Pranchal Shah

pranchx@gmail.com | 681-242-7542 | Linkedin.com/pranchals | github.com/p-shah256 | shah256.dev | Boston, MA

Education

MS in Computer Science | *Northeastern University* | *Boston, MA* **Jan 2023 – May 2025**

Relevant Courses: Network Security, Kernel implementation, Computer Systems, OOP, Data Structures, Low level Software Security, Algorithms, Database Management

B.Arch in Architectural Engineering | *Sardar Patel University* | *India* **Aug 2014 – Dec 2019**

Technical Skills

Languages	: Go, C/C++, Python, TypeScript, JavaScript, Java
Web Technologies	: Node.js, Django, Express, SpringBoot, React, RestAPI, GraphQL, gRPC, HTMX
Database & Storage	: PostgreSQL, MongoDB, Redis, Elasticsearch, Kafka, RabbitMQ
DevOps & Cloud	: AWS, GCP, Azure, Kubernetes, Docker, Terraform, Pulumi, Jenkins
Tools & Observability	: Git, Jenkins, Prometheus, Grafana, DataDog, Github Actions

Professional Experience

FoxyAI | **May 2024 – Present**

Software Engineer Intern *Boston, MA*

- Architected a distributed pipeline with Golang and Docker, running 12M+ daily images, reduced processing time by 73%
- Improved ML model inference speed by 12% using NVIDIA MPS on Kubernetes GPU scheduling on A100 clusters
- Implemented cross-cloud caching layer(AWS/GCP) reducing processing costs by 35%, processing 90M+ images weekly
- Introduced serverless concurrent E2E testing with real-time alerts on Slack using Jest+TS, decreasing test time by 65%
- Developed AWS SQS and PostgreSQL-based audit trail system handling 3000 QPS with 0.8ms average latency

Sangath LLP | **Dec 2020 – Jul 2022**

Computational Engineer *Ahmedabad, India*

- Engineered extensible C# plugin framework reducing manufacturing simulation time by 33% across modelling workloads
- Led 5-person team building manufacturing analytics pipeline for $22M project, reducing material waste by 27%
- Automated Python workflows across 20+ projects, transforming 40-hour processes to 15-min computations of estimates

Excellence Org | **Dec 2019 – Dec 2020**

Software Engineer (Backend) *Mumbai, India*

- Built internal developer platform with CI/CD using Github actions reducing deployment time from 2 hours to 15 minutes
- Optimized high-traffic PostgreSQL cluster achieving 85% lower latency through partitioning and query optimization
- Spearheaded Redis caching implementation with cache invalidation patterns, achieving 40% reduction in database load

Projects

Operating Systems Components | *C, Assembly, Pthreads, Systems* **Sept 2024 - Present**

- Created 16-bit CPU emulator supporting 20+ instructions with 64KB virtual memory and 8 registers management
- Engineered user-level threading library with custom scheduler, and 90ns context switching wite 1000+ parallel threads
- Programmed a Unix shell supporting 10+ pipes and process management with I/O redirection and 15+ bash features

Home lab Infrastructure | *Docker, Prometheus, Grafana, Nginx* **Jan 2024 - Present**

- Initiated and deployed social forum serving 900+ weekly active users with 99.9% uptime on Raspberry Pi hardware
- Implemented Prometheus and Grafana observability stack monitoring 1000+ daily requests with 100ms p95 latency
- Deployed production-grade infrastructure with nginx, fail2ban, personal blog and git server handling 10k+ weekly visits

Secure Messaging Protocol | *Python, OpenSSL, Cryptography, TCP/IP* **Mar 2024 - Apr 2024**

- Designed custom end-to-end encrypted messaging protocol with perfect forward secrecy handling 100+ msg per second
- Developed PKI infrastructure creating and exchanging X.509 certificates with RSA key exchange and digital signatures
- Created secure session management with TLS-like handshake with achieving less than 50ms connection establishment

draw Sync | *Node.js, WebSocket, Redis, React, Docker, Websockets* **Sept 2023 - Dec 2023**

- Designed WebSocket and CanvasJS based collaborative drawing app supporting 50+ concurrent users with 25ms latency
- Enhanced Redis pub/sub architecture for real-time state sync and messages handling 43 updates/second across clients
- Introduced connection pool to improve reliability with automatic failover handling 1000+ WebSocket reconnections