Krish Patel

krishspatel.com • github.com/krisapa • linkedin.com/in/krisapa • kspatel8@icloud.com • (980) 226-7642

Education

University of North Carolina at Chapel Hill May 2026

B.S. Computer Science, B.S. Statistics and Analytics *Current GPA: 3.89*

Courses: Data Structures & Algorithms, Operating Systems, Computer Organization, Database Systems, Cryptography, Computer Networks, Distributed Systems, Algorithms & Analysis, Digital Logic & Computer Design

Skills

Languages: Python, C++, C, CUDA, Go, SystemVerilog, SQL, JavaScript/TypeScript, Swift, Java
Systems & Networking: Linux, TCP/IP, Wireshark, scapy, gRPC, perf/ftrace
DevOps & Cloud: AWS, Docker, PostgreSQL, Redis
ML & Data: PyTorch, Core ML, NumPy, Pandas

Experience

Fidelity Investments | *Software Engineering Intern* Jun. 2025 – Aug. 2025

- Working on Code Intelligence & AI Platform, building infrastructure to transform multi-language codebases into knowledge graphs for LLMs

UNC CS Experience Labs | *Backend Engineer (Campus Dev Team)* Aug. 2024 – Dec. 2024

- Led architecture overhaul by containerizing backend services, integrating OAuth SSO, and automating deployments to improve security and reliability for 100+ daily users
- Standardized OpenAPI contracts, added monitoring with Prometheus and OpenTelemetry, and worked with cross-functional teams to reduce page latency by 30% and accelerate release cadence

Technical Projects

CAN-Cuda Logger | *C++, CUDA, SocketCAN, LZ4*

- Built a real-time Linux SocketCAN bus logger that batches frames and compresses them on an NVIDIA GPU with LZ4, sustaining 250 MB/s test throughput with zero dropped frames and a 65% reduction in log size

Chrome Dino on FPGA | *SystemVerilog, MIPS Assembly, Xilinx Vivado*

- Built a 32-bit MIPS computer on a Nexys A7 FPGA, integrating CPU, accelerometer, keyboard, VGA (640×480 @ 60 Hz, 25 MHz pixel clock) and PWM audio
- Implemented Chrome Dino in MIPS assembly and optimized SystemVerilog in Vivado to achieve 100 MHz timing closure for smooth graphics and sound

PeerBeam | *Go, SvelteKit, TypeScript*

- Built a peer-to-peer file-transfer app (web UI & CLI), leveraging WebRTC data-channels to bypass relay servers and achieve 90 MB/s LAN throughput for fast, private file sharing
- Used Wireshark + tcpdump to trace UDP flows over 802.11ac; tuned ICE keep-alive timing to drop retransmissions 20% on congested Wi-Fi

BGP-Lite Router | *C, Python, Netlink*

- Implemented a lightweight BGP daemon in C that manages eBGP sessions and programs routes into the Linux kernel via Netlink; achieves full convergence of 3000 prefixes in about 1.2 seconds on a 2-vCPU VM
- Built a Docker pytest harness that spawns Quagga neighbors; perf/ftrace tuning reduced route install latency 18%

Lock-Free Graph Store | *C++*

- Implemented lock-free adjacency lists with atomic CAS enabling concurrent node/edge insertions without blocking
- Benchmarked at 10 M ops/s on 32 vCPUs (AWS c5.4xlarge); memory <200 MB for a 1 M-node graph; validated lock-free correctness with ThreadSanitizer

Wifi Beacon Heatmap | *Python, Wi-Fi, matplotlib*

- Built a Wi-Fi spectrum analyzer that continuously scans beacons, parses SSID–channel–RSSI, and renders a live heatmap for channel congestion diagnosis

Neural Network From Scratch | *C++*

- Implemented a high-performance feedforward neural network in C++ (no external libs), coding matrix operations, forward/backward propagation, and stochastic gradient descent from scratch