

JUDE GAO

jude.gao@uwaterloo.ca • 519-781-2577 • github.com/gaojude • linkedin.com/in/jude-gao

EDUCATION

UNIVERSITY OF WATERLOO, Ontario, Canada
Bachelor of Computer Science, Honours with Major GPA: 3.7

09/2017 – 06/2020

SUMMARY OF QUALIFICATIONS

- Profile: C++ Programmer & Web Developer in MERN (Mongo, Express, React, Node.js)
- Languages: C++, Java, Scala, JavaScript (in React & Node.js)
- Knowledge: OOP Design Patterns, Algorithms & Data Structures, Foundational Concepts

PROFESSIONAL EXPERIENCE

Project Team Member [Java, Scala]

UNIVERSITY OF WATERLOO, Ontario, Canada

01/2020 – 04/2020

- Implemented a compiler that compiles Java 1.3 into i386 assembly & contributed over 6000 LOCs in Scala
- Transformed parse trees into a well-designed abstract syntax tree that significantly saves development time
- Made strategic decisions in forgoing certain parts to meet the deadline of overall project
- Drafted 5 reports that summarize the work at each 1/5 point of the total duration of project

Intern Developer [JavaEE]

GOLDEN WEB KING, Zhejiang, China

05/2017 – 08/2017

- Built J2EE backend for a previously manually operated site, mitigating constant redeployment
- Wrote a data visualizer component using open-source library ECharts
- Onboarded in half the time typically allotted to interns & learned Java Spring with Oracle database
- Served as key point of contact to clients & responsible for installing authentication devices

PROJECTS

Realtime Blog [MERN] [\[Site\]](#) [\[Repo\]](#)

05/2020 – Present

- Implemented in MERN stack and utilized Redux for state management
- Supports online editing (w/ CodeMirror) in markdown & displays posts with Math equations (w/ MathJax)
- HTTP requests from the frontend were handled asynchronously through Axios & Redux Sagas and routed to the backend where a Node.js server with Express would handle the CRUD operations
- React app builds were hosted to GitHub & Heroku and MongoDB Atlas to host Node.js and MongoDB

Tetris [C++] [\[Repo\]](#)

03/2018 – 04/2018

- Classic Tetris game implemented in C++ with simultaneous CLI & GUI views
- Utilized modern C++ 14 with heavy use of STL containers & smart pointers
- Effectively planned for project, using OOP concepts and design patterns (Observer, Factory, Command)
- Illustrated UML & discussed challenges and resilient to changes in reports

“A C-Like Language” Compiler [C++]

08/2018 – 12/2018

- Compiles a C-like language into MIPS, including Context-Sensitive Analysis & Code generation
- Designed an $O(n)$ Dynamic Programming algorithm to efficiently type check parse trees
- Actualized theoretic practice by simulating DFA & Knuth's $LR(1)$ Transducer
- Implemented in perfect conformity with specifications & one of 1% to submit