

প্রজেক্ট নং(Project no):১

তারিখ(Date):

প্রজেক্টের নাম(Project Title): HTML, CSS, and JavaScript ব্যবহার করে টুডু লিস্ট

তৈরি করা।

ভূমিকা(Introduction):

বর্তমান সময়ে আমাদের দৈনন্দিন কাজগুলোকে গুছিয়ে রাখার জন্য ডিজিটাল টুলস খুবই গুরুত্বপূর্ণ। এই প্রজেক্টের মূল লক্ষ্য হলো একটি ইউজার-ফ্রেন্ডলি **Todo List Application** তৈরি করা, যার মাধ্যমে একজন ব্যবহারকারী তার প্রতিদিনের কাজের তালিকা তৈরি করতে পারবেন, কাজ সম্পন্ন হলে তা চিহ্নিত করতে পারবেন এবং অপ্রয়োজনীয় টাস্ক মুছে ফেলতে পারবেন।

প্রজেক্টের উদ্দেশ্য(Objectives):

1. একটি পরিষ্কার এবং রেসপনসিভ ইউজার ইন্টারফেস (UI) তৈরি করা।
2. HTML-এর মাধ্যমে পেজের গঠন (Structure) তৈরি করা।
3. CSS-এর মাধ্যমে অ্যানিমেশনটিকে আকর্ষণীয় ডিজাইন দেওয়া।
4. JavaScript ব্যবহার করে ডাইনামিক ফিচার (যেমন: টাস্ক অ্যাড করা, ডিলিট করা) যুক্ত করা।

ব্যবহৃত প্রযুক্তি(Technologies Used):

অ্যাসাইনমেন্টের কোড তৈরিতে নিচের প্রযুক্তিগুলো ব্যবহার করা হয়েছে :

- HTML5: ওয়েব পেজের কঙ্কাল বা স্ট্রাকচার তৈরির জন্য।
- CSS3: পেজটিকে সুন্দর ও রেসপনসিভ করার জন্য।
- JavaScript (ES7+): অ্যানিমেশনে প্রাণ দেওয়ার জন্য বা লজিক হ্যান্ডেল করার জন্য।
- VS Code: কোড এডিটর হিসেবে।

ফিচারসমূহ(Key Features):

- ❖ Add Task: ইনপুট ফিল্ডে লিখে নতুন কাজ যোগ করা।
- ❖ Delete Task: কোনো কাজ ভুল হলে বা প্রয়োজন না থাকলে তা মুছে ফেলা।

❖ **Responsive Design:** এটি মোবাইল এবং কম্পিউটার-উভয় ডিভাইসেই সুন্দরভাবে দেখা যাবে।

❖ **Task Management:** কাজের লিস্ট সিরিয়াল অনুযায়ী দেখা যাবে।

কোড ইমপ্লিমেন্টেশন (Code Implementation):

HTML:

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <script src="main.js" defer></script>
    <title>Document</title>
  </head>
  <body>
    <div class="container">
      <form id="task-form" class="task-form">
        <input
          name="newTask"
          class="task-input"
          type="text"
          placeholder="Add a new task!"
        />
        <button type="submit">
          <svg
            xmlns="http://www.w3.org/2000/svg"
            width="24"
            height="24"
            viewBox="0 0 24 24"
            fill="none"
            stroke="currentColor"
            stroke-width="2"
            stroke-linecap="round"
            stroke-linejoin="round"
            class="lucide lucide-plus-icon lucide-plus"
          >
            <path d="M5 12h14" />
            <path d="M12 5v14" />
          </svg>
        </button>
      </form>
    </div>
  </body>
</html>
```

```
    </form>
    <ul id="todo-list" class="todo-list"></ul>
  </div>
</body>
</html>
```

CSS:

```
@import
url("https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100..900;1,100..
900&display=swap");

* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  background: #eeaeca;
  background: radial-gradient(
    circle,
    rgba(238, 174, 202, 1) 0%,
    rgba(148, 187, 233, 1) 100%
  );

  font-family: "Roboto", sans-serif;
}

.container {
  background-color: #eef4fc;
  padding: 20px;
  border-radius: 10px;

  width: 40%;
  height: 80vh;

  margin: 0 auto;
  margin-top: 30px;

  overflow: scroll;
  scrollbar-width: none;
}
```

```
.task-input {
  width: 100%;
  height: 30px;
  background-color: #eef4fc;
  border: 1px solid #00171f51;
  border-radius: 6px;
  padding: 5px 10px;
}

.task-input::placeholder {
  color: #00171f51;
}

.todo-list {
  list-style: none;
  display: flex;
  flex-direction: column;
}

.todo-item {
  display: flex;
  align-items: center;
  gap: 8px;
  border-bottom: 1px solid #00171f51;
  padding: 16px 0;
}

.todo-item p {
  flex: 1;
  line-height: 0;
}

.delete-button {
  background-color: rgba(255, 0, 0, 0.25);
  border: 0;
  padding: 5px;
  display: flex;
  align-items: center;
  justify-content: center;
  width: 30px;
  height: 30px;
  border-radius: 5px;
  cursor: pointer;
}
```

```
.delete-button svg {
  color: rgba(255, 0, 0, 0.8);
  width: 15px;
}

.task-form {
  display: flex;
  align-items: center;
  gap: 10px;
}

.task-form button {
  background-color: rgba(148, 187, 233, 1);
  border: 0;
  padding: 5px;
  display: flex;
  align-items: center;
  justify-content: center;
  width: 30px;
  height: 30px;
  border-radius: 5px;
  cursor: pointer;
}

.task-form button svg {
  color: white;
  width: 15px;
}
```

Java Script:

```
// Store tasks
let tasks = JSON.parse(localStorage.getItem("TODOLIST_TASKS")) || [];

console.log(tasks);

// Selecting elements
const taskForm = document.querySelector("#task-form");
const todoList = document.querySelector("#todo-list");

// Displaying the todo list UI
function displayTodoList() {
```

```

todoList.innerHTML = null;

tasks.forEach(function (task) {
  const todoItem = document.createElement("li");
  todoItem.setAttribute("class", "todo-item");
  todoItem.innerHTML = `
    <input type="checkbox" ${task.isComplete ? "checked" : ""} />
    <p ${task.isComplete ? 'style="text-decoration: line-through"' :
""}>${task.task}</p>
    <button class="delete-button">
      <svg xmlns="http://www.w3.org/2000/svg" width="24" height="24"
viewBox="0 0 24 24" fill="none" stroke="currentColor" stroke-width="2" stroke-
linecap="round" stroke-linejoin="round" class="lucide lucide-trash2-icon lucide-
trash-2"><path d="M10 11v6"/><path d="M14 11v6"/><path d="M19 6v14a2 2 0 0 1-2
2H7a2 2 0 0 1-2-2V6"/><path d="M3 6h18"/><path d="M8 6V4a2 2 0 0 1 2-2h4a2 2 0 0
1 2 2v2"/></svg>
    </button>
  `;

  handleTaskCompletion(todoItem, task);
  handleTaskDeletion(todoItem, task);

  todoList.appendChild(todoItem);
});
}

function handleTaskCompletion(todoItem, task) {
  const todoCheckbox = todoItem.querySelector("input");

  todoCheckbox.addEventListener("click", function () {
    const foundTaskIndex = tasks.findIndex(function (currentTask) {
      return currentTask.task == task.task;
    });

    const status = tasks[foundTaskIndex].isComplete;

    if (status) {
      tasks[foundTaskIndex].isComplete = false;
    } else {
      tasks[foundTaskIndex].isComplete = true;
    }

    localStorage.setItem("TODOLIST_TASKS", JSON.stringify(tasks));
    displayTodoList();
  });
}

```

```

}

function handleTaskDeletion(todoItem, task) {
  const deleteButton = todoItem.querySelector("button");

  deleteButton.addEventListener("click", function () {
    tasks = tasks.filter(function (currentTask) {
      return currentTask.task !== task.task;
    });

    localStorage.setItem("TODOLIST_TASKS", JSON.stringify(tasks));
    displayTodoList();
  });
}

// New task input
taskForm.addEventListener("submit", function (event) {
  event.preventDefault();
  const userInput = taskForm.querySelector("input");
  const userInputValue = userInput.value.trim();

  const taskAlreadyExists = tasks.some(function (currentTask) {
    return currentTask.task === userInputValue;
  });

  if (taskAlreadyExists) return;

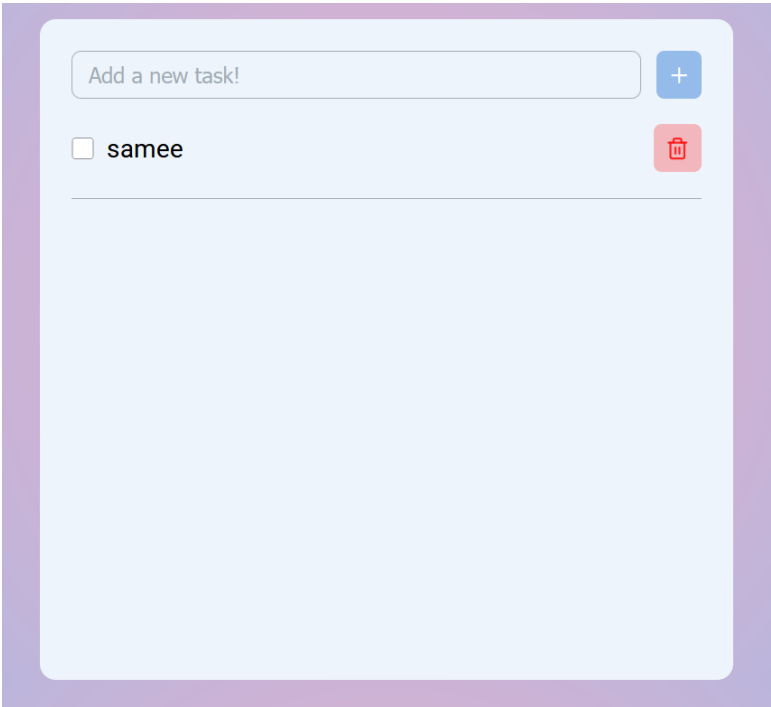
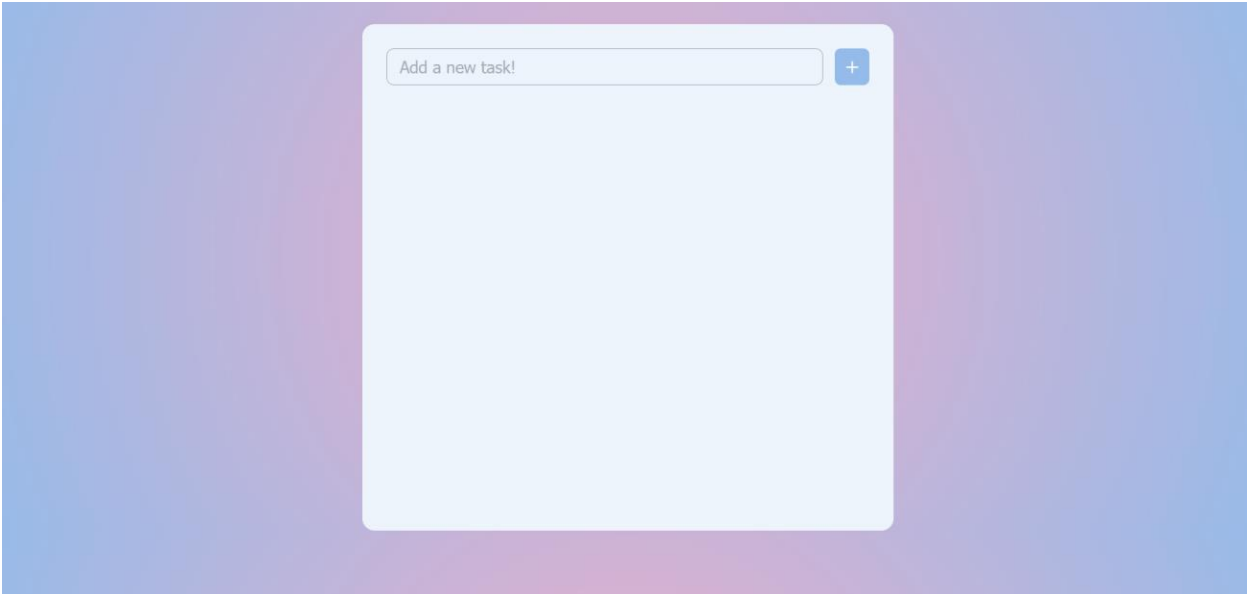
  const newTask = {
    task: userInputValue,
    isComplete: false,
  };

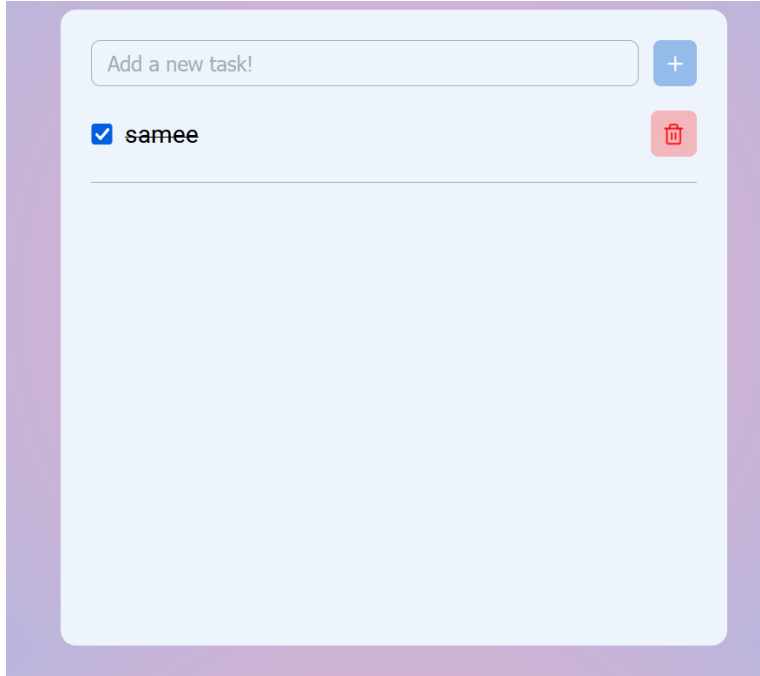
  tasks.push(newTask);
  localStorage.setItem("TODOLIST_TASKS", JSON.stringify(tasks));
  userInput.value = null;
  displayTodoList();
});

displayTodoList();

```

আউটপুট(Output):





উপসংহার (Conclusion):

এই প্রজেক্টটির মাধ্যমে আমি ওয়েব ডেভেলপমেন্টের মৌলিক বিষয়গুলো, বিশেষ করে **DOM Manipulation** এবং ইভেন্ট হ্যান্ডলিং সম্পর্কে বাস্তব অভিজ্ঞতা অর্জন করেছি। ভবিষ্যতে এই অ্যাপ্লিকেশনে **Local Storage** যুক্ত করার পরিকল্পনা রয়েছে যাতে ব্রাউজার রিফ্রেশ দিলেও ডেটা হারিয়ে না যায়।