Jacob Tucker

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

EDUCATION

Northwestern University **GPA: 3.96/4.00**

B.A Computer Science Expected Graduation: 2023

- Selected Courses: Web Development, Algorithms

Open World Builders: Blockchain Bootcamp (onflow.org) July-August 2020

- Built a capstone project in Cadence, a new smart contract language for the Flow blockchain, and served as the technical lead in a group project of 3 to design & develop a new way to leverage loyalty rewards on the blockchain.

EXPERIENCE

Undergraduate Teaching Assistant July-August 2020

Northwestern CS211

- Helped instruct a Computer Science course (CS211) in C/C++ and oversaw the completion of multiple mini-game final projects.

PROJECTS

SpotifyAid (spotifyaid.netlify.app) August 2020

Personal Project

- Built a spotify tool that allows users to view & manage their profile, which displays 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.
- Handled 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 to prevent malicious logins.
- Learned & utilized Sass and tied it to a React.js front end.

Search Algorithm Tool (searchalgorithmtool.netlify.app) July 2020

Personal Project

- Implemented a visual tool for learning graph search algorithms using React.js hooks that handled the state of the walls, the nodes currently being searched, the discovered nodes, and the start & end points simultaneously.
- Wrote 5 graph traversal algorithms, including A*, Dijkstra (w and w/o 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.

EasyTalkv2 (easytalkv2.netlify.app) May-June 2020

Personal Project

- Built an instant messaging chat application that allows users to send messages back and forth simultaneously through WebSockets (using Socket.io library), via a Node.js server.
- Enabled the ability to continue conversations at any time by retrieving user profiles and chat conversations stored on a NoSQL MongoDB Atlas cluster, using AWS as the cloud provider.
- Designed user personalization by giving users the ability to change the styling of their chat.
- Successfully built RESTful APIs that handle client-server communications that are protected using JSON authentication tokens stored through browser cookies.

SKILLS

Front-end: React.js, JavaScript, HTML5, CSS3, Sass, Axios
Back-end: Node.js, MongoDB (NoSQL), REST APIs, JSON, Python, Socket.io
Tools: AWS EC2, Docker, Git, Heroku, Netlify