# Ivan Zaplatar

(512)-239-8224 | zaplatarivan@gmail.com | https://www.linkedin.com/in/ivan-zaplatar/ |
https://github.com/ivzap

## EDUCATION

**Texas A&M University** | **GPA:** 4.0 / 4.0 **College Station, Texas**
*Bachelor of Science in Computer Engineering* Expected Dec 2024

- **Relevant Coursework:** CSCE 120, CSCE 222, CSCE 221, ECEN 214, ECEN 248, STATS 211, MATH 251, MATH 308

## SKILLS

**Languages:** C/C++, Python, ARMv8, Java, Rust, JavaScript, Verilog, BASH, HTML, CSS, SQL
**Technologies:** Git, Linux/Unix, LLVM, Clang, GCC, Cmake, Docker, React, Node.js

## EXPERIENCE

**Kershner Trading Group** **Austin, Texas**
*Software Engineer Intern* June 2023 – Aug 2023

- Developed and deployed a Python and C++ call graph tool for traders' scripts, leveraging the Abstract Syntax Tree (AST) modules in both languages, and pushed into production using Docker
- Call graph tool assisted in identifying circular dependencies and risky market orders within multiple traders' scripts
- Wrote documentation for the company's market simulator SDK, subsequently gaining familiarity with CMake
- Converted trader/quant Python scripts to C++ for low latency, which improved speeds by over 200%

**DV Trading** **Austin, Texas**
*Software Engineer Intern* May 2022 – Aug 2022

- Developed the frontend and backend of stock algorithm optimization software using Electron.js
- Reduced algorithm development time by 40% by generating C++ code with optimal parameters
- Improved an algorithm by 60% via statistical techniques, resulting in a daily profit of +$10,000
- Wrote, refactored, and documented over 10,000 lines of code in Python, Rust, and JavaScript

**Kershner Trading Group** **Austin, Texas**
*Software Engineer Intern* May 2021 – Aug 2021

- Built frontend and backend of stock analysis software with Python, JavaScript, Node.js and C++
- Utilized in-depth understanding of the company's tech stack and documentation
- Automated an alert system to detect down servers, saving the IT department 7 hours weekly
- Wrote and documented over 5,000 lines of code for various libraries and APIs in Python and C++

## PROJECTS & EXTRACURRICULARS

**Custom Programming SHELL**

- Wrote an entire programming shell that supports sign expansion and i/o redirection using IPC
- Implemented using C/C++ in addition to unix system api calls such as fork, exec, and pipe

**OpenJDK Systems Research (Texas A&M University)**

- Integrated marking of RDD objects into the OpenJDK G1 garbage collector source code using the Java Native Interface (JNI) to enable caching of Spark RDDs
- Required knowledge of G1 heap allocation(eden, survivor, old), object tracing, and marking phase