

Ambareesh Shyam Sundar

ambareesh@gatech.edu • (408) 744-6798

 ambareesh.sh

 ambareesh1510

 ambareesh-shyam-sundar

Education

Georgia Institute of Technology • GPA: 3.96 / 4.0

Atlanta, GA

B.S. in Computer Science; B.S. in Mathematics

August 2023 - December 2026

Selected coursework: Operating Systems, Processor Design, Computer Architecture, Embedded Systems, Advanced Algorithms, Natural Language Processing, Computer Networking, Probability and Statistics

Skills and Awards

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

Technologies: Linux, Git, Docker, QEMU, GDB, AWS (EC2, RDS, S3), MySQL, PostgreSQL, MongoDB, Nix

Awards: Putnam score of 23; 5x Georgia Tech Faculty Honors; College of Sciences Multidisciplinary Award Nominee

Experience

1Password

May 2026 – August 2026

Software Engineering Intern – Data Security

Rust, Go, TypeScript, Nix

- Implemented cross-platform hardware security key support in the open-source Rust library `passkey-rs`, enabling vault unlock using CTAP2 authenticators like YubiKeys, Touch ID, and Windows Hello for over 150k 1Password users.
- Designed a pluggable cryptographic-backend architecture for `passkey-rs`, enabling FIPS-validated signature algorithms and contributing to 1Password becoming the first password manager to support post-quantum passkeys.
- Built mock-based unit tests and expanded cross-platform CI with dependency checks and network monitoring, preventing regressions and hardening the build pipeline against supply-chain attacks.

Georgia Tech Open Source Program Office

May 2024 – August 2024

Software Engineering Intern

PyTorch, Python, C++

- Diagnosed and resolved defects in PyTorch's TorchDynamo JIT compilation pipeline, improving correctness and stability.
- Contributed code fixes and technical documentation to upstream PyTorch repositories through an IBM-sponsored open-source mentorship.

Georgia Tech Mathematical Neuroscience Group

January 2024 – May 2026

Research Assistant, Active Sensing in Masked Autoencoders

PyTorch, computer vision

with Dr. Hannah Choi and Abdo Sharaf

- Designed and built a two-stage vision pipeline on top of Masked Autoencoders (MAE) that performs uncertainty-aware patch selection prior to encoding, reducing redundant computation and improving downstream performance.
- Validated the system on ImageNet by replacing random patch masking with intelligent selection, demonstrating up to a 4.5% improvement in classification accuracy with minimal additional inference overhead.

Georgia Tech School of Computing Instruction

August 2026 – December 2026

Teaching Assistant – Systems for Machine Learning

C++, Python

Projects

Foxtail • ambareesh1510/foxtail

November 2025 – present

Unix-like operating system kernel for 32-bit x86.

C, bare-metal programming, QEMU, GDB

- Designed and implemented a fully functional Unix-like x86 operating system kernel and user-space from scratch, supporting preemptive multitasking, real-time graphics, and execution on real hardware.
- Developed a flexible process management interface, a custom filesystem with support for files, directories, and symbolic links, and inter-process communication via pipes.
- Built a complete memory management system with paging, virtual memory, demand allocation, and strong isolation/safety guarantees enforced by hardware protection rings.

Kalshi Crypto Prediction-Market Trading Bot

June 2026 - August 2026

Autonomous paper trading on Kalshi's cryptocurrency prediction markets.

C++, Python, XGBoost, REST/WebSockets

- Built a C++20 research and paper-trading system for Kalshi's 15-minute BTC/ETH markets, integrating authenticated production WebSocket data, historical replay, and simulated execution, and demonstrating positive profit after fees.
- Developed a four-class gradient-boosted model to estimate joint BTC/ETH settlement probabilities and fair values using Kalshi prices, CF Benchmarks spot data, momentum, volatility, and cross-asset correlation features.
- Evaluated 24 model and feature configurations through backtests across weeks of collected market data, using statistical methods to select the best-performing configuration based on settlement-outcome accuracy.