**Eric Modi Senior Software Developer**

807 Telluride Dr, Piscataway, NJ | (903) 272-1775 | techericmodi@gmail.com

M.S. New Jersey Institute of Technology, New Jersey, 2020

<https://www.linkedin.com/in/eric-modi-891b9a228>

Senior Software Developer with 15+ years of progressive experience specializing in the React ecosystem. Possess a proven track record of leading full-cycle development of complex, high-stakes web applications across diverse sectors including Finance, Healthcare, and Manufacturing. Expertise in architecting scalable solutions, optimizing performance, and mentoring teams to drive project success and technological innovation.

* **Frontend:** React.js, TypeScript, Angular, JavaScript (ES6+), HTML5, CSS3, Context API, React Query, Redux Toolkit
* **Testing:** Jest, React Testing Library, Cypress
* **Build Tools:** Webpack, Vite, Babel, ESLint, Prettier
* **Styling:** CSS-in-JS, Tailwind CSS, Sass, Material-UI
* **Backend & DB:** Node.js, Express.js, RESTful APIs, GraphQL, Dynamo DB, MongoDB
* **DevOps & Cloud:** AWS, AWS Lamda, CI/CD (GitHub Actions), Docker
* **Agile & Tools:** Jira, Confluence, Scrum, Kanban, Git, Figma, Storybook

**Professional Experience**

**Senior Software Developer** | **Berkshire Hathaway** | Apr 2023 - Present

* Lead the frontend architecture for a high-frequency trading analytics dashboard, utilizing React.js, TypeScript ensures maximum performance real-time data rendering for financial analysts.
* Developed dynamic and responsive user interfaces by building component-based applications with React, leading to improved code reusability and maintainability.
* Architected and lead the development of a high-frequency, low-latency trading analytics microservice using Node.js, Express, and Socket.IO, processing over 1 million real-time market events per minute with an average latency of under 50ms.
* Developed high-performance, scalable backend services and RESTful APIs using Node.js and the Express.js framework to serve millions of daily requests. Developed scalable Node.js backend services powering high-traffic Angular and React front-end applications.
* Built RESTful APIs in Node.js integrated with AWS services like S3, DynamoDB, and Lambda for full-stack solutions. Implemented Node.js clustering and load balancing, ensuring optimal performance for real-time data-heavy applications. Designed and deployed full-stack applications on AWS, integrating Angular, Node.js, and serverless backend services.
* Managed IAM roles and security policies in AWS, ensuring least-privilege access for applications and developers. Implemented AWS CloudFormation and CDK for Infrastructure as Code, automating provisioning of services across environments. Built event-driven serverless microservices in AWS Lambda integrated with API Gateway, DynamoDB, and S3.
* Optimized Lambda execution performance by tuning memory allocation and leveraging provisioned concurrency. Built modular Angular applications with lazy loading, significantly improving initial load times and enhancing user experience. Implemented Angular Reactive Forms with complex validations and custom directives, ensuring data accuracy across enterprise apps.
* Optimized Angular performance using change detection strategies (OnPush) and trackBy, reducing unnecessary DOM re-renders. Developed Angular services and interceptors for secure API communication, implementing JWT handling and error recovery mechanisms. Implemented Lambda event triggers for DynamoDB streams SQS queues, real-time backend processing.
* Managed source code repositories in Git across multiple teams, enforcing branch protection rules and pull request workflows. Established Git branching strategies (GitFlow, trunk-based development) to ensure smooth collaboration in full-stack projects. Automated Git hooks for linting, formatting, and security scans, improving code quality before merge.
* Implemented end-to-end CI/CD pipelines in Jenkins for Angular and Node.js applications, ensuring automated builds and deployments. Designed Jenkins declarative pipelines with scripted stages for build, test, and deploy processes. Integrated SonarQube and security scans in Jenkins pipelines, enforcing code quality and compliance.
* Wrote comprehensive UI tests with React Testing Library, validating component behavior and user flows. Developed mock REST services for integration tests in React and Angular applications. Implemented end-to-end testing pipelines with React Testing Library, production stability.
* Designed data models in DynamoDB optimized for mobile and web application access patterns. Implemented DynamoDB queries and scans using Node.js and AWS SDK, ensuring efficient data access. Built DynamoDB streams with Lambda triggers, event-driven backend processing.
* Lead the full software development lifecycle for multiple Node.js microservices, from design and coding to deployment and monitoring in cloud environments. Optimized Node.js application performance by profiling refactoring event-loop bottlenecks, implementing clustering, database queries, resulting in a 40% reduction in response time.
* Engineered serverless transaction processing system on AWS Lambda, leveraging DynamoDB peak loads of 5,000+ transactions per second while ensuring data consistency and integrity.
* Managed application state effectively using React's built-in useState and useContext hooks, ensuring predictable data flow and minimizing prop drilling across components.
* Utilized React Router to implement seamless navigation and create single-page applications with a fluid, app-like user experience. Enhanced performance of React applications by implementing optimization techniques such as React.memo to reduce unnecessary re-renders.
* Engineered a custom, reusable charting library built on D3.js integrated with React to visualize complex market data, resulting in a 40% improvement in data interpretation speed for end-users.
* Implemented rigorous security protocols and code obfuscation to protect sensitive financial data and intellectual property within the frontend application.
* Optimized critical rendering paths leveraged React.memo handle real-time WebSocket data streams, reducing UI latency over 50ms. Mentored junior developers advanced TypeScript patterns state management best practices, the team's code quality reducing bugs by 25%.
* Wrote comprehensive unit and integration tests using Jest and React Testing Library, achieving, and maintaining 95% test coverage for all new components.
* Automated the build and deployment process using Jenkins and Docker, creating a reliable CI/CD pipeline for seamless production releases. Collaborated with product owners to translate complex regulatory and business requirements into detailed technical specifications and agile user stories.

**Senior Software Developer** | GAF | Dec 2020 – Feb 2023

* Developed and maintained a large-scale supply chain and logistics management system using React, Redux Toolkit, and Material-UI to track inventory across global warehouses in real-time.
* Integrated React applications with backend RESTful APIs and GraphQL endpoints using async/await and the useEffect hook for efficient data fetching and side-effect management.
* Built reusable and scalable UI component libraries in React, promoting design consistency and accelerating development velocity across multiple projects. Wrote comprehensive unit and integration tests for React components and hooks using testing libraries like Jest and React Testing Library to ensure reliability and reduce bugs.
* Created Node.js middleware to centralize error handling, logging, and performance monitoring.
* Designed Node.js microservices with Express.js, supporting modular service deployments within AWS. Refactored monolithic backend systems into Node.js microservices, improving scalability and reducing time-to-deploy.
* Integrated AWS S3 and CloudFront with Angular apps for optimized static content delivery. Monitored applications using AWS CloudWatch dashboards and alarms, enabling proactive incident resolution. Migrated legacy front-end code to modern Angular versions, improving maintainability and aligning with enterprise best practices.
* Leveraged Angular Material and PrimeNG to create reusable UI components with consistent theming and accessibility standards. Integrated Angular apps with RESTful and GraphQL APIs, ensuring seamless data flow between client and backend. Enhanced Angular testing coverage using Jasmine and Karma, delivering stable and production-ready front-end releases.
* Designed multi-environment deployments on AWS, streamlining CI/CD workflows. Secured Lambda functions with least-privilege IAM execution roles, reducing security risks. Designed error-handling and retry strategies in Lambda with DLQs for reliable message processing.
* Integrated Lambda with third-party APIs to extend application features keeping infrastructure lightweight. Led code reviews and Git merge workflows, mentoring developers on best practices. Integrated Git with Jenkins and CI/CD pipelines, ensuring automated builds and deployments.
* Designed multi-repo and mono-repo strategies with Git to optimize large-scale mobile and web projects. Automated container builds with Jenkins and Docker, pushing images to AWS ECR for ECS deployments. Configured Jenkins multi-branch pipelines for GitHub, parallel environment builds.
* Monitored Jenkins pipeline executions and optimized build times with caching strategies. Created custom test utilities with React Testing Library, streamlining repetitive testing patterns across apps.
* Integrated React Testing Library into CI/CD pipelines, ensuring automated test execution before deployments. Refactored existing Enzyme tests into React Testing Library, aligning with modern testing best practices. Applied DynamoDB partition key and sort key strategies, improving query performance and scalability.
* Implemented TTL and GSI in DynamoDB for efficient data management. Automated DynamoDB backups and point-in-time recovery, ensuring data resilience and compliance. Spearheaded development real-time data ingestion platform using Node.js, collecting sensor data from 10,000+ factory floor devices monitor equipment health predictive maintenance schedules.
* Containerized Node.js applications using Docker and orchestrated them with Kubernetes, ensuring consistent environments from development to production and simplifying scalability.
* Designed and implemented a real-time communication layer using Node.js and Socket.IO, enabling features like live chat, notifications, and data streaming for a seamless user experience.
* Established robust CI/CD pipelines specifically for Node.js projects using Jenkins/GitHub Actions, automating testing, linting, and deployment processes to increase release frequency.
* Implemented advanced React patterns like custom hooks to abstract complex logic into reusable functions, simplifying components and streamlining development.
* Designed and implemented a drag-and-drop interface for factory floor planning, improving operational efficiency for plant managers. Lead the migration of a legacy AngularJS application to a modern React, improving maintainability and page load performance by 60%.
* Championed the adoption of React Query for asynchronous state management, simplifying data fetching logic and eliminating redundant network requests. Integrated Mapbox GL JS with React to create real-time asset tracking maps, providing logistics teams live visibility shipment locations.
* Conducted performance audits using Lighthouse, identifying, and resolving bottlenecks related to memory leaks and large bundle sizes. Created a shared component library with Storybook, ensuring design consistency and accelerating development across multiple product teams.

**Senior Software Developer** | Medline Industries | Jan 2017 – Nov 2020

* Developed critical features for a patient portal and electronic health record (EHR) system using React.js, ensuring strict compliance with HIPAA regulations and data security standards.
* Configured and managed React application builds and deployments using modern tooling like Create React App, Webpack, and Vite for an efficient development workflow. Applied asynchronous programming with Node.js to optimize concurrent workflows.
* Wrote comprehensive unit and integration tests in Node.js with Mocha/Chai, increasing service reliability. Leveraged AWS Elastic Beanstalk and ECS to deploy containerized Node.js applications for high scalability. Implemented AWS Cognito authentication in Angular and Node.js apps, ensuring secure and seamless user sign-in flows.
* Automated Lambda deployments using Jenkins CI/CD pipelines, ensuring zero-downtime rollouts.
* Instrumented CloudWatch logs and X-Ray tracing in Lambda, improving observability of distributed workflows. Managed Git submodules and dependencies across interconnected Angular and Node.js repositories. Resolved complex Git conflicts in large codebases, ensuring smooth collaboration across distributed teams.
* Set up Jenkins distributed build agents for faster, scalable build execution. Automated blue/green and canary deployments with Jenkins, ensuring minimal downtime during production releases.
* Ensured accessibility testing in React Testing Library by validating ARIA roles, labels, and keyboard navigation. Mentored team members in writing clean, maintainable test cases with React Testing Library, improving coverage and reducing flaky tests. Integrated DynamoDB with GraphQL APIs in AWS AppSync, providing scalable serverless data access for front-end apps.
* Tuned DynamoDB read/write capacity units and caching with DAX, optimizing application performance under heavy load. Developed and maintained critical, HIPAA-compliant backend services for a patient portal and EHR system using Node.js, Express, and PostgreSQL.
* Migrated legacy monolithic applications to a modern microservices architecture built with Node.js, improving developer agility, system resilience, and deployment independence.
* Secured Node.js APIs by implementing authentication with JWT, authorization middleware, input validation, and rate limiting to protect against common vulnerabilities.
* Built highly accessible (WCAG) UI components to ensure the application was usable by patients and providers with disabilities. Worked extensively with React, Redux to manage complex application state involving patient data, medical history, and appointment details.
* Contributed to a monorepo codebase using Lerna and Yarn workspaces to manage shared dependencies across multiple frontend projects. Wrote extensive unit and end-to-end tests with Jest and Cypress to guarantee the reliability and accuracy of healthcare-related functionalities.
* Participated in a 24/7 on-call rotation to provide support for critical healthcare applications, demonstrating responsibility and problem-solving under pressure. Collaborated with backend teams to design RESTful APIs for fetching and updating sensitive patient health information.

**Software Developer** | PWC | June 2013 – Dec 2016

* Developed internal and client-facing web applications for audit, tax, and advisory services using JavaScript, React, and Node.js.
* Aligned frontend error messages and alert behavior in React with backend exceptions tested, ensuring consistency in full-stack error handling and UX. Used Jasmine's fakeAsync to simulate time-based interactions and verify React component behavior under controlled timing scenarios.
* Developed highly dynamic dashboards using ReactJS with D3.js and Chart.js, integrating real-time data feeds and responsive graphs using WebSockets and REST APIs.
* Migrated legacy jQuery and AngularJS views to modern ReactJS architecture, including routing and performance optimizations using React concurrent features. Optimized web performance by handling simple DOM updates JavaScript, bypassing React rendering for non-critical components.
* Created pure JavaScript components using class-based inheritance and custom elements, enabling smooth integration with React via ref and dangerouslySetInnerHTML. Used Webpack Bundle Analyzer to identify and reduce bundle bloat by removing unused packages and optimizing third-party dependencies in React applications.
* Contributed to full-stack development tasks, including building Node, Express.js APIs and working with MongoDB databases. Participated in code reviews and quality assurance processes, ensuring deliverables met PwC's high standards for client work.

**Software Developer** | American Express | Feb 2011 – May 2013

* Developed and maintained dynamic, data-rich user interfaces for the American Express online services portal using a stack centered on jQuery, Backbone.js, and Handlebars templating, ensuring a seamless and interactive customer experience.
* Engineered complex single-page application (SPA) features by implementing the MVC architecture with Backbone.js (Models, Views, Collections) and utilized Underscore.js for key functional programming utilities to manipulate data and manage client-side logic.
* Implemented scalable JavaScript architecture using RequireJS manage dependencies, significantly improving code maintainability, team collaboration, and application performance.
* Built responsive and cross-browser compatible layouts with HTML5 and CSS3, leveraging modern techniques while ensuring graceful degradation for older browsers to support a diverse user base.
* Utilized JavaScript templating engines Handlebars.js to cleanly separate presentation logic from business logic, dynamically generating HTML content from JSON data returned by RESTful APIs.
* Wrote comprehensive unit tests for JavaScript modules using QUnit or Jasmine to validate functionality, ensure code quality, and prevent regressions across development cycles.
* Collaborated closely with back-end developers to integrate front-end components with RESTful web services, efficiently parsing JSON responses and managing asynchronous data flow using jQuery’s AJAX Deferreds and Promises.
* Improved front-end development workflow by adopting CSS pre-processors like SASS or LESS to utilize variables, mixins, and nested rules, leading to more organized and efficient stylesheets.
* Participated in code reviews and contributed to the evolution of front-end development standards and best practices within team, advocating for performance optimization accessibility guidelines.
* Automated repetitive tasks such as code linting, minification, and concatenation using Grunt.js, contributing to a more efficient build process and optimized assets for production deployment.