Tyler Huynh

(717) 816-2919 | tyler02huy@gmail.com | LinkedIn | GitHub | Portfolio

Education

University of California, Davis — Davis, CA
Bachelors of Science in Computational Cognitive Science (Computer Science Emphasis) — *August 2025*

Technical Skills

Languages: Python, TypeScript, JavaScript, HTML/CSS, SQL, Java, C, C#, C++
Frameworks & Libraries: React, Next.js, Node.js, RadixUI, TensorFlow/Keras, Gradio, OpenCV, NumPy, Pydantic
Databases & Platforms: PostgreSQL, MySQL, Supabase, Vercel, Render, Cloudflare, Azure, AWS
Developer Tools: Cursor, Git, Docker, Vite, ESLint, Prettier, Figma, Storybook, pytest, n8n
Core Concepts: Data Structures, Algorithms, Object-Oriented Programming, Full Stack Development, System Design, Version Control, Agile Methodologies, Testing and Debugging, RESTful/GraphQL APIs, WebSockets, Asynchronous Programming, LLM Integration, Prompt/Context Engineering, Deep Learning, Computer Vision

Experience

Chevron — Davis, CA
Full Stack Software Engineer Intern — *April 2025 – July 2025*
- Collaborated with product managers and designers to deploy a web-scraping tool that automated data collection
- Integrated **OpenAI's API** with a **Python**-based scraping pipeline to automate extraction and summarization of metrics, and built an analytics dashboard displaying key insights using **Next.js, FastAPI,** and **PostgreSQL**
- Designed PostgreSQL database schema and directed frontend development by establishing **React** component guidelines with **TypeScript** interfaces; reviewed teammates' code to ensure consistency with UI/UX principles
- Delivered a product that cut research time by **88%** (4 hours to 30 minutes) with **95%** summarization accuracy

CodeLab — Davis, CA
Open-Source Software Developer — *February 2025 – April 2025*
- Contributed reusable **Progress Bar** and **Stepper** components to the club's UI library using **TypeScript** and **Tailwind CSS**, and wrote technical documentation of behaviors, states, and accessibility in **Storybook**
- Led to selection for Team Chevron, resulting in cross-functional collaboration with other developers and designers

UC Davis: Information and Educational Technology — Davis, CA
Junior Computer Room Consultant — *September 2024 – August 2025*
- Resolved technical issues across seven computer labs, prioritizing clear communication and user experience
- Provided on-site troubleshooting for hardware, software, network, and account issues to students and faculty, maintained classroom equipment functionality, and consistently updated internal knowledge base and live tickets
- Supported instruction by proactively addressing issues and minimizing technical disruptions before escalation

Projects

AI-Powered Resume Auditor (GitHub)
Python, OpenAI API, Anthropic API, asyncio, pytest — *November 2025 – December 2025*
- Developed a CLI tool that parses resumes against job descriptions using AI to identify missing keywords, generate optimized bullet rewrites, and recommend quick-learn skills, streamlining the process of tailoring job applications
- Designed an async pipeline coordinating GPT-4 and Claude with **QA feedback loops**, using **Pydantic** for type-safety and **HTTPX** for simultaneous API calls that cross-verify LLM outputs to prevent hallucinations

Real-Time Hand Gesture Recognition (GitHub) (Demo)
Python, TensorFlow/Keras, OpenCV, Gradio — *September 2025 – November 2025*
- Built a full-stack computer vision application using machine learning to classify 12 hand signs with **93% accuracy**, enabling live webcam streaming that delivers instant prediction feedback through an interactive web interface
- Implemented an end-to-end ML workflow from data collection (custom **OpenCV** script) to deployment: trained **VGG16** model with transfer learning and data augmentation, and deployed inference backend on **Gradio**

CoDraw (GitHub) (Demo)
React, TypeScript, Vite.js, Tailwind CSS, Cloudflare Workers and Durable Objects, WebSockets — *October 2025*
- Developed a lightweight collaborative chalkboard web application that enables multiple users to draw simultaneously through WebSocket-based synchronization, while achieving a **sub-500ms** load time
- Utilized a serverless backend with **Cloudflare Workers** and **Durable Objects** to manage real-time room-based connections, and implemented state sync protocol ensuring consistent canvas state for all concurrent users