






# Rohan Mishra

## Full-Stack Web Developer (React, Node.js / Express.js)


 mydearluffy093@gmail.com    7000476533    Satna, India    Rohan Mishra    RohanMishra47

 Rohan Mishra    Portfolio: Rohan Mishra

### Profile

**Full-Stack Developer** specializing in **Next.js (React)** and **TypeScript** for building resilient web applications. Proficient in architecting scalable **Node.js/Express** backends and utilizing **Prisma** for reliable database interaction. Experienced in managing end-to-end **Vercel/Render deployment pipelines**, demonstrating a commitment to professional production quality. Eager to contribute to dynamic teams and grow through real-world challenges.

### Education



**Bachelor of Computer Applications (BCA)** Satna, India  
Makhanlal Chaturvedi National University of Journalism and Communication, Bhopal — Satna, India   
*Expected Graduation: 2027*



### Skills



<b>Languages &amp; Frameworks:</b> JavaScript, TypeScript, React.js, Next.js, Node.js, Express.js, Prisma, PostgreSQL, MongoDB, HTML5, CSS3, Tailwind CSS	<b>Tools &amp; Platforms:</b> Git, Docker, REST APIs, VS Code, pnpm, Vercel, Render, CI/CD Principles
--	--

### Projects

**Note: Portfolio projects use free-tier hosting and may have brief initial load times**

**SyncBoard — Real-Time Collaborative Whiteboard**  
Tech Stack: Next.js, TypeScript, Socket.io, Express.js, Prisma, PostgreSQL, Zustand  
• [\[Live Site\]](#)  | [\[GitHub\]](#)   
• Built a **real-time collaborative whiteboard** using **Socket.io**, achieving **sub-100ms synchronization** via WebSocket event broadcasting and optimized state reconciliation across multiple concurrent users.  
• Implemented **collaborative undo/redo** using the **Command Pattern**, allowing users to selectively revert their own actions while preserving others' work, with support for **1,000-entry action history**.  
• Optimized network performance by **reducing traffic by 40%** through throttled emissions (**60 FPS**) and minimized database writes by **90%** using debounced auto-save, maintaining a responsive UX for **5+ simultaneous users**.  
• Designed a scalable **Zustand-based state architecture** with **room-based PostgreSQL persistence**, shareable room URLs, and **PNG/SVG export** functionality, deployed on **Vercel (frontend)** and **Railway (backend)**.

**Personal Job Tracker**  
Tech Stack: React, Tailwind CSS, Express.js, MongoDB, Docker, Hugging Face API (LLM Integration)  
• [\[Live Site\]](#)  | [\[GitHub\]](#)   
• Built a **full-stack job management dashboard** to create, update, and organize job applications with CRUD operations.  
• Integrated **authentication and a drag-and-drop Kanban board** for intuitive job status management.  
• Implemented **robust backend APIs supporting advanced filtering, sorting, and pagination** via query parameters.  
• Enabled **deep linking** by syncing frontend state (filters, sort, and pagination) with URL parameters for reproducible views.  
• **Optimized performance and maintainability** by offloading data operations to the backend.  
• Integrated an **AI-powered Job Fit Score** feature for intelligent, data-driven role matching.

**API Monitoring Dashboard**  
Tech Stack: Next.js, Node.js/Express, PostgreSQL Prisma, TypeScript, Vercel, Render  
• [\[Live Site\]](#)  | [\[GitHub\]](#)   
• **Designed and deployed a full-stack, real-time monitoring dashboard** for API endpoints, built with Next.js (Vercel) and Node.js/Prisma (Render).  
• Implemented **automated endpoint monitoring** with custom check intervals to provide continuous **uptime tracking** and immediate status updates.  
• Developed comprehensive **response time analytics** functionality, including historical charts for identifying performance trends and regressions.  
• Engineered a reactive, **auto-refreshing dashboard** UI to display current endpoint status and latency with minimal user interaction.  
• Utilized **Prisma** for reliable schema management and PostgreSQL interaction, ensuring data integrity across all monitoring checks.  
• Managed the professional deployment pipeline, including secure, dynamic **CORS policies** to whitelist Vercel origins.