Here are some of the React features I’ve used. While I could have completed this basic task with just fundamental features, I’ve incorporated advanced ones to showcase my skills within a limited timeframe.

1. **Functional Components**: Used [Seat](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and SeatMap to define reusable UI components.
2. **React Hooks**: Used [useState](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [useRef](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), and [useEffect](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") for state, DOM references, and side effects.
3. **React Router**: Used useNavigate for programmatic navigation between pages.
4. **Props**: Passed data like [seat](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [onSelect](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") to child components for dynamic rendering.
5. **Conditional Rendering**: Rendered elements like tooltips or seat availability based on conditions.
6. **Event Handling**: Handled user interactions with [onClick](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and other event handlers.
7. **Dynamic Class Names**: Applied CSS classes dynamically using template literals.
8. **React Icons**: Used react-icons (e.g., [FaChair](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")) for visual elements.
9. **Context API**: Accessed global styles using [useGlobalStyles](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") from a context.
10. **JSX**: Used JSX syntax to structure and style components.
11. **Array Mapping**: Iterated over arrays (e.g., [seat?.rawSeatCharacteristics.map(...)](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)) to render lists.
12. **Inline Styles**: Applied inline styles for specific element positioning and design.
13. **Fragments**: Used React fragments to group elements without adding extra DOM nodes.
14. **Component Composition**: Combined components like [Seat](vscode-file://vscode-app/c:/Users/k1s1s/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and StepNav to build the UI.
15. **Error Boundary**: Used to catch and handle errors gracefully in the component tree.

Here is the Live link, can click and see: <https://deluxe-toffee-e6db6d.netlify.app/>

Here is the git link for download : <https://github.com/saif364/SeatPlan.git>

**How to run the project in local.**

1. Go to command line or CMD from your operating system.
2. **Clone from above git link. By** - git clone <repository-url>
3. **Go to project folder-** cd SeatPlan
4. **Install Dependencies:** npm install
5. **Run the Development Server:** npm run dev
6. **Enjoy!!!!!!**