

Michael Huang

Oakland, CA | (415) 505-8933 | michael.micah.huang@gmail.com | www.linkedin.com/in/michael-micah-huang

EDUCATION

California Polytechnic State University - San Luis Obispo <i>Bachelor of Science in Computer Science, Minor in Philosophy; Graduation: August 2025</i>	San Luis Obispo, CA GPA: 3.40
• Relevant Coursework: Data Structures & Algorithms, Computer Architecture, Operating Systems, Systems Programming, Database Systems, Deep Learning	

PROFESSIONAL EXPERIENCE

Dynamic Experts <i>Contract AI Engineer</i>	December 2024 – January 2025 Remote
• Delivered a fully documented, production-ready codebase with 157 unit tests (Vitest), ensuring seamless handoff to the client engineering team	
Solidigm (Formerly Intel NAND) <i>Automation and Product Engineering Intern</i>	June 2022 – August 2023 Folsom, CA
• Migrated legacy Python 2.7 automation frameworks to Python 3.x, modernizing the codebase for 600+ production-line tests and significantly reducing false-positive failures	
• Engineered end-to-end automation scripts that increased test coverage, saving the engineering team approximately 8 hours of manual verification per SKU during testing cycles	
• Performed root cause analysis on non-product failures, implementing patch fixes that improved overall testing stability and reduced infrastructure downtime	

HACKATHON

Before the Ballot <i>Gemini, ElevenLabs, TypeScript</i>	Jan 2026
• Won 2nd place at Before the Ballot hackathon, building a real-time election misinformation detection system	
SkepticScript – Claude Code Hack Day <i>Azure Speech, Gemini, TypeScript</i>	Jan 2026
• Won 2nd place (Silver Medalist) out of 25 teams at Claude Code Hack Day @ AWS with Jam.dev	
• Architected a real-time speech verification engine that cross-references live audio against LLM knowledge bases	
• Engineered a JSON Schema validation layer using Gemini to ensure deterministic, error-free outputs from non-deterministic LLMs, solving structural hallucination issues	
• Led orchestration of four microservices (Azure Speech, Gemini, Backend, Frontend) and shipped a fully integrated, crash-resistant MVP in a 9-hour sprint	

PROJECTS

Generative Adversarial Network (GAN) <i>Python, PyTorch, Sklearn, Jupyter Notebook</i>	
• Implemented a GAN on the Frey dataset to generate realistic facial images	
Convolutional Neural Network (CNN) <i>Python, PyTorch, Sklearn, Jupyter Notebook</i>	
• Built a CNN to classify images from a complex dataset. Incorporated data augmentation techniques like rotation, flipping, and zooming to improve generalization and reduce overfitting	
• Used pretrained models like VGG19 to extract style and content features and applied optimization techniques to generate blended images. Experimented with different styles and evaluated the quality of the generated images	
Neural Network (NN) Optimization <i>Python, PyTorch, Sklearn, Jupyter Notebook</i>	

- Experimented with various hyperparameters (learning rate, batch size, number of layers, and activation functions) to optimize neural network performance. Evaluated results using accuracy, precision, and recall metrics
- Enhanced the neural network's generalization by implementing regularization methods such as L1/L2 regularization, dropout layers, and early stopping to reduce overfitting

Personal Finance Dashboard (MERN) | MongoDB, Express, React.js, Node.js, JavaScript, TypeScript

- Built a full-stack budgeting application with real-time transaction syncing and automated expense categorization using the MERN stack
- Implemented secure user authentication and RESTful APIs to manage user accounts and transaction data reliably
- Designed a responsive dashboard with visual spend breakdowns and filters so users can quickly identify patterns and adjust decisions

Custom Linux Shell & System Programming | C, POSIX

- Implemented a custom Linux shell supporting pipes and process management to mirror core Unix behavior
- Built a text-based client-server messaging tool between two Linux machines using sockets
- Developed a tape archiver and Huffman encoder/decoder to explore low-level compression techniques

Huffman-Encoding/Decoding | C, Linux

- Implemented Huffman Encoding and Decoding in C to compress and decompress text files. Built frequency tables, constructed Huffman trees, and generated encoded binary output
- Included functionality to decode binary files back into original text format and compare file sizes before and after compression, achieving typical compression ratios of 40-60%

TECHNICAL SKILLS

Languages: Python, C/C++, Java, JavaScript (ES6+), TypeScript, SQL, HTML/CSS, Bash/Shell

Web Development: React.js, Node.js, Express, MongoDB, RESTful APIs, Next.js, HTML5, CSS3

AI/ML: PyTorch, TensorFlow, Gemini API, Azure Speech SDK, Sklearn, Jupyter Notebook, LLMs, Generative AI, Neural Networks, Deep Learning, NLP, Computer Vision

Backend & APIs: Node.js, Express, MongoDB, PostgreSQL, REST APIs, GraphQL, Microservices, MCP, Authentication

Systems & Infrastructure: Linux, POSIX, Unix, System Programming, Sockets, Process Management, Git, Docker, CI/CD

Tools & Other: Git, GitHub, Vitest, Jest, Pytest, Confluence, Jira, Agile, AWS, Azure, Firecrawl API