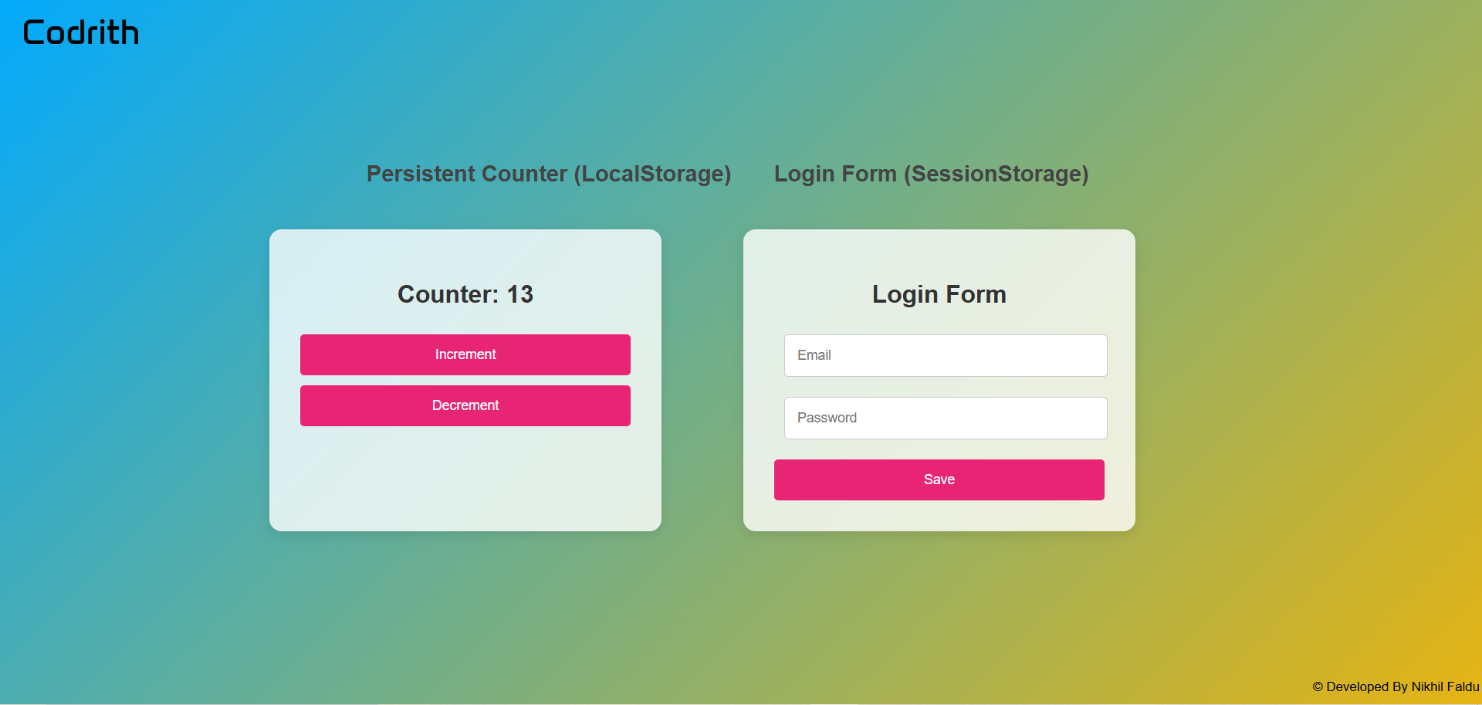
**Local and Session storage:**

1. **Task Description**

This project demonstrates the use of Local Storage and Session Storage in React.js through two interactive components, showcasing the differences in data persistence between these storage options.

1. **Task Output Screenshot**
2. **Description of Widget/Algorithm**

**Widgets Used:**

1. **Local Storage for Counter**:
   * Counter value is initialized using localStorage.getItem.
   * Counter value is saved to local storage using localStorage.setItem on every update.
2. **Session Storage for Email and Password**:
   * Email and password fields update state with setEmail and setPassword.
   * On form submission, values are stored in session storage using sessionStorage.setItem.
   * Fields reset on tab reload to simulate session expiration.

**Algorithm:**

* **Counter Initialization and Update**:
  + Check if a counter value exists in local storage.
  + Load the value if found; otherwise, start at 0.
  + Save updated value back to local storage on change.
* **Email and Password Management**:
  + On form submission, save email and password in session storage.
  + Reset fields on page load to mimic session expiration.