RICHARD HUANG

richard_huang@ucsb.edu | github.com/huangderful | linkedin.com/in/huangderful/

EDUCATION

University of California, Santa Barbara *Expected Graduation with MS Computer Science: June 2025* | GPA: 3.9

University of California, Santa Barbara *Bachelor's Degree Computer Science: Sept 2021 - June 2024* | GPA: 3.84

Relevant Coursework: Machine Learning, Data Structures & Algorithms, Parallel Computing, Operating Systems, Networks, Advanced C++, Object Oriented Prog, Advanced Graphics, Database, Compilers

SKILLS

Certifications: AWS Certified Cloud Practitioner, Splunk Core Certified User
Languages/CMD: Java, JavaScript, Swift, TypeScript, Node.js, Ruby, C, C++, C#, SQL, Python, Rust, XML, HTML/CSS/JS, JSON, Splunk SPL, Kotlin, Unix, Powershell, Bash, Shell Scripting
Cloud/DevOps: AWS, Azure, GCP, Docker, Terraform, Ansible, CI/CD, Automation Controller, Splunk, QEMU
Machine Learning: TensorFlow, Pytorch, AzureML, Scikit-Learn, BERT, Mistral, NLP, HuggingFace, GenAI
Libraries/Misc: React.js, Vue.js, Express.js, TailWind CSS, jQuery, Rails, Tsung, Java Spring Boot, Unity, OpenGL, REST, discord.js, SpigotMC, Jira, Confluence, Slack, Github, Git, JUnit, PyTest, Mocha, Microservices

EXPERIENCE

Cloud Engineer Intern @ Nintendo *June 2024 – August 2024 | Ansible, Automation Controller, Nutanix Beam, ServiceNow CMDB, HTML, AWS Lambda, SNS, Cloud Formation, Event Bridge, Azure AD, Microsoft Graph API*

Created 4 automation projects annually saving 120 hours in cost reporting and 27 hours for rotations, notably:

- Prod v Nonprod Spending – With AWS cost categories, lambda, and event scheduler, created a Cloud Formation template which automatically measures prod v nonprod account spending
- SAML Replacement – Using Ansible Playbooks, created automatic rotation of SAML certificates on Azure AD to AWS Identity Center heightening the security of AWS accounts

Hypergraph Random Walk Graph Neural Networks *October 2024 - December 2024 | GNN, VGAE*

- A project for CS292F at UCSB which expands on the original concept of the Random Walk Graph Neural Network by creating hypergraph of hidden graphs and applying a variational graph auto encoder
- Formal paper and presentation, code

Schrodinger's Full Stack *October 2024 - December 2024 | AWS, Ruby on Rails, Tsung*

- A project for CS291A at UCSB which is a Twitter-like Full Stack application in Ruby on Rails, load tested it via Tsung, and performed web optimizations like pagination with 3 other people
- Report with screenshots of app, tsung tests, and optimizations, code

Co-Founder @ GTO Nexus (gtonexus.com) *May 2023– Present | AWS, Rust, TypeScript, Vue, Next.js, Node.js, Stripe*

- Created a game theory optimal poker software that was built on with rust functions that connect to a Vue front end via Tauri. There are two main features to the app: the trainer which looks through a tree of poker solutions generated by the rust backend and the solver which "solves" a poker game which then display results to the user in a professional intuitive manner (gtonexus.com/downloads to try free)
- Created landing site (gtonexus.com) with Next.js, the functionality to create users on our backend, and talk to Stripe with webhooks
- Created and manage a secure Cloud environment in AWS and wrote lambda functions to automatically send emails, create and store license keys, store logging information, and verify accounts

Software Engineer Intern @ First Republic Bank *June 2022 – August 2022 | Splunk, Ansible, Terraform, AWS*

- Migrated on-prem Splunk systems to cloud via Terraform and Ansible and created Service Control Protocols in AWS to enhance the security posture

Full-Stack SWE Intern and Interviewer @ Invi Grid (invigrid.com) *June 2021 – June 2022 | Google Cloud SDK, Google Cloud APIs, HTML, CSS, JS, Node.js, React, Express*

- Spearheaded the initial framework for Invigrid as the first full-stack developer
- Used Node.js googleapis to automatically detect security noncompliance against industry level CIS security benchmarks within a GCP project for the backend and display those vulnerabilities on a frontend portal via React and vanilla HTML, CSS, and JS.