import React, { Component, useState } from 'react';

import './App.css';

// TaskManager Component (Class Component)

class TaskManager extends Component {

constructor(props) {

super(props);

this.state = {

tasks: [

{ id: 1, text: 'Sample Task 1', completed: false },

{ id: 2, text: 'Sample Task 2', completed: true },

],

newTaskText: '',

};

}

// Handle new task input change

handleInputChange = (e) => {

this.setState({ newTaskText: e.target.value });

};

// Add new task to the list

addTask = (e) => {

e.preventDefault();

if (this.state.newTaskText.trim() === '') return;

const newTask = {

id: Date.now(),

text: this.state.newTaskText,

completed: false,

};

this.setState((prevState) => ({

tasks: [...prevState.tasks, newTask],

newTaskText: '',

}));

};

// Toggle task completion

toggleCompletion = (id) => {

this.setState((prevState) => ({

tasks: prevState.tasks.map((task) =>

task.id === id ? { ...task, completed: !task.completed } : task

),

}));

};

// Delete a task

deleteTask = (id) => {

this.setState((prevState) => ({

tasks: prevState.tasks.filter((task) => task.id !== id),

}));

};

render() {

return (

<div className="task-manager">

<h1>Task Manager</h1>

<TaskForm

newTaskText={this.state.newTaskText}

onInputChange={this.handleInputChange}

onAddTask={this.addTask}

/>

<TaskList

tasks={this.state.tasks}

onToggleCompletion={this.toggleCompletion}

onDeleteTask={this.deleteTask}

/>

</div>

);

}

}

// TaskForm Component (Controlled Component)

const TaskForm = ({ newTaskText, onInputChange, onAddTask }) => (

<form onSubmit={onAddTask} className="task-form">

<input

type="text"

value={newTaskText}

onChange={onInputChange}

placeholder="Add a new task..."

/>

<button type="submit">Add Task</button>

</form>

);

// TaskList Component (Functional Component with Conditional and Looping Constructs)

const TaskList = ({ tasks, onToggleCompletion, onDeleteTask }) => (

<div className="task-list">

{tasks.length === 0 ? (

<p>No tasks to display.</p>

) : (

tasks.map((task) => (

<TaskItem

key={task.id}

task={task}

onToggleCompletion={onToggleCompletion}

onDeleteTask={onDeleteTask}

/>

))

)}

</div>

);

// TaskItem Component (Individual Task Component)

const TaskItem = ({ task, onToggleCompletion, onDeleteTask }) => (

<div className={`task-item ${task.completed ? 'completed' : ''}`}>

<input

type="checkbox"

checked={task.completed}

onChange={() => onToggleCompletion(task.id)}

/>

<span>{task.text}</span>

<button onClick={() => onDeleteTask(task.id)}>Delete</button>

</div>

);

export default TaskManager;