



INTRO & TECH STACK



Vite + React + TypeScript

Fast development with type safety



Material-UI

Professional components with mobile-responsive, accessible design out of box



RTK Query

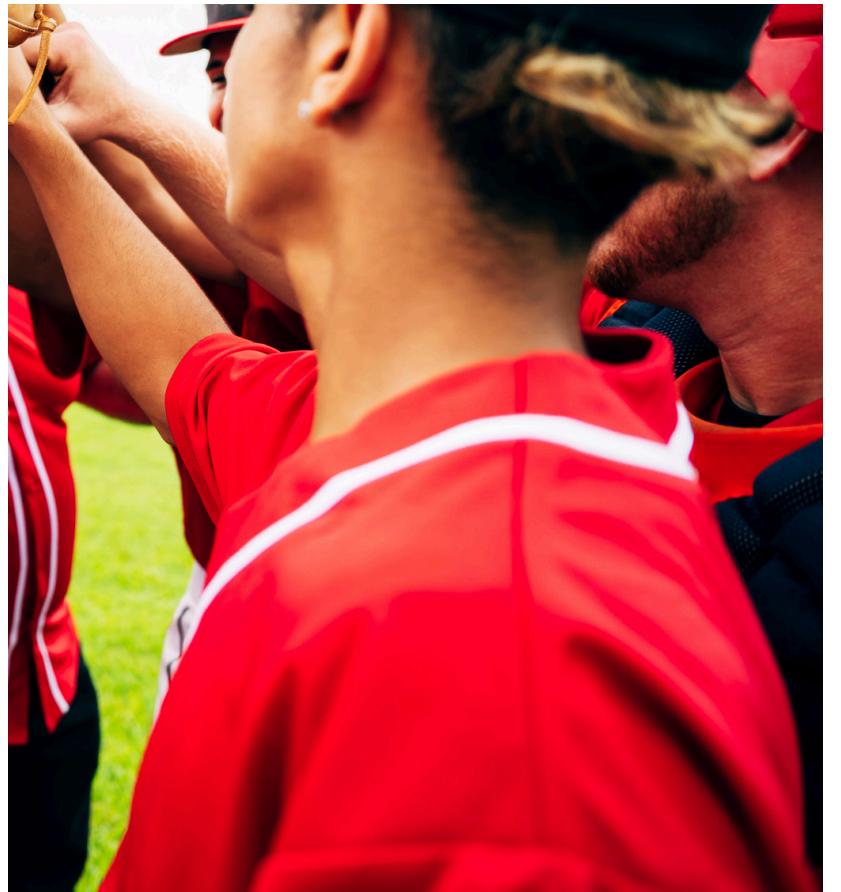
Data fetching with built-in and automatic caching and polling



FEATURES

DEMO

- Date Navigation
 - Loads today's games by default
 - User can select other dates
- Affiliate Coverage
 - Shows all Marlins affiliates
 - "NO GAME" clearly displayed if no matchup
- Game States
 - Preview → time, venue, opponent, probable pitchers
 - Final → final score, winning/losing/save pitchers
 - Live → current score, inning, outs, runners on base, pitcher & batter (Video)
- Team & Opponent Info
 - Team name & level (e.g., AA, AAA, 1A)
 - Opponent name + parent MLB club



HIGHLIGHTS

Error boundaries for resilience

Mobile-responsive via MUI

Consistent Marlins theme, fonts, and colors

• • • •

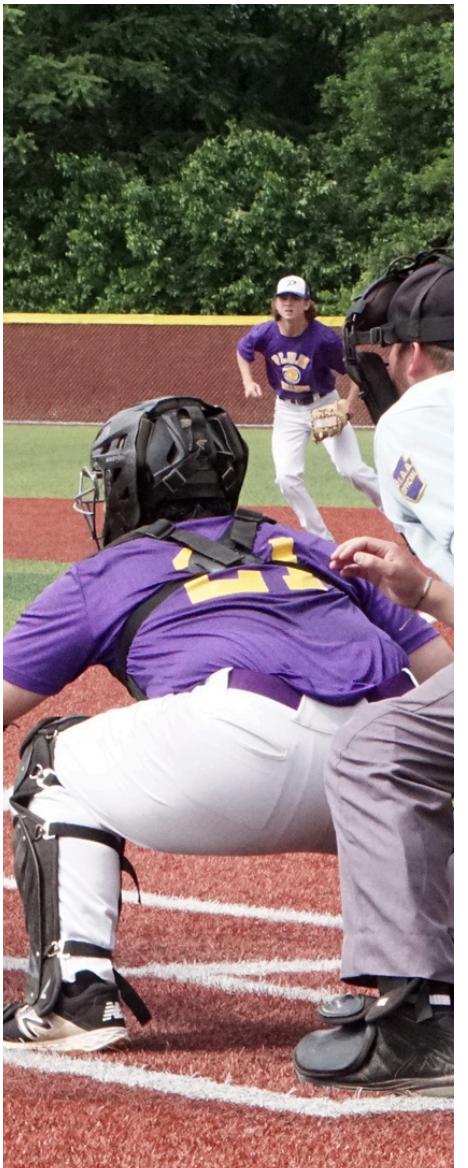
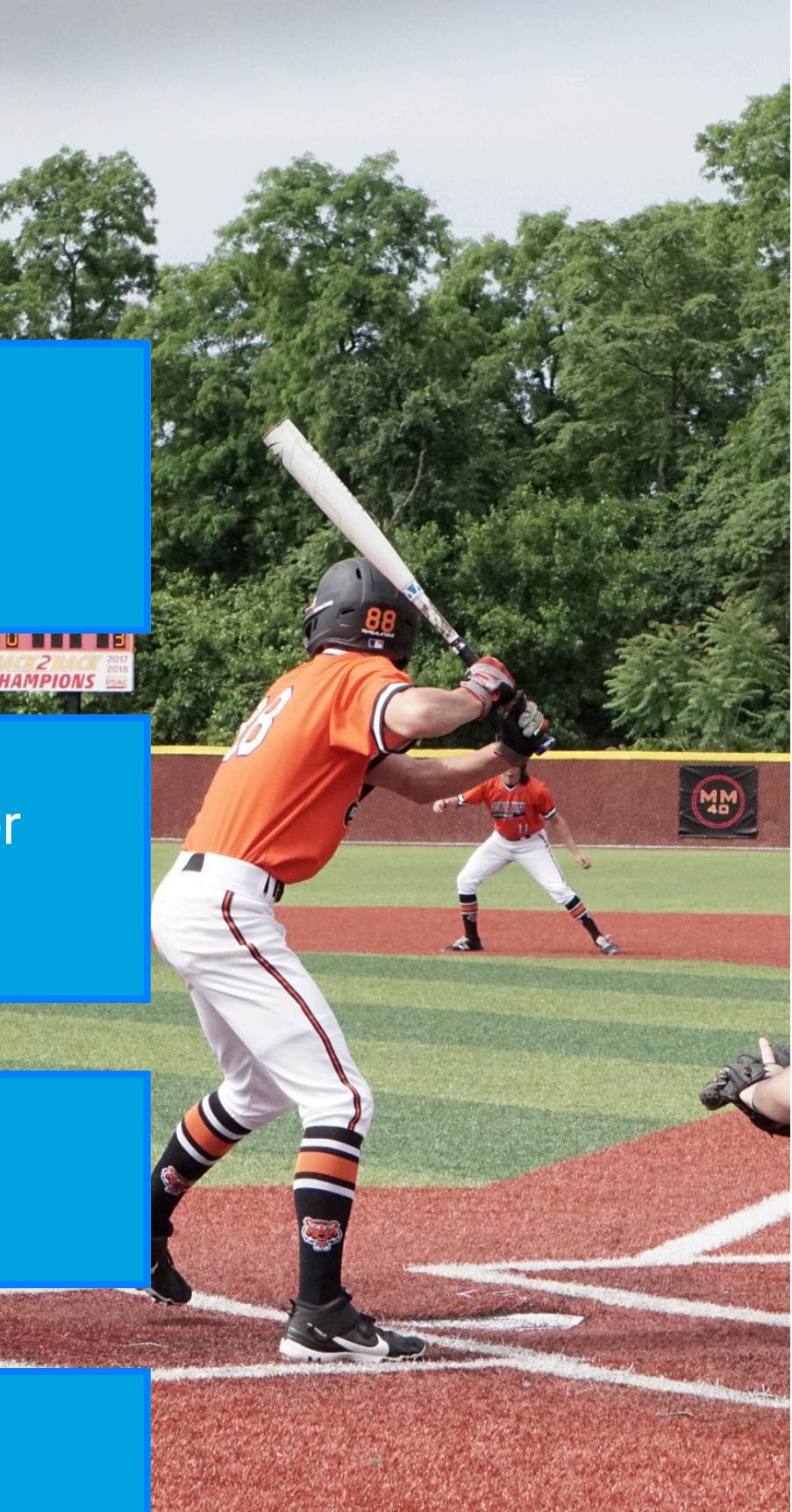
ARCHITECTURE DECISIONS

Tech stack: Vite + React + TS for speed & type safety; MUI for accessibility; RTK Query for caching; Redux store for API state.

Component structure: Schedule.tsx → grid, GameRow.tsx → per team, useGameDetails.ts → live feed.

API integration: Schedule endpoint, Game Feed, Team Info.

Layout: CSS Grid for 2×3 design





LONG-TERM THINKING

MISSED OPPORTUNITIES / FUTURE WINS



Extras



Bases Visualization

On-base diamond



Tests

Unit testing with React/Jest

TEAMMATES

LETS GET BETTER



| Tool | Pros | Cons | Best Fit |
|-----------------------|---|--|---|
| Context API | Built-in, simple, familiar | No enforced patterns, performance issues with frequent updates | Small/medium apps, simple global state (theme, auth) |
| Zustand | Simple mental model, minimal boilerplate, performant | Can become inconsistent in big teams without patterns | Medium apps needing simplicity + power |
| Jotai | Granular control, great for complex interdependent state | Unfamiliar model, risk of atom sprawl | Complex UIs with many interdependent state pieces |
| Redux (RTK/RTK Query) | Strong conventions, debugging tools, built-in caching & polling | Higher learning curve, more verbose | Large apps, team collaboration, frequent server-state updates |

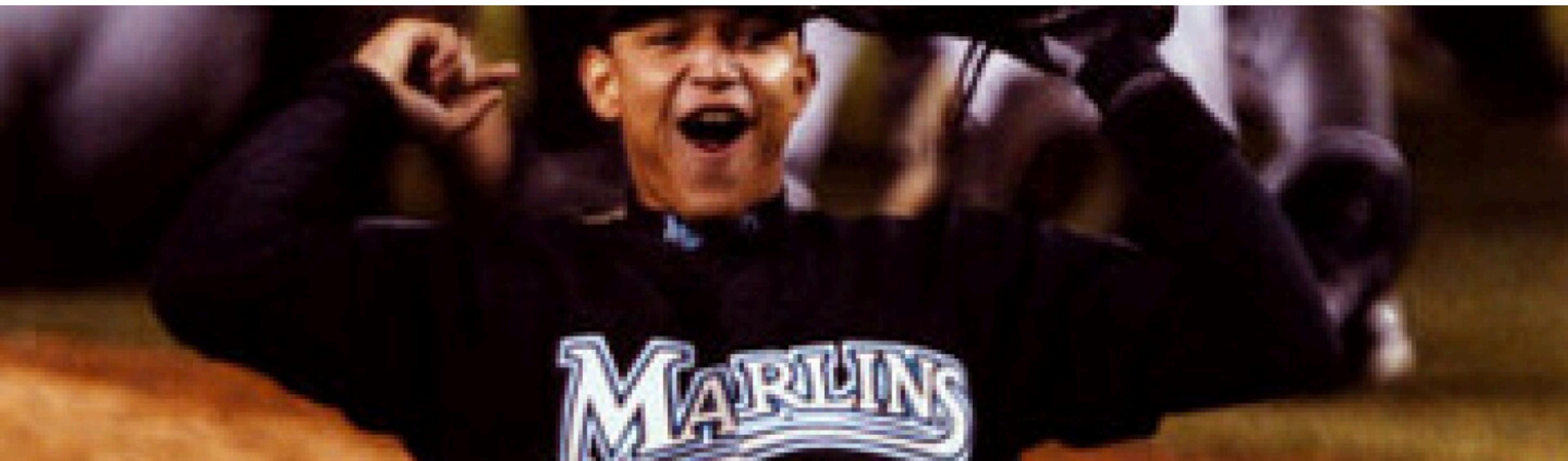
Source:^{*}

https://dev.to/hijazi313/state-management-in-2025-when-to-use-context-redux-zustand-or-jotai-2d2k#:~:text=Zustand%20*%20Pros:%20Simple%20mental%20model%2C%20flexible,inconsistency%20in%20larger%20teams%20without%20established%20patterns



THIS PROJECT SHOWS HOW I:

Innovate and build for scale + UX future WINS





THANK YOU BASEBALL OPERATIONS!



Ismail Khair - GitHub URL

<https://github.com/IshLovesLucy/mlb-schedule-app>