Brad Yin

Senior Software Engineer

7732579347 | bradyinrt@gmail.com | Linkedin | Portfolio

I'm a Experienced full-stack engineer with 5+ years building secure, production-ready web apps using TypeScript, React, Node.js, and AWS. Proven track record integrating third-party services, debugging in high-stakes environments, and modernizing legacy systems. Adept at writing testable, maintainable code, supporting production rollouts, and working across hybrid Agile teams. Strong communicator comfortable interfacing with stakeholders and translating technical details to cross-functional teams.

- Senior Software Developer with an eye for visual design, and focused on creating great user experiences.
- Over 5 years of experience building modern web and mobile applications using React, React Native, TypeScript, Node.js, and supporting technologies like Redux, React Router, Expo, HTML5, CSS3, Sass, and Bootstrap — with hands-on work in offline support, camera integration, and background syncing.
- Practical experience using Golang to build lightweight backend microservices for file processing and secure API integrations.
- Graduated from NYU with a bachelor degree in Computer Science and Data Science
- Extensive experience with ReactJS, NodeJS, and their core principles.
- 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.
- Professionalism developing server-side applications using Express, OAuth 2.0, Passport in Node.js.
- Proficient 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 efficient version control as well as task management using Git and JIRA.
- Experienced in creating responsive and animated web designs on Figma or Framer.
- Comfortable collaborating with cross-functional stakeholders to gather requirements, deliver tailored solutions, and support end-user adoption in client-facing environments.
- Self-driven and capable of managing multiple priorities under pressure.
- Leverage AI tools like Cursor or custom built assistants to speed up refactoring, catch edge cases, and explore alternate solutions during development.

SKILLS

Cloud / DevOps: AWS (Lambda, ECS, IAM, DynamoDB, CloudFront), Docker, GitHub, CI/CD, Celery, RabbitMQ

Frontend: TypeScript, React 18, Angular, Redux, React Query, RXJS, HTML5, CSS3, Webpack, Cypress, Jest, Jasmine

Backend: Node.js, Express.js, Nest.js, Java (Spring Boot), RESTful APIs, OAuth2, PostgreSQL, MongoDB, DynamoDB, GoLang

Monitoring: Splunk, Chrome DevTools, Lighthouse

Testing: Jest, Cypress, React Testing Library, PyTest, TDD, Integration & E2E Testing

Workflow: Agile/Scrum, Figma, Framer, Storybook, Jira, Postman

Work History

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

Company: AMD is a leading semiconductor company producing high-performance GPUs, CPUs, and display technologies for gaming, professional visualization, and data centers.
Project: Radeon Display Insights Portal – Internal telemetry dashboard for display validation
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.

Key Contributions:

- Joined a team of 5 to construct the Radeon Display Insights Portal using React 18 and TypeScript 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.
- Co-developed an internal React Native mobile app for QA engineers to scan display serials, view test telemetry, and log faults while moving between test setups.
- Implemented QR-based pairing between test displays and mobile devices, with local SQLite
- caching and background sync to the main dashboard.
- 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.

Key Contributions:
- Collaborated with legal staff to define structured data models for traffic tickets and build an OCR workflow tuned to jurisdiction-specific layouts and constraints.
- Participated in vendor evaluations (Azure, Google Document AI), choosing a hybrid approach combining cloud OCR with custom post-processing for accuracy and cost efficiency.
- Developed a Golang microservice to manage OCR task routing and webhook ingestion from Azure's Vision API. The service handled retries, rate limiting, and job-level telemetry.
- Used Go's concurrency model to process and normalize OCR results in parallel, significantly reducing latency for multi-page ticket batches.
- Integrated the Go service with Celery and RabbitMQ as part of an asynchronous pipeline, ensuring traceable OCR job states and recovery on failure.

- Built the frontend dashboard in TypeScript + React, featuring reusable table components, a diff viewer HOC, and lazy-loaded PDF previews to inspect OCR outputs.
- Used React Query for server-state sync and Redux slices to support offline drafts and optimistic updates across complex legal form flows.
- Embedded the portal into the firm's Django site via SSR for seamless navigation and improved first paint metrics.
- Connected the React frontend to a Django REST Framework API, with PostgreSQL-backed models for ticket tracking and audit revisions.
- Implemented automated test suites across the stack: PyTest for Django, Jest for UI logic, and Cypress for end-to-end QA coverage.
- Conducted iterative UX testing with non-technical staff to refine form design, highlight OCR confidence levels, and reduce training friction.

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 *PWA* 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, cutting auditing preparation from days to hours.

Key Contributions:
- 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.
- Developed a hybrid PWA and React Native interface to support check-in on low-end Android tablets used in classrooms, enabling full offline flow and later sync.
- Worked on native integrations like device camera access for parent signature capture and background sync via WorkManager.
- 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.
- Styled the app using SCSS with live previews in Storybook.
- Optimized performance by code-splitting heavy calendar components using React.lazy and Suspense, delivering assets via Cloudflare CDN.
- Practiced TDD with Jest, React Testing Library, and Cypress, maintaining high test coverage in CI.
- Used GitHub and trunk-based development, participated in two-week sprints and product demos with teachers for iterative feedback.