SHAWN BOYLE

(801) 541-6884 shawn.boyle01@gmail.com PORTFOLIO GITHUB LINKEDIN

SKILLS

JavaScript / TypeScript / Python / React / Redux / Node.js / Flask / SQLAlchemy / HTML5 / CSS3 / PostgreSQL / Docker

RELEVANT WORK EXPERIENCE

Junior Front-End Developer | *BrainStorm* April 2022 - Jun 2022

- Spearheaded **Video.js** overhaul for our product, updating vanilla **JavaScript** functionality, as well as creating **custom plugins**.
- Created performance charts and visualized statistics for partners using **Redux Toolkit, Redux Saga,** and **D3.js.**
- Replicated **Zeplin** designs from the **UX** team for our **React, TypeScript** & **Sass** frontend, communicating through **Jira** tickets.
- Migrated **Enzyme tests** to **React Testing Library/Jest**, increasing **test coverage** and maintainability.
- Upgraded to **React v16.8**, updating legacy **class component**s to **functional**, modernizing our platform for a new launch.
- Practiced **SDLC/Scrum** methodology through **2 week sprints**, communicating regular product updates to shareholders.

Junior Software Engineer | *Flapmax (Contract)* Nov 2021 - Dec 2021

- Engineered optimization for AI models via **FastAPI** for **RESTful** routes to validate the models, then running **automated tests** on **Azure Virtual Machines** to measure and optimize output, resulting in improved throughput, latency, and memory footprint, while maintaining accuracy.
- Constructed an intuitive, comprehensive **frontend UI** through **React**, allowing clients to upload and optimize models, then displaying data through rendered charts and statistics, providing clients with clarifying data and an enhanced product.
- Integrated **Azure cloud storage** services for **PostgreSQL and MongoDB** providing **SQL** and **NoSQL** storage, allowing for efficient, flexible data management for our **backend API.**

L1 Technical Support Engineer | *Instructure* May 2020 - Dec 2020

- Utilized internal Canvas technologies to troubleshoot **API** integrations into virtual classrooms by replicating bugged environments for **testing and debugging,** then communicating and applying a fix to live environments**.**
- Assisted admins, teachers, and students to resolve 80+ tickets per 20 hour weeks on the Canvas platform, by navigating a thorough troubleshooting process centralized around various bugs and common issues.

TECHNICAL PROJECT EXPERIENCE

PROGRAMMATIC | *Group Project built with **React, Redux, Flask-SQLAlchemy, AWS S3, PostgreSQL, Docker*** Live Site | GitHub

- Engineered a visualization of **Dijkstra's Algorithm** by calculating the shortest path for a weighted graph of nodes, rendering the graph and edges with **a React/CSS** grid, then animating the traversal and shortest path for an enhanced **UI/UX.**
- Programmed a fluent search algorithm with **React** to match user queries with comprehensive filtering, returning results utilizing the **useLocation hook** for accurate DOM rendering.
- Developed a seamless, responsive discussion section for users to read feedback as well as engage in discussion by creating, editing, or deleting their comments utilizing **React**/**Redux** to manipulate the front-end dynamically and efficiently.

CRESCENDO | *Solo Project utilizing **React, Redux, Node.js, Express, Sequelize, AWS S3, PostgreSQL*** Live Site | GitHub

- Developed a relational database with 12 tables held by **PostgreSQL,** which I queried on the **Node.js** backend's **Express** server with the **Sequelize ORM**, allowing server data to have clarity and accessibility.
- Utilized the audio element to construct a feature-rich, application-wide song player, giving users audio progress and extensive controls over the current playlist, album, artist, or song.
- Formed a cohesive **Redux** store through efficient **API** queries, cutting down on server load and resulting in a quick, optimal load time for data rendered on the frontend with **React**.

EDUCATION

App Academy | ***Full- Stack Web Development*** Oct 2021

- *Rigorous 1000-hour software development course with <3% acceptance rate*
- *Topics include: **TDD (Test Driven Development)**, scalability, algorithms, **OOP (Object Oriented Programming), REST API's**, security, single-page apps, pair programming, coding style, design patterns, and web development best practices.*