K. S. Sushanth

+91-63791 19460 | k.s.sushanth06@gmail.com | github.com/SushanthKS06 | linkedin.com/in/sushanthks

Education

Bachelor of Technology in Information Technology Sep. 2023 – May 2027 (Expected)
Sri Krishna College of Engineering and Technology *Coimbatore, India*
Relevant Coursework: Distributed Systems, Operating Systems, Data Structures & Algorithms, DBMS, Computer Networks, Software Engineering.

Technical Skills

Primary Languages: Java, Python, C++, Golang
Core CS: Data Structures & Algorithms, Multithreading, Concurrency, Object-Oriented Design, Complexity Analysis
Backend & APIs: Spring, SpringBoot, Django, FastAPI, gRPC, REST APIs, GraphQL, Thread Pools, Caching
Datastores: PostgreSQL, Redis, MongoDB, MySQL
Data & Analytics: SQL, Data Modeling, Metrics Design, Exploratory Data Analysis
Infrastructure & Tooling: AWS, GCP, Linux, Git, Docker, Kubernetes, CI/CD, Github Actions
Testing & Reliability: Unit & Integration Testing, Load Testing, Logging, Profiling

Experience

Data Analyst Intern Jun 2025 – Aug 2025
Elevate Labs *Bangalore, India*

- Worked with **3 product managers and 2 engineering teams** to define success metrics for A/B experiments and quarterly growth initiatives.
- Analyzed **5M+ rows** of user event data using SQL and Python to evaluate engagement across core user flows and feature variants.
- Findings from controlled experiments led to deprioritizing two planned features and reallocating effort toward higher-impact changes in the next release cycle.
- Contributed analysis that resulted in a $\sim 12\%$ **improvement in a core engagement KPI** measured over a 4-week window following rollout to active users.
- Built reusable analysis pipelines and dashboards used in weekly product reviews, reducing ad-hoc reporting turnaround time by **40%**.
- Partnered with backend engineers to validate data pipelines, resolve data-quality issues, and align analytics outputs with system behavior.

Projects

Concurrent Event Processing Engine | *Java, Redis, Multithreading*

- Defined throughput and latency targets with contributors for an event ingestion pipeline designed for bursty producer workloads.
- Identified throughput regressions caused by lock contention and queue saturation in an initial fixed thread-pool design.
- Redesigned scheduling and queueing strategy to eliminate contention hotspots, sustaining **10K+ events/sec** for 30+ minutes on a 4-core system with **100 concurrent producers** and zero data loss.
- Reduced p95 end-to-end latency by **38%** through profiling-driven thread pool tuning and queue threshold calibration.
- Added structured logging and runtime metrics for queue depth and worker utilization; shared load-testing harnesses and design notes, reducing future tuning effort by $\sim 30\%$.

Polyglot Microservices System | *Java, Python, gRPC, Kubernetes*

- Collected API usability and backward-compatibility requirements from consumers across **5 internal services**.
- Implemented schema-evolving gRPC services sustaining **hundreds of requests per second** during pre-production load tests over sustained multi-hour runs.
- Achieved approximately **7× faster serialization** compared to REST under equivalent traffic and payload sizes.
- Introduced versioning guidelines and review checkpoints to prevent breaking changes during schema evolution, reducing contributor ramp-up time by $\sim 20\%$.
- Deployed services to Kubernetes with automated integration tests, health checks, and scaling policies, validating operational readiness prior to production rollout.

Mini CI/CD Testing Pipeline | *Python, Git Hooks*

- Built CI workflows enforcing tests, coverage thresholds, and static analysis across **4 repositories** used by multiple contributors.
- Prevented **30+ regressions** over a 3-month period by catching failures pre-merge and providing fast developer feedback.
- Reduced test flakiness by **25%** through dependency isolation and deterministic mocking, shortening CI turnaround time.
- Standardized pre-commit workflows, increasing first-pass approvals and reducing review cycles.

Open Source & Achievements

Open Source Contributions: Performance and stability improvements merged into FastAPI, LangChain, and TaiPy following maintainer review.
Problem Solving: Solved **500+** data structure and algorithm problems on LeetCode and Codeforces, with emphasis on graph algorithms and performance constraints.
Developer Tooling: Built custom load-testing utilities and a job-queue visualizer to debug and analyze distributed workflow behavior.
Certifications: AWS Certified Developer – Associate; Google Cloud Fundamentals; DeepLearning.AI Multi-Agent Systems.