Timothy Cai

timcai.tyc@gmail.com | github.com/TAMUTim | 832-951-7889 | www.timcai.dev

Education

- **Texas A&M University** College Station, TX
  Bachelor of Science in Computer Science *Graduating May 2025*
  - **Relevant Coursework:** Data Structures and Algorithms, Competitive Programming, Probability, Discrete Mathematics, Analysis of Algorithms, Formal Languages and Automata, Computer Organization, Distributed Networks and Systems, Airport Systems and Design, Operating Systems

Experience

- **Frogslayer** College Station, TX
  Junior Software Developer *June 2022 - June 2024*
  - Software consulting firm, with projects focusing on everything from **Angular** with **.NET Core** to **Azure Cosmos** and **Vue**
  - Rewrote calls to graph database, eliminating unefficient queries and reducing overall load by **270%**
  - Responsible for tight turnaround in fast paced development environment,
  - Wrote more than **30 API** endpoints and over **150 Angular, Vue, React** components
- **Senseye** Austin, TX
  Software Engineer & Machine Learning Intern *May 2023 - Aug. 2023*
  - Designed and implemented internal **Python** library to autonomously pull data from **AWS Athena/Glue S3**
  - Lightweight and flexible in ML pipeline, with maximum bandwith up to **600 Mb/s**
  - Filters out a peak **82%** of bad data through hashing, while costing nothing in terms of compute resources

Projects

- **Found in Translation** *Tools Used: Python, JavaScript, Slack Bolt, Flask, Node.js, Co:here, Pinecone, Azure*
  - **Overall Winner** at the Cohere 2023 Semantic Search Hackathon
  - Intelligent Slack bot that can **semantic search** for messages across languages as well as analyze emotions in a server
  - Utilized **Flask** to train **co:here** models on the Google GoEmotion dataset, with storage through **Pinecone**
  - Developed user interface with **Slack Bolt API**, and hosted everything on **AWS ec2** instances
- **Multithreaded Web Crawler** *Tools Used: C++, WinSock, TCP, DNS*
  - Designed and implemented web crawler scalable to **10000 threads** concurrently with Visual C++
  - Fully memory safe and robust to errors, crawling HTTP standard urls with TCP over windows sockets
  - With robots.txt detection and DNS handling, was able to parse over **1 million urls** in less than 5 minutes
- **Mock Shell** *Tools Used: C++, PHP*
  - Mock Linux shell accepting multiple commands and flags, with **multithreading** capability
  - Utilized **POSIX** standard to implement and maintain low level, efficient C++ code
  - Developed and designed systems for piping, file I/O redirection, background processes
- **NUC Legion** *Tools Used: MetalLB, Ansible, Kubernetes / k8s*
  - Built a highly available, 6 node Kubernetes cluster with leftover Intel NUCs through **k8s** and **Ansible**
  - Running in 1 master / 5 worker node configuration, with up to 3 node down tolerance.
- **Small Distributed Social Network** *Tools Used: C++, gRPC, glog, cmake*
  - Implementation of a social network service utilizing **Chubby lock system** with 3 server clusters communicating over **grpc**
  - Designed to be scalable, fault tolerant, and highly available with up to 1 down cluster
  - Server clusters split into master / slave, with **Chubby** running as coordinator, yielding close to **99%** uptime.

Technical Skills

- **Skills**: Competitive Programming (C++), Full Stack, Networks and Distributed Engineering
- **Languages**: *(Proficient):* Python, Java, C++, Typescript, HTML, CSS *(Familiar):* Go, Scheme, SQL, Rust
- **Technologies**: Pinecone, Pandas, React, Angular, .NET Core, Node.js, Firebase, Heroku, gRPC, glog, cmake, Azure, AWS, Vue, Svelte

Leadership

- **Roles**: Aggie Competitive Programming Club Officer