ISHAN BHAT
7656949978 | ibhat.141004@gmail.com | LinkedIn | GitHub

EDUCATION

Purdue University May 2027
Master of Science, Computer Science (GPA: 3.6/4.0)

Purdue University Dec 2025
Bachelor of Science, Computer Science, Mathematics, Statistics (GPA: 3.8/4.0)

SKILLS

• Languages: Python, Java, C/C++, JavaScript, TypeScript, HTML, CSS, Go, SQL
• Systems & Infrastructure: AWS, Azure, Docker, Kubernetes, GCP, IAM, CI/CD, Bash, Linux, Distributed Systems, Networking, NoSQL
• AI & Agentic Workflows: LLM Inference Serving, LangChain/LangGraph, ADK, Context Optimization, Agentic Systems Design
• Development Practices: Data Structures & Algorithms, Object-Oriented Design, SDLC, Version Control, Debugging, AI-Assisted Development Tooling

WORK EXPERIENCE

Amazon, Prime Air Apr 2026 - Present
Systems Engineering Intern Seattle, WA
• Engineered a self-service drone-operations platform unifying 15+ ground-system and onboard telemetry sources, collaborating with cross-functional teams and following SDLC practices to give engineers real-time visibility into component health, navigation state, and fleet-wide metrics.
• Designed scalable data schemas and mathematical models for core operational metrics, including cost-per-package, standardizing measurement across teams and eliminating hours of ad hoc manual analysis per release cycle.
• Built an agentic natural-language query interface over telemetry and operational data, collapsing multi-hour debugging investigations into seconds and accelerating root-cause analysis across the engineering org.

Purdue University Jan 2026 - Present
Agentic Systems Researcher – Prof. Vamsi Addanki West Lafayette, IN
• Designed and implemented an Earliest-Deadline-Late (EDL) scheduling policy for LLM inference, improving goodput and deadline adherence across multi-step, latency-sensitive agent workflows.
• Architected a KV-cache-aware request router using custom data structures to track cache locality across nodes, lifting simulated cache-hit rate by 8% and cutting redundant prefill computation.

Capital One Jun 2025 - Aug 2025
Software Engineering Intern Dallas, TX
• Built an automation framework that parses Infrastructure-as-Code repositories into abstract syntax trees via the GitHub GraphQL API, surfacing service ownership and cross-team cloud dependencies at scale.
• Designed a recursive parser extracting component configurations and 10+ distinct AWS service relationships, normalizing infrastructure metadata into a containerized NoSQL DynamoDB pipeline.
• Shipped a Vue.js dashboard built with JavaScript, HTML, and CSS, plus a companion CLI, for resource monitoring and engineer onboarding, adopted by 50+ engineers and cutting incident-investigation time by 2+ hours per case.
• Incorporated feedback from code reviews and design discussions to raise code quality and align on architecture across the engineering team.

Cylus May 2024 - Aug 2024
Software Engineering Intern Remote
• Built a real-time packet-capture (pcap) streaming tool in Elixir, Phoenix, and LiveView, cutting R&D analysis turnaround time by 50%.
• Reduced annual AWS spend by $3K through targeted Postgres index tuning, Lambda refactoring, and S3 storage optimization.

• Engineered a custom multithreaded Driver-Machine Interface (DMI) server on JMRI, demonstrated live at InnoTrans 2024 to prospective enterprise clients.

PROJECTS

Custom Compiler
• Built an SSA-based intermediate representation in C++ with LLVM, implementing liveness analysis, dead-code elimination, constant folding, and speculative loop-invariant code motion to optimize compilation performance.
• Designed a register allocator using interference graphs to support variable spilling and reachability tracking, improving code-generation efficiency.

Distributed Key-Value Store - Raft Consensus
• Implemented Raft consensus protocol in Python using object-oriented design and advanced data structures, supporting leader election, log replication, and heartbeat-based failure detection across a multi-node cluster.
• Designed and implemented log compaction and snapshotting mechanisms to bound memory growth, and exposed a client-facing API for linearizable read and write operations.
• Conducted chaos testing by simulating network partitions and node failures to validate consistency and fault tolerance in an AWS-deployed, multi-region environment.
• Deployed the service on AWS using Docker and Kubernetes, and established CI/CD pipelines with GitLab CI for automated testing and rolling deployments.