SAGAR INSTITUTE OF RESEARCH AND TECHNOLOGY – EXCELLENCE

Department of Computer Science & Engineering

A Project

On

"To Do List"

Using HTML, CSS and JavaScript.

Submitted

By

Devesh Tiwari

HTML Coding

- The below code is saved by the file name index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>To-Do List</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
   <div class="container">
      <h1>To-Do List</h1>
      <div class="input-wrapper">
         <input type="text" id="taskInput" placeholder="Add a task">
         <button id="addTaskBtn">Add</button>
      </div>
      <button id="markAllCompletedBtn" class="button">Mark All
Completed</button>
      <ul id="taskList"></ul>
   </div>
 <script src="script.js"></script>
   </body>
</html>
```

CSS Coding

- The below code is saved by the file name style.css

```
* {
   box-sizing: border-box;
}

body {
   font-family: Arial, sans-serif;
   background-color: #f5f5f5;
   margin: 0;
   padding: 0;
}

.container {
   max-width: 500px;
   margin: 30px auto;
   background-color: #fff;
   padding: 20px;
   border-radius: 5px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h1 {
```

```
    text-align: center;
    color: #333;
}

.input-wrapper {
    display: flex;
    margin-bottom: 20px;
}

.input-wrapper input[type="text"] {
    flex: 1;
    padding: 10px;
    font-size: 16px;
    border: none;
    border-radius: 5px 0 0 5px;
}

.input-wrapper button {
    padding: 10px 15px;
    background-color: #007bff;
    border: none;
    color: #fff;
    font-size: 16px;
    border-radius: 0 5px 5px 0;
```

```
    cursor: pointer;
}

ul {
    list-style: none;
    padding: 0;
}

li {
    display: flex;
    align-items: center;
    margin-bottom: 10px;
    padding: 10px;
    background-color: #f9f9f9;
    border-radius: 5px;
}

li span {
    flex: 1;
}

li button {
    padding: 5px;
    background-color: #dc3545;
```

```
    border: none;
    color: #fff;
    font-size: 14px;
    margin-left: 10px;
    border-radius: 50%;
    cursor: pointer;
}

li.completed {
    background-color: #d4edda;
    color: #155724;
    text-decoration: line-through;
  }

  li.completed button {
    background-color: #dc3545;
  }

.button {
    display: inline-block;
    padding: 10px 15px;
    background-color: #007bff;
    border: none;
    color: #fff;
```

```
  font-size: 16px;
  border-radius: 5px;
  cursor: pointer;
  transition: background-color 0.3s ease;
}

.button:hover {
  background-color: #0056b3;
}
 #markAllCompletedBtn {
  margin-right: 10px;
}
```

JavaScript Coding

- The below code is saved by the file name script.js

```
// Get required elements from the DOM
const taskInput = document.getElementById('taskInput');
const addTaskBtn = document.getElementById('addTaskBtn');
const taskList = document.getElementById('taskList');

// Add event listener to the "Add" button
addTaskBtn.addEventListener('click', addTask);

// Function to add a new task
function addTask() {
   const task = taskInput.value.trim();

   if (task !== '') {
      // Create new task item
      const li = document.createElement('li');
      li.innerHTML = `<span>${task}</span><button
onclick="removeTask(this)">X</button>`;
// Append task item to the list
      taskList.appendChild(li);

      // Clear the input field
```

```
        taskInput.value = '';
    }
}

// Function to remove a task
function removeTask(button) {
    const li = button.parentNode;
    li.parentNode.removeChild(li);
}

// Function to toggle task completion
function toggleTaskCompletion(checkbox) {
    const li = checkbox.parentNode.parentNode;
    li.classList.toggle('completed');
  }

  // Function to add a new task
  function addTask() {
    const task = taskInput.value.trim();

    if (task !== '') {
      // Create new task item
      const li = document.createElement('li');
```

```
    li.innerHTML = `<span>${task}</span><input type="checkbox"
onchange="toggleTaskCompletion(this)"><button
onclick="removeTask(this)">X</button>`;

    // Append task item to the list
    taskList.appendChild(li);

    // Clear the input field
    taskInput.value = '';
  }
 }

// Function to add a new task
function addTask() {
  const task = taskInput.value.trim();

  if (task !== '') {
   // Create new task item
   const li = document.createElement('li');
   li.innerHTML = `<span>${task}</span><input type="checkbox"
onchange="toggleTaskCompletion(this)"><button
onclick="removeTask(this)">X</button>`;

   // Append task item to the list
   taskList.appendChild(li);
```

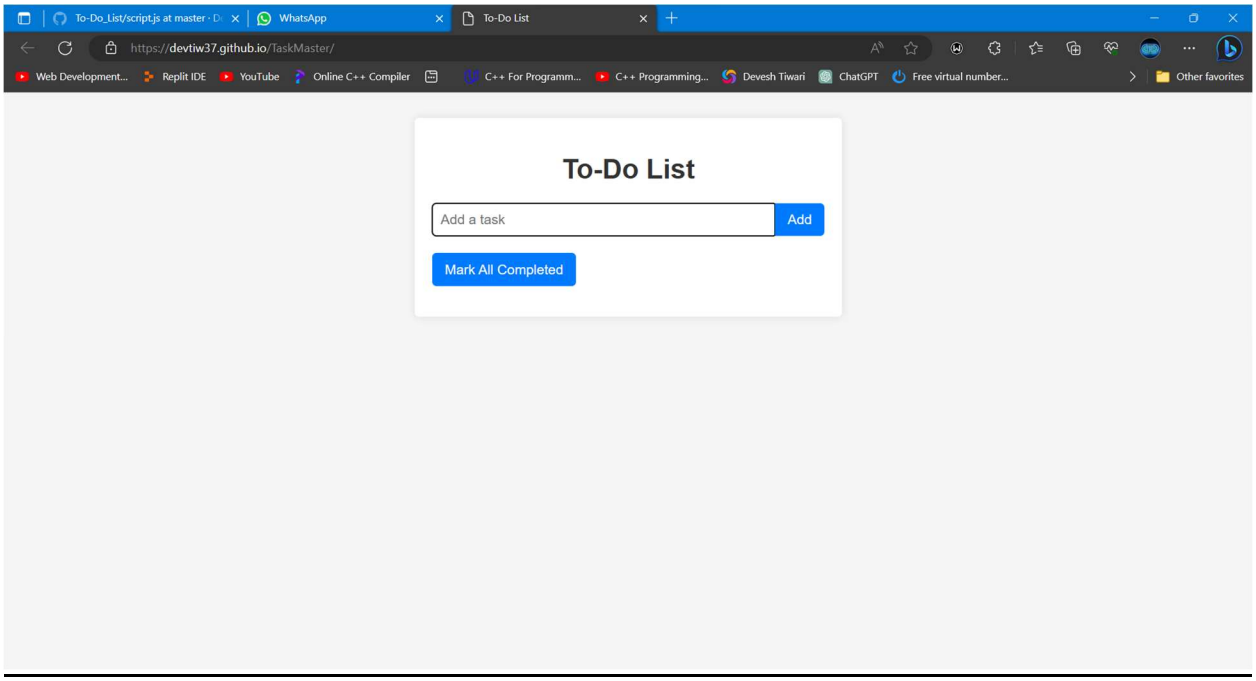
```
    // Save tasks to local storage
    saveTasksToLocalStorage();

    // Clear the input field
    taskInput.value = '';
  }
}

// Function to remove a task
function removeTask(button) {
  const li = button.parentNode;
  li.parentNode.removeChild(li);

  // Save tasks to local storage
  saveTasksToLocalStorage();
}

// Function to toggle task completion
function toggleTaskCompletion(checkbox) {
  const li = checkbox.parentNode.parentNode;
  li.classList.toggle('completed');

  // Save tasks to local storage
  saveTasksToLocalStorage();
```

```
}

// Function to save tasks to local storage
function saveTasksToLocalStorage() {
  const tasks = Array.from(taskList.getElementsByTagName('li')).map(li => ({
    task: li.getElementsByTagName('span')[0].textContent,
    completed: li.classList.contains('completed')
  }));

  localStorage.setItem('tasks', JSON.stringify(tasks));
}

// Function to load tasks from local storage
function loadTasksFromLocalStorage() {
  const tasks = JSON.parse(localStorage.getItem('tasks'));

  if (tasks) {
    tasks.forEach(task => {
      const li = document.createElement('li');
      li.innerHTML = `<span>${task.task}</span><input type="checkbox"
onchange="toggleTaskCompletion(this)"><button
onclick="removeTask(this)">X</button>`;

      if (task.completed) {
        li.classList.add('completed');
```

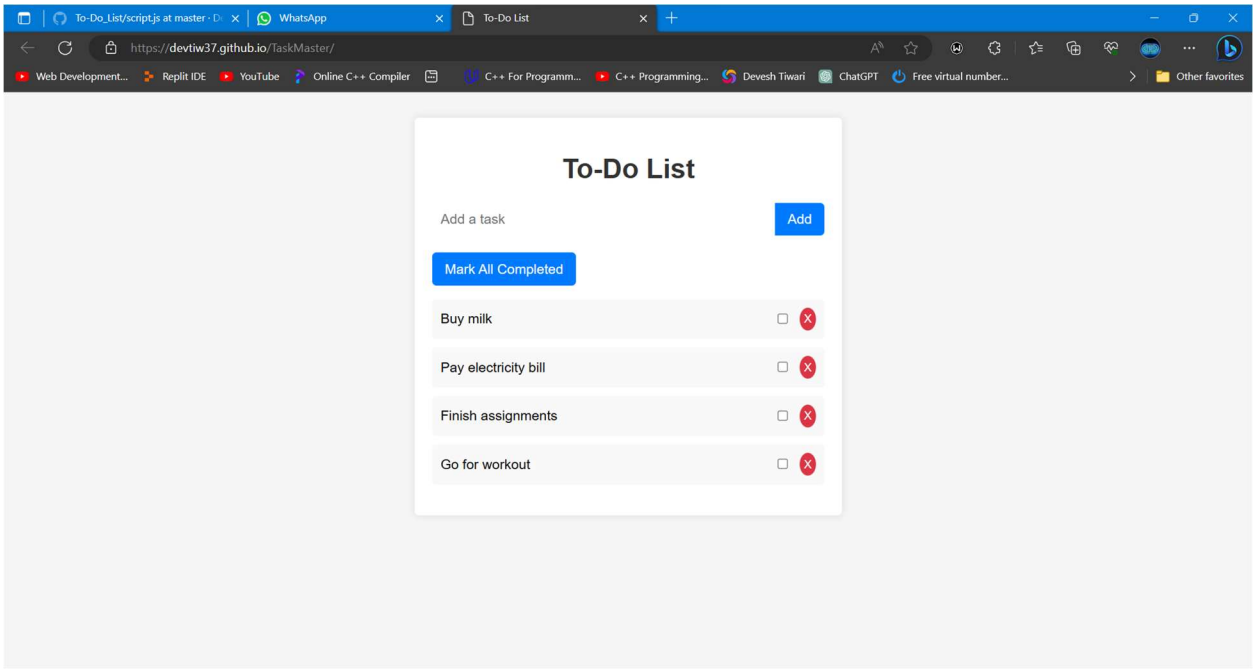
```
      }
       taskList.appendChild(li);
    });
  }
}

// Load tasks from local storage when the page loads
loadTasksFromLocalStorage();

// Add event listener to the "Mark All Completed" button
const markAllCompletedBtn =
document.getElementById('markAllCompletedBtn');
markAllCompletedBtn.addEventListener('click', markAllTasksCompleted);

// Function to mark all tasks as completed
function markAllTasksCompleted() {
  const tasks = Array.from(taskList.getElementsByTagName('li'));
  tasks.forEach(li => {
    li.classList.add('completed');
  });
// Save tasks to local storage
  saveTasksToLocalStorage();
}
```

Overview Of UI

- This is the input box where task is entered.



- You can add as many tasks as required.



- Once the given task is finished, you can click on the check box or click on “Mark All Completed” to finish all tasks at once.