





Matt Zhang

 github.com/mzhang  matt@zhang.software  zhang.software  mattzhang-

EDUCATION

University of Waterloo

2020 – 2024 | GPA: 91.2%

Bachelor of Software Engineering (BSE)

PROFESSIONAL EXPERIENCE

Incoming @ Twitter

May 2023 | San Francisco, CA

Backend Engineer // Go · Scala

Spatial

Sep 2022 – Jan 2023 | San Francisco, CA

Software Engineer // Typescript · React · Go · C#

- Engineered in-browser visualization of 3D spaces using **89%** less bandwidth than existing solution
- Lead end-to-end integration of a chat platform for **150k** MAU leading to **21%** increase in time spent by users
- Architected and implemented user avatar interactions system accumulating over **2 million** uses weekly

Immigrate.ai

Dec 2021 – Apr 2022 | Toronto, ON

Software Engineer // Typescript · React · React Native

- Created **React Native** cross-platform NLP chatbot portal resolving **87%** of user inquiries
- Authored full-featured task management platform in **React** to empower & organize 12 internal teams
- Optimized rendering & memoization, reducing API calls by **28%** and load time by **32%**
- Implemented traveling salesman heuristic to reduce average runtime by **27%**

Imagine Communications

May 2021 – Sep 2021 | Toronto, ON

Fullstack Developer // C# · Python · Typescript

- Triaged & fixed critical authentication bugs in **C#** that blocked **35%** of users from accessing products
- Reduced complexity and file size of product deployment by **63%** using **WebSockets** and **RabbitMQ**
- Launched pipeline to normalize **PostgreSQL** databases, reducing storage by **25%** and query time by **33%**

PROJECTS

Automated Proof Checker

- Designed & implemented scanner, validator and checker for mathematical proofs in **C#**
- Modified Shunting-Yard algorithm to parse n-ary inputs, reducing time complexity from cubic to linear

Scala Compiler

- Built high-level programming language compiler converting Scala-like syntax to MIPS assembly
- Supports type checking, functions(nesting, scoping and recursion), closures and garbage collection

Modular Turn-Based Game Engine & CPU

- Architected & built turn-based game engine in **C++** supporting creation of arbitrary games at runtime
- Implemented minimax with alpha-beta tree pruning for AI capable of playing arbitrary user-created games

LANGUAGES & TECHNOLOGIES

Typescript

Go

Python

C#

C++

React/React Native

SQL

MongoDB