

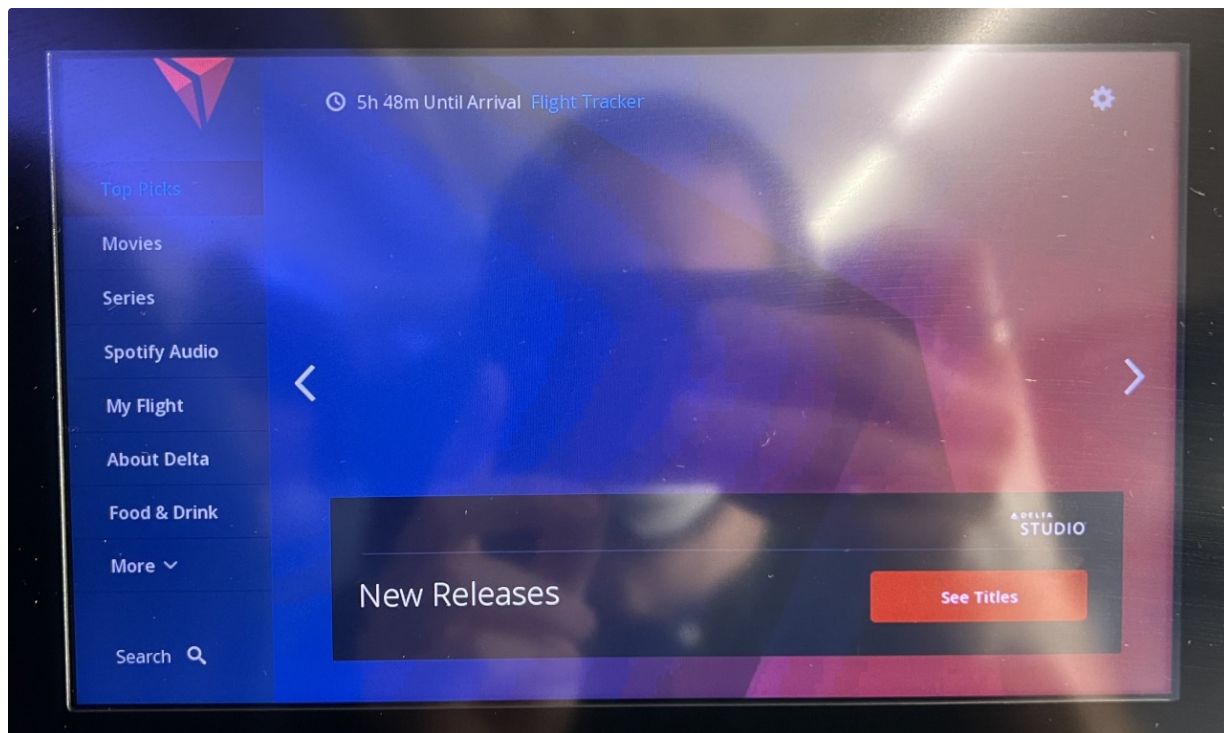
Senselessness

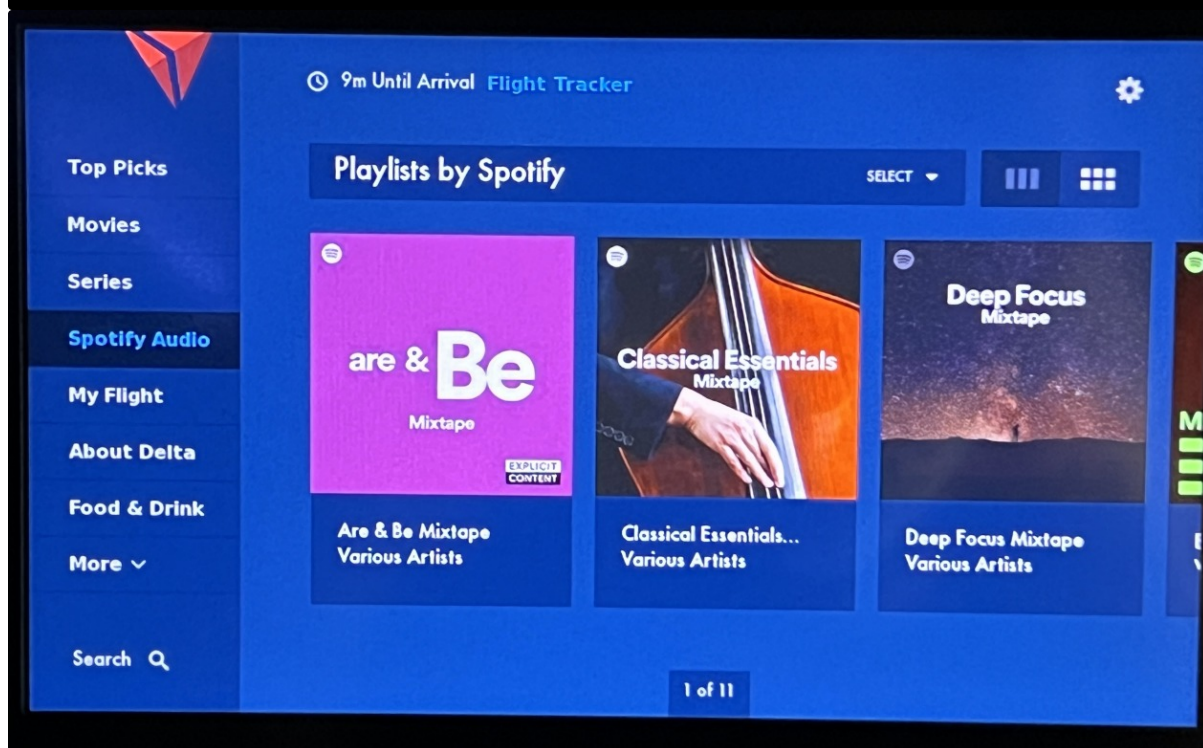
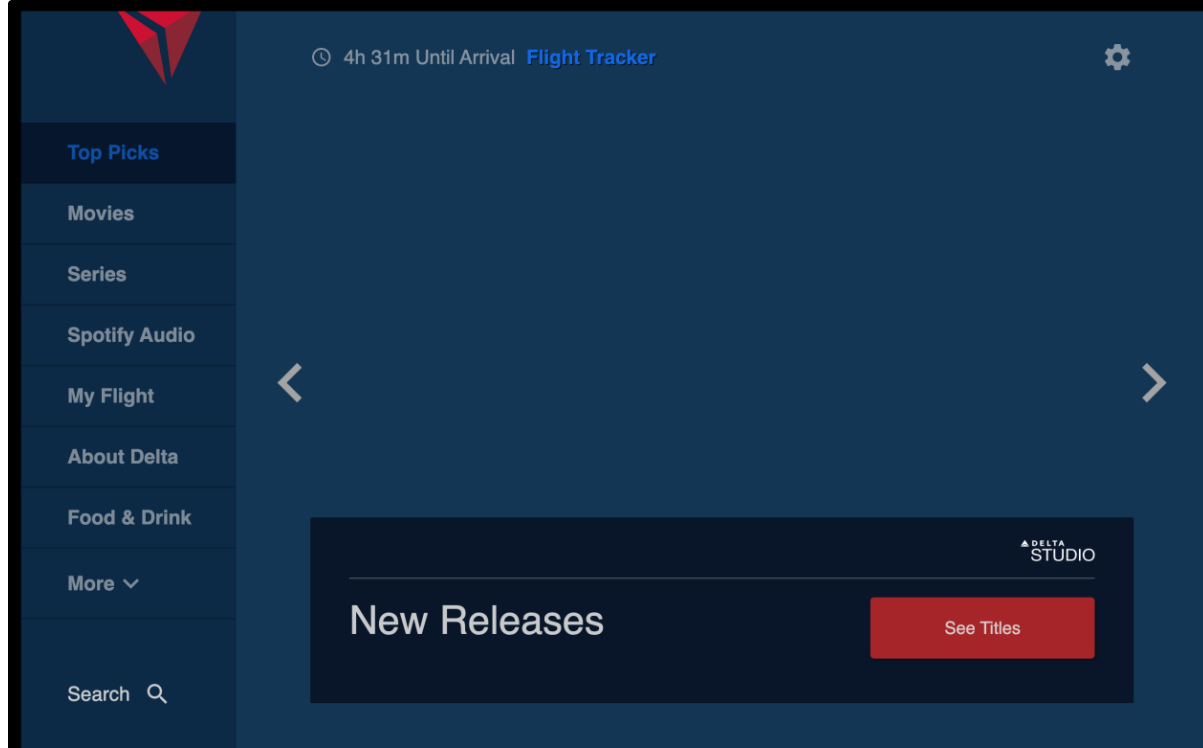
The Inflight Entertainment Challenge

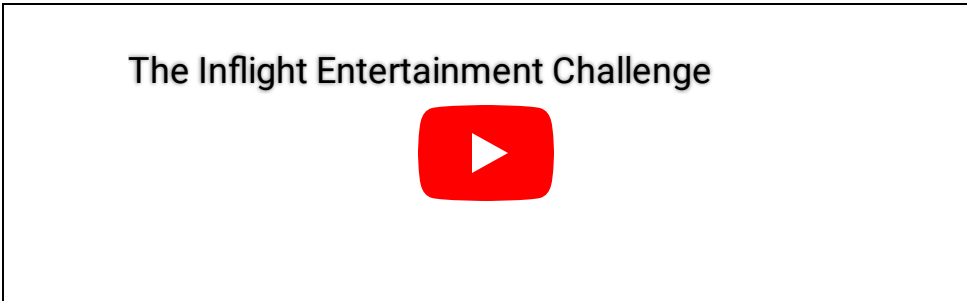
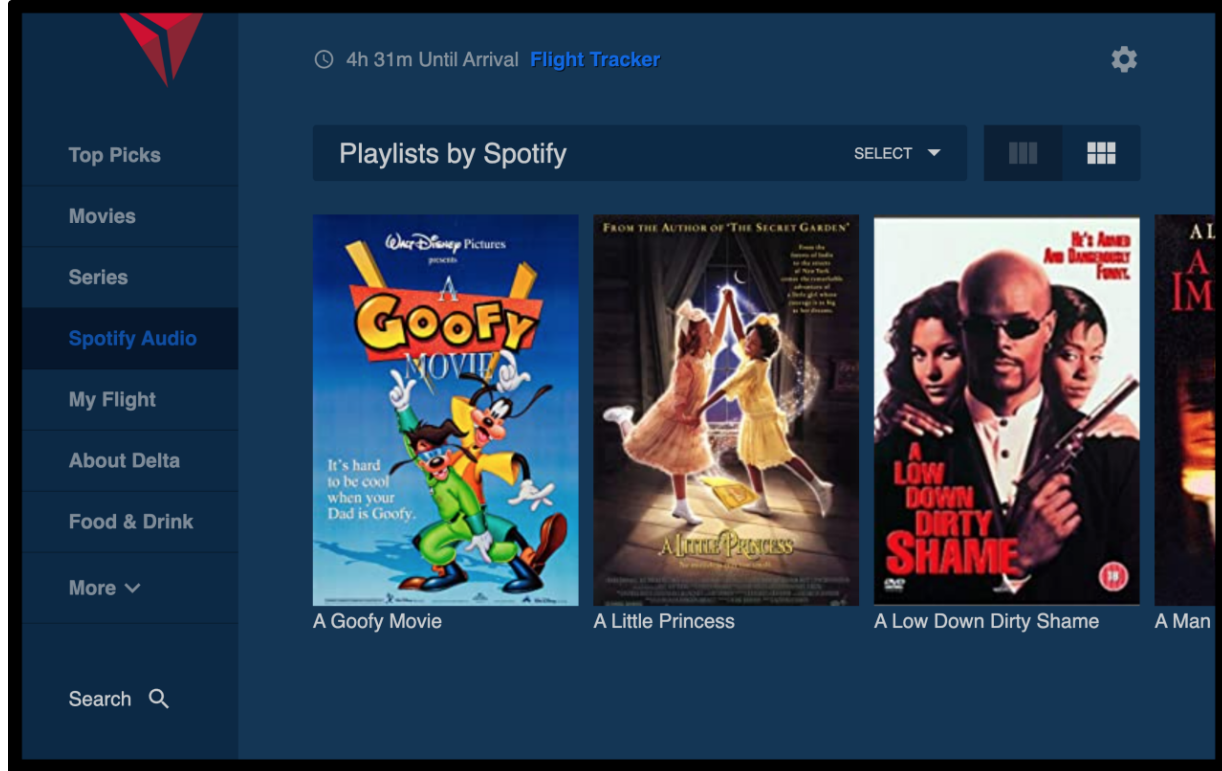
On a recent flight from NYC -> SFO, I decided to try to re-create as much of the Delta in-flight entertainment system as I could.

Try the final result here: <https://delta-inflight.vercel.app>

And here's the code: <https://github.com/paulmand31/delta-inflight>







Why do something like this?

Rapidly translating UIs into code is a core skill for UX engineers and design technologists. As I've moved more into management I've been doing less and less coding and wanted to test my skills. Obviously this isn't a perfect facsimile of real-world development, but it was an interesting challenge and a fun way to spend the flight.

Constraints

- Started with a blank [Vite](#) project initialized with the react-ts template.
- No pre-work aside from installing off-the-shelf npm libraries ([recoil](#), [lodash](#), [mui icons](#), etc) and downloading any data sets you'll need (e.g. a bunch of movie posters).
- No pre-built component libraries.

- No internet access except for the in-flight wifi portal and the airline's website (useful for grabbing logos). No StackOverflow and no new NPM libraries.
- Don't worry about production quality, just re-create as much of the look and feel of the UI as quickly as possible. No accessibility, responsive design, localization, etc; you just want something you can run through a user test.
- Project-specific code can only be written from when they say you can use your laptop (once the plane has reached 10k feet) to final approach when they make you put laptops away.

What did I learn?

Going in, I thought that not having access to StackOverflow was going to be the biggest challenge. But that actually turned out to be ok! Aside from forgetting the syntax for CSS linear gradient backgrounds and how to ellipsize multiple lines of text I was able to remember most of my syntax and usage (and VSCode helped me fill in the gaps).

The biggest problem turned out to be not having access to new NPM libraries. Slick draggable scrolling is a key part of many screens in Delta's in-flight entertainment system (complete with momentum-based flicks, over-scroll, page-snapping, etc). Because I didn't have access to a component library and hadn't thought to pre-install a good drag scroll library, I ended up having to build my own. It went ok but took a long time and isn't nearly as good as many off-the-shelf options. Next time I try this I'll be sure to come better-prepared.

Another challenge was the sizes, spacings and colors. Usually when I build a prototype I'll include a hide-able screenshot of the comp to make it easy to achieve pixel-perfection. Without the internet I couldn't match the fonts and not having access to the raw visuals made it hard to compare directly. Sometimes I ended up trying to hold my laptop up to the screen which got me some funny looks.

One other major takeaway was feeling very good about my current prototyping toolchain. React, typescript, a vs code extension I wrote to quickly create new React components, recoil, css modules, stylus, and MUI make it really, really easy and fast to quickly build prototypes. They may

not always be the right choice for production, but dang do they feel good to use.

Conclusion

This challenge was a lot of fun and I definitely want to do it again. I made good progress but didn't get as far as I wanted and I'm excited to take what I learned and try to do better next time.

If you're also a front-end engineer or work with designers I highly recommend giving this a try!

MAR. 19 2023
#DELTA AIRLINES #JAVASCRIPT #PROTOTYPING #REACTJS #TYPESCRIPT #INFLIGHT
ENTERTAINMENT #YOUTUBE

[Home](#) [Archive](#)

OBSERVER THEME BY ZACK SULTAN