
Education

Bachelor of Science in Electrical and Computer Engineering
University of Texas at Austin

May 2020

Experience

Software Engineer at Saronic Technologies in *Austin, TX (On site)*

Sep 2023 - Present

Focused work on Saronic Echelon Command and Control (C2) software frontend and backend with additional work on boat software.

- Rewrote entire UI in **TypeScript** using **React**, **styled-components**, **MapBox** and **Three.js** to implement a modern designs.
- Implemented a feature across the stack from boat software to C2 backend to C2 frontend which allows the user to change which radios of the boat are active for a segment of the mission, with robust error handling and recovery.
- Design and implement dataflow in frontend to render hundreds of active assets on the map and allow the user to control all of them at once or individually.
- Wrote playwright integration tests running with our simulation environment to ensure the most of the stack from C2 frontend/backend to object tracking and sim boat mission execution works as expected.
- Rewrote the backend from **JavaScript** to **Rust** and **SQLite** to improve code explorability (go to definition), developer iteration time and runtime speed and memory usage (12MB for rust vs 48MB each for 4 js services).
- Utilize AWS services:
 - **EC2** - instances, VPCs, subnets, security groups and Amazon Load Balancer for separating services and routing traffic from clients to serve the frontend and API requests to the backend running in EC2 instances
 - **S3** - storing logs of data sent over the bus in the boat
 - Automate workflows using **Python** scripts utilizing **boto3** to call AWS APIs

Backend Developer at Amazon/Whole Foods in *Austin, TX (Hybrid)*

Oct 2022 - Aug 2023

Work on Whole Foods Merchandising/Selection and Team Member Shift Management

- Work with a team of 5 developers in 2 week sprints to create new services for merchandising and selection and maintain existing services supporting team member shift management and respond to alerts or outages when on call.
- Design and implement new services using **Java** for services involving planograms (shelf layout and arrangement in each store):
 - Catalog export service to generate data imported to planogram design software
 - Planogram import service to import planograms from design software
 - Planogram compliance service to check real world planograms against the expected planogram
- Create and maintain unit tests using **Junit5** and **Mockito** to ensure 90% branch/statement/function coverage of production code at all times
- Automate regular workflows and triaging of alerts using **Python** scripts utilizing boto3 to call AWS APIs
- Utilize AWS services like **Lambda**, **Fargate/ECS**, **AppSync**, **API Gateway**, **S3**, **DynamoDB**, **SQS**, **SNS**, **CloudWatch** **CDK/CloudFormation**

Fullstack Developer at Spreetail in *Austin, TX (Remote)*

Jun 2021 - Sep 2022

Worked on Spreetail vendor portal microservices

- Worked with a team of 5 developers in 2 week sprints to fix, update, and build new and existing microservices to replace functionality from the original monolithic application and responded to alerts or outages when on call
- Created unit tests using **Jest** and **React Testing Library** to ensure at least 85% branch/statement/function coverage of production code at all times
- Implemented REST API endpoints as **Azure** functions for actions like file uploading, account setting modification and sales/invoice data querying ...etc
- Sped up development and improved readability of new endpoints by consolidating common Okta authentication/authorization, body parsing and validation into middleware.
- Wrote database queries using **Sequelize** and **Prisma** ORM
- Created a common report API across various report frontends (Excel, CSV, HTML tables) and created an implementation to output an Excel file to avoid column mismatches across report frontends.
- Migrated microfrontends from **single-spa** to use **Webpack 5** Module Federation.
- Spearheaded integration testing and process automation by building demo integration tests in **Cypress**, **Puppeteer** and **Playwright** resulting in choosing **Playwright** for future integration testing.

Fullstack Developer at LeanDNA in *Austin, TX (Remote)*

Mar 2020 - Jun 2021

Worked on the LeanDNA ERP analytics application.

- Worked with a team of 10 developers in 1 week sprints to deliver bug fixes and features to users of a very data heavy analytics application.
- Implemented a REST API with authenticated with API keys that allow programs to interact with the application on users' behalf. This required writing **SQL** migration scripts, **Java** backend API endpoints and new **SQL** queries for the backend to access the database as well as a new API management UI written using **TypeScript**, **React**, **Redux**, and **Material-UI**
- Added new reports to provide customers with further insights into their item demand and inventory data. This required adding new Angular controllers in **JavaScript** in addition to creating new API endpoints and SQL queries to provide the report data to the frontend.
- Implemented several new workflows for users in the frontend in **TypeScript** and **React** using **Material-UI** and **Redux** requiring integrations with the old **Angular** frontend. I implemented requirements specified in Figma, Jira tickets, and Confluence pages into the application very few bugs making it into production.

Software Engineer at X-Microwave in *Austin, TX*

Feb 2016 - May 2019 Aug 2019 - Mar 2020

- X-MW Controller (on Raspberry Pi 3 B)
 - Created and developed software in **Python 2.7** using the **Kivy** GUI library to provide a touchscreen interface to use GPIO pins to control various RF parts
 - Replicated evaluation boards (\$3000 boards) and software with the X-MW controller and X-MW blocks (\$300 boards)
 - Enabled the QA department to automate testing of physical parts with the controller software through an alternative TCP network interface
 - As the main developer for the project I maintained consistent code style (PEP8), and large test coverage with unit tests and integration tests to enable easier transition to other developers at the company and greater confidence that code changes have not caused regressions
 - Provided support to both external customers as well as users within X-Microwave
 - Ported to **Python 3.6**, and a newer version of Kivy to take advantage of newer features and better support
- X-MW CASCADE (Layout and simple simulation web application written in **JavaScript**, **LESS**, and **HTML**)

- Added a lot of functionality to an existing Layout tool to allow customers and employees to quickly layout designs and gather simple performance characteristics
- Created a REST API server to provide data to the Layout tool from the internal database so that the tool is up to date with the newest parts
- Added advanced convenience features like the following: auto placement of walls, auto placement of GSGs and anchors, and auto placement of control boards required by RF boards
- Added unit tests (using **Mocha** and **Chai**) and integration tests (using **Nightwatch** browser automation library) to check for and avoid regressions
- Greatly increased sales of X-Microwave as well as X-Microwave quoting capacity to meet those sales by enabling customers to understand what parts they need for their applications.
- Used GitLab CI/CD to setup automated deployments whenever a merge to the master or staging branch is made
- Leveraged AWS CloudFront, S3, and EC2 to deploy the frontend and backend applications separately

Research & Development Intern at TrendMicro TippingPoint in *Austin, TX*

May 2019 - Aug 2019

- Worked in C on TippingPoint's fork of OpenSSL to remove a copy operation to improve performance of SSL inspection
- Integrated `ssl_read_zero_copy`, and other buffer allocation functions that I created in OpenSSL, into the TippingPoint SSL inspection code path
- Learned a lot about OpenSSL library architecture and how to use GDB effectively to debug embedded applications
- Worked with a team of other interns on an internal web app tool in **JavaScript**, **CSS** and **HTML** using **React.js** and **MongoDB**. This tool helps developers search and find commits in the version control history in github as well as in the user story and bug tracker in Jira.

Skills

Programming/Markup/Styling Languages	Web Frameworks and Libraries	Other Desktop or native libraries
• C++	• Docker	• Boost C++
• C#	• Express.js	• OpenGL/WebGL
• CSS/LESS/...etc	• Flask	• Kivy & Tkinter
• HTML/SVG/XML	• Kubernetes	• SDL1.2 & SDL2
• Java	• Material-UI	• Unity (game engine)
• JavaScript/TypeScript	• Mocha.js & Chai.js	
• Python 2 & 3	• NGINX	
• Rust	• Node.js	
• Solidity	• React.js	
• SQL (MariaDB/MySQL)	• Selenium/Nightwatch.js	

Selected Independent Projects (to see more of my projects visit <https://github.com/IsaacSteadman/>)

C++ Compiler and the StackVM Virtual Processor Architecture

<https://github.com/IsaacSteadman/IsaacCompiler>

- Wrote a **Python 2.7** application to compile **C++**
- Wrote the Lexer/Tokenizer, then the parser and context container, then the code generator
- Created StackVM, a virtual stack based 64-bit architecture for the compiler to target, to learn how processor instruction sets are architected, then later added support for virtual memory and interrupts to StackVM
- Thoroughly documented the ISA of StackVM with a collection of HTML documents
- Translated the entire project to **Python 3.6** and made the code PEP8 Compliant
- Increased my knowledge and appreciation for the language of **C**, and **C++** in developing this compiler
- Learned a lot about developing and debugging a large project
- Other Notes/Continued Work
 - working on this project further by developing an operating system for StackVM and a C++ implementation of StackVM
 - researched how to implement StackVM as an Out of Order, Pipelined, Register/Stack Renaming CPU in Verilog
 - learned a lot about how to implement this for a stack based system since most literature is for register based ISAs

Music Player

<https://github.com/IsaacSteadman/MusicPlayer3>

- Wrote a music player desktop application using **Pygame**, **Python**, and **C++** that has features lacking from other music players
- Integrated my **C++** Utility library to make multithreaded buffering available to **Python** to stream music over the internet
- Learned how to read MP3 metadata, learned how impactful network latency can be when performing file operations that depend on previous file operations
- Learned cryptographic principles to enable secure encrypted music streaming and remote file system access
- Wrote a GUI and widget library on top of pygame to ease development of this project and learn about how GUIs work

Online Password Manager

<https://github.com/IsaacSteadman/PassManJs>

- Wrote a web application entirely in **TypeScript**, **LESS**, and **HTML** using **NodeJS**, and **ExpressJS**
- Used PBKDF2 on the client side to derive 2 different keys using different salts: Authentication Key (sent to the server), and AES-256 Encryption Key (never sent to the server).
- Created a simple generic table interface so that arbitrary tables can be stored in the password vault and not just password. (I use it to store some todo lists)
- Added password generator that allows you to specify the number of bits of entropy and types of symbols you want.
- Deployed the application on my website under a subdomain with a wildcard ssl certificate using **NGINX** as a reverse proxy.
- Added "policy spaces" that can be used to limit account creation and account size to prevent denial of service attacks.
- The frontend is entirely a single page application and does not store any cookies.

Sudoku Helper

<https://github.com/IsaacSteadman/SudokuWeb>

- Wrote a web application with in **Python** using **Flask** (server side) and in **JavaScript/CSS/HTML** using **jQuery** (client side)
- Added auto generation of hints/notes whenever the board changes as well as other convenience features to solve the puzzle faster in order practice more advanced puzzle solving
- Added support for undo and redo and created script to scrape websites for new sudoku puzzles
- Rewrote as an ExpressJS application using **TypeScript** and added even more helpers

Logic Simulators

<https://github.com/IsaacSteadman/PygameLogicSim>, and <https://github.com/IsaacSteadman/PygameLogicSim2>

- Wrote a logic simulation and diagramming application with in **Python** using **Pygame**
- The first version does not have the concept of buses and the UI and simulation are highly integrated and hard to separate.
- The newer version adds buses that can link to multiple inputs and outputs simultaneously, and the simulation backend can be run without the GUI since it is decoupled.
- Used these logic simulators to help with my logic design class, and I learned a lot about proper **Python** code practices.