# Justin Aguirre

Email: jw.aguirre@rice.edu | LinkedIn: jw.aguirre Website: jwaguirre.xyz | Github: jwaguirr

## EDUCATION

**Rice University |** Computer Science
Houston, TX | Class of 2027

**Southwest Texas Junior College |** Associate Of Science/Associate of Arts
Uvalde, TX | Class of 2023

---

## EXPERIENCE

**Steel Dynamics,** *Software Engineering Intern*
Sinton, Texas | May 6 – August 16 2024

- Using Python, React, and SQL- I focused on enhancing the efficiency and accuracy of coil production monitoring. My primary project involved developing and refining tools to better visualize over 1,000 points of measurement data from PLC's, which helped streamline the process of tracking production quality.

**Vestaboard,** *Software Engineering Intern*
San Francisco, California | June 22 – August 6 2022

- Using Python, Flask, React, and other various platforms–I was responsible for developing an installable feature for over 10,000 Vestaboard+ users that dynamically displays and updates the board like a clock in real-time. This project required me to ensure seamless integration with the existing system while maintaining optimal performance and user experience.

---

## SKILLS & INTERESTS:

**Skills:** Virtualization, Server Deployment, Python, React, Javascript, Java, SQL

**Interests:** Software Engineering, Backend Engineering, Object Oriented Programming

---

## PROJECTS:

**Metal Sniffer** — *Rust, Tauri*

- A macOS packet sniffer with real-time traffic filtering, protocol inspection, and source-destination grouping. Built with performance and visibility in mind.

**Digital A* Live Search** — *MongoDB, Python, Typescript, Next.js*

- Built for Rice University COMP 140 course, this web-based A* pathfinding system uses QR-code navigation and color-based group routing with real-time route validation.

**Dubtrack** — *PostgreSQL, Docker, React, Tailwind, Next.js, NGINX*

- A URL shortening platform with integrated QR code generation and real-time analytics. Dubtrack tracks scan metadata including timestamp, geolocation, and user-agent fingerprinting to provide detailed usage insights..

**Reed-Solomon Error Correction for QR Codes** — *Python*

- I implemented a Reed-Solomon error correction algorithm for QR codes by manipulating polynomials over $Z_{256}$ arithmetic. I created message and generator polynomials and applied polynomial division to generate the error correction code, successfully encoding the message with the required redundancy.