John C. Egnatis

(214) 537-1564 * jce180001@utdallas.edu * Richardson, TX

https://www.linkedin.com/in/john-egnatis * https://github.com/chrisegnatis

EDUCATION

UNIVERSITY OF TEXAS AT DALLAS
GPA 3.934
Bachelor's Computer Science (Graduate 2023)

L'SPACE NPWEE ACADEMY
Spring 2022
Academy Run by NASA to train collage undergraduates to be able to write effective proposals

SKILLS

Programming languages: Java, JavaScript, MIPS, HTML, CSS, Python, C/C#/C++
Technical Skills: Visual Studio Code, React.js, Node.js, .NET Framework, Microsoft Office
Coursework: Computer Architecture, Data Structures and Algorithms, Computer Science 1, 2, & 3, Probability and Statistics, Software Engineering, Systems Programming in UNIX and Other Environments, Advanced Algorithm and Design Analysis, Discrete Math

PROJECTS

Portfolio Website .... *January 2022*
Developed a web application to act as a personal portfolio using the React.js framework. The UI was designed with the help of the tailwindcss package, and the website was deployed using on GitHub Pages. Routing, conditional rendering, and components were utilized. The link to my portfolio can be found here.

Resume Filter *(Hackreason 2022)* .... *January 2022*
Developed an AI solution with my team to assist companies in automated resume filtering. Based off certain credentials, the solution was able to decide who deserves a chance at an internship. The solution was created in Prolog and further tested and displayed with the help of s(CASP). My team's solution was among the finalists at Hackreason 2022.

Falling Asteroids 2D JavaScript game .... *December 2021*
Developed a 2D JavaScript canvas game where a space rover must collect several valuable stones while avoiding projectiles. The JavaScript source code implements random generation of objects, smart hitboxes with buffers, and object-oriented programming. The game is rendered through HTML and a demonstration can be found on my portfolio.

Computer Science Calculator in Assembly Language .... *November 2021*
Programmed a project in MIPS to engineer the process of converting numbers from one base system to another. Good coding practices for assembly language were used such as protecting the scope of variables, creating meaningful procedures, support for error handing and input variation, and strong documentation. The calculator supports up to 32-bit calculations (2s complement) and currently supports binary, decimal, and hexadecimal calculation.

LANGUAGES

- English (native)
- Spanish (learned to conversational fluency)