

ReactJS (CODE)

```
import React, { useState, useEffect } from 'react';

import './App.js'; // CSS for styling the Kanban board

export default function App() {

  // Sample data simulating an API response

  const sampleData = {

    tickets: [

      { id: "CAM-1", title: "Update User Profile Page UI", tag: ["Feature request"], userId: "usr-1", status: "Todo", priority: 4 },

      { id: "CAM-2", title: "Add Multi-Language Support", tag: ["Feature Request"], userId: "usr-2", status: "In progress", priority: 3 },

      { id: "CAM-3", title: "Optimize Database Queries", tag: ["Feature Request"], userId: "usr-2", status: "In progress", priority: 1 },

      { id: "CAM-4", title: "Implement Email Notification System", tag: ["Feature Request"], userId: "usr-1", status: "In progress", priority: 3 },

      { id: "CAM-5", title: "Enhance Search Functionality", tag: ["Feature Request"], userId: "usr-5", status: "In progress", priority: 0 },

      { id: "CAM-6", title: "Third-Party Payment Gateway", tag: ["Feature Request"], userId: "usr-2", status: "Todo", priority: 1 },

      { id: "CAM-7", title: "Create Onboarding Tutorial for New Users", tag: ["Feature Request"], userId: "usr-1", status: "Backlog", priority: 2 },

      { id: "CAM-8", title: "Implement Role-Based Access Control (RBAC)", tag: ["Feature Request"], userId: "usr-3", status: "In progress", priority: 3 },

      { id: "CAM-9", title: "Upgrade Server Infrastructure", tag: ["Feature Request"], userId: "usr-5", status: "Todo", priority: 2 },

      { id: "CAM-10", title: "Conduct Security Vulnerability Assessment", tag: ["Feature Request"], userId: "usr-4", status: "Backlog", priority: 1 },

    ],

    users: [

      { id: "usr-1", name: "Anoop Sharma", available: false },

      { id: "usr-2", name: "Yogesh", available: true },

      { id: "usr-3", name: "Shankar Kumar", available: true },

      { id: "usr-4", name: "Ramesh", available: true },

    ]

  }

}
```

```
    { id: "usr-5", name: "Suresh", available: true },  
  ],  
};
```

```
const [groupingOption, setGroupingOption] = useState('status');  
const [groupedTickets, setGroupedTickets] = useState({});
```

```
// Inline function to group tickets based on selected option
```

```
useEffect(() => {  
  setGroupedTickets(  
    sampleData.tickets.reduce((acc, ticket) => {  
      const key = ticket[groupingOption];  
      if (!acc[key]) {  
        acc[key] = [];  
      }  
      acc[key].push(ticket);  
      return acc;  
    }, {})  
  );  
}, [groupingOption]);
```

```
// Inline function to fetch user by ID
```

```
const getUserById = (userId) => sampleData.users.find((user) => user.id === userId);
```

```
return (  
  <div className="App">  
    <h1>Kanban Board</h1>  
    <div className="group-by">  
      <label>Group By:</label>  
      <select value={groupingOption} onChange={(e) => setGroupingOption(e.target.value)}>  
        <option value="status">Status</option>
```

```

    <option value="priority">Priority</option>
    <option value="userId">User</option>
  </select>
</div>

<div className="kanban-columns">
  {Object.keys(groupedTickets).map((group) => (
    <div key={group} className="kanban-column">
      <h2>
        {groupingOption === 'userId'
          ? getUserById(group)?.name || 'Unknown User'
          : group}
      </h2>
      {groupedTickets[group].map((ticket) => (
        <div key={ticket.id} className="kanban-card">
          <h3>{ticket.title}</h3>
          <p>Status: {ticket.status}</p>
          <p>Priority: {ticket.priority}</p>
          <p>Tags: {ticket.tag.join(', ')}</p>
          <p>Assigned to: {getUserById(ticket.userId)?.name || 'Unknown'}</p>
        </div>
      ))}
    </div>
  ))}
</div>

  ))}
</div>
</div>
);
}

```

CSS (CODE)

```
/* App.css */  
  
.App {  
  font-family: 'Arial', sans-serif;  
  padding: 20px;  
}
```

```
.kanban-board {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}
```

```
.group-by {  
  margin-bottom: 20px;  
}
```

```
.group-by label {  
  margin-right: 10px;  
}
```

```
.kanban-columns {  
  display: flex;  
  justify-content: space-around;  
}
```

```
.kanban-column {  
  background-color: #f4f4f4;  
  padding: 15px;
```

```
border-radius: 8px;  
width: 30%;  
border: 1px solid #ccc;  
}
```

```
.kanban-column h2 {  
text-align: center;  
background-color: #007bff;  
color: white;  
padding: 10px;  
border-radius: 4px;  
}
```

```
.kanban-card {  
background-color: #fff;  
padding: 10px;  
border: 1px solid #ddd;  
border-radius: 5px;  
margin-bottom: 10px;  
}
```

```
.kanban-card h3 {  
margin: 0;  
font-size: 1.1em;  
}
```