Matthew Okner

(408) 219-6593 | mokner@berkeley.edu | linkedin.com/in/mokner | github.com/maokner

Education

University of California, Berkeley **Expected Graduation: May 2028**

Bachelor of Science – Electrical Engineering and Computer Science *Berkeley, California*

- Honors Scholar — **GPA: 4.0**
- Courses: Machine Learning, Probability and Random Processes, Computer Architecture and Machine Structures, Data Structures and Algorithms, Signals and Information Processing, Discrete Mathematics and Probability Theory, Foundations of Data Science, Blockchain for Developers ‘
- Activities: **Blockchain at Berkeley** — Developer, Solidity / On-Chain Development

Work Experience

Wrodium **Jan 2026 – March 2026**

Software Engineering Intern *Berkeley, California*

- Built a **content synchronization pipeline** that parses external URLs, computes structural diffs against internal mirrors, and generates **approve/reject change proposals** via an inline review interface
- Designed a **retrieval-augmented content refresh system** integrating the **Perplexity Deep Research API** with **PostgreSQL**, enabling automated detection of stale content and generation of revision suggestions
- Developed **scalable scraping and backfill pipelines** in **Python** and **TypeScript** using **BeautifulSoup**, **Cheerio**, and **n8n**, with **scheduling, retry logic, and deduplication** across 500+ documents

San Jose State University **May 2023 – August 2023**

Software Researcher *San Jose, California*

- Trained a **multilayer perceptron (MLP)** on **15,000+ labeled samples** across **10 malware families**, hitting **91% accuracy** on held-out test data
- Built **feature extraction pipelines** from **opcode sequences**, **byte sequences**, and **byte histograms** to produce structured inputs for model training

Projects

Warden | *Source Code* **TypeScript | Node.js | OpenAI Agents SDK**

- Built **Warden**, a TypeScript guardrail library for **OpenAI Agents SDK** apps that intercepts **100% of wrapped tool calls** through a custom `guardTools()` middleware before execution
- Designed a **risk-based policy engine** that classifies agent actions across **11 risk categories** and enforces allow, deny, approval, or redaction decisions before tools can reach external systems
- Implemented **fail-closed approval and audit workflows** with terminal/Telegram/callback review, secret redaction, configurable timeouts, and **JSONL logs** for policy decisions

Sage.ai | *Devpost* | *Source Code* **Twilio | Deepgram | Anthropic API | MongoDB**

- Built a **real-time voice agent** using **Twilio**, **Deepgram** ASR, and the **Anthropic API**, achieving audio-in to spoken-response latency under **1s**
- Designed an **event-driven pipeline** for call handling: transcription, LLM inference, and **stateful conversation tracking** across turns
- Wired **Google Calendar** into the conversation layer so the agent could book appointments mid-call without dropping context

Automatic Irrigation System | *Research Paper* | *Source Code* **Flask | Raspberry Pi**

- Developed an automated irrigation system using a **Raspberry Pi**, soil moisture sensors, and a Flask server to trigger data-driven watering decisions
- Built a **remote monitoring dashboard** hosted on the Raspberry Pi, enabling users to view sensor data and control irrigation through a web interface

Technical Skills

Languages: Python, TypeScript, JavaScript, C++, Java, Solidity, SQL, ARM Assembly

Libraries/Frameworks: TensorFlow, PyTorch, NumPy, Pandas, Scikit-learn, BeautifulSoup4, Flask, ReactJS, NodeJS

Developer Tools: Git, Docker, Raspberry Pi, Arduino, MongoDB, Google Cloud, Anthropic API

Operating Systems: macOS, Linux, Windows