Siddhant Munjamkar

IIT Roorkee, Roorkee, Uttarakhand – 247667

☎ +91-9403982378 ✉ siddhant@mt.iitr.ac.in LinkedIn GitHub Portfolio

Summary

Final-year engineering student at IIT Roorkee with hands-on experience building production-grade backend systems and distributed architectures. Delivered real-world projects handling **10K+ orders/day** and **5K+ emails/day** using Node.js, TypeScript, Kafka, Redis, and PostgreSQL. Strong foundation in system design, microservices, and event-driven architecture with proven ability to ship at scale.

Education

Indian Institute of Technology Roorkee (IIT Roorkee) Roorkee, Uttarakhand
Bachelor of Technology (B.Tech) – Final Year *Sep. 2022 – May 2026*

Technical Skills

Languages: TypeScript, JavaScript, C++, SQL

Frameworks & Libraries: React.js, Next.js, Node.js, Express.js, Tailwind CSS

Databases & Caching: PostgreSQL, MongoDB, Redis, Kafka, BullMQ

DevOps & Cloud: Docker, Kubernetes, AWS, CI/CD

Concepts: REST APIs, WebSockets, Microservices, Event-Driven Architecture, System Design

Experience

Full Stack Developer Sep. 2024 – Mar. 2025
Raizzify *Remote*

- Engineered web applications using **React.js, Next.js, and TypeScript**, improving page load performance by **40%**.
- Designed **RESTful APIs** with Node.js and Express.js, delivering **5+ production features** including authentication and notifications.
- Managed global state across **15+ components** using Context API and Redux, reducing complexity by **35%**.
- Optimized frontend via React hooks in an **Agile** environment, reducing re-renders by **50%** and resolving **20+ UI issues**.

Independent Backend Developer 2025
Self-Employed | via Fiverr *Remote*

- Built an **AI-driven image generation pipeline** using **n8n** with **95%+** prompt success reliability.
- Architected automated workflows processing **100+ user prompts** and triggering image generation via **REST** APIs.
- Integrated backend with frontend delivering generated images in **near real-time** (<3s response time).
- Optimized **CI/CD pipeline** and workflow execution using **Docker**, reducing processing latency by **30%**.

Projects

Exchange System | *Node.js, TypeScript, Redis, WebSocket, Docker, TimescaleDB* |

- Built a real-time crypto exchange handling **10K+ orders/day** with microservices for API, matching engine, and WebSockets.
- Leveraged Redis Pub/Sub for async communication enabling **sub-100ms** order propagation across services.
- Integrated TimescaleDB to persist high-frequency trade data supporting time-series analytics for price, volume, and depth.
- Containerized **6+ services** with Docker, reducing environment setup time by **60%** and ensuring consistent deployments.

Email Scheduling & Delivery SaaS | *Node.js, TypeScript, BullMQ, Redis, PostgreSQL, Prisma, Next.js* |

- Built a production-grade platform processing **5K+ emails/day** using Redis-backed BullMQ with precise delayed delivery.
- Implemented **per-sender rate limiting (200+ emails/hour)**, throttling, and auto-rescheduling to handle burst traffic.
- Designed **idempotent, restart-safe workers** with Redis locks guaranteeing exactly-once delivery across retries and crashes.
- Separated API and worker services for **horizontal scaling**; built a Gmail-like Next.js dashboard to monitor email statuses.

Event-Driven Notification System | *Node.js, TypeScript, Kafka, Redis, PostgreSQL, Docker* |

- Built an event-driven system processing **10K+ events/day** using Kafka-based microservices architecture.
- Implemented retries, dead-letter queues, and idempotent consumers, achieving **99%** reliable message delivery.
- Used Redis for caching, deduplication, and rate limiting, reducing duplicate notifications by **40%**.
- Persisted **100K+** records in PostgreSQL and containerized **5+ services** with Docker for consistent deployments.

Open Source Contributions

pg-wire-mock – Open Source PostgreSQL Wire-Protocol Mock Server
Hacktoberfest 2025 Contributor *Oct. 2025*

- Improved connection handling and protocol behavior in **pg-wire-mock**, an open-source PostgreSQL mock server.
- Implemented connection pooling and optimized request handling, enhancing stability under concurrent client connections.
- Collaborated via pull requests, code reviews, and issue discussions during Hacktoberfest following open-source best practices.