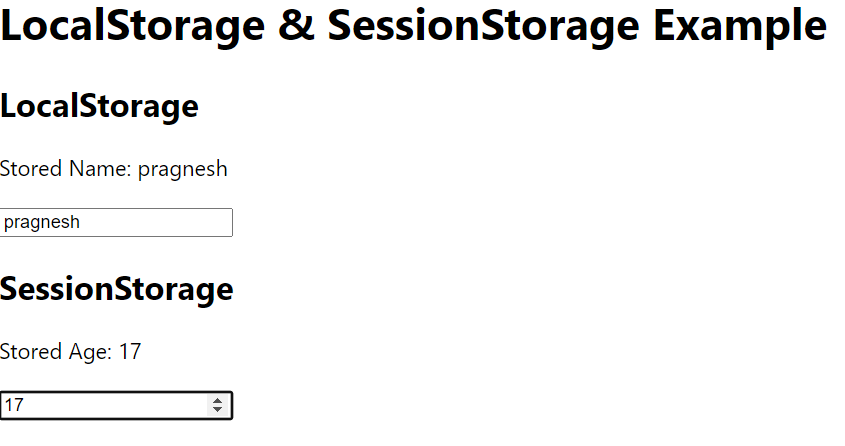
### **Task Description**

Create a custom React hook that simplifies the management of browser storage (either localStorage or sessionStorage). This hook provides an easy way to save, retrieve, and remove data from the storage, making state persistence between page reloads more convenient.

### **Task Output Screenshot**



### **Widget/Algorithm Used in Task**

* **Custom Hook (useStorage)**: A custom hook is created that abstracts the logic for managing localStorage or sessionStorage. The hook takes in two parameters: the key and the storage type (local or session).
* **useState and useEffect Hooks**:
  + useState is used to store the value in memory.
  + useEffect is used to sync the state with the storage, ensuring that any changes made are reflected in localStorage or sessionStorage.
* **Storage Type Parameter**: The hook accepts a parameter to determine whether to use localStorage or sessionStorage. This allows flexibility in how data is stored across sessions.
* **Set and Get Functions**:
  + setValue: A function to store a value in the specified storage.
  + getValue: A function to retrieve the value from the storage based on the provided key.
* **Data Persistence**: The custom hook ensures that even after a page reload, the stored data can be retrieved from the chosen storage method.
* **Error Handling**: The hook includes error handling for storage limits and cases where access to storage may not be allowed (e.g., in private browsing).