

Harsh Karia

hnkaria@ucdavis.edu | 669-336-2170 | [Personal Portfolio](#) | [LinkedIn](#) | [GitHub](#)

Education

University of California, Davis, B.S. in Computer Science Sept 2022 – June 2026

- **Relevant Coursework:** UNIX Development, Machine Learning, Computer Networks(TCP/IP), Computer Vision, Algorithm Design and Analysis, Data Structures and Algorithms, Statistics, Cybersecurity, Linear Algebra

Skills

Technical: Python, C++ , JavaScript, React, Machine Learning, Network Protocols, WireShark, TCP/IP, Metasploit, AWS EC2, Linux, Docker, MySQL, Data Modeling, Deep Learning, Network and Browser Security

Experience

Software Engineering Intern, Corgi AI (YC) Sep 2024 - Oct 2024

- Introduced automations for insurance **data parsing** and email marketing using OpenAI's LLM models and built internal APIs for **financial data modeling**, reducing resource requirements by **100x**
- Developed insurance policy pricing by building in-house risk management algorithms and **created databases** for property managers using **PostgreSQL and Python**

Software Engineering Intern, OurDate Dec 2023 – Jan 2024

- Partnered with senior leadership to assess AI usage, designing and **implementing prompts** that accelerated time to final results by **200%** through **query optimization**
- Engaged with beta testers and customers, gathering feedback to refine prompt customization, and **led training initiatives** for employees to enhance prompt generation efficiency by **10x**

Software Engineering Intern, [American Wild Horse Campaign](#) June 2023 – Sept 2023

- Spearheaded end-to-end development of a **mobile application** (IOS and Android) using **React Native** and **JavaScript**, facilitating **data collection** for a nonprofit and successfully launching the app on the App Store
- Designed and deployed frontend user interfaces using **Figma** and integrated **object detection models** with **95% accuracy** to automate wild horse identification in images using Azure Cloud

Software Engineering Intern, SchedGo Nov 2022 – Mar 2023

- Revamped meeting type display on SchedGo's web application to allow students to optimize schedules based on class times using **TypeScript** and **React.js**
- Enhanced app functionality for over **2500 monthly users** across **4 universities** by refining design and integration processes for schedule import/export and resolving outstanding bug issues

Projects

SharkProof [GitHub](#)

- Integrated Hume AI, Whisper, Google Gemini, and Groq models to **analyze 48 user emotions** and 36 facial expressions, **generating real-time** interview feedback
- Created real-time **web sockets** to route live interview data between the Hume AI and frontend, improving **data flow efficiency** by 25% and **reducing feedback latency** for users.

WebTrace [GitHub](#)

- Developed **DNS client** using socket APIs to resolve domain names and extract DNS records, manually **constructing DNS and HTTP packets**, and measuring round-trip times for DNS resolution and HTTP requests.
- Automated web crawling and **HTTP traffic analysis** for 1,000 websites using Selenium with **BrowserMobProxy** utilizing **HAR files** to identify top 10 third-party domains and uncover **online privacy** practices

CourtCheck [GitHub](#)

- Engineered automated tennis analysis tool using **Python**, **PyTorch**, **OpenCV**, and **Detectron2** to achieve **95% accuracy** in tracking court boundaries and ball positions, **reducing human error** in line calls by **90%**
- Utilized **NVIDIA A100 GPUs** to train a Detectron2 model with a **ResNet-50 backbone**, processing over **10,000 annotated images** from tennis video film for robust keypoint detection; integrated TrackNet for ball tracking, enabling precise in/out call simulations and ball **trajectory analysis with 85% accuracy**

MeMonitor [GitHub](#)

- Designed and implemented **network monitoring system** using Python and **dpkt library** to analyze **500+** network packets, enhancing visibility into network traffic, resulting in **30% increase** in network reliability
- Developed **TCP proxy server** and **UDP client-server architecture** to facilitate message routing and **measure throughput** for 100 MB batches of data transfer, **optimizing network performance** and response time by 25%