ALL CARDS

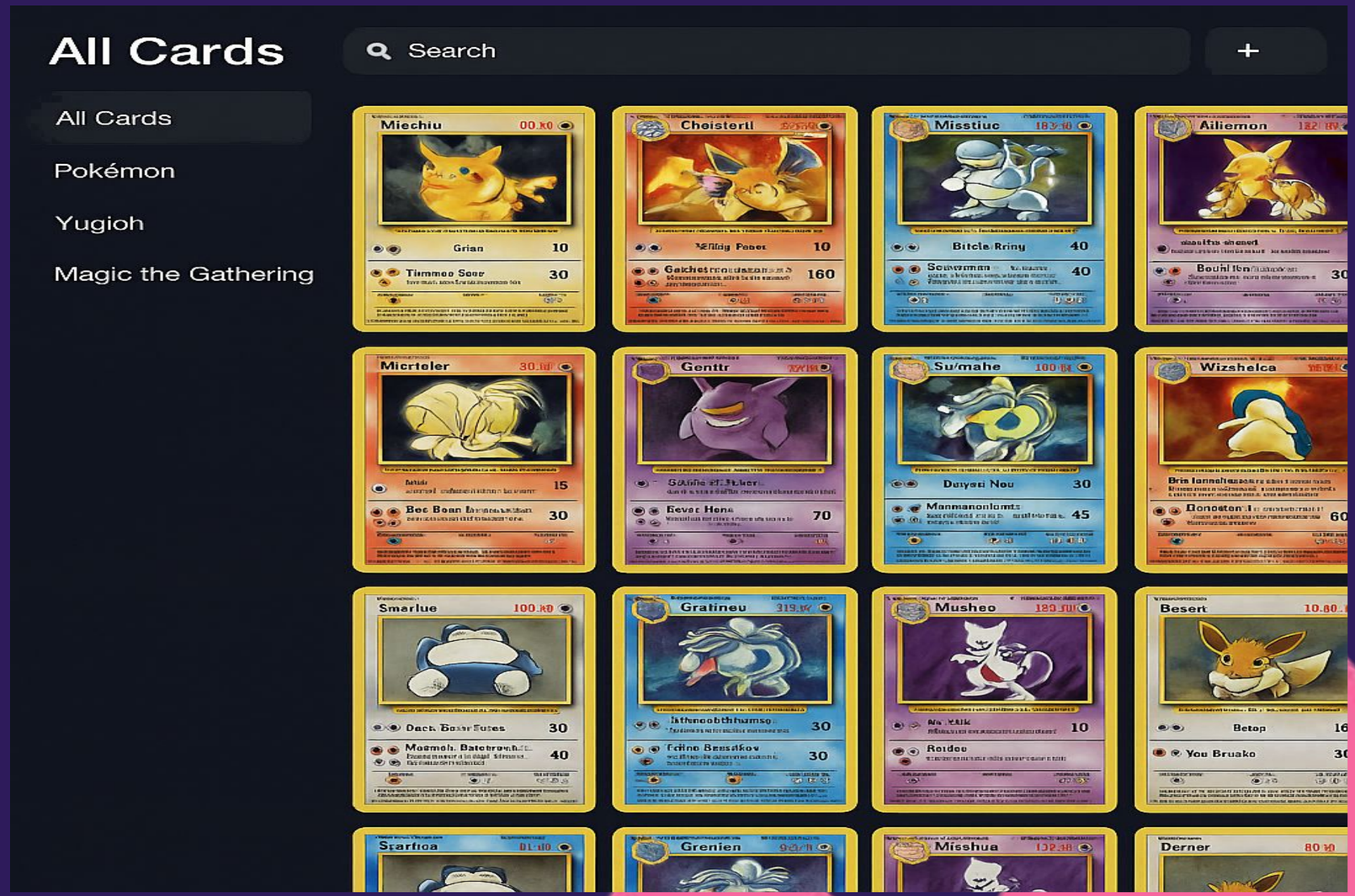
Anisur Rahman, Blaise Pierre,

Richard Vilcinsh, & Ryan O'Connor

Capstone CSCI 499

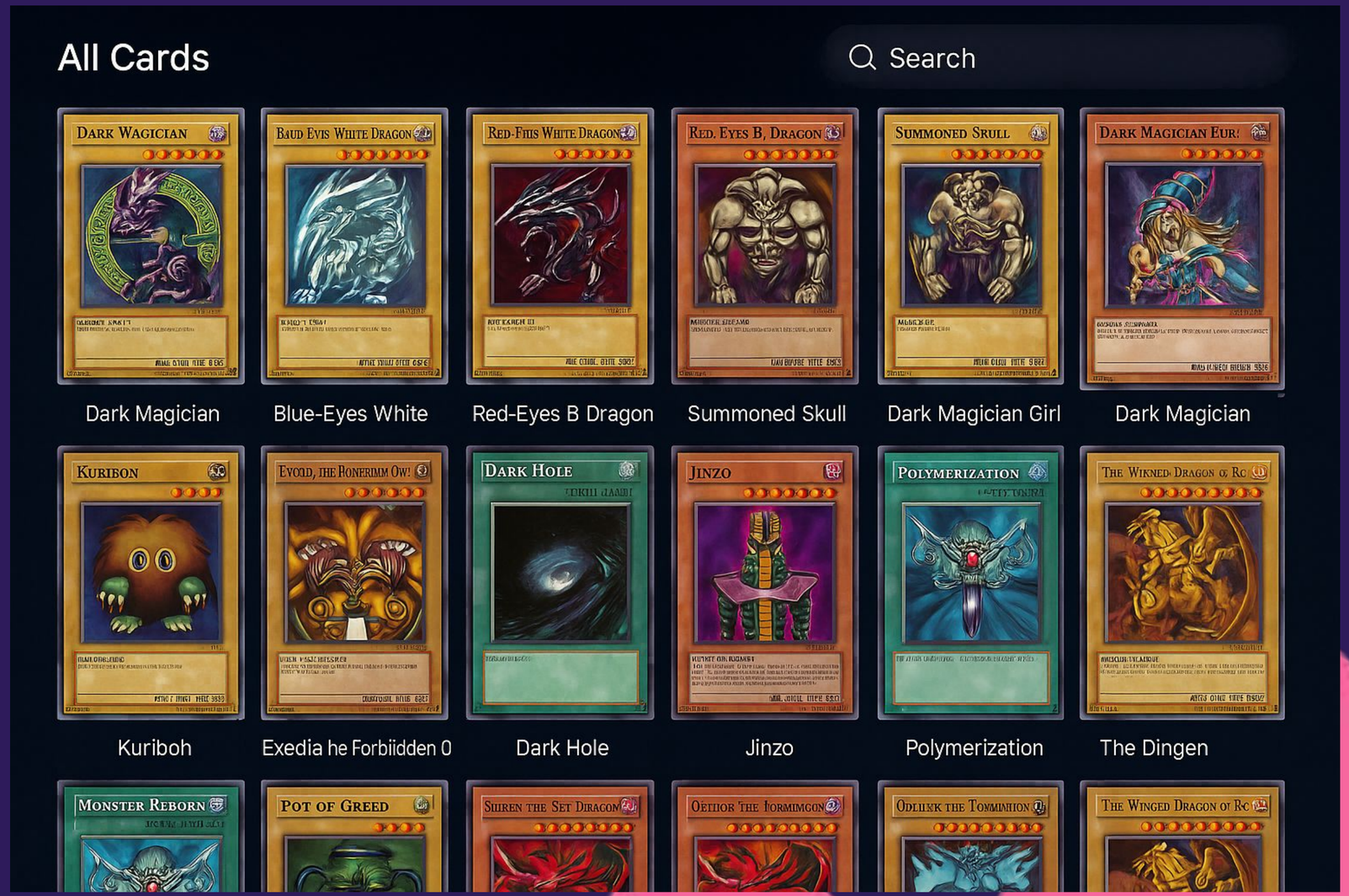
Ui Example

- All Cards is a web app that allows you to store and search cards you have in real life. A digital Library of your real life cards.



A real life example

- Let’s say you opened up a box of yu-gi-oh cards. You get mostly bad cards, but you get some good cards you want to use to improve your deck. As you are building your deck, you are wondering if you have any extra spirit blossoms in your binder or any containers. Instead of trying to look for them, you can open up your digital binder and find how many extra copies you have and where it is located.



Vision & Target Audience

Problems trying to solve

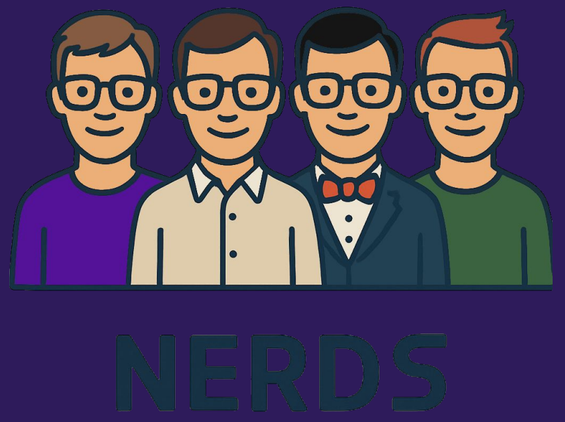
- Physical Binders are limited and not easily shareable
- Online databases are often not personalized
- Many users track cards across multiple platforms
- No dedicated tool that combines visual collection browsing with personal tagging and favoriting

Vision

To create the most intuitive and personalized digital binder experience for collectors and fans of Trading Card Games. To make it easier to discover, favorite and organize the cards they love

Target Audiences

- Casual and Hardcore TCG Collectors
- Newcomers
- Competitive players
- Collectors of specific card series
- Collectors of multiple card series



How does it differ?

There is no platform where you can you can have different card series in one place. Our app focuses on collectors of all types offering beautiful and customizable binder experience that feels personal, playful and powerful

Tech Stack

Tech Stack

Web application

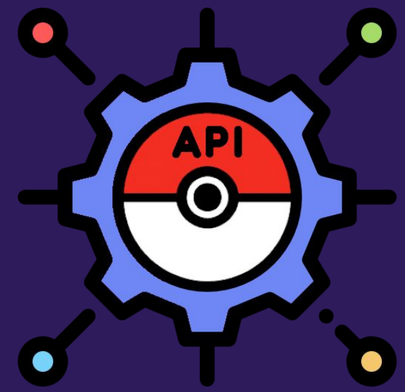
Frontend/ UI: React + TailwindCSS

Backend: Firebase

DataBase infrastructructure: Firebase

API used: Pokemon TCG API, more as needed

Languages: HTML, CSS, JavaScript



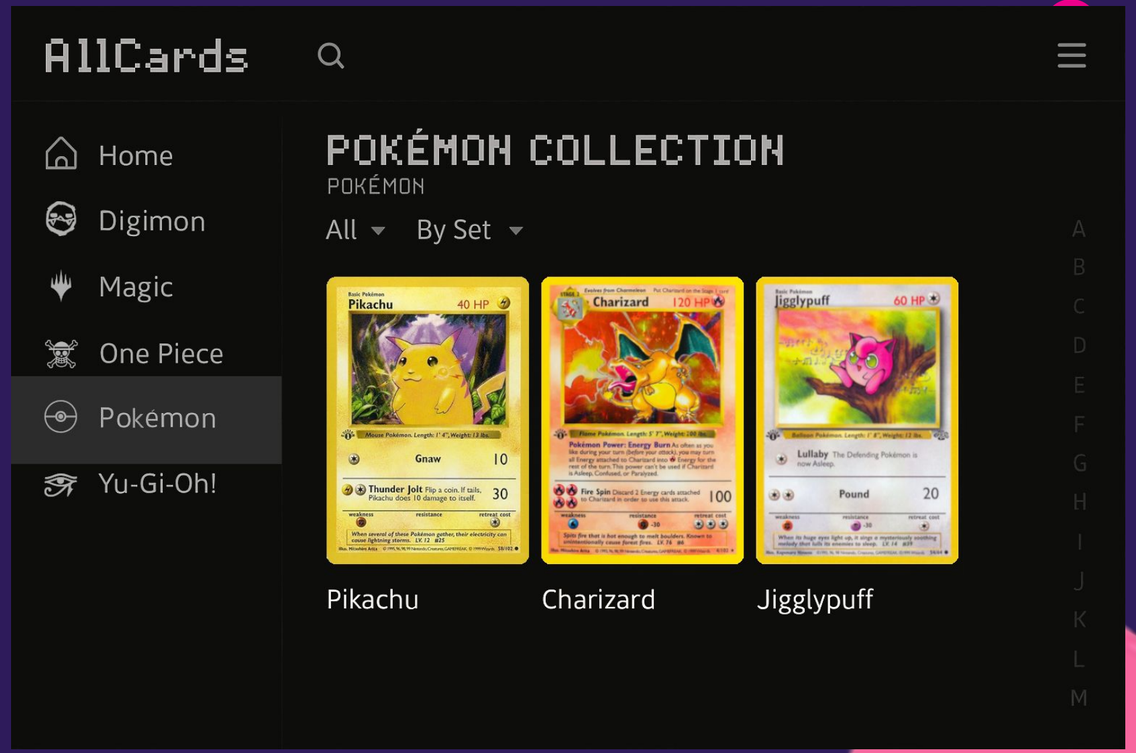
MVP

MVP

- Upload card images to add to Virtual Binder
- View by collection/set or by a built deck
- Organize by Trademark
- Simple login system

Stretch

- Advanced filters (rarity, series, tags)
- Social features (public collections, sharing)
- Showcase mode / carousel
- API integration for auto-tagging or verification



Timeline

Timeline (tentative)

Week 1: Project Setup & Planning

- Finalize scope, goals, and tech stack
- Assign roles (frontend, backend, database, styling, etc.)
- Create GitHub repo, design shared workflow (branches, issues)
- Initial wireframes / UI sketches

Week 2: UI Mockups & Infrastructure

- Build mock UIs (Home page, collection view, card viewer)
- Choose and configure framework (React + Tailwind)
- Set up Firebase project or backend server
- Basic file structure + routing scaffold

Week 3: Authentication & Basic Card Upload

- Implement login (Firebase Auth or custom)
- Develop upload feature: allow image + metadata entry
- Start storing data in Firestore or MongoDB
- Basic upload > display card flow (no tags/search yet)

Week 4: Collection Viewing

- Create card list/gallery UI (by franchise)
- Add support for browsing by franchise (Pokémon, etc.)
- Improve card layout: name, set, rarity display
- Begin refining component styling

Week 5: Organization Features

- Implement sorting (e.g., by set, alphabetically)
- Add basic tagging or categorization
- Introduce folders/collections (e.g., Base Set, Jungle)
- Enable card deletion and editing

Timeline (tentative)

Week 6: MVP Polishing

- Bug fixes, responsive layout adjustments
- UI refinement: transitions, spacing, typography
- Conduct user testing (within group)
- Document MVP code and features

Week 7: Advanced Search + Filtering

- Add search by name/tag
- Filter by franchise, rarity, custom tags
- Index cards for performance if needed

Week 8: Showcase / Sharing Options

- Add slideshow view ("Showcase Mode")
- Start implementing public viewing (optional)
- Prepare stretch feature demos

Week 9: Final Polish & Presentation

- Final styling, responsiveness tweaks
- Prepare slides & script for final presentation
- Record a demo video or deploy to Vercel
- Conduct a team review and clean up codebase