You're almost there — sign up to start building in Notion today. Sign up or login

YoPrint React Coding Project Spec

Anime Search App

> **AI Usage Policy**
> We actively encourage the use of AI tools (ChatGPT, Claude, Copilot, etc.) throughout this project. Using AI effectively is a valuable skill in modern development, and candidates who leverage AI tools thoughtfully may receive preferential consideration during evaluation.
>
> If you use AI assistance:
>
> - Create a `PROMPTS.md` file in your project root
> - Document all prompts you used with AI tools
> - Include context about which parts of the project each prompt helped with
> - This helps us understand your problem-solving approach and how you collaborate with AI

Overview

This mini-project verifies your React proficiency and should take a few hours to complete. You'll build an Anime Search App - a two-page application where users can search for anime and view details.

Page 1: Search page displaying results

Page 2: Detail page for selected anime

API: Jikan API - free, no authentication required

Technical Requirements

Core Stack:

- React 18 or higher
- React hooks only (no class components)
- TypeScript
- react-router-dom for navigation
- Redux for state management
- UI library of your choice
- Single Page App only (no Next.js)

Functionality:

- Server-side pagination on the search page
- Instant search with debouncing (see details below)
- Must use redux for state management

Instant Search Implementation

The search bar should work without requiring users to press Enter or click a button:

- Debounce API calls to 250ms intervals to avoid excessive requests
- Cancel any in-flight API requests if the user continues typing
- This prevents making calls on every keystroke while keeping search responsive

Submission Requirements

Package Manager and Setup (CRITICAL):

- **You MUST use npm only** - do not use yarn, pnpm, or any other package managers
- Your project must be runnable with these two commands only:

```
npm install npm run dev
```

- The dev server must start on **port 4000**
- **Do not use environment variables** - the app should work immediately after installation

- **Failing to meet these requirements will result in automatic disqualification** - if we cannot simply run `npm install` and `npm run dev` and use your app right away, your submission will not be accepted

Deployment:

- **You must host your project on a free hosting platform** so we can view it live
- You can use any free hosting service you prefer (Vercel, Render, GitHub Pages, etc.)
- We recommend **Netlify** if you're unsure, as it is free and convenient
- Include the live URL in your submission

Evaluation Criteria

Your submission will be evaluated on:

1. **Correct Implementation** - All features work as described, proper routing, and state management
2. **TypeScript Usage** - Proper typing throughout with minimal use of 'any' types
3. **Code Organization** - Logical folder structure, reusable components, and clear separation of concerns that makes it easy for other developers to extend your work
4. **Code Quality** - Clean, well-formatted code following React and TypeScript best practices
5. **React Best Practices** - Proper hook usage, avoiding anti-patterns, efficient re-rendering

Bonus Points (Optional)

Stand out from other applicants by including:

If you implement any bonus features, please list them in your README under a "Bonus Implementation" header to help us evaluate your submission quickly.

User Experience:

- Creative UI with unique "wow" factor
- Skeleton loaders or meaningful loading states
- Empty state and no results handling with helpful messaging
- Mobile responsiveness

- Additional features that enhance the project

Technical Excellence:

- Proper error handling (network failures, rate limiting, invalid API responses)
- Race condition handling
- Unit or integration tests

Submission Checklist

Before submitting, ensure:

- [ ] Project uses npm only (no yarn/pnpm)
- [ ] `npm install` and `npm run dev` starts the app successfully
- [ ] Dev server runs on port 4000
- [ ] No environment variables required
- [ ] Project is deployed and accessible via live URL
- [ ] All core functionality works as described
- [ ] Code is written in TypeScript
- [ ] Redux is properly implemented for state management
- [ ] If bonus features implemented, they are listed in README under "Bonus Implementation" header