

Daniel Briseno

Brisenodan12@gmail.com | (219)-798-5777 | Graduation: May 2027

Education

Purdue University, Northwest: B.S. Computer Science

Relevant Coursework: Data Structure & Algorithms, Computer Architecture & Assembly, Computer Science Principles, C Programming, Object-Oriented Programming in Java, Operating Systems, Artificial Intelligence, Databases, Networking Programming

Technical Skills

Languages: JavaScript, TypeScript, Java, C, Python, SQL

Frameworks & Tools: React, React Router, Vite, Tailwind CSS, TanStack React Query, Redux Toolkit, React Hook Form, Node.js, Express, PostgreSQL, Supabase (Auth, DB, Storage), Git, GitHub, Postman, Chrome DevTools, Jira, Vercel, npm, ESLint, Prettier, VS Code

Work Experience

Full-stack Developer - Self-Employed (January 2024 – Present)

- Built and deployed full-stack web apps using React + Supabase (Auth/DB/Storage) + PostgreSQL, implementing authentication, uploads, and feed-style features
- Designed responsive, component-based UIs with React Router and custom hooks; improved UX with smooth state-driven updates and clear loading/error handling
- Developed and maintained RESTful APIs, designed relational schemas, and managed end-to-end delivery using Git/GitHub (branching, code reviews, deployments)

Projects

[SchoolEm \(University Social Feed Platform\)](#) – Full Stack Developer

- Stack: React 18, React Router, Vite, TanStack React Query, Tailwind CSS, Supabase (Auth/DB/Storage), PostgreSQL (PL/pgSQL, triggers, RLS)
- Built a university social feed with authentication + protected routes, profiles, follow system, and unified content types (posts/threads/quickies).
- Added responsive design and dynamic UI updates using async Supabase queries and cache refetching to improve user experience
- Implemented Postgres feed + detail functions (cursor pagination + per-user “liked” flags via auth context) and trigger-driven notifications/counters for likes/comments/follows.
- Improved performance with targeted indexes (author/publication, followers by username, comments/likes lookups) and client-side image downscaling/compression (multi-MB → ~sub-100KB typical) for faster rendering + lower storage/bandwidth.

[BlogIt \(Dynamic Blog Post Application\)](#) – Full Stack Developer

- Stack: Node.js, Express.js, PostgreSQL, EJS, Passport.js, Multer
- Added responsive design and asynchronous UI updates to improve user experience
- Built a CRUD blog platform with likes/comments, secure authentication (Google OAuth), and image uploads

[WorldWise](#) – Front End

- Stack: React 18, React Router, Vite, Leaflet/React-Leaflet
- Optimized load time with lazy-loading, improved user engagement with interactive maps.
- Implemented global state (Context/useReducer), custom hooks, ProtectedRoute, and JSON Server.

Volunteering

[Chicago Family Health Center](#) (August 2011 – Present)

- Actively involved in the continuous development of culture and community service, including assisting patients, organizing and executing events, and catering to children and those in need.

License & Certifications

- Programming with JavaScript — Meta (JavaScript)
- Complete Web Development Bootcamp — Udemy (HTML/CSS, JavaScript, React, Node/Express, PostgreSQL)
- Java Data Structures & Algorithms Masterclass — Udemy (Data Structures, Algorithms)
- Ultimate React Course — Udemy (React, Next.js, Redux, Tailwind CSS, Styled-components, Supabase)