

# ATULYA KUMARI

Pune, India • +91-9123177821 • [atulyap1234@gmail.com](mailto:atulyap1234@gmail.com)

[LinkedIn](#) • [GitHub](#) • [atulyarai.space](#)

Fullstack Engineer with roughly 5 years of hands-on experience shipping fintech and enterprise products from scratch to production. I work primarily with Node.js, NestJS, React, and Next.js, building everything from payment backends and real-time chat systems to accessible, fast-loading frontends. Comfortable owning features end-to-end, setting up CI/CD pipelines, and working closely with cross-functional teams to get things out the door on time.

## PROFESSIONAL EXPERIENCE

---

### Infocusp Innovations LLP

2024 – Present

#### Software Engineer

- Own the backend for a fintech platform built on Node.js and NestJS. It handles user accounts, invoicing, and payment flows processing around 5,000 transactions daily against a PostgreSQL database I designed from the ground up
- Built the entire frontend in React, Next.js, and TypeScript. Brought initial page load down by roughly 40% after introducing code splitting, lazy loading, and server-side rendering where it actually mattered
- Locked down the API layer with JWT auth, role-based permissions, rate limiting, and thorough input validation. No security incidents since launch
- Wired up Stripe and Razorpay with webhook-based reconciliation and retry logic, which cut payment failures by about 25% compared to the previous integration
- Set up the monorepo (Turborepo) and GitHub Actions CI/CD from scratch. Build times dropped 60%, and three separate teams now deploy independently without stepping on each other

### Cybage Software Pvt. Ltd.

2021 – 2024

#### Software Engineer (Client: J.P. Morgan Chase)

##### Project 1: Corporate Travel Booking Engine

**Stack:** React.js, Next.js, TypeScript, GraphQL, Jest, REST APIs

- Wrote and maintained the Node.js/Express APIs powering flight, hotel, and car rental searches. Handled 10K+ lookups per week with average response times under 200ms thanks to query tuning and Redis caching
- Built accessible, responsive interfaces with React and Next.js that worked well on both mobile and desktop. User-reported UI bugs dropped about 30% after my accessibility pass
- Handled the payment integration, search filtering, and booking management flows end-to-end, all built to PCI-compliant standards
- Put together a solid test suite using Jest and Supertest. Got coverage above 85% on API endpoints and the key UI paths, which caught several regressions before they hit production
- Worked closely with backend, frontend, and QA folks across biweekly sprints. Shipped features on time through 12+ release cycles without any major rollbacks

##### Project 2: Real-Time Chat & Review Platform

- Built the messaging backend with Node.js and WebSockets from scratch. Supported 500+ users chatting simultaneously with message delivery consistently under 100ms
- Created the chat UI in React with Tailwind CSS, focusing on smooth animations and instant message rendering. Users rated the experience 4.7/5 in internal feedback surveys
- Implemented JWT-based auth with fine-grained access controls. Passed every internal security audit without critical or high-severity findings
- Improved real-time throughput by adding connection pooling, message queuing, and smart debouncing on the client side. Cut overall server load by around 35%
- Deployed everything to production with proper monitoring dashboards. Kept an eye on performance metrics weekly and reduced P1 incidents by half over six months

## PROJECTS

---

### FlowForge: No-Code AI Workflow Automation Platform

**Stack:** Next.js, Node.js, NestJS, PostgreSQL, Redis, Docker, Microservices, OpenAI API

- Built a drag-and-drop automation platform where users chain together AI-powered workflows. Connects email triggers to GPT-based summarizers, Slack notifications, and database writes, all without writing code
- Designed the system as a set of loosely coupled microservices: a workflow orchestrator, a task execution engine, an auth service, and a webhook listener, each deployed independently via Docker Compose
- The frontend is a Next.js app with a custom node-based visual editor (built on React Flow) where users wire up automation steps. State management runs on Zustand to keep interactions fast
- Implemented a job queue with BullMQ and Redis to handle async task execution, retries on failure, and scheduling. Workflows run reliably even when individual steps time out or throw errors
- Added OAuth2-based user auth, workspace-level permissions, and execution logging so teams can share and audit automations collaboratively

### VoxMirror: Real-Time AI Voice Cloning and Synthesis Agent

**Stack:** Next.js, Node.js, Express, Python (FastAPI), WebSockets, Docker, Microservices, ElevenLabs API

- Created a voice cloning application that lets users upload a short audio sample, train a custom voice model, and then generate speech in that cloned voice from any text input
- Split the system into three microservices: a Next.js frontend for recording and playback, a Node.js API gateway handling auth and job management, and a Python/FastAPI service wrapping the voice synthesis models
- Implemented real-time audio streaming over WebSockets so users hear generated speech as it is produced rather than waiting for the full file. Cuts perceived latency significantly
- Built a clean dashboard where users manage their voice profiles, view generation history, and tweak parameters like speed, pitch, and emotion. All built in Next.js with Tailwind CSS
- Containerized every service with Docker and set up a shared message bus (Redis Pub/Sub) for inter-service communication. The whole stack spins up locally with a single docker-compose command

## EDUCATION

---

**Bharati Vidyapeeth Deemed University, Pune, Maharashtra**

2017 – 2021

*Bachelor of Technology (B.Tech) – Information Technology | GPA: 8.20 / 10.0*

## SKILLS

---

- **Languages & Frontend:** JavaScript, TypeScript, React.js, Next.js, HTML5, CSS3, Tailwind CSS, Responsive Design, WCAG Accessibility
- **Backend & APIs:** Node.js, Express.js, NestJS, REST APIs, GraphQL, gRPC, WebSockets, Microservices Architecture
- **Databases:** PostgreSQL, MongoDB, Redis, SQL, Query Optimization, Schema Design
- **Auth & Security:** JWT, OAuth2, Role-Based Access Control, Input Validation, XSS/CSRF Protection, Rate Limiting, PCI Compliance
- **Testing & QA:** Jest, Supertest, Postman, End-to-End Testing, API Testing, 85%+ Code Coverage
- **DevOps & Infra:** Git, GitHub, GitHub Actions, Docker, CI/CD Pipelines, Turborepo, Monorepo Architecture, Linux
- **AI & Integrations:** OpenAI API, ElevenLabs API, Stripe, Razorpay, Webhook Integration, Third-Party API Design
- **State & Performance:** React Context, Zustand, Redux Toolkit, Code Splitting, Lazy Loading, SSR, BullMQ, Job Queues