

# Brad Yin

## Senior Software Engineer

[bradyinrt@gmail.com](mailto:bradyinrt@gmail.com) | 7732579347

- Experienced Software Developer with an eye for visual design, and focused on creating great user experiences.
- Skilled at collaborating with cross-functional stakeholders to gather requirements, deliver tailored solutions, and support end-user adoption in client-facing environments.
- Over 5 years of experience developing modern web applications using HTML5, CSS3, Javascript, React, Redux, React Router, Node.js, TypeScript, Babel, Sass, Bootstrap, jQuery
- Extensive experience with ReactJS, NodeJS, and their core principles.
- Deep knowledge of modern authorization mechanisms, such as JWT.
- Solid mastery using React Redux to manage various increasingly complex UI states of single page applications such as active routes, selected tabs, spinners, pagination controls, etc.
- Strong expertise building scalable, reusable CSS templates following the BEM naming methodology
- for easier maintenance using the CSS preprocessor like SCSS.
- Good grasp of web security issues such as SQL injection, CORS, cross-site scripting, etc.
- Professionalism developing server-side applications using Express, OAuth 2.0, Passport in Node.js.
- Demonstrated proficiency in developing interactive applications involving web services by utilizing RESTful services and creating web service APIs adhering to REST architectural constraints.
- Fluency managing all phases of the software development life cycle using the Agile/Scrum development methodologies.
- Broad experience writing unit tests and creating test cases using test runners such as Jest, Enzyme, and React-Testing-Library.
- Ability to perform successful and efficient version control as well as task management using Git and JIRA.
- Self-driven and capable of managing multiple priorities under pressure.

## SKILLS

**Front-End:** JavaScript, TypeScript, React, Redux, React Router, HTML5, CSS3, SASS/SCSS, Styled-Components, Responsive Design, Cross-Browser Compatibility, Webpack, Babel

**Back-End:** Node.js, Express.js, Java (Spring Boot, Spring MVC), RESTful API design, Microservices architecture, MongoDB, MySQL, PostgreSQL

**DevOps & Testing:** Git & GitHub, CI/CD pipelines (GitHub Actions, Jenkins), Docker & Containerization, Unit Testing (Jest, JUnit), Integration/UI Testing (React Testing Library, Cypress),

Performance profiling (Chrome DevTools, Lighthouse)

**Tools & Workflows:** Agile/Scrum methodology, JIRA project tracking, Figma, Framer, Postman/Swagger (API testing), VS Code/IntelliJ IDE

## Work History

### **Front-End Developer – AMD, TX Hybrid, (June 2023 – Present)**

**Project: Radeon Display Insights Portal** – Internal telemetry dashboard for display validation

**Company Overview:** AMD is a leading semiconductor company producing high-performance GPUs, CPUs, and display technologies for gaming, professional visualization, and data centers.

**Project Overview:** The Radeon Display Insights Portal is an internal web tool built to help AMD's Display QA teams monitor and validate real-time display data across a wide range of devices. It was created to support new technologies like DisplayPort 2.1, HDR10+, and the Radiance Display Engine in RDNA 3/4 GPUs, replacing outdated tools for tasks like FreeSync certification, HDR compliance, and firmware regression tracking.

- Built a React 18 + TypeScript dashboard to visualize real-time telemetry data from AMD GPUs, using secure REST APIs and WebSocket streams.
- Developed the HDR Activation Timeline Viewer to graph HDR metadata frame-by-frame and identify color space mismatches.
- Created the FreeSync Link Status Monitor to track refresh rate drift, state changes, and recovery attempts in real time.
- Designed and implemented a Display Faults Heatmap showing historical blanking, flickering, and sync loss across GPU models and firmware versions.
- Collaborated with firmware and driver teams to align telemetry schemas and ensure forward compatibility as RDNA 3/4 display technologies evolved.
- Managed frontend state and caching using React Query, enabling responsive navigation even with large telemetry payloads.
- Integrated debounced filter inputs and batched state updates to maintain UI performance under heavy data streams.
- Optimized rendering performance with lazy loading, memoized selectors, and virtualized tables for massive telemetry datasets.
- Engineered real-time WebSocket listeners, reducing telemetry delay.
- Built component-level error boundaries to gracefully handle and report front-end failures without interrupting QA workflows.
- Designed nested navigation and advanced filters with React Router v6, streamlining root-cause analysis for QA engineers.
- Aligned the dashboard UI with AMD's corporate design system and collaborated with QA, firmware, and validation teams in bi-weekly Agile sprints.
- Ran feedback sessions with QA users to iterate on interface usability and telemetry visualizations.

---

## **Full-Stack Engineer – Rosenblum Law, NY & NJ (Jul 2021 – May 2023)**

**Project: CaseSight Portal** – AI-powered OCR and case intake platform

**Company Overview:** Rosenblum Law specializes in traffic ticket defense and criminal law across New York and New Jersey, handling high-volume case intake from both in-person and online submissions.

**Project Overview:** Rosenblum’s marketing/intake was supported by tools like Intaker for lead capture. Once potential clients submitted their scanned traffic tickets (via upload forms, email, or in-person), the firm’s internal staff would process the documents. **The CaseSight Portal** automated this back-office step, applying OCR and post-processing to reduce manual data entry and speed up case intake into the firm’s CMS.

- Collaborated with attorneys, paralegals, and stakeholders to define legal data requirements and design an OCR-driven case intake workflow tailored to traffic ticket variability and compliance standards.
- Participated in vendor evaluations, prioritizing services that offered strong privacy by prohibiting data reuse for model training by third-party providers.
- Evaluated and integrated third-party OCR services (Azure Computer Vision, Google Document AI), selecting a custom post-processing model approach to maximize accuracy and control processing costs.
- The frontend is designed to avoid storing any raw scanned images or extracted data in the browser’s cache or local storage, reducing client-side data vulnerability.
- Built a TypeScript dashboard featuring a reusable table component, diff viewer HOC, and a lazy-loaded OCR preview pane for large PDFs.
- Managed frontend state using React Query for server cache synchronization and Redux slices for offline draft storage and optimistic UI rollbacks.
- Embedded the portal into the firm’s Django-based website using a Django-React SSR bridge for improved SEO and faster load times.
- Connected frontend workflows to a Django REST Framework API managing Tickets and Revisions, backed by PostgreSQL.
- Integrated with a containerized OCR service using Docker, Azure Computer Vision API, and asynchronous Celery/RabbitMQ job queues.
- Conducted usability testing with legal staff, iterating UI and OCR confidence thresholds based on real-world feedback.
- Implemented automated testing with PyTest, Jest, and Cypress.

## **Front-End Engineer – Ironbound CC, Newark, NJ (Feb 2020 – May 2021)**

**Company Overview:** Ironbound Community Corporation provides early childhood education services and community programs across Newark, NJ.

**Project: Roster PWA** – Offline-first child check-in and staff scheduling platform

**Project Overview: Roster** replaced paper-based child drop-offs, staff time tracking, and payroll preparation with an offline-first web application. The system enabled real-time local recording of check-ins and shifts, syncing seamlessly to the server once Wi-Fi was available — cutting auditing preparation from days to hours.

- Developed accessible semantic HTML5 forms with responsive Flexbox/Grid layouts, validated for WCAG color contrast and mobile compatibility.
  - Built and maintained React 18 functional components with hooks, wrapping legacy class components in error boundaries for stability.
  - Managed app-wide state using Redux Toolkit and redux-thunk, treating synced Firestore documents as the single source of truth following the Flux pattern.
  - Designed nested navigation for Children, Staff, and Payroll views with React Router, implementing client-side routing for instant UX feedback.
  - Optimized performance by code-splitting heavy calendar components using React.lazy and Suspense, delivering assets via Cloudflare CDN.
  - Styled the application using SCSS with theme-based mixins, maintaining live component previews in Storybook for faster QA sign-off.
  - Practiced Test Driven Development (TDD) with Jest (unit tests), React Testing Library (UI tests), and Cypress (E2E flows), maintaining >85% test coverage in CI pipelines.
  - Used GitHub for version control, following a trunk-based development model. Participated in two-week Scrum sprints with regular product demos to daycare teachers for feedback and iteration.
-