

Samuel Quaye

(614)-805-7088 · quayes955@gmail.com · Canal Winchester, OH · <https://github.com/squaye6756> · <https://squaye6756.github.io/>

Summary

I'm a software engineer and a graduate of the General Assembly bootcamp. I have an analytical mind and can quickly assess problems and come up with solutions. I always strive to complete given tasks to the fullest of my ability and provide outstanding results.

Education

Franklin University

Bachelor of Science, Computer Science

COMP 215 - Programming Language: Principial/Practice

COMP 281 - Database Management Systems

COMP 294 - Computer Science Practicum

COMP 311 - Object Oriented Data Structures and Algorithms

COMP 323 - Fundamentals of Operating Systems

WEBD 325 - Mobile Programming

2023

Skills

Software: Javascript, HTML, CSS, React.js, Angular, C, Java, Python, MongoDB, Mongoose, PostgreSQL, MySQL, django, JQuery, Express.js, Node.js, Bootstrap, Git/Github, Trello, Microsoft Office (Word, PowerPoint)

Data Structures (Java): BSTs: AVL Trees, Heaps (Priority Queues), Stacks, Queues, Hashmaps, Linked Lists, Arraylists

Sorting Algorithms: Merge Sort, Quick Sort, LSD Radix Sort, Insertion Sort, Selection Sort, Bubble Sort

Software Development Projects

Brackets-R-Us (<https://brackets-r-us.herokuapp.com>)

- Utilized React to create-the front-end for this application, allowing users to sign up for, post, and display the results of tournaments
- Implemented React-Router to display each tournament, the tournament list page, and the user page upon successful login
- Used axios calls to apply user authentication and CRUD operations on the tournament data

Streetball (<https://streetball-front.herokuapp.com>)

- Utilized React to create the front-end for this application, allowing users to post about their street basketball team's skills
- Used axios calls to communicate with this application's backend utilizing the MERN (MongoDB, Express, React, Node.js) stack

Coin Sharing Collection (<https://murmuring-dusk-27318.herokuapp.com>)

- Created a RESTful app utilizing an Express, MongoDB, and Node.js backend which handled create, read, update, and destroy (CRUD) operations
- Utilized Mongoose to create a one-to-many relationship between two schemas, User and Coin and employ user authentication
- Implemented EJS to create an MPA

Frames, Frames, Frames (<https://competent-keller-89a6f8.netlify.app>)

- Using Ajax calls, this app uses the Kurogane Hammer API (<https://api.kuroganehammer.com/swagger/index.html#/>) and allows the user to sort through a character's frame data
- JQuery functions used to update the display once response from api is returned

MyDrawingTracker

- Utilized a .NET framework and JQuery to develop an app that makes drawings