

Experiment 8

Develop a Reminder Application that allows users to efficiently manage their tasks. The application should include the following functionalities: Provide a form where users can add tasks along with due dates. The form includes task name, Due date, An optional description. Display a list of tasks dynamically as they are added. Show relevant details like task name, due date, and completion status. Include a filter option to allow users to view all Tasks and Display all tasks regardless of status. Show only tasks marked as completed. Show only tasks that are not yet completed.

App.js

```
function App() {  
  return (  
    <div>  
      <Reminder />  
    </div>  
  );  
}  
export default App;
```

Reminder.js

```
import React, { useState } from "react";
import './App.css';

function Reminder() {
  const [tasks, setTasks] = useState([]);
  const [task, setTask] = useState({ name: "", due: "", desc: "" });
  const [filter, setFilter] = useState("all");
  const addTask = () => {
    if (!task.name || !task.due) return;
    setTasks([...tasks, { ...task, id: Date.now(), completed: false }]);
    setTask({ name: "", due: "", desc: "" });
  };
  const toggleCompletion = (id) => {
    setTasks(tasks.map(t => t.id === id ? { ...t, completed: !t.completed } : t));
  };
  const filteredTasks = tasks.filter(t =>
    filter === "all" ? true : filter === "completed" ? t.completed :
    !t.completed
  );
  return (
    <div className="container">
      <h2>Add New Task</h2>
```

```
<input placeholder="Task Name" value={task.name} onChange={e
=> setTask({ ...task, name: e.target.value })} />
```

```
<input type="date" value={task.due} onChange={e => setTask({
...task, due: e.target.value })} />
```

```
<textarea placeholder="Description (optional)" value={task.desc}
onChange={e => setTask({ ...task, desc: e.target.value })} />
```

```
<button onClick={addTask}>Add Task</button>
```

```
<h3>Filter</h3>
```

```
<select value={filter} onChange={e => setFilter(e.target.value)}>
```

```
<option value="all">All Tasks</option>
```

```
<option value="completed">Completed Tasks</option>
```

```
<option value="pending">Pending Tasks</option>
```

```
</select>
```

```
<ul>
```

```
{filteredTasks.map(t => (
```

```
<li key={t.id} className={`task-item ${t.completed ? "completed" :
"pending"}`} >
```

```
<div className="task-header">
```

```
<strong>{t.name}</strong>
```

```
<button onClick={() => toggleCompletion(t.id)}>
```

```
{t.completed ? "Mark as Pending" : "Mark as Completed"}
```

```
</button>
```

```
</div>
```

```
<div><strong>Due:</strong> {t.due}</div>
```

```
{t.desc && <div><strong>Note:</strong> {t.desc}</div>}
```

```
<div className="status-label">Status: {t.completed ? "Completed" :  
"Pending"}</div>
```

```
</li>
```

```
)))}
```

```
{filteredTasks.length === 0 && <p className="no-tasks">No tasks  
to display</p>}
```

```
</ul>
```

```
</div>
```

```
);
```

```
}
```

```
export default Reminder;
```

App.css

```
body {
```

```
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
background-color: #f0f4f8;
```

```
margin: 0;
```

```
padding: 20px;
```

```
color: #333;
```

```
}
```

```
.container {
```

```
max-width: 600px;
```

```
margin: auto;
```

```
background-color: #ffffff;  
padding: 30px;  
border-radius: 12px;  
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);  
}
```

```
h2, h3 {  
color: #2c3e50;  
margin-bottom: 15px;  
}
```

```
input, textarea, select, button {  
width: 100%;  
margin-bottom: 15px;  
padding: 10px;  
font-size: 15px;  
border: 1px solid #ccc;  
border-radius: 8px;  
box-sizing: border-box;  
}
```

```
textarea {  
resize: vertical;  
}
```

```
button {  
background-color: #4a90e2;  
color: white;
```

```
font-weight: bold;
border: none;
cursor: pointer;
transition: background-color 0.3s ease;
}
button:hover {
  background-color: #357abd;
}
ul {
  list-style: none;
  padding: 0;
}
.task-item {
  padding: 15px;
  margin-bottom: 12px;
  border-radius: 8px;
}
```