Anandapadmanabhan Santhosh

Email : ananduap19122001@gmail.com
LinkedIn : https://www.linkedin.com/in/anandapadmanabhan-santhosh
Mobile : +1(774)999-9455
GitHub : https://github.com/anandu-0116

Education

Stony Brook University — Stony Brook, NY, USA
Master of Science, Computer Science — *Aug 2024 – May 2026*
Coursework: Distributed Systems, Operating Systems, System Security, Storage Systems

Birla Institute of Technology and Science, Pilani — Hyderabad, India
Bachelor of Engineering, Computer Science — *Aug 2019 – Jul 2023*
Coursework: Data Structures & Algorithms, Database Systems, Object Oriented Programming, Computer Networks

Technical Skills

Languages: Golang, Java, Python, C++, C, JavaScript, HTML, CSS

Database Technologies: SQL, MongoDB, DynamoDB, Cloud Firestore

Frameworks: Spring Boot, Flask, Node.js, React, Angular

Cloud Technologies, DevOps & Version Control: AWS S3, Git, Docker, Jenkins, Grafana

Projects

Distributed Banking System with Paxos Consensus — GitHub
Go, gRPC — *Nov 2025 – Dec 2025*

- Architected a distributed **banking system** in **Golang** using **gRPC** and **Protobuf**, partitioning data across **9 servers** into **3 shards** to support high-throughput concurrent transaction processing.
- Engineered a **Multi-Paxos consensus protocol** with stable-leader optimization to guarantee linearizability and fault tolerance for **intra-shard transactions**, ensuring data consistency across replicas.
- Implemented **Two-Phase Commit (2PC)** combined with **Two-Phase Locking (2PL)** to orchestrate **cross-shard transactions**, coordinating between clusters to ensure data consistency and atomicity across shards.
- Designed a crash-recovery mechanism utilizing **Write-Ahead Logging (WAL)**, enabling the system to roll back aborted transactions and recover state after node failures.

Byzantine Fault Tolerant Banking System — GitHub
Go, gRPC — *Oct 2025 – Nov 2025*

- Implemented the **Linear PBFT consensus protocol** in Go for a **7-node** replicated banking application, guaranteeing state ordering and safety with tolerance for **Byzantine faults**.
- Developed the full PBFT **view change protocol** to ensure liveness, allowing the cluster to automatically elect a new primary and resume operation upon **leader failure**.
- Implemented transaction processing for **10 concurrent clients**, maintaining a distributed **key-value datastore** with real-time balance updates across servers.
- Built a robust **checkpointing** and **state transfer** mechanism that allows lagging or recovering nodes to safely catch up to the cluster's current state using signed snapshots.
- Validated system resilience against Byzantine attacks (equivocation, dark, signature) and measured performance using a custom SmallBank benchmark with high-contention (Zipfian) workloads.

Safex: Read Access Control LSM — GitHub
C, LSM — *Jun 2025 – July 2025*

- Developed a custom **Linux Security Module** from scratch to enforce read access control using LSM hooks, with denylist loading via deferred workqueues and kernel-level file path resolution.

Professional Experience

Nykaa — Bangalore, India
Software Development Engineer — *Jul 2023 – Aug 2024*

- Engineered the decoupling of the Catalog Feed from a legacy monolith into a standalone Microservice, reducing data synchronization latency by **90%** (8 hours to 45 mins) and ensuring real-time product availability.
- Developed the backend infrastructure for the **Virtual Try-On (VTO)** feature using Python (Flask) by integrating Modiface AR APIs, enhancing user engagement and interaction capabilities.

Licious — Bangalore, India
Software Engineer Intern — *Jan 2023 – Jun 2023*

- Spearheaded the decomposition of a monolithic architecture by extracting the Order Management module into an isolated **Spring Boot** microservice, improving fault tolerance and maintainability.
- Refactored legacy API pathways into optimized service-oriented architectures, reducing redundant calls and enhancing system scalability.