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 (M.S.), Computer Science — *Los Angeles, CA*
Relevant: Analysis of Algorithms, Geospatial Information Management, Web Technologies

University of Mumbai — Aug 2018 – May 2022
Bachelor of Engineering, Computer Science — *Mumbai, India*
Relevant: Distributed Computing, Object-Oriented Programming, Operating Systems

Experience

Software Engineer Intern, *Supplyframe, a Siemens Company* — Los Angeles, CA — May 2025 – Present
- Engineered and deployed a production-grade OAuth2 sign-in flow within an 8-person team by resolving state mismatches, enforcing CSRF controls, adding **Redis-backed scope caching**, and hardening the auth flow for replay/state-based attacks.
- Optimized API performance by implementing exponential backoff on Swagger token retries and streamlining Solr-bound request payloads, lowering Solr traffic and **cutting p95 latency by 20% (480 ms → 380 ms)** at scale.
- Redesigned dev.hackaday.io into a modular EJS design system with Storybook; collaborated with design and QA to improve Core Web Vitals and cut average PR review iteration cycles by 25%.

Senior Software Development Engineer, *Deutsche Bank* — Pune, India — Aug 2023 – Jul 2024
- Improved payment success by **21%** on a **$300M+/day** platform by adding a **Kafka** reprocessing pipeline with idempotency keys, transactional outbox, DLQs, and offset managed retries with jitter, reducing reconciliations and on call alerts.
- Migrated Fabric from on-prem to **GCP** using Infrastructure as Code with zero-downtime blue-green deployments, tuned autoscaling policies and multi-zone placement, and **reduced run cost by 17%** while improving failover stability across 3 zones.
- Integrated **Azure MFA** with a custom **OIDC** provider to federate external users, apply Conditional Access with step up MFA for sensitive scopes, standardize claims and token lifetimes, and reduce login escalations.
- Mentored two new grads through code reviews and debugging, speeding up onboarding and improving PR quality.

Software Development Engineer, *Deutsche Bank* — Pune, India — Aug 2022 – Jul 2023
- Maintained SLAs on critical paths using **circuit breakers**, idempotent retries, and full **observability** (metrics, logs, tracing).
- Decomposed a monolith into **Spring Boot microservices** using JPA and Hibernate; introduced thread-safe concurrency controls to eliminate race conditions and standardized Maven builds, **cutting build times by 35%**.
- Enhanced **React + Redux** architecture with optimized state handling, memoization, and lazy loading; implemented GraphQL/REST integration with caching and pagination, reducing re-renders by **40%** and improving responsiveness by **25%**.
- Led UAT cycles with 10+ users and analysts, prioritizing feedback that enabled delivery of **3** key features in the next release.

Software Development Engineering Intern, *Deutsche Bank* — Pune, India — Jun 2021 – Aug 2021
- Built a sharded cache with **Redis Cluster** + **Spring Boot** using consistent hashing, Raft-style leadership, and quorum reads/writes, scaling to **millions of keys** and serving **10+** downstream services without data loss under node failures.
- Used **Merkle trees**, gossip, and Bloom filters to eliminate cache stampedes and remove thousands of redundant calls per day.

Selected Projects

Distributed Observability Pipeline — Go, Kafka, OpenTelemetry, Elasticsearch (GitHub) — Feb 2025 – April 2025
- Engineered a distributed observability pipeline using **Go** microservices and **Kafka** to ingest and index high-volume **OpenTelemetry** traces and logs (millions of events/minute) into **Elasticsearch** for real-time analysis.
- Deployed on **Kubernetes** with fine-grained **RBAC** for scalable and secure operations; built in fault tolerance and automated alerting, resulting in **99.9%** pipeline uptime.

MCP SafeSQL — TypeScript/Node.js, SQLite/Postgres (GitHub) — Dec 2024 – Jan 2025
- **Model Context Protocol server** exposing a read only `sql.query` with allowlisted views, parameterized queries, and PII masking across **40 + columns**; adds RBAC, audit logs, pooling, and OpenTelemetry for SQLite/Postgres
- ***Safe EXPLAIN*** returns the operator tree with cost and row estimates without executing, redacts identifiers and literals, fingerprints statements for caching, and flags full scans or missing indexes on **50+** test queries

Technical Skills

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