



FAIZAL RAHEEM

Full Stack Developer | MERN Stack | Microservices

faizalraheem76@gmail.com | linkedin.com/in/faizal-r | github.com/Faizal-R | +91 8089396776

PROFESSIONAL SUMMARY

Full Stack Developer specializing in the MERN stack with hands-on experience building and deploying scalable web applications across microservices and monolithic architectures. Skilled in TypeScript, real-time systems, event-driven design, and cloud deployments, with a strong focus on clean code, SOLID principles, and shipping production-ready products. Proficient in AI-assisted development tools to accelerate delivery workflows and maintain high code quality.

TECHNICAL SKILLS

Frontend Development: React.js, Next.js, JavaScript (ES6+), TypeScript, HTML5, CSS3, Tailwind CSS, Redux Toolkit, Zustand

Backend Development: Node.js, Express.js, TypeScript, REST APIs, JWT Authentication, Bcrypt, WebSockets, Socket.io, RabbitMQ, Vitest

Databases: MongoDB, PostgreSQL, Redis

Cloud and DevOps: Google Cloud Platform (GCP), AWS EC2, AWS S3, Docker, Docker Compose, CI/CD, GitHub Actions, Nginx, Linux

Integrations: Razorpay, Firebase, Cloudinary, Multer, WebRTC, Postman

Architecture and Patterns: Microservices Architecture, Modular Monolithic Architecture, Clean Architecture, MVC Pattern, Repository Pattern, SOLID Principles, Dependency Injection, InversifyJS, Factory, Singleton, Observer, Strategy Design Patterns

Concepts: Data Structures and Algorithms, Object-Oriented Programming (OOP)

Tools: Git, GitHub, Vercel, Render, Figma, GitHub Copilot, OpenAI Codex, Cursor, Antigravity

PROJECTS

Stratiffi — SaaS Technical Interview Outsourcing Platform

[Live Demo](#) | [Frontend](#) | [Backend](#)

A SaaS platform enabling startups to outsource candidate technical interviews to expert industry interviewers. Implements a two-stage evaluation pipeline: an AI-powered mock interview followed by a live one-to-one technical session for shortlisted candidates. Built end-to-end with TypeScript on both frontend and backend.

- Real-time video interviews built with **WebRTC** and **Socket.io** for in-session chat, supporting concurrent multi-user sessions with low-latency communication.
- Designed a **two-stage interview pipeline**: AI mock interview for initial screening followed by a live technical interview, reducing manual recruiter workload.
- Backend architected with **Repository Pattern** and **Clean Architecture**, implementing JWT authentication and RBAC for three user roles: admin, interviewer, and candidate.
- Subscription billing via Razorpay with plan management and an interviewer wallet and earnings system.
- Deployed on **Google Cloud Platform** with automated **CI/CD pipelines** via GitHub Actions and Nginx reverse proxy for production routing.
- **MongoDB Aggregation pipelines** powering analytics dashboards: interview completion rates, interviewer performance metrics, and revenue reporting.
- **TypeScript** used across the entire codebase (Next.js frontend and Node.js/Express.js backend) for end-to-end type safety and maintainability.

Tech Stack: TypeScript, Next.js, Node.js, Express.js, MongoDB, WebRTC, Socket.io, Razorpay, GCP, Docker, GitHub Actions

FigurLedger — Banking and Finance Platform using Microservices Architecture

[Source Code](#)

A fintech platform simulating modern banking systems using microservices architecture with fully decoupled, independently deployable services communicating via message queues. Built with TypeScript across all services for strict type safety in financial data handling.

- **Microservices architecture** across 7 domains: Authentication, Accounts, Transactions, Loans, Payments, Notifications, and Reporting — each following Clean Architecture and Repository Pattern.
- Centralized **Auth Service** with JWT-based authentication, RBAC, and secure cross-service token validation.

- **API Gateway** for unified service routing, request validation, and inter-service communication.
- **Event-driven communication** using RabbitMQ for asynchronous processing of financial transactions and notification delivery.
- **Redis caching** layer for frequently accessed account and transaction data, significantly reducing database query load.
- Containerized with **Docker and Docker Compose** for consistent local development and production deployment environments.
- **TypeScript** used across all microservices (Next.js frontend and all Node.js/Express.js backend services) ensuring type-safe API contracts between services.

Tech Stack: TypeScript, Next.js, Node.js, Express.js, MongoDB, PostgreSQL, Redis, RabbitMQ, Docker, Docker Compose

Pentaluxe — Full-Stack E-Commerce Platform

[Live Demo](#) | [Frontend](#) | [Backend](#)

A production-grade e-commerce application handling online product sales with secure authentication, payment processing, and a complete admin operations dashboard. TypeScript used on the React.js frontend for type-safe component development and API integration.

- **TypeScript frontend** built with React.js for type-safe components, props validation, and reliable API integration with the backend.
- Secure authentication and authorization workflows using **JWT** and bcrypt password hashing with protected route middleware.
- Dynamic product browsing with filtering, search, and real-time cart functionality using React state management.
- Razorpay payment gateway integration with order tracking and payment status handling.
- Admin dashboard for product CRUD operations, order management, inventory tracking, and sales reporting.
- Cloudinary image uploads via **Multer** middleware with server-side optimization and client-side preview.
- **MVC architecture** maintaining a modular, maintainable Express.js and Node.js backend structure.

Tech Stack: TypeScript, React.js, Node.js, Express.js, MongoDB, Razorpay, Cloudinary, Multer

EDUCATION

Professional Training — Full Stack Development (MERN Stack)

2024 – 2026

Brototype, Thiruvananthapuram [View Certificate](#)

- Built and deployed multiple production-grade applications including a SaaS interview platform, a fintech microservices system, and an e-commerce platform with real-time features and payment integrations.
- Designed scalable backend systems using Node.js, Express.js, and TypeScript, following Clean Architecture, Repository Pattern, and SOLID principles.
- Applied microservices architecture, event-driven communication with RabbitMQ, Redis caching, Docker containerization, and GCP cloud deployments in hands-on projects.
- Strengthened skills in debugging, performance optimization, and professional development workflows using Git, CI/CD, and AI-assisted tooling.