To-Do List Web App

1. Project Title

To-Do List Web Application

2. Introduction

The To-Do List Web Application is a simple productivity tool that allows users to add, manage, and delete tasks. It helps users keep track of their daily activities in an organized way. This project was chosen because it demonstrates the core concepts of web development, including HTML, CSS, and JavaScript, while also being highly practical for everyday use.

3. Working

1. **The user opens the web application in a browser.**
2. **The user can enter a task in the input field and click the "Add Task" button.**
3. **The task gets displayed in the task list below.**
4. **Users can mark tasks as completed or delete tasks when they are done.**
5. **The tasks are dynamically updated using JavaScript, giving an interactive user experience.**

4. Uses

- **Helps users organize daily tasks.**
- **Improves productivity and time management.**
- **Can be extended to store tasks permanently in local storage or a database.**

5. Advantages & Disadvantages

Advantages:

- **Simple and easy-to-use interface.**
- **Lightweight, works on any browser.**
- **Interactive and responsive.**

Disadvantages:

- **Tasks are lost when the page is refreshed (unless connected to storage).**
- **No user authentication (anyone can use it).**

`

6. Future Scope

- **Add Local Storage / Database support to save tasks permanently.**
- **Add User Login system so that different users can have their own to-do lists.**
- **Add features like due dates, reminders, and categories for better task management.**
- **Create a mobile app version for cross-platform use.**

7. Code Implementation

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 App</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <h2>To-Do List</h2>
    <input type="text" id="taskInput" placeholder="Enter a task">
    <button onclick="addTask()">Add</button>
    <ul id="taskList"></ul>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

style.css

```
body {
  font-family: Arial, sans-serif;
  background: #f4f4f4;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  background: #fff;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0,0,0,0.2);
  width: 350px;
}

h2 {
  text-align: center;
}

input {
  width: 70%;
  padding: 8px;
  border: 1px solid #ddd;
  border-radius: 5px;
}

`
```

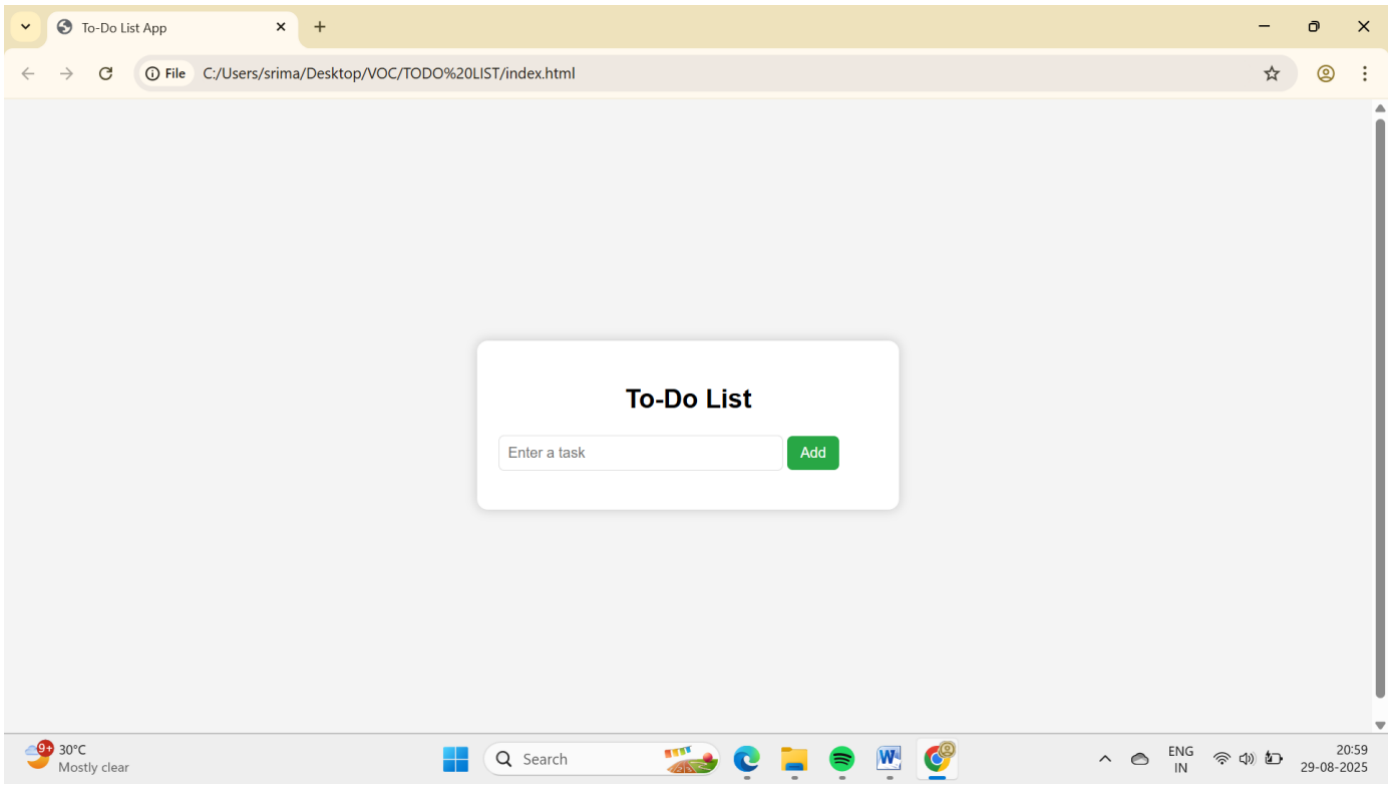
```
button {
  padding: 8px 12px;
  border: none;
  background: #28a745;
  color: white;
  border-radius: 5px;
  cursor: pointer;
}

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

li {
  background: #eee;
  padding: 8px;
  margin: 5px 0;
  border-radius: 5px;
  display: flex;
  justify-content: space-between;
}

li.completed {
  text-decoration: line-through;
  color: gray;
}

li button {
  background: #dc3545;
  border: none;
  color: white;
  padding: 5px 10px;
  border-radius: 5px;
  cursor: pointer;
}
```

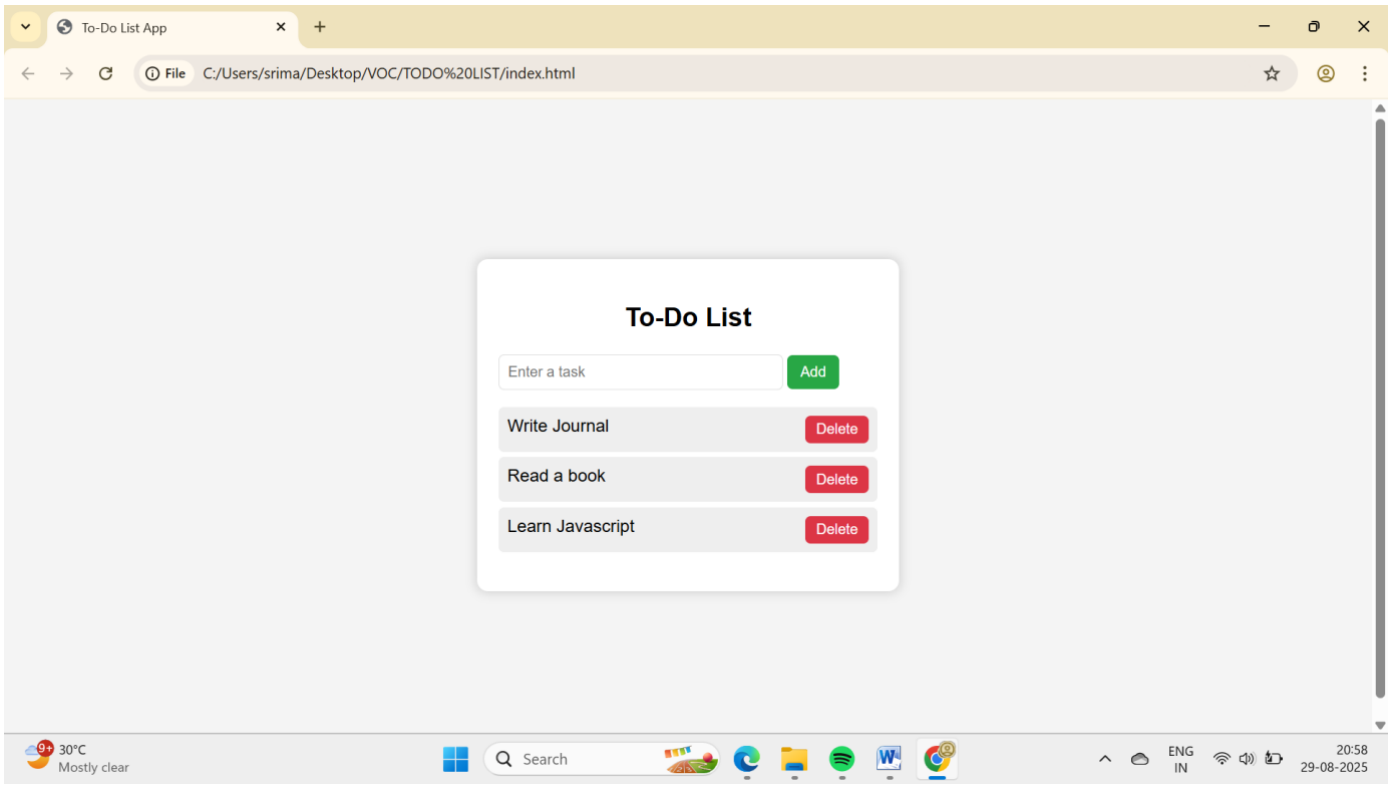
script.js

```
function addTask() {
  let input = document.getElementById("taskInput");
  let taskText = input.value.trim();
  if (taskText === "") return;

  let li = document.createElement("li");
  li.textContent = taskText;

  li.addEventListener("click", () => {
    li.classList.toggle("completed");
  });

  let delBtn = document.createElement("button");
  delBtn.textContent = "Delete";
  delBtn.onclick = () => li.remove();

  li.appendChild(delBtn);
  document.getElementById("taskList").appendChild(li);
  input.value = "";
}
```

`

8. Output





Name: SR.ARCHANA

Email: sr.archana01@gmail.com

Github: Archana01-07

`