

James Houghton

CONTACT INFORMATION

ADDRESS: 1720 Chesterbrook Vale Ct., McLean, VA 22101
PHONE: +1 (571) 242 9362 | EMAIL: jhoughton@virginia.edu
WEBSITE: jhoughton.me | GITHUB: github.com/jamesthoughton

EXPERIENCE

- JUN 2016 - PRESENT** | **Lead Web Developer at INSIGHT INTERFACES**
Full-Stack Development and Cloud Deployment
Created a browser-based remote teleconferencing application built on top of WebRTC. Developed with the Django web application framework, the Node.js runtime, Socket.IO, and Redis. Became familiar with cloud deployment with AWS (EB, EC2, and RDS) via Docker containers. Honed front-end JavaScript development and webpage design skills in CSS.
- SEPT 2013 - JUN 2017** | **Student Systems Administrator at TJHSST**
Improved and maintained computer systems used by faculty and students including web servers, workstations, and parallel computing clusters. Made heavy use of AFS, KVM, slurm, and Nginx. Contributed to the student intranet project, a Django-based web application.

EDUCATION

- MAY 2020** | **University of Virginia - B.S. in COMPUTER SCIENCE, B.A. in PHYSICS - GPA: 4.0**
Relevant Coursework: Computer Architecture, Program and Data Representation, Discrete Math, Linear Algebra, Differential Equations, Probability Theory
- JUN 2017** | **Thomas Jefferson High School for Science and Technology**
Fairfax County Advanced Studies Diploma - GPA: 4.5 (Weighted)
Relevant Coursework: Artificial Intelligence, Parallel Computing, Computer Vision, Quantum Mechanics and Electrodynamics, Advanced Math Techniques for Scientists and Engineers

SKILLS

- WEB APPLICATION DEVELOPMENT:** JavaScript, CSS3, HTML, Django, AWS
PROGRAMMING: C++, C, MPI, OpenMP, Java, Python
VERSION CONTROL AND BUILD SYSTEMS: git, CMake, GNU Make
LINUX & SYSTEM ADMINISTRATION: Nginx, Salt, Docker, Kerberos, KVM, Bash, GPG

PROJECTS

Multi-threaded Wikipedia indexer & Wikipedia game solver (Jan 2018)

Worked with mutexes, condition variables, and atomic variables to create performant, synchronized multi-threaded code, resulting in a 4x speed up over single-threaded code. Wikipedia indexing is done using parallelized DFS, and HTML parsing is done using regular expressions.

Console-based video viewer (May 2017)

Used OpenCV to read video frames, and displayed them using color escape sequences in supported terminals.

Persistent High Frequency Audio Removal in Music (Mar 2017)

Used `scipy` and `numpy` to run a DFT on many samples in audio files to find and remove persistent frequencies greater than 12kHz, common in studio recordings from the early 2000s.