

Jacob Tucker

Website: jacob-tucker.github.io | GitHub: github.com/jacob-tucker | Email: jacobtucker818@gmail.com | Phone: (914) 217-8139

EDUCATION

Northwestern University

B.A Computer Science

GPA: 4.00/4.00

Expected Graduation: 2022

EXPERIENCE

Decentology

September 2020 - Present

Blockchain Developer & Software Architect

- Wrote the smart contracts, transactions and scripts for [MotoGP Ignition](#).
- Serving as the lead Flow developer by architecting and writing all smart contracts, transactions, and scripts in Cadence.
- Constructed [DappStarter](#), a platform that kickstarts developers in creating decentralized applications by providing them with an out-of-the-box full stack application including server-side tests, a UI harness & everything blockchain related.
- Contributed to product & business strategy through daily standup calls.

Undergraduate Teaching Assistant

April 2020 - Present

Northwestern CS211

- Helped instruct multiple Computer Science courses (CS211 & CS213) in C/C++ and x86-64 Assembly Language, respectfully.

PROJECTS

ReKt ([rekt.netlify.app](#))

February - March 2021

Personal Project

- Connected Metamask wallet to React.js front-end using Web3, allowing users to sign transactions with test Ethereum on the Ropsten Test Network.
- Wrote and deployed an upgradeable smart contract in Solidity onto the Ethereum Ropsten Test Network that handles the uploading & storage of moments to the blockchain.
- Stored all videos on IPFS, a decentralized storage network, allowing the smart contract to only store a hash that can be retrieved by making API requests on the front-end through Infura.

Search Algorithm Tool ([searchalgorithmtool.netlify.app](#))

July 2020

Personal Project

- Implemented a visual tool for learning graph search algorithms.
- Utilized React.js hooks that handled the state of the walls, the nodes currently being searched, the discovered nodes, and the start & endpoint simultaneously.
- Wrote 5 graph traversal algorithms, including A*, Dijkstra (with and without a Binary Heap), Greedy Best-First Search, and Depth-First Search using priority queues and other data structures.
- Programmed a maze generator using recursive division along with a random maze generator.

SpotifyAid ([spotifyaid.netlify.app](#))

August 2020

Personal Project

- Built a Spotify tool that allows users to view & manage their profile, their playlists, followers, the artists they follow, their tracks, and more.
- Made GET/POST requests to the Spotify Web API to retrieve statistical data on any track a user would like to look up, along with their own top tracks & artists.
- Implemented Spotify user authentication tokens in Node.js by making GET/POST requests to Spotify's REST API and storing the access/refresh tokens in browser cookies.

SKILLS

Front-end: React.js, JavaScript, HTML5, CSS3, Sass, Axios

Back-end: Node.js, MongoDB (NoSQL), REST APIs, JSON, Python, Socket.io, JWT

Other Languages/Tools: Metamask, Infura, Web3, C++, C, Cadence, DSSL2, Docker, Git, Heroku, Netlify