Keegan Smith

📞 +1 (832) 773-8754
✉ keeganasmith2003@gmail.com
🌐 keegancodes.com | github.com/keeganasmith

Education

Texas A&M University — College Station, TX

Master of Science — Computer Science *Jan 2025 — May 2026*
GPA: 3.8
Bachelor of Science — Computer Science *Aug 2021 — Dec 2024*
GPA: 4.0

Selected Coursework:

Deep Learning (PyTorch, Tensorflow, Pandas)	Machine Learning (PyTorch)	Software Engineering (Ruby on Rails)
Computer Architecture (C++)	Parallel Algorithms (C++, Open MPI, YGM)	Computer Networking (Visual C++)
Operating Systems (C++)	Quantum Algorithms	Programming Languages (Java, Scheme)

Work Experience

- **High Performance Computing (HPC) Research Fellow** *May 2023 - Present*
  Texas A&M University High Performance Research Computing College Station, TX

  - Benchmarked popular scientific algorithms such as multi-core FFT and sparse matmul in C++ (**FFTW**, **oneMKL**) on Intel Optane configured as swap space
  - Helped researchers from Fermilab train neural nets for speeding up High Level Synthesis for FPGA development by converting their kubernetes workflows to slurm scripts on ACES cluster
  - Developed an Open OnDemand interactive application for assisting researchers with programming tasks, utilizing local llm inference to do so. Local inference was distributed across multiple nodes and performed on 4 Intel GPU Max 1100s per node. (Python)
  - Accelerated science research by shipping an Open OnDemand interactive app to automate machine learning pipelines with **AutoGluon** on Slurm; standardized job specs and reduced setup time for researchers. (JavaScript / Vue.js / Python / FastAPI)
  - Automated the ticketing system by integrating with Slack using the Slack web API. Created a robust workflow CLI to allow ticket workflow changes to be made without modifying code. (Python)
  - Resolved a wide range of HPC support tickets from researchers, including software troubleshooting, job scheduling issues, and environment configuration across diverse scientific domains including chemistry, biology, and atmospheric science.
  - All development done on RHEL 8 machines

- **iOS Developer** *May 2023 - present*
  AKW Ventures College Station, Texas

  - Created and maintained backend for "College Football: Rapid Roster" iOS app (Swift / JavaScript / Cloudkit JS) which displays robust college football statistics in a user friendly interface
  - Application supports thousands of active users with an average monthly recurring revenue of $2500

- **Software Engineer Intern** *May 2024 - August 2024*
  The Coca-Cola Company Atlanta, Georgia

  - Built a segmentation & analytics web app (Vue.js, Flask, Pandas) to profile 8,861 retail outlets; engineered features from transactions, product mix, and geography to support cluster-based analyses.
  - Developed a promotion–outlet matching algorithm that generated 124,054 relevant recommendations; designed the data pipeline (cleaning, feature generation, candidate generation, ranking) and exposed results via an analyst-friendly UI with downloadable cohorts.

Research

- **Parallel Bridge Finding** *January 2025 - Present*
  Texas A&M University, under Dr. Roger Pearce

  - Developed and iterated upon a novel parallel bridge finding algorithm (graph theory)
  - Achieved near linear scaling up to 768 cores - 15.36 times as many cores as the current literature supports
  - Distributed computing on Lawrence Livermore National Labs systems, notably the Dane cluster
  - Implemented in C++ using the You've Got Mail library (an asynchronous parallel computing library built on OpenMPI)

Projects

- **Temporal Frequency Analysis on CPU, GPU, FPGA** *August 2024 - December 2024*
  Texas A&M University capstone project done in collaboration with Sandia National Labs

  - Optimized an existing temporal frequency analysis algorithm for detecting drone activity on CPU, GPU, and FPGA architectures
  - Won 3rd place in the Texas A&M Engineering Project Showcase Fall 2024
  - Our GPU implementation achieved the highest throughput, achieving a 125x throughput improvement over the original FPGA implementation
  - Increased throughput of FPGA implementation by 4.23x
  - Used C++, Vitis HLS, CUDA, OpenMP, JavaScript, Express.js, React.js

- **Push Battle Bot Artificial Intelligence** *November 2024*
  Texas A&M University Datathon

  – Developed an AI for playing the push battle board game
  – Utilized the minimax algorithm with custom heuristics
  – Won 2nd place out of more than 300 contestants
  – Used Python

- **M-Height Approximation** *May 2025*
  Texas A&M University CSCE 636 (Deep Learning)

  – Created a large training dataset (approximately 100 million m height problems and solutions) using OpenMPI
  – Trained a dense neural net with 1 million parameters on multiple A30 GPUs
  – Achieved a log loss of 1.20
  – Used PyTorch and OpenMPI

Technical Skills

DevOps: Docker, Singularity, Github Actions, Git, GitHub, CI/CD
Skills: PyTorch, C++, Python, Java, Bash; CUDA, Open MP, Open MPI, Slurm, YGM; Docker, Singularity, Linux, RHEL 8; Flask, Vue.js, Express.js, React.js, Vim, JSON, AWS Amplify, AWS Lambda Functions, AWS EventBridge, AWS DynamoDB, MongoDB
Methodologies: Agile, Waterfall, Object Oriented Programming, Functional Programming, REST APIs