

Ankit Mukhopadhyay

San Francisco, CA • amukhopadhyay@dons.usfca.edu • [GitHub](#) • [LinkedIn](#)

EDUCATION

University of San Francisco, San Francisco, CA

MS in Computer Science

Expected May 2028

GPA: 4.0; graduate coursework in NLP, Systems Foundations, and Software Development

BS in Computer Science, Concentration in AI, Minor in Film Studies

Expected May 2027

GPA: 3.98 (4.0 in all CS courses); Dean's Honor Roll (6 semesters)

EXPERIENCE

ClearOne Advantage

February 2026 to May 2026

Associate Software Engineer

Built and shipped an AI voice agent that lets employees practice live phone calls against a simulated AI client. Owned the STT/TTS pipeline and Twilio telephony integration for real-time voice, and iterated on quality through review cycles with the people who used it.

University of San Francisco, San Francisco, CA

August 2024 to Present

Computer Science Teaching Assistant

TA across the CS curriculum as an undergraduate: graduate Natural Language Processing (CS 486/686), Operating Systems (CS 326), Computer Architecture with RISC-V (CS 315), and introductory Java (CS 112, CS 111). Also a walk-in CS tutor.

PROJECTS

OSlings: RISC-V Kernel Teaching CLI

A Rust CLI that teaches operating systems by guiding learners through building a pedagogical RISC-V kernel from scratch, with a full-screen terminal UI, live test-watching that reruns verification on every save, progressive hints, and QEMU-based verification. Used in USF's Operating Systems course (CS 326), where students build a kernel incrementally through exercises released one at a time.

Sparse Auto-Encoder: LLM Interpretability

[GitHub](#)

Built an interpretability pipeline in PyTorch over pre-trained sparse autoencoders to identify which internal features of a transformer activate for a given prompt, using feature extraction and activation patching, with a written explainer on model internals.

Muxwriter: AI Screenwriting Desktop App

[GitHub](#)

Full-stack app (TypeScript, Tauri, Rust) with a context-aware AI collaborator via Anthropic's Claude API; a three-tier context architecture and accept-or-reject diff edits keep the writer in control. Open source, built with Claude Code.

Focusix7: Android Focus App

[GitHub](#)

Shipped a gamified focus app using native Android Accessibility Services, backed by a 283-test Jest suite covering logic, native bridging, and cross-manufacturer behavior; in closed testing on Google Play.

SKILLS

Technical: Python, Java, TypeScript, C, Rust, RISC-V Assembly, JavaScript, SQL, Git, Linux; PyTorch, BERT embeddings, LLM APIs (Anthropic), agent systems, QEMU, Tauri; daily development with Claude Code

Languages: English (fluent), Hindi (fluent), Bengali (fluent)