# **Experiment: Open-Ended Lab**

**Todo list Application**

**Web Engineering**

**Submitted to**

Lab Engr Hafsa Awan

**Submitted By**

Rameesha Shafiq

Javeria Javed

Sabahat Farhat

Military College of Signals (NUST)

December 23, 2024

**Objective:** To design an application to help users manage their daily tasks efficiently. It supports operations like adding, deleting, editing and mark tasks.

**Introduction**

In the fast-paced and dynamic landscape of modern life, the effective management of tasks and responsibilities is paramount to personal and professional success. As individuals navigate through a multitude of commitments, from work assignments to personal goals, the need for an organized and efficient task management system becomes increasingly evident. This underscores the significance of a to-do list—an indispensable tool designed to streamline and enhance the way we approach our daily activities. This introduction sets the stage for the exploration of a to-do list application, a digital solution meticulously crafted to address the challenges of contemporary task management. By delving into the intricacies of its functionalities, design principles, and user interface, this report aims to illuminate the role of a to-do list in fostering productivity, prioritization, and overall well-structured time management.

## **Features**

1. **Add Tasks**: Users can add new tasks by clicking the "Add" button.
2. **Update Tasks**: Users can update existing tasks by clicking the edit icon.
3. **Delete Tasks**: Users can delete tasks by clicking the "X" button next to the task.
4. **Mark Tasks as Done**: Tasks can be marked as "done" by clicking on a checkbox, which will visually indicate completion.

**Implementation of Todo App:**

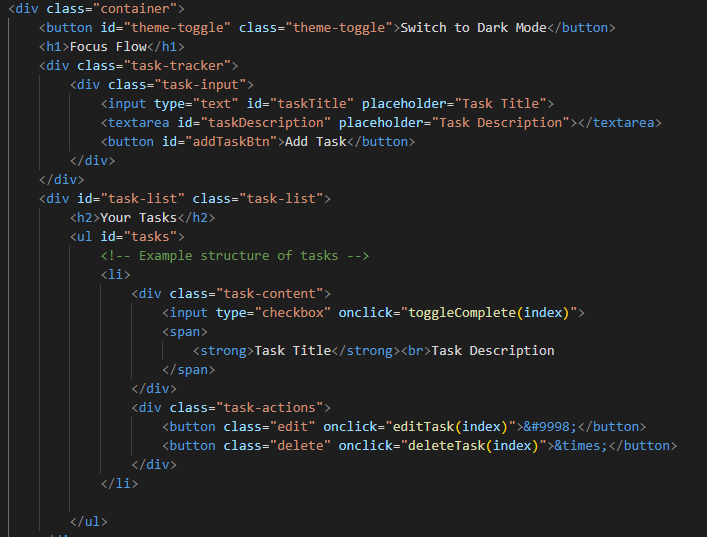
We have implemented todo app using html,css and js.

**Html code:**

The HTML defines the structure of the application.

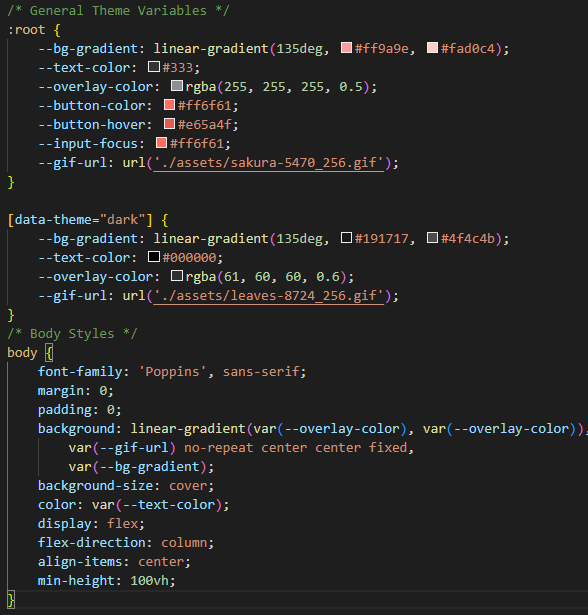
**Key Sections**:

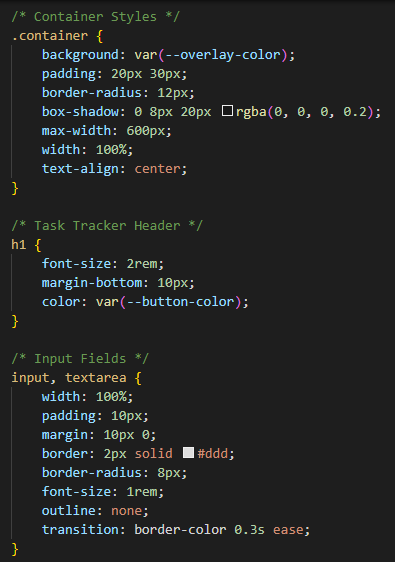
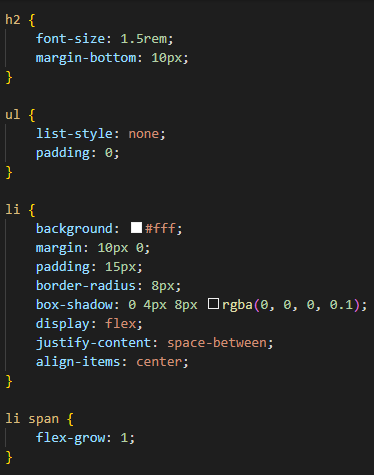
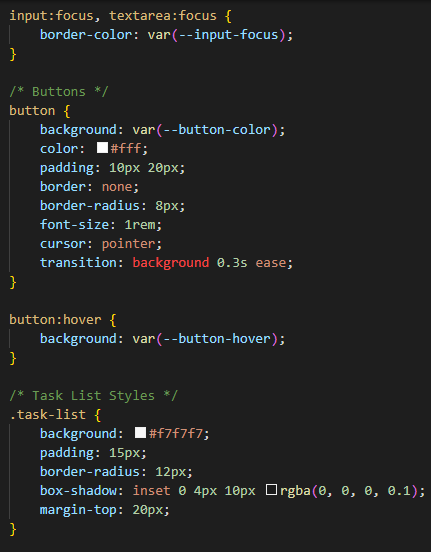
* **Task Input**: Users can input a task title and description.
* **Task List**: Displays a list of tasks dynamically generated from the JavaScript logic.
* **Theme Toggle**: A button to switch between light and dark modes.



**CSS code:**

It is used for the styling of the application.



**JavaScript code:**

The logic of the application is implemented in script.js.

#### **Key Features**

1. **Adding Tasks**:
   * Captures user input for the task title and description.
   * Validates inputs and prevents duplicate titles.
   * Saves tasks to local storage and updates the UI.



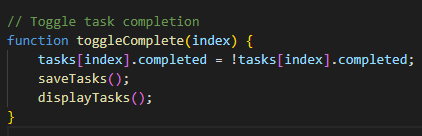
1. **Displaying Tasks**:

* Dynamically generates and displays task items.
* Each task includes a checkbox, edit, and delete button.



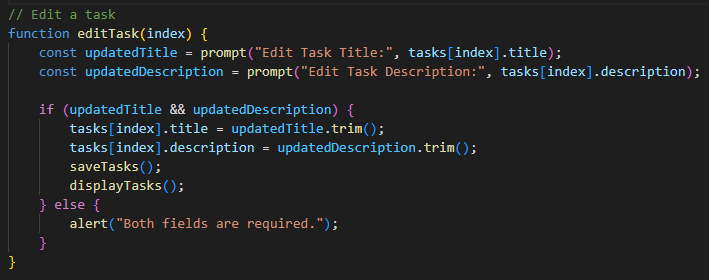
1. **Marking Tasks as Completed**:

* Toggles the completed state of a task when the checkbox is clicked.



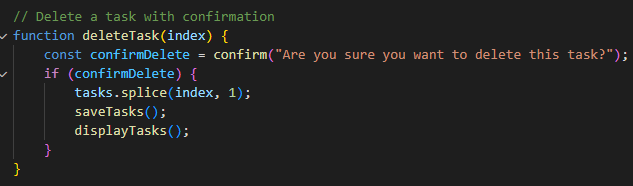
1. **Editing Tasks**:

* Allows users to edit the title and description of tasks.



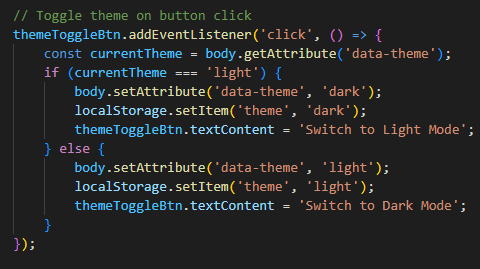
1. **Deleting Tasks**:

* Removes tasks after user confirmation.



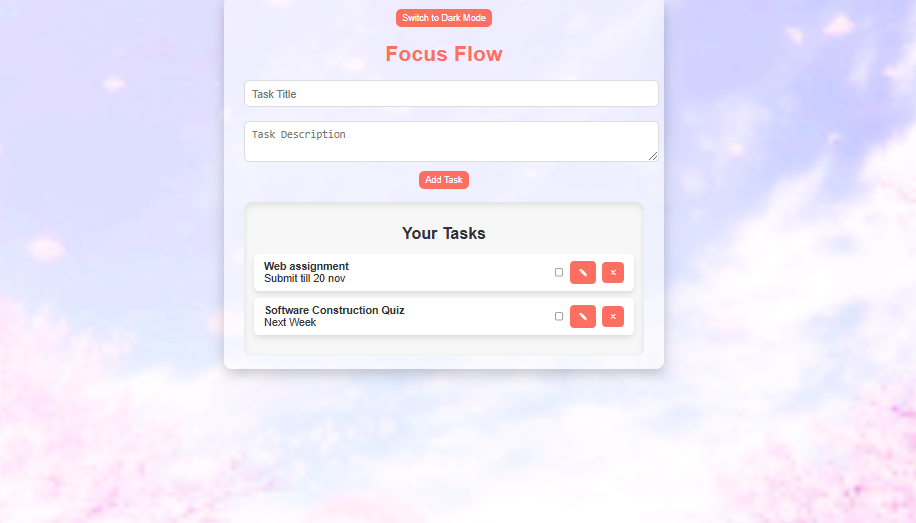
1. **Theme Toggle**:

* Toggles between light and dark modes.
* Persists the selected theme in local storage.

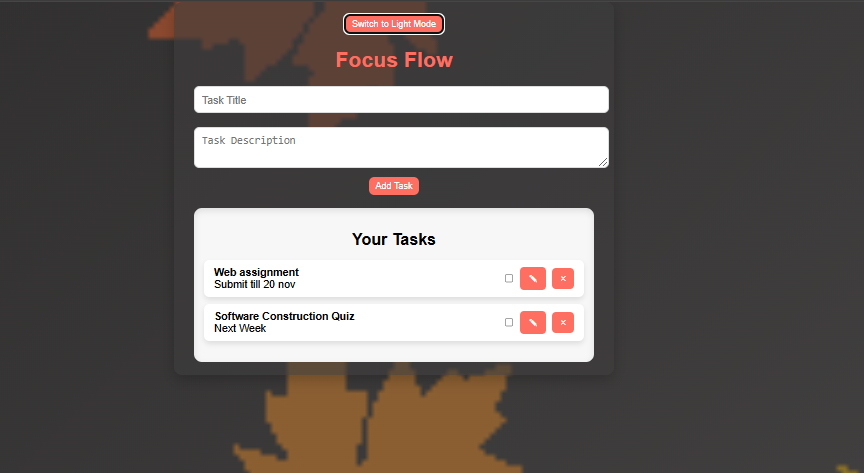


**GUI(output):**

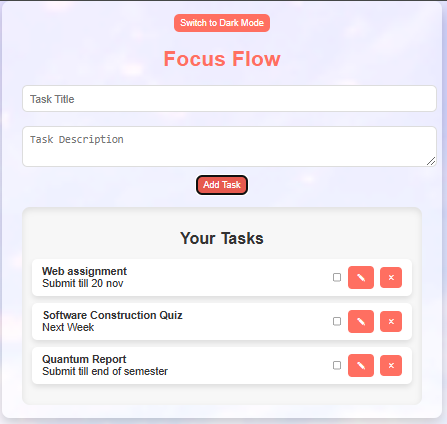
**Light Mode**

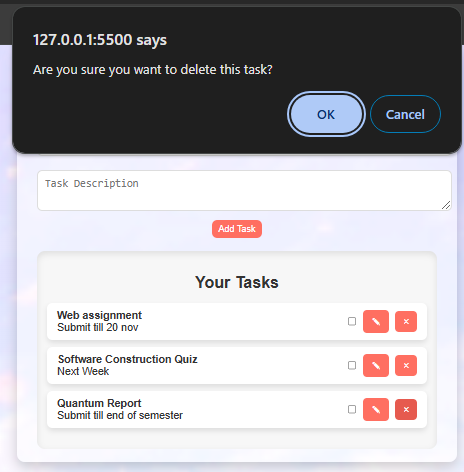
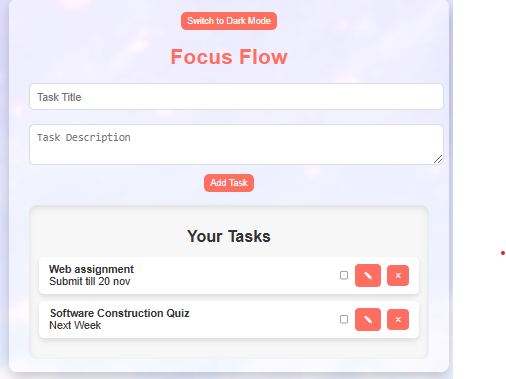


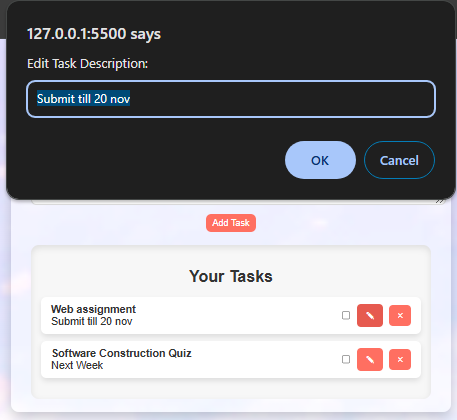
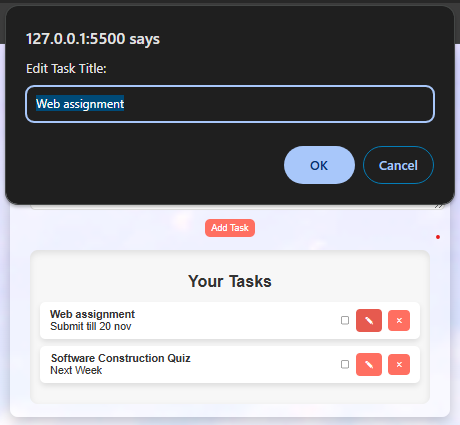
**Dark Mode:**

****

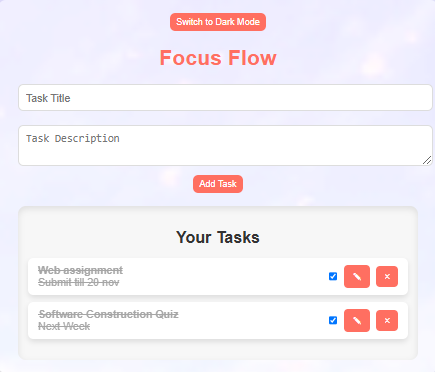
**Add task**

****

**Delete Task**:

**Edit Task**

**Mark Task:**

****

### **Future Enhancements**

* Add task prioritization (e.g., high, medium, low).
* Enable task sorting based on priority or completion status.
* Add due date functionality.
* Integrate a search/filter bar to quickly find tasks.