



# RecoMedia

Media that matches your moments.

## High Fi Prototype ReadMe

Calvin L   Chris K   Jack R   Whayden D

## Tools Used

Our final prototype faithfully adheres to the original design specifications, delivering a fully functional, production-ready experience without relying on placeholder or simulated elements (i.e., no Wizard of Oz techniques or hard-coded data). From end to end, the system dynamically processes user input, retrieves recommendations via AI-driven APIs, and serves real-time media content and events.

### Tools Used; Pros and Cons

#### *Frontend:*

We built the frontend using React Native and TypeScript through Expo Go, allowing for a smooth development workflow, rapid iteration, and a robust type system that minimized runtime errors.

Pros: Rapid prototyping, excellent developer experience, and strong type safety.

Cons: Limited direct native module customization without ejecting from Expo and a frustrating navigation system.

#### *Backend & Data Storage:*

User authentication, search queries, and historical results were managed with Supabase, providing a straightforward, scalable database solution.

Pros: Easy integration, real-time capabilities, and built-in auth.

Cons: Requires careful schema design and a bit of learning with PostgreSQL.

#### *Custom API Endpoints:*

We leveraged Make.com (a low-code integration platform) to implement a custom endpoint, `getRecommendations`. This endpoint processes a user's search query via Anthropic's Claude, applies a carefully refined prompt to generate media recommendations, then retrieves matching posters and images from the OMDB API (for movies and series) and the Pexels API (for event and location visuals).

Pros: Rapid, code-light iteration; modular endpoint creation; free tier enabled extensive experimentation.

Cons: Limited low-level control compared to traditional backend frameworks and a long-term reliance on the platform in the future.

#### *AI Assistance:*

AI proved to be immensely helpful in all aspects of the development process. We leaned heavily on ChatGPT and Claude to help us with debugging, and found it very helpful in getting

started. Once we got our code to a position we felt customizing from, we went to work and were able to make things fly. Cursor with Claude enabled was also a huge help with debugging across files.

Pros: Significantly accelerated development, reduced boilerplate coding, offered quick solutions to complex frontend issues.

Cons: AI-driven debugging was limited for custom API logic since the model lacked direct implementation visibility.

## Operating Instructions

Our prototype currently only works on an IOS device, though it will support android devices later on. To access the app on an IOS device:

1. Pull source code from GitHub: <https://github.com/calvinstanford2024/recomedia>
2. Open in editor (VS code, etc.)
3. Install required software: `npm install`
4. Start project: `npx expo start`
5. Follow terminal instructions to open in IOS simulator or scan QR code to open on Expo Go mobile app
6. Create account or use existing test account:
  - a. Email: **laughlincalvin+test@gmail.com**
  - b. Password: **Password!**
7. Enjoy!

## Limitations

Our app currently does not support the ability to save content to profile, though this is something that we plan on implementing in the future.