Laksh Jethani

lakshjethani.dev | +1 (352) 256-5058 | ljethani@usc.edu | linkedin.com/in/laksh-jethani | github.com/lakshjethani

Education

University of Southern California Aug 2024 – May 2026
Master of Science, Computer Science *Los Angeles, CA*
Relevant: Data Structures and Algorithms, Natural Language Processing, Web Technologies

University of Mumbai Aug 2018 – May 2022
Bachelor of Engineering, Computer Science *Mumbai, India*
Relevant: Parallel and Distributed Systems, Network Security, Operating Systems

Experience

Software Engineer Intern, *Supplyframe, a Siemens Company* — Los Angeles, CA May 2025 – Present

- Rebuilt dev.hackaday.io and Hackaday.io into a modular, **Storybook-driven** system using **React/Angular** and **AJAX**, extending backend workflows in **Node.js/Express**; improved UI consistency and **accelerated** design–QA cycles by **25%**.
- Reduced **p95 API latency by 20%** (480ms → 380ms) by implementing exponential-backoff for Swagger token requests and optimizing the Solr payload flow, lowering request volume and boosting responsiveness on high-traffic paths.
- Rolled out a hardened **OAuth2 sign-in flow** with **CSRF safeguards** and **Redis-based scope caching** for Hackaday's developer platform **(1M+ users)**, closing replay attack vectors and strengthening auth consistency.

Senior Software Development Engineer, *Deutsche Bank* — Pune, India Aug 2023 – Jul 2024

- Engineered a **Kafka-driven retry pipeline** with idempotency keys, a transactional outbox, DLQs, and jittered offset-managed retries, lifting payment success by **21%** on a **$300M+/day** platform and lowering reconciliation and on-call escalation rates.
- Migrated **Fabric** from on-prem to **GCP** through Infrastructure as Code and seamless blue-green deployments, refining autoscaling rules and multi-zone topology to lower run cost by **17%** and strengthen failover resilience across three zones.
- Integrated **Azure MFA** with a custom **OIDC** provider to federate external users, enforce Conditional Access with step-up authentication for sensitive scopes, and unify claims and token lifetimes, reducing login issues.
- Leveraged **Copilot** and **GPT-4** to automate documentation and tests, boosting developer productivity.

Software Development Engineer, *Deutsche Bank* — Pune, India Aug 2022 – Jul 2023

- Decomposed a monolith into **scalable Spring Boot microservices** and introduced **concurrency controls**, **circuit breakers**, idempotent retries, and full **observability** (metrics, logs, tracing), improving reliability on critical payment paths.
- Enhanced **React + Redux** architecture with optimized state handling, memoization, and lazy loading; integrated **GraphQL** and REST APIs with caching and pagination, cutting re-renders by **40%** and boosting responsiveness by **25%**.
- Led UAT cycles with 10+ users and analysts, turning feedback into engineering priorities and coordinating fixes across teams; mentored new grads during onboarding to help deliver **3 key features** in the next release.

Software Development Engineering Intern, *Deutsche Bank* — Pune, India Jun 2021 – Aug 2021

- Built a sharded cache on **Redis Cluster** (Spring Boot), applying consistent hashing, Raft-style leader election, and quorum reads/writes to scale to **millions of keys** and serve **10+ downstream services** with no data loss during node failures.
- Applied **Merkle trees**, gossip, and **Bloom filters** to stop cache stampedes and cut thousands of redundant calls daily.

Selected Projects

Distributed Observability Pipeline — Go, Kafka, OpenTelemetry, Elasticsearch (GitHub) Feb 2025 – Apr 2025

- Built a distributed observability pipeline in **Go**, using **Kafka** to ingest and index high-volume **OpenTelemetry** traces and logs (millions of events/min) into **Elasticsearch** for real-time analysis.
- Added an **AI-powered DevOps assistant** using the **OpenAI API** and **Slack SDK** to surface logs, answer queries, and suggest fixes, reducing debugging and on-call effort by **30%**.

MCP SafeSQL — TypeScript/Node.js, SQLite/Postgres (GitHub) Dec 2024 – Jan 2025

- **Model Context Protocol (MCP) server** exposing a read-only `sql.query` with allowlisted views, parameterized queries, and PII masking across **40+ columns**; added **RBAC**, audit logging, pooling, and OpenTelemetry for SQLite/Postgres.
- **Safe EXPLAIN**: Provided an explain plan (operator tree with cost and row estimates) without executing the query, redacted identifiers and literals, fingerprinted statements for caching, and flagged full scans or missing indexes on **50+** test queries.

Technical Skills

Languages: **Java**, **C++**, Go, Python, Android (Kotlin), **TypeScript**, JavaScript, SQL, HTML, CSS
Backend: **Spring Boot**, **Node.js**, Express, REST, **Apollo GraphQL**, Django, JPA, Hibernate, OpenAPI
Frontend: **React**, Redux, EJS, SCSS, Storybook, **AJAX**, Bootstrap, WCAG 2.1
Data & Messaging: **PostgreSQL**, MySQL, MongoDB, **Redis**, Apache Solr, **Kafka**, Celery
Cloud & DevOps: **GCP**, **AWS** (EC2, S3, Elastic Beanstalk), **Docker**, **Kubernetes**, GitHub Actions, Jenkins
Observability & Quality: **OpenTelemetry**, distributed tracing, JUnit, Mockito, Jest, Playwright, ESLint, Prettier, Nginx
Auth & Security: **OAuth2**, **OIDC**, CSRF, Azure MFA