

Kamran M. Ahmed

+1 (239) 810-3586 · kamranahmed@stanford.edu
[linkedin.com/in/kamran-m-ahmed](https://www.linkedin.com/in/kamran-m-ahmed) · [kamranmahmed.com](https://www.kamranmahmed.com)

PROFESSIONAL SUMMARY

Recently graduated Master's student in Computer Science from Stanford University with a passion for computer networking and systems programming. Experienced in building network protocols, distributed systems, program analysis tools, and cloud-based applications.

EDUCATION

Stanford University, School of Engineering **September 2022 – June 2024**
Master of Science, Computer Science, Systems GPA 4.1/4.0

Coursework: Computer Networking, Operating Systems, Distributed Systems, Parallel Computing, Programming Languages, Program Analysis and Optimization, Cryptography

Harvard University, Graduate School of Arts and Sciences **August 2020 – May 2022**
Graduate Student, Neuroscience GPA 4.0/4.0

Awards: National Science Foundation Graduate Research Fellowship

Coursework: Advanced Practical Data Science, Machine Learning, Deep Learning for Biomedical Image Analysis

University of California, Berkeley, College of Letters and Science **August 2015 – May 2019**
Bachelor of Arts, Molecular and Cell Biology, Neurobiology with Honors GPA: 4.0/4.0

Coursework: Structure and Interpretation of Computer Programs

SKILLS

Programming Languages: Python, C, C++, Rust, Go, JavaScript, TypeScript, Java, SQL

Technologies and Frameworks: FastAPI, Flask, Django (DRF), PostgreSQL, SQLAlchemy, Redis, Apache Kafka, React, Node.js, Next.js, Express, MongoDB, CUDA, TensorFlow, PyTorch

Cloud and Tooling: AWS (EC2, Fargate, Lambda, SQS, S3, DynamoDB), GCP (Compute Engine, Cloud Storage, Cloud Run, Cloud Functions, BigQuery, Cloud Build, Vertex AI), Docker, Git, GitHub Actions, Jenkins

PROFESSIONAL EXPERIENCE

Capital One **McLean, VA**
Software Engineer Intern June 2023 – August 2023

- Optimized Capital One's step-up authentication decision engine and fraud risk monitoring system, substantially reducing customer authentications during call sessions. This enhancement in customer satisfaction yielded annual savings of over \$3M for real-time fraud decision-making platforms.
- Built a high-throughput, streaming-based serverless application that efficiently handled Kafka events, sent them to an SQS queue, and processed them asynchronously with Lambda functions.
- Demonstrated expertise in a wide range of technologies, including AWS (Lambda, Fargate, SQS, IAM, CloudWatch, and DynamoDB), Apache Kafka, Python, pytest, Behave, Splunk, and Jenkins.

HiHome (Startup) **Boston, MA**
Software Engineer Intern May 2022 – August 2022

- Spearheaded technical development of a real estate workflow management application, architecting backend solutions, including an API service, an event and task manager, and third-party integrations with Google APIs and Follow Up Boss.
- Enhanced the home-matching platform by optimizing multiple services, resulting in improved frontend usability and enhanced backend API performance, reducing geolocation query times by 85%. Collaborated with a remote development team on HiHome's FastAPI service and Elasticsearch scoring engine.
- Applied expertise in FastAPI, SQLAlchemy, PostgreSQL, Redis, and Docker.

TEACHING EXPERIENCE

CS 107, CS 142, and CS 144, Stanford University

Stanford, CA

Course Assistant

April 2023 – June 2024

- Served as a course assistant for foundational courses in Stanford’s CS curriculum: CS 107 (Computer Organization and Systems) with Jerry Cain, CS 142 (Web Applications) with Professor Mendel Rosenblum, and CS 144 (Computer Networking) with Professor Keith Winstein.
- Guided students to enhance problem-solving skills and acquire in-depth knowledge during office hours. Led effective discussion sections and aided students in understanding complex design decisions and tradeoffs. Supported instructional staff with grading projects and assessments for ~200 students.

CS 109A, Introduction to Data Science, Harvard University

Cambridge, MA

Teaching Fellow

September 2021 – December 2021

- Served as a teaching fellow for a data science course for ~300 computer science students.
- Synthesized course material, collaborated with students, and built foundational concepts during weekly office hours. Supported instructors and teaching staff with grading and debugging.

PROJECTS

Advanced Systems Projects – C, C++, Rust, Docker

- Pintos Operating System. Developed thread management, virtual memory, and file system functionality in C. Implemented system calls to enable interaction between user programs and the kernel.
- Transmission Control Protocol. Recreated a full-fledged TCP networking stack in user space with C++. Created a traceroute-like network analysis tool to examine how packets are routed through the Internet.
- Peer-to-Peer VPN. Created a VPN similar to Wireguard with Rust and C. Encrypted IPv4 traffic with AES-GCM and implemented a sliding window protocol to mitigate replay attacks.
- DNS Server. Built a DNS recursive resolver in Rust. Created an LRU cache to optimize query resolution and a custom parser and serializer to minimize data copying overhead.

Sidekick – C++, CMake, Mininet

- Built a sidekick protocol that assists secure end-to-end transport protocols over asymmetric network paths as part of a replication study of the Sidekick NSDI paper ([writeup](#), [code](#)).
- Implemented a selective acknowledgment mechanism to efficiently acknowledge packets without access to cleartext sequence numbers using polynomial power sums and modular arithmetic.
- Replicated the original study’s results using Mininet and real-world testing, achieving a 51% reduction in de-jitter latency of a low-latency audio stream over a lossy network in emulation (vs. 52% in the study) and a 79% reduction in real-world scenarios (vs. 91% in the study), demonstrating the proxy’s effectiveness across both environments.

Catamaran – Go

- Engineered a fault-tolerant and distributed DNS nameserver that replicates DNS resource records using a custom Raft implementation ([writeup](#), [code](#)).
- Integrated Dynamic DNS (DDNS) updates to allow services with continuously changing IP addresses to update resource records without manual intervention.
- Conducted extensive evaluation of query and update latency, fault tolerance, and replication costs compared to BIND 9. Found that while Catamaran had a low cost of replication for reads, it incurred a significantly higher cost for writes due to the sequential processing of DNS updates forwarded to the Raft leader, resulting in greater latency compared to BIND 9.

Tainted – Python, pytest

- Developed a program analysis tool for Python to dynamically identify and track the flow of sensitive data in consumer-facing applications ([writeup](#), [code](#)).
- Implemented dynamic taint tracking through program instrumentation and a custom runtime library, minimizing interference with normal program execution.
- Conducted comprehensive unit and performance testing, including microbenchmarks for SQL injection and cryptographic key leakage scenarios to demonstrate the tool’s effectiveness.