Name :- Sukanya Mahendra Kumbhar.

E-mail ID:- sukanyakumbhar09@gmail.com

Task Title:- To-Do List

Task Description:- The goal of this project is to develop a user-friendly to-do list application that allows users to efficiently manage their tasks and priorities. The application should have features for creating, editing, updating, and deleting tasks.

Steps Taken:-

Step 1: Set Up the HTML Structure

Start by creating the basic structure of your to-do list application using HTML. This will involve setting up the input field for task entries, buttons for adding tasks, and possibly sorting or filtering tasks. You will also need a container where the tasks will be displayed.

Step 2: Style the Application with CSS

Use CSS to style your HTML structure. Define styles for your task input, buttons, and the task list itself. Make it visually appealing and user-friendly.

Step 3: Add Interactivity with JavaScript

JavaScript will handle the functionality of the to-do list, such as adding new tasks, marking tasks as complete, or deleting them. This involves manipulating the DOM based on user interaction.

Step 4: Test the Application

Run your application in a browser and test it to make sure that tasks can be added, displayed, and removed correctly. Check responsiveness and cross-browser compatibility if needed.

Step 5: Enhancements and Debugging

Based on your testing, make necessary enhancements. You can add features like editing tasks, sorting by priority, or filtering completed tasks. Debug any issues that come up during testing.

Challenges faced:-

1. Manipulating the Document Object Model (DOM) efficiently as users add, remove, or modify tasks. JavaScript must handle these tasks without causing performance lags or user interface freezes, especially as the list grows.
2. Retaining the state of the to-do list (e.g., what tasks have been added or completed) across browser sessions without a backend server.

Solutions implemented:-

1. Use efficient DOM manipulation strategies, such as minimizing reflows and repaints by updating elements in batches or using Document Fragments.
2. Utilize the browser's local storage or session storage to save the state of the to-do list. This ensures that when the user returns to the application, their tasks are still visible.

Project Update:-

The TaskTracker project is progressing well, with the core functionalities of adding and removing tasks fully operational. Recent enhancements include the integration of local storage for persistence

and the implementation of a responsive design for better mobile accessibility. Current efforts are focused on adding task sorting capabilities and improving user interface interactions. Anticipated completion of these features is expected by the end of the next development sprint.