

Ex. No: 5	React App Development
22.08.2023	

**Aim:**

To create an app using React with Components, Rendering, and Data Sharing.

**Program:**

App.jsx

```
import { useEffect, useState } from "react"
import { NewTodoForm } from "../NewTodoForm"
import "../styles.css"
import { TodoList } from "../TodoList"

export default function App() {

  const [todos, setTodos] = useState(() => {
    const localValue = localStorage.getItem("ITEMS")
    if(localValue == null) return []
    return JSON.parse(localValue)
  })

  useEffect(() => {
    localStorage.setItem("ITEMS", JSON.stringify(todos))
  }, [todos])

  function addTodo(title) {
    setTodos(currentTodos => {
      return [
        ...currentTodos,
        { id: crypto.randomUUID(), title, completed: false },
      ]
    })
  }

  function toggleTodo(id, completed) {
    setTodos(currentTodos => {
      return currentTodos.map(todo => {
        if (todo.id === id) {
          return { ...todo, completed }
        }
        return todo
      })
    })
  }
}
```

```

function deleteTodo(id, completed) {
  setTodos(currentTodos => {
    return currentTodos.filter(todo => todo.id !== id)
  })
}

return (
  <>
    <NewTodoForm onSubmit={addTodo}/>
    <h1 className="header">Todo List</h1>
    <TodoList todos={todos} toggleTodo={toggleTodo} deleteTodo={deleteTodo}/>
  </>
)
}

```

## NewTodoForm.jsx

```

import { useState } from "react"

export function NewTodoForm({onSubmit}) {

  const [newItem, setNewItem] = useState("")

  function handleSubmit(e){
    e.preventDefault()
    if (newItem === "") return
    onSubmit(newItem)

    setNewItem("")
  }

  return (
    <form onSubmit={handleSubmit} className="new-item-form">
      <div className="form-row">
        <label htmlFor="item">New Item</label>
        <input
          value={newItem}
          onChange={e => setNewItem(e.target.value)}
          type="text"
          id="item"
        />
      </div>
      <button className="btn">Add</button>
    </form>
  )
}

```

## TodoList.jsx

```
import { TodoItem } from "./TodoItem"

export function TodoList({todos, toggleTodo, deleteTodo}) {
  return (
    <ul className="list">
      {todos.length === 0 && "No Todos"}
      {todos.map(todo => {
        return (
          <TodoItem
            {...todo}
            key = {todo.id}
            toggleTodo={toggleTodo}
            deleteTodo={deleteTodo}
          />
        )
      })}
    </ul>)
}
```

## TodoItem.jsx

```
export function TodoItem ({completed, id, title, toggleTodo, deleteTodo}) {
  return (
    <li>
      <label>
        <input
          type="checkbox"
          checked={completed}
          onChange={e => toggleTodo(id, e.target.checked)}
        />
        {title}
      </label>
      <button
        onClick={() => deleteTodo(id)}
        className="btn btn-danger">
        Delete
      </button>
    </li>
  )
}
```

## Styles.css

```
* {
  font-family: Arial, Helvetica, sans-serif;
  box-sizing: border-box;
}

body {
  background: #333;
```

```
    color: hsl(200, 100%, 90%);
    max-width: 400px;
    padding: 1rem;
    margin: 0 auto;
}

.new-item-form {
    display: flex;
    flex-direction: column;
    gap: 0.5rem;
}

.form-row {
    display: flex;
    flex-direction: column;
    gap: 0.1rem;
}

.btn {
    background: hsl(200, 100%, 50%, 0.1);
    border: 1px solid hsl(200, 100%, 50%);
    color: hsl(200, 100%, 50%);
    padding: 0.25em 0.5em;
    border-radius: 0.25em;
    cursor: pointer;
    outline: none;
}

.btn:hover,
.btn:focus-visible {
    background: hsl(200, 100%, 50%, 0.2);
}

.btn.btn-danger {
    background: hsl(0, 100%, 40%, 0.1);
    border: 1px solid hsl(0, 100%, 40%);
    color: hsl(0, 100%, 40%);
}

.btn.btn-danger:hover,
.btn.btn-danger:focus-visible {
    background: hsl(0, 100%, 40%, 0.2);
}

.new-item-form input {
    outline: none;
    border: 1px solid hsl(200, 100%, 40%);
    background: hsl(200, 100%, 30%);
    border-radius: 0.25em;
    padding: 0.25em 0.5em;
    color: hsl(200, 100%, 90%);
}
```

```
}

.new-item-form input:focus {
  border: 1px solid hsl(200, 100%, 70%);
}

.header {
  font-size: 1.5rem;
  margin-top: 1.5rem;
  margin-bottom: 0.5rem;
}

.list {
  margin: 0;
  padding: 0;
  margin-left: 1rem;
  list-style: none;
}

.list li:has(input:checked) label {
  color: hsl(200, 20%, 40%);
}

.list {
  display: flex;
  flex-direction: column;
  gap: 0.3rem;
}

.list li {
  display: flex;
  gap: 0.5rem;
  align-items: center;
}

.list li label {
  display: flex;
  gap: 0.25rem;
  cursor: pointer;
  align-items: center;
}

.list li:has(input:focus-visible) label {
  outline: 1px solid hsl(200, 100%, 50%);
}

.list li input {
  outline: none;
  width: 0;
  height: 0;
  appearance: none;
}
```

```

    pointer-events: none;
    position: absolute;
}

.list li label::before {
    content: "";
    display: block;
    width: 0.9rem;
    height: 0.9rem;
    background: hsl(200, 100%, 90%);
    border-radius: 0.25em;
    display: flex;
    justify-content: center;
    align-items: center;
}

.list li label:hover::before {
    background: hsl(200, 100%, 80%);
}

.list li:has(input:checked) label::before {
    content: "✓";
    background: hsl(200, 100%, 40%);
    color: hsl(200, 100%, 90%);
    font-size: 0.75rem;
    font-weight: bold;
}

.list li:has(input:checked) label:hover::before {
    background: hsl(200, 100%, 30%);
}

```

## Output:

New Item



## Todo List

- ☒ Item 1
- ☐ Item 2
- ☐ Item 3

**Result:**

Therefore, we've successfully an app using React with Components, Rendering, and Data Sharing.