Eric Chi - Full Stack Developer

ciredeveloper@gmail.com ❖ Boston, MA ❖ Github

SKILLS

Frontend: HTML5, CSS3, SCSS, Tailwind, Browser APIs, Astro, React, React Router, React Hook Form, Radix UI, SWR, React Query, Framer Motion, Recharts, D3, Zustand, Vite, understanding of a11y
Backend: Node, Express, SQL, Redis, PostgreSQL, Postgres.js, Drizzle ORM, OWASP (session mgmt, csrf protection, auth)
Cloud: AWS - S3, CloudFront, Beanstalk, EC2
Common: JavaScript, TypeScript, Zod
Testing: Jest, Vitest, React Testing Library
Other: Git, CI/CD, Docker, Nginx, Bash

PROJECTS

Propel CRM - Real Estate CRM using TypeScript, React, Node, Express, PostgreSQL, Docker, Nginx, & AWS
- React Router / Express middleware to only render protected routes when valid user is returned from server
- using React Query to handle server state, optimistic updates, paginated/filtered queries, and cache invalidation
- Custom authentication that handles user sessions, CSRF protection, account recovery, and email verification
- Dockerized /nginx and /server in a monorepo architecture, connected via docker compose for easy deployment
- Deployment to AWS using S3, CloudFront, and Elastic Beanstalk (EC2)

Pokedex Table - Filterable and paginated data table for all pokemon from public API
- Implemented useSWR to handle cache management and improve load times and UX via pagination
- Implemented Zustand to handle client side state such as filter options and pagination state
- Walked through various client side data fetching methods, useEffect / useState, custom hooks, and useSWR to learn the pros and cons of each method

Portfolio - React SPA Showcase site using React Router and Framer Motion
- Typesafe single source of truth for content rendered to view
- Using React Router and Framer Motion for smooth SPA functionality
- Utilized React.lazy, .woff2 fonts, and webp images that took a > 2MB initial payload down to ~300kb

Food Service Demo
- Used Astro to optimize loading of JavaScript only when needed
- Given a design file, was able to recreate a 1:1 website at given spec

Todo App
- Utilized Vanilla JavaScript DOM Manipulation to handle rendering of different projects and their associated todos, while handling performant DOM updates when a todo is updated
- MVC client side architecture to handle separation of concerns in a vanilla JavaScript context
- Worked with LocalStorage API to handle persistence of application state across browser sessions

EDUCATION

University of Massachusetts - Lowell

- Completed 65/120 credits.

CERTIFICATIONS / COURSE WORK

- Responsive Web Design - freeCodeCamp, Containers - Full Stack Open, Algorithms - Frontend Masters