Alexander Antonio Alvarado-Barahona
alexaab@stanford.edu • linkedin.com/in/alexaab • github.com/SirHauler • (515) 608-9032

EDUCATION

Stanford University, Stanford, CA | Class of 2025

• B.S Computer Science, Systems Track

Relevant Coursework: Computer Networking (TCP/IP/Bytestreams, C++), Operating Systems (C and Unix), Reinforcement Learning (Tensorflow), Deep Learning for Computer Vision & Natural Language Processing (PyTorch)

SKILLS

Coding: C/C++, Python, Java, JavaScript/TypeScript, HTML, CSS, x86 Assembly, Unix, SQL

Technologies: AWS, Google Cloud, React & React Native, Jupyter Notebook

WORK EXPERIENCE

Amazon Web Services | Jersey City, NJ | SDE Intern | Summer 2023

• Worked in AWS Fintech with a team that specializes in aggregating revenue, profit & loss, and other metrics for all AWS products.

• Designed a utility that synchronizes changes from production level financial databases to lower level environments: Gamma, Alpha & Beta. The feature helps developers test new features/services with up to date financial data. Utilized AWS Athena (SQL), API Gateway, Boto3, IAM, and DynamoDB.

Amazon Web Services | Seattle, WA | SDE Intern | Summer 2022

• Worked with an AI team that specializes in storing and managing datasets that are used to train AI models for Amazon Lex and Amazon Rekognition.

• Designed a labeling feature (Java) that uses API Gateway and a Lambda function to create a reference to a dataset with labels in an ElasticSearch Cluster. The feature supports thousands of labels per dataset for the existing 400,000+ datasets and helps internal teams label and discover their datasets.

• Utilized AWS Lambda, DynamoDB, API Gateway, Cloudformation, and ElasticSearch.

PROJECTS

LocalEats

• Built a mobile app using React Native to help users find their favorite vendors who are always on the move. Utilized AWS Amplify, DynamoDB, GraphQL and Google Map APIs to provide up to date locations and user specific recommendations for new vendors based on their ratings for similar vendors.

Notes

• Collaborated on a multiplayer piano sight-reading game similar to NitroType in which users can connect their electric keyboards and compete while playing popular pieces. Utilized Unity 2D and C#.

Offside-O-Matic

• Implemented a pipeline to detect offsides in soccer matches using video captured by tactical cameras. The pipeline utilizes a deep neural network to classify players (which teams they play for), and the ball. This information is then used to calculate the offside line and make a decision.

Ping Pong Prodigy

• Built a 2D ping-pong simulation in Python and implemented an Actor-Critic reinforcement learning algorithm using Keras and Tensorflow to train an agent to play against a hard-coded AI agent. RL Agent learned to play better than a near perfect AI Agent (AI Agent only makes mistakes 5% of the time).