JACOB WILLIAM CROUCH

1355 Blanchette Dr. ⋄ East Lansing, MI 48823
https://jcbcrouch.github.io/ ⋄ 517-897-3450 ⋄ jwcrouch@umich.edu

EDUCATION

University of Michigan (Senior) *September 2015 - Present*
Major: Computer Science (3.32 GPA)

SKILLS

Skills c/c++, Python, javascript, SQL, AJAX, Jinja, React, Flask, Rest API
Languages Spanish: Limited Working Proficiency

COURSEWORK

- System Design of a Search Engine in c++
- Web Systems and Design
- Data Structures and Algorithms
- Foundations of Computer Science
- Introduction to Statistics and Data Analysis
- Database Management Systems
- Introduction to Computer Security
- Cryptography
- Programming Introductory Data Structures
- Computer Organization

Projects

- **Search Engine**: Currently building a search engine in c/c++ linux with a small team of students using our own libraries. As of now, I've built a query compiler and HTML parser. This project emphasizes project planning, system design, multi-threading, sockets, system calls, data storage, and atomicity.
- **Web App**: Built a mock instagram web app with client-side dynamic pages using Flask, React/JS, python, sqlite, and REST API. Included infinite scrolling and sessions.
- **Creative AI**: Used natural language processing to create new Beatles songs from their lyric catalog and new Nintendo music. This project taught me how to work with python and its different libraries. I also learned basic NLP techniques. I enjoyed this project because it combined coding with music.
- **Virtual Stock Exchange**: Created a virtual market of buyers and sellers with priority queues. I also added features such as a time traveler mode to go back and make maximal profit, a trader-info mode where the activity of each trader was recorded, and a median mode where I maintained a running median of daily stock prices. This project really tested my debugging skills.
- **Image Processing**: Implemented a seam carving algorithm to resize images while keeping the most important pixels. For this project I had to read a paper on the seam-carving algorithm and then implement it. Not only was the project very cool, but I learned how to read technical papers.
- **Python MapReduce Server**: A single machine, multi-process, multi-threaded server that executes user-submitted MapReduce jobs. Spawned each new worker as its own process with a thread connected to the master to send UDP heartbeat messages. All other communication done between the master and workers was done over TCP. The master assigns tasks to workers based on their state. This server is fault tolerant because the master thread reassigns tasks from workers that miss 5 heartbeat pings.

INTERESTS

- Backend
- Databases
- Software Development
- Web Development
- Security
- Soccer