

JATIN SHARMA

Frontend-Oriented Full Stack Engineer | SDE | 3 YOE | **Immediate Joiner**

+91 9368558623 | raghavjatin245@gmail.com | [linkedin.com/in/raghav1924](https://www.linkedin.com/in/raghav1924) | github.com/raghav1924 | jatinsharma.in

SUMMARY

Frontend-oriented Full Stack Engineer with around 3 years of experience building scalable, production-grade web applications using **React.js**, **Next.js**, **JavaScript**, and **TypeScript**, backed by **Node.js** serverless APIs on **Microsoft Azure**. **Reduced** client onboarding time from **7–8 months to under 30 minutes** and **improved** feature delivery speed by **43–67%** through reusable component systems. Focused on building fast, intuitive, and accessible UI experiences with end-to-end ownership from design to production deployment.

TECHNICAL SKILLS

• React.js • Next.js • JavaScript (JS) • TypeScript (TS) • HTML • CSS • Tailwind CSS • Redux • Node.js • REST APIs • GraphQL • Azure Functions (Serverless) • NestJS (foundational) • JWT • RBAC • Authentication and Authorization • PostgreSQL • MySQL • Supabase • Azure Cosmos DB • DBMS (Database Management Systems) • Database Design and Schema Design • Row Level Security (RLS) • Microsoft Azure (Blob Storage, AKS, APIM, Synapse) • AWS (foundational) • Docker • CI/CD • GitHub • Redis (foundational) • Zod (Validation) • Responsive Design • Component-Based Architecture • Micro-Frontend Architecture • System Design • Data Structures and Algorithms (DSA) • Object-Oriented Programming (OOPs) • Low Level Design (LLD) • Design Patterns • Unit Testing • Performance Optimization • Agile • SDLC • Problem Solving

EXPERIENCE

Frontend-Oriented Full Stack Developer

Aug 2023 – Apr 2026

Cogitate Technology Solutions

- **Problem:** Teams rebuilding identical UI components repeatedly, slowing feature delivery.
Solution: Built and integrated reusable **React.js** / **Next.js** UI components with **TypeScript** for Nexus, an internal **micro-frontend** library; enforced **component-based architecture**, wrote adoption docs, and led code reviews on **GitHub**.
Result: Adopted across **4+ applications**, cutting delivery time by **43–67%**.
- **Problem:** Client onboarding required 7–8 months of manual setup with no automated frontend or backend workflows.
Solution: Built the complete frontend using **React.js** / **Next.js** and integrated it with **Node.js** backend on **Azure Functions**, **Azure Cosmos DB**, **Azure APIM**, and **GraphQL** layer; applied **responsive design** and **performance optimization** throughout.
Result: Cut client setup time from **7–8 months to under 30 minutes**.
- **Problem:** No analytics visibility for contract decisions; UI lacked data-driven views.
Solution: Built the full frontend and data layer for Customer Scorecard using **Next.js**, **Azure SQL**, **MySQL**, and **Azure Synapse** in 4 weeks; implemented **observability** through logging and monitoring; iterated UI directly with stakeholders.
Result: Delivered production-grade analytics dashboard **within 4 weeks**.
- **Problem:** APIs lacked consistent authentication, validation, and documentation across services.
Solution: Designed **REST** and **GraphQL** APIs using **Node.js** with full middleware (**JWT**, **RBAC**, **Zod** validation, error handling); applied **Design Patterns** in a **Microservices** setup deployed via **CI/CD** on **AKS**.
Result: Delivered reliable, documented APIs reused across multiple frontend applications.

PROJECTS

Draftly - Collaborative Document Workspace | *Next.js 15, TypeScript, Supabase, PostgreSQL, TipTap* |

2026

- **Problem:** Needed a production-ready collaborative document editor with secure multi-user access.
Solution: Built and deployed using **Next.js 15**, **TypeScript**, **Supabase**, **PostgreSQL**, and **TipTap** rich-text editor; implemented document CRUD, **authentication**, **Row Level Security (RLS)**, Owner/Editor/Viewer **RBAC**, debounced autosave, **Zod** validation, file import (.txt/.md), and **responsive design**.
Result: Deployed on Netlify with complete technical documentation; production-grade UI with secure **authorization** and persistent storage.

GraphQL Utility Tool | *Next.js, Node.js, GraphQL, TypeScript* |

2024

- **Problem:** Developers needed a reusable utility to test GraphQL schemas and queries.
Solution: Built and deployed a live web tool using **React/Next.js** and **Node.js** end-to-end, open for all developers to use.
Result: Live in production; used by developers for GraphQL schema testing and validation.

EDUCATION

NIIT StackRoute - PG Program in Full Stack Software Engineering (Sep 2022 – Feb 2023) | **AKTU University** - B.Tech (Aug 2018 – Jul 2022)