

Nguyen Hai Trinh

ReactJS Engineer

GitHub | LinkedIn

Email: n.haitrinh2810@gmail.com

Mobile: 0384945724

Ho Chi Minh City, Vietnam

PROFESSIONAL SUMMARY

- **Goal-oriented Frontend Developer** with strong experience in building scalable web applications using modern technologies. In the next **3 years**, I aim to become a **Senior Frontend Developer**, mastering system design and leading frontend best practices. Within **5 years**, I plan to take on the role of a **Tech Lead**, driving product and engineering decisions. My long-term vision, in the next **7–10 years**, is to **grow into a Software Architect or Engineering Manager**, designing robust, scalable, and maintainable software systems across large-scale platforms.

EDUCATION

- **Van Lang University** Ho Chi Minh City
B.S. in Software Engineering 2019 – 2024

LANGUAGES

- **English:** Intermediate (Working proficiency – can read/write documentation and discuss requirements with clients)

TECHNICAL SKILLS

- **Languages:** HTML5, CSS3, SCSS, JavaScript (ES6+), TypeScript
- **Frontend:** React.js, Next.js, React Router, Redux Ecosystem (Redux Toolkit, Redux Thunk, Redux Saga), Tanstack Query, Apollo Client (GraphQL), Tailwind CSS, MUI, Ant Design, React Hook Form, Formik
- **Testing:** Jest, React Testing Library
- **Backend:** Express.js, Spring Framework, MySQL, MongoDB
- **Tooling:** Webpack, Vite, Rsbuild, Docker, Git, Bitbucket Pipelines, ESLint, Prettier, Biome

EXPERIENCE

- **Bolt Technologies** Contract Freelance
Fullstack Engineer **April 2025 – Now**
 - Spearheaded Nano Trading platform (derivatives) with Vietnam Commodity Exchange APIs and Backoffice system for account management. Built real-time trading UIs with WebSocket and TradingView, ensuring high performance and successful production launch.
 - Extended platform into F1 ecosystem: developed CMS dashboard (Vue 3/TypeScript) and web portal with content management, approval workflows, and Spring Boot APIs supporting multi-platform trading operations.
- **Eye Design Sydney** Contract Remote
Software Enginner (Frontend Lead) **Dec 2024 – Now**
 - Lead the frontend development of KOMO Booking and KOMO Admin, establishing a scalable architecture, optimizing build and deployment pipelines, and improving performance and usability through clean code practices, close collaboration with backend teams, and continuous refinement of UI/UX to ensure seamless booking experiences and efficient salon management.
- **WATER AND ENVIRONMENT SOLUTIONS SINGAPORE (WNE)** Ho Chi Minh City
Frontend Developer (IoT Platform) **Nov 2023 – May 2025**
 - I actively participated in the development of innovative solutions aimed at efficiently managing electricity and water consumption in buildings through the integration of cutting-edge IoT devices such as smart meters. In addition to my role as a Frontend Developer, I embraced the Agile/Scrum methodology, collaborating closely with cross-functional teams to deliver high-quality results.
- **ISB Vietnam** Ho Chi Minh City
Frontend Developer (Banking Platform) **Dec 2022 – Nov 2023**
 - I was involved in developing a banking system in Japan, with our client being a leading financial conglomerate. In this project, I took on the roles of both Frontend Developer and Manual Tester, ensuring the quality and accuracy of the product. I carried out the work following the Agile/Scrum methodology, closely collaborating with other team members to achieve our goals and desired outcomes.

- As a Software Engineer, maintained legacy systems, optimized code performance, and implemented new data-driven features including reporting dashboards and revenue tracking modules for Sales and Marketing.

PROJECTS

◦ **F1 Trading Ecosystem**

Team Size: 6 members

Description: Public web + CMS + backend services extending the trading platform; content operations, onboarding, and real-time board prices.

My Role:

- * **Public Web (f1_web):** Took ownership of app architecture and UX for onboarding, account, and content modules; introduced guarded routing, server-driven pagination/filters, lazy loading/code-splitting, standardized error/loading patterns, and WebSocket consumption for live prices to keep pages responsive under load.
- * **CMS Frontend (f1_cms):** Led build of a TypeScript-based admin with role-aware navigation and reusable table/form primitives; integrated CKEditor 5 (tables/media/paste-from-office); implemented end-to-end approval workflows; established UI/state conventions with Pinia, Tailwind, PrimeVue/AntD for consistent, maintainable features.
- * **Core Services (f1_be):** Designed and implemented essential REST endpoints and authentication (JWT access/refresh), basic rate limiting, CORS, and WebSocket/STOMP channels for real-time board prices; aligned API contracts with FE, unified response envelopes/error codes, and applied MySQL schema/index tuning with MongoDB for logs/cache.
- * **CMS Services (f1_cms_be):** Delivered posts/jobs CRUD, categories/brokers, search with pagination, and approval lifecycle; added validation/DTO mapping, structured logging, and rate-limit protection; standardized Maven profiles and Docker images for predictable builds and multi-env deployments.

Technologies Used:

- * **Frontend:** Vue 3, TypeScript, Vite, Vue Router, Pinia, Tailwind CSS, PrimeVue, Ant Design Vue, Font Awesome, CKEditor 5 (tables, media, paste-from-office), Flatpickr, Axios, Virtual Scroller, lazy loading, code splitting, SEO/meta handling.
- * **Backend Services:** Java 21, Spring Boot 3.x (Web, Security, Validation), JWT (access/refresh), CORS, basic rate limiting, WebSocket/STOMP for live board prices, JPA/Hibernate, Lombok, Logback.
- * **Data:** MySQL (schema design, indexing, pagination), MongoDB (logs/cache, token/OTP where applicable).
- * **Integrations:** eKYC/eContract (FPT), Banking/QR payments, SMS/OTP, Telegram Bot.
- * **DevOps & Tooling:** Maven (mvn), Docker, Jenkins (CI), Git, Yarn, Pnpm, multi-environment configs (staging/production), structured logging, deploy scripts.

◦ **Trading Nano**

Team Size: 6 members

Description: A derivatives trading platform integrated with the official APIs of the Vietnam Commodity Exchange, featuring **Advanced Charts (TradingView)** for technical analysis. Backend built with **NestJS microservices, PostgreSQL, RabbitMQ, Go (FIX Protocol)**. Released to production with real traders and revenue generation.

My Role:

- * Bootstrapped the frontend codebase from scratch, including project architecture, routing, state management, and build setup with Vite.
- * Implemented GraphQL (Apollo Client) for API integration, cache management, and optimized data fetching with real-time updates.
- * Built responsive and consistent UIs using React 19, TailwindCSS, and MUI, aligned with the design system.
- * Optimized frontend performance, reducing bundle size, improving chunk splitting, applying lazy loading and tree shaking for efficient builds.
- * Integrated TradingView Advanced Charts with a custom datafeed to load and render market data seamlessly.
- * Connected WebSocket streams to enable real-time chart updates, synchronized order book, and trade executions.
- * Integrated React-Toastify for real-time notifications, including order status and system alerts.
- * Closely collaborated with backend teams (NestJS, Go, RabbitMQ, PostgreSQL) to refine API contracts, ensure reliable real-time data, and support a high-frequency trading environment.

Technologies Used: React.js (v19), TailwindCSS, MUI, React-Toastify, React-Router, Vite, GraphQL (Apollo Client), WebSocket(Socket.io), TradingView Advanced Charts, Jenkins, NestJS microservices, GraphQL server, PostgreSQL, TypeORM, RabbitMQ, Go (order service with FIX Protocol)

○ KOMO BOOKING & KOMO ADMIN

Team Size: 10 members

Description:

- * **KOMO** is an end-to-end beauty service booking platform consisting of two separate web applications, designed to connect customers with salons and help salon owners streamline their operations.
- * **KOMO Booking** is a web-based user-friendly application that enables customers to discover, browse and book beauty services such as haircuts, nail care, massages, and more at nearby salons. Users can view available time slots, select preferred services, and manage their bookings through a clean and intuitive interface optimized for both mobile and desktop devices.
- * **KOMO Admin** is a management dashboard for salon owners to oversee all aspects of their operations, including branch information (locations), service listings, staff (therapists) and customer bookings. The platform helps digitize the salon management workflow, reduces manual errors, saves time, and improves operational efficiency for medium to large salon chains and franchises.
- * The system aims to modernize the traditional salon booking experience, offering a seamless experience for customers and a powerful management tool for salon owners.

My Role:

- * Took full ownership of designing and implementing the frontend architecture for both KOMO Admin and KOMO Booking from scratch, focusing on scalability, reusability, and maintainability.
- * Replaced Create React App with modern build tools — used Rsbuild for the Admin app and Vite for the Booking app — to optimize build performance and development speed.
- * Structured the entire codebase, including routing, layouts, reusable components, hooks, store, and service layers; enforced clean folder organization and naming conventions across both apps.
- * Built key product features based on requirements from the Product team, ensuring business logic was accurately implemented and UI matched the provided Figma designs pixel-perfectly.
- * Used Material UI (MUI) as the design foundation, customizing components according to the project's design system to ensure consistency and responsiveness across all viewports.
- * Collaborated closely with backend developers to align API contracts, handle async logic, validate and normalize response data effectively.
- * Optimized performance in production builds with code splitting, lazy loading, tree shaking, and efficient dependency management.
- * Integrated a full CI/CD pipeline using Bitbucket Pipelines, automating deployment to staging and production environments.
- * Established and enforced code quality standards using ESLint, Prettier, and Biome; maintained a clean and readable codebase for easy onboarding and team collaboration.
- * Frequently initiated code refactors, reviewed pull requests, and contributed feedback to improve product quality and team workflows.

Technologies Used: React.js, Typescript, Javascript, React-Router, Axios, Redux Toolkit, MUI, Vite, Rsbuild, Figma, Formik, Yup, Lodash, moment, React Big Calendar, Bitbucket Pipelines(CI CD)

○ WEB APPLICATION SYSTEM FOR IOT

Team Size: 12 members

Description: Our current project focuses on developing solutions to manage electricity and water consumption for buildings using advanced IoT devices such as smart meters. We are building this platform with the aim of serving future smart city projects, promising efficiency and convenience for the community.

My Role:

- * I participated in developing new features for the project, improving the project's UI/UX, and addressing lingering bugs.
- * Utilized Next.js for server-side rendering to improve application performance and SEO optimization.
- * Collaborated with backend developers to enhance site speed, manage errors, and customize interface and features based on client specifications.
- * Contributed to product development, debugging, and source code optimization for optimal performance.

Technologies Used: React.js, Lodash, Redux Core, Redux Saga, Redux Form, React i18next, React Hook Form, Redux Saga, React Router Dom, Moment, Bootstrap, Material UI, Yup, Jest, React Testing Library, Webpack, Axios, Firebase, Google Map React, Prop-types, Rechart, Webpack and more.

○ BANKING WEB APPLICATION SYSTEM Position: Frontend Developer and Manual Tester

Team Size: 30 members

Description: This is a project involving a banking system in Japan, with our client being a prominent financial conglomerate. Presently, the system has been operationalized, although there remain several modules requiring maintenance and upgrades. Due to the sensitive nature of this project and my adherence to a confidentiality agreement, I am unable to provide extensive details about it.

My Role:

- * Throughout the project, I conducted thorough testing on roughly 100 pages, leading to the discovery and logging of over 100 issues. These challenges encompassed various aspects, such as incorrect currency and text formats, form validation errors, missing fields in API requests, and incomplete data returned by APIs.
- * In addition to testing, I actively contributed to the system maintenance phase by addressing various issues and implementing additional features across more than 50 different pages, as per client requirements.
- * Furthermore, I collaborated closely with backend developers to enhance site performance, troubleshoot errors, and adapt interface elements and functionalities based on client feedback.
- * My role also involved participating in code reviews, debugging sessions, and optimizing source code to ensure optimal performance.

Technologies Used: Next.js(version 8) with Typescript, Redux Ecosystem(Redux core, Redux logger, Redux Saga, Reselect), immer, Formik, Yup, Ramda, React-i18next, Axios ,Styled component, Material UI, Storybook, Jest.