Task Manager Web Application

Project Overview

This project is a **simple Task Manager web application** built using **HTML, CSS, and Vanilla JavaScript**.
The application allows users to add, complete, and delete tasks dynamically in the browser.

Project Goals and Objectives

- Create an interactive task management interface
- Allow users to add new tasks with validation
- Enable marking tasks as completed
- Support task deletion
- Provide a clean and user-friendly UI
- Demonstrate core DOM manipulation and event handling concepts

---

Setup Instructions

This is a frontend-only project and does not require any external dependencies.

Installation and Configuration Steps

1. Download or clone the project files.
2. Ensure the following files are present in the same directory:
   - index.html
   - style.css
   - script.js
3. Open index.html in any modern web browser (Chrome, Firefox, Edge).

No server, database, or additional configuration is required.

Code Structure

```
task-manager/
|
├── index.html # HTML structure and layout
├── style.css # Styling and layout definitions
```

```
 ├── script.js # Application logic and interactivity
└── README.md # Project documentation
```

File Responsibilities

- **index.html**
  - Defines the layout of the application
  - Contains task input form, task list container, and empty state message
  - Links external CSS and JavaScript files
- **style.css**
  - Handles visual styling, spacing, colors, and layout
  - Defines styles for completed tasks, buttons, and empty state
- **script.js**
  - Manages task creation, deletion, and completion
  - Handles form submission and click events
  - Updates UI dynamically using DOM manipulation

Technical Details

Architecture

- **Client-side only architecture**
- Separation of concerns:
  - HTML for structure
  - CSS for presentation
  - JavaScript for behavior

Algorithms and Logic

- **Task Creation Algorithm**
  - Capture form submission
  - Validate input (non-empty task)
  - Dynamically create and append task elements to the DOM
- **Event Delegation**

    - A single click listener on the task list handles:
        - Task deletion
        - Task completion toggling
    - Improves performance and scalability

Data Structures

- Tasks are represented implicitly as DOM elements (<li>)
- No external data storage or complex data structures are used

---

Testing Evidence

Testing was performed using **manual functional testing**.

Sample Test Cases

Test Case	Description	Expected Result
Add Task	Enter text and click Add	Task appears in list
Empty Task	Submit empty input	Alert shown, task not added
Complete Task	Click checkbox or task text	Task marked as completed
Delete Task	Click Delete button	Task removed from list
Empty State	Delete all tasks	“No tasks yet” message appears

Validation Results

- Input validation prevents empty tasks
- UI updates instantly without page reload
- Application works consistently across modern browsers