Samuel Diaz

FRONTEND ENGINEER

(347) 994-4690 samueldiaz216@gmail.com samueldiazdev.netlify.app

SKILLS

Languages
Javascript(ES6), HTML, CSS/ Sass, Java

Databases
MongoDB, SQL

Libraries & Frameworks
React, Bootstrap, Mongoose, Node.js, Express

Tools
Git/Github, Heroku, Netlify

EDUCATION

CUNY Lehman College
BS Computer Science
May 2020

RELEVANT COURSEWORK

Studied data structures such as arrays, stacks, queues, trees and graphs. Wrote algorithms that made use of these data structures, including sorting, searching, and memory management algorithms. Built software that implemented these data structures and algorithms, while taking best practices into considertion.

WORK EXPERIENCE

Web Dev Tutor — **Wyzant**
(February 2019 - Present)

- Administer one-on-one tutoring sessions to students of various levels on Javascript, HTML, CSS, React, Node.js, and Express
- Provide supplemental assistance to students completing class projects.
- Create individualized lesson plans for students depending on their level of knowledge

Peer Mentor — **MSEIP-RISE**
(July 2016 - May 2017)

- Led a peer mentor group in the creation of lesson plans for high school level introductory computer programming using Lego Mindstorm robots
- Prepared students at Philip Randolph high school and The Urban Assembly Maker Academy for the 2017 RoboCup Junior competition. Assisted students in building competition robots
- Implemented Lego Mindstorms hardware training for new hire peer mentors, instructed peer mentors on how to use drag and drop programming software to deliver lesson plans
- Presented progress of students to program directors during weekly meetings

PROJECTS

The Way Back Machine Online Clothing Store

- Used MongoDB to create a database of product listings.
- Used React.js to build a user-friendly interface that reflects the constantly changing list of available products in the database and allows the user to add and remove items from their shopping cart

Ghost Multiplayer Word Game Web Application

- Used Node.js and Express to create an API that searches through the Collins Scrabble Words dictionary, to check if the user input is a word or can become a word
- If the user input is a word, the definition of that word is returned to the frontend, where it is displayed using React.js.
- Made an API request to check user inputs and validate whether or not an input is a word. If true, the API would return the definition of the word which was rendered in the React.js application.
- Used Socket.io to emit messages between the players when the game score or turn changes, in order to keep the game state consistent among the opponents.