

Frontend Challenge – Blip Hackathon 2025

Introduction

Sports events viewer and betting demo platform. Here's what it does:

- Displays a list of sports events (matches), each showing:
 - The names of both teams
 - The date and time of the event
 - The odds for team A to win, a draw, or team B to win

Files

app.jsx — Main app logic, fetches and displays events.
event-card.jsx — Renders each event card (teams, date, odds).
handlers.js, events.js — Mock API and event data.

- Allows you to sort the list by team name or by event date.

Files

app.jsx — Handles sorting state and logic.
sort-dropdown.jsx — Dropdown UI for selecting sort order.
sorting.js — Sorting functions.

- Lets you filter events by team name using a search field.

Files

app.jsx — Handles filter state and logic.
filter-form.jsx — Search/filter input form.

- Lets you customize the background color of the event cards (color picker).

Files

app.jsx — Manages background color state and passes it to cards.
event-card.jsx — Applies the background color to each card.

- Has a “Goal Overview (WIP)” button on each event, which opens a modal window indicating that the feature is “Work In Progress”.

Files

event-card.jsx — Renders the button for each event.
overview-modal.jsx — Modal component shown when button is clicked.

In summary, it's a demo app for exploring, filtering, and viewing sports events and their odds, with some customization and interactive features.

Bugs

1 – Broken Mock API Handlers

Problem

The handlers.js file only had commented-out code, so the mock API for /api/events didn't work.

Fix

Replace the comments with real code that:

- Imports http and HttpResponse from MSW.
- Imports the mock data (events).
- Exports a handler for /api/events that returns the mock events.
- Result:
- The frontend can now fetch event data from the mock API.

2 – Invalid Event Validation Logic

Problem

The validation function checked for event.eventId (lowercase 'd'), but the data uses event.eventID (uppercase 'D'). This caused all events to be filtered out as invalid.

Fix

I changed the check to event.eventID !== null && event.eventID !== undefined and used optional chaining for safety.

Result

Only valid events (with ID, both teams, and date) are shown.

3 – Broken String Sorting

Problem

Sorting by team names used subtraction (a.details.teamA - b.details.teamA), which doesn't work for strings.

Fix

Replace the subtraction with localeCompare, which correctly sorts strings alphabetically.

Result

Sorting by team names now works as expected.

4 – Non-functional Dropdown

Problem

The sort dropdown used onChange on buttons, which doesn't trigger on click.

Fix

Change onChange to onClick for the dropdown items.

Result

Clicking a dropdown item now changes the sort order.

5 – Broken Form Submission

Problem

The filter form didn't prevent the default form submission, so the page would reload and filtering didn't work.

Fix

Added a handleSubmit function that calls e.preventDefault() and then triggers the filter logic.

Result

Filtering by team name now works without reloading the page.

6 – App State Management Issues

Problem

- There were duplicate useEffect calls for fetching events.
- The loading state wasn't managed properly.
- The effect depended on sortKey, causing unnecessary re-fetches.

Fix

- I removed the duplicate useEffect.
- I set loading to true before fetching and false after.
- I made sure the effect only runs once on mount.

Result

The app shows a loading indicator correctly and only fetches events once.