the list.

6. UPDATE AND MANAGE TASKS

- Update the task list in real-time as tasks are added, completed, or deleted.
- Ensure the app reflects the current state of the task list at all times.
- If using localStorage, update storage whenever the task list changes.

7. STYLE AND POLISH

- Use CSS to create a clean, modern, and visually appealing layout.
- Ensure the app is responsive for various screen sizes:
- Use media queries for layout adjustments.
- Prefer responsive units (% , rem , em , vh , vw) over fixed pixels.
- Utilize Flexbox or CSS Grid for flexible layouts.
- Add creative touches: color schemes, spacing, animations, and transitions.

8. BONUS CHALLENGES (OPTIONAL)

- Implement task filtering (e.g., show all, active, or completed tasks).
 - Add an edit feature to modify existing tasks.
 - Allow the user to clear all completed tasks with a single click.
 - Add a date/time property to each task.
 - Sync tasks with a backend or external API for advanced persistence.
-

Responsiveness

With the growing variety of devices and screen sizes, making your To-Do App responsive is essential. **Responsiveness** ensures your app looks and works well on mobile phones, tablets, laptops, and desktops. Rather than targeting every possible device size, focus on:

- Using responsive units (rem, em, %, vh, vw) for dimensions and font sizes.
- Applying media queries only when structural changes are needed.
- Leveraging Flexbox and Grid for adaptable layouts.
- Testing your app on different screen sizes and adjusting margins, padding, and alignment as needed.

Remember, your first attempt at responsiveness doesn't have to be perfect. Aim to make your app as flexible as possible, and iterate as you learn more.

Important Instructions

- **Do not copy code** from external sources. Use references for learning, but write every line yourself for maximum benefit.
- Experiment with different CSS units and properties to achieve a highly responsive and visually appealing design.
- Add creative elements: animations, color patterns, and thoughtful spacing.
- If you're new to web development, break the project into small sections and tackle each one step by step.
- It's okay if you don't finish every feature—focus on learning and doing your best.
- Web development offers vast opportunities. Use this project as a stepping stone to explore more advanced concepts and projects.

Good luck, and enjoy building your interactive To-Do App!