**Project Documentation**

**To-Do List Application**

Overview:

**The To-Do List application is a lightweight, user-friendly web application designed to help users manage their daily tasks efficiently. It allows users to add tasks, mark them as completed, and delete them. The application is built using HTML, CSS, and JavaScript, ensuring compatibility across most modern browsers without requiring additional libraries or frameworks.**

**Features:**

**Core Functionalities:**

* **Add Tasks: Users can input tasks into a text field and add them to the to-do list.**
* **Mark as Complete: Tasks can be marked as completed, applying a strikethrough style for visual distinction.**
* **Delete Tasks: Tasks can be removed from the list when no longer needed.**

**Design Enhancements:**

* **Modern and responsive design suitable for desktop and mobile.**
* **Visual feedback for interactive elements (e.g., button hover effects).**
* **Clear separation of completed and active tasks.**

**Technologies Used:**

* **HTML: Structures the application and defines user interface components.**
* **CSS: Styles the application, ensuring an attractive and consistent appearance.**
* **JavaScript: Adds interactivity, including task addition, completion, and deletion.**
* **Vs code: which is used to run the html , css , js codes.**

**Detailed Code Explanation:**

**HTML (index.html):**

**Defines the basic structure of the application:**

* **A title for the webpage.**
* **An input field and a button for adding tasks.**
* **A container (ul) to hold task items.**

**Key elements:**

* **<input>: For user input of tasks.**
* **<button>: To trigger task addition.**
* **<ul>: Dynamically populated with task items.**

**CSS (styles.css)**

**Handles the visual styling:**

* **Responsive Design: Ensures usability across devices.**
* **Hover Effects: Buttons change color when hovered over for better UX.**
* **Task States: Differentiates completed tasks using the completed class.**

**Key CSS Features:**

* **Flexbox for layout alignment.**
* **Shadow effects for modern aesthetics.**
* **Transitions for smooth interactions.**

**JavaScript (script.js)**

**Adds interactivity:**

* **Task Addition: Dynamically creates a new <li> element for each task.**
* **Completion Toggle: Adds/removes the completed class to strike through tasks.**
* **Deletion: Removes the parent <li> of the delete button when clicked.**

**Key Functions:**

1. **add\_item: Main function for adding tasks and attaching buttons.**
   * **Validates non-empty input.**
   * **Creates task elements dynamically.**
2. **Button Event Listeners:**
   * **Complete Button: Toggles task completion.**
   * **Delete Button: Removes the task.**

**Example Workflow:**

**Add a Task:**

* + **Input text into the text box.**
  + **Click the "Add" button.**
  + **The task appears in the list with "Complete" and "Delete" buttons.**

**Complete a Task:**

* + **Click the "Complete" button for a task.**
  + **The task gets a strikethrough effect.**

**Delete a Task:**

* + **Click the "Delete" button for a task.**
  + **The task is removed from the list.**

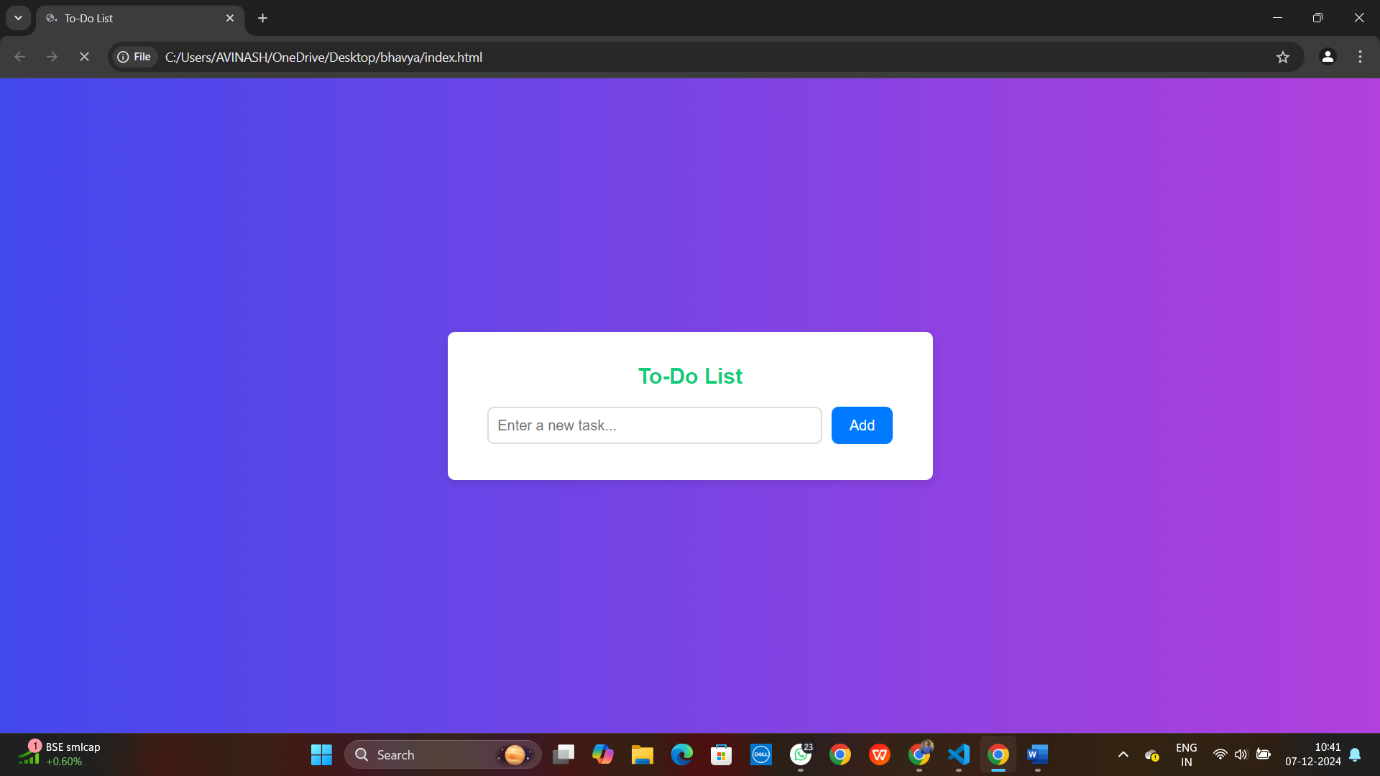
**Future Enhancements:**

**Planned Features:**

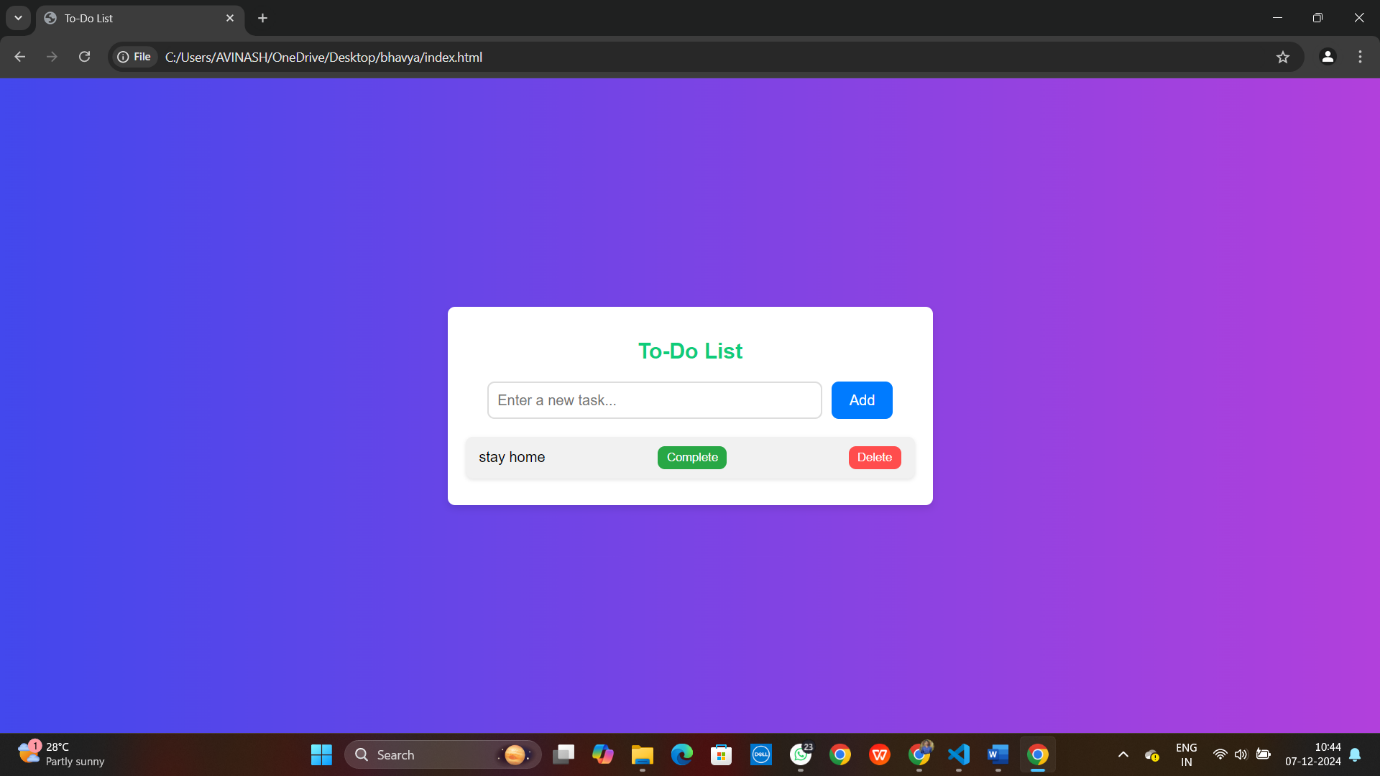
* **Task Editing: Allow users to edit tasks directly in the list.**
* **Persistence: Use local storage or a database to save tasks across sessions.**
* **Sorting Options: Enable sorting by status (completed/active) or alphabetical order.**
* **Due Dates: Add an option to set and display due dates for tasks.**

**Screenshots**

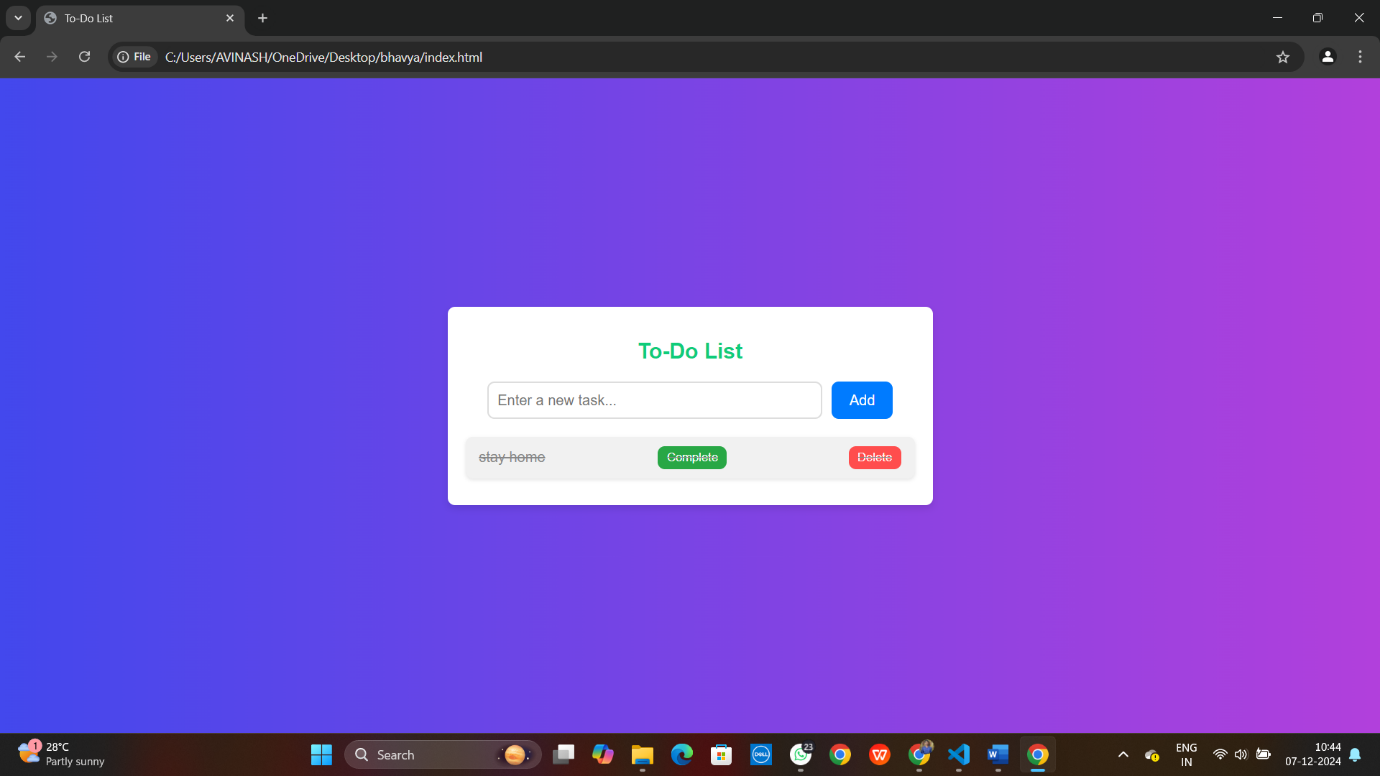
**Initial State:**



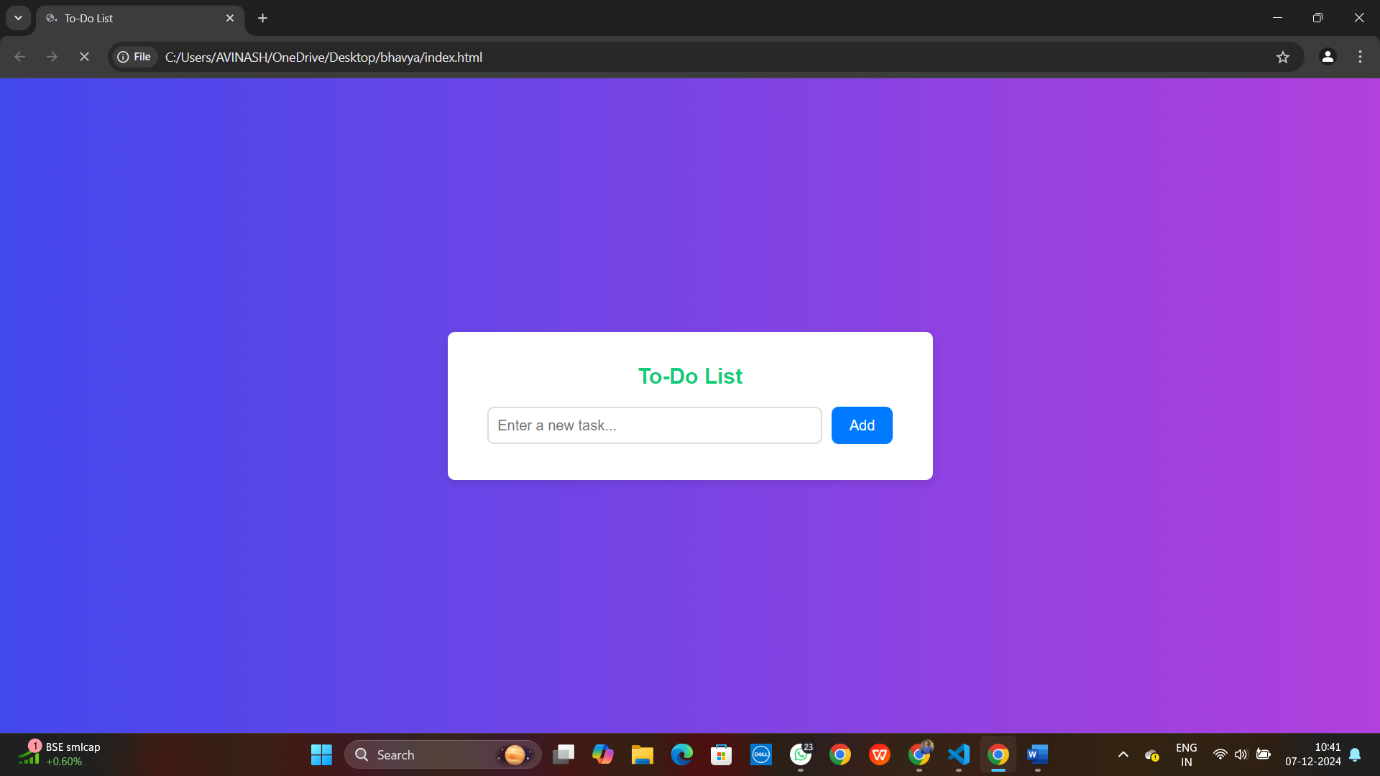
**Adding a Task:**



**Marking as Complete:**



**Deleting a Task**



**Conclusion:**

**The Enhanced To-Do List is a simple yet powerful web application for managing tasks. Its clean interface, responsive design, and interactive features make it a practical tool for daily task management. The project provides a solid foundation for further enhancements, including persistence and advanced features.**