

Felix Alvarado

Phone 281-467-6493

felixalvarado653@gmail.com

falvarado.com

[LinkedIn](#)

[Github](#)

SKILLS JavaScript, React, , React Hooks, Redux, Node.js, MongoDB, Express, Passport, Python, Flask, SQL, SQLite, PostgreSQL, CSS3, Sass, HTML5, HTML Canvas, jQuery, Git, Bash, Socket.io

WORK SAMPLES

PokeQuiz - Full Stack Developer

[Live Site](#) | [Github](#)

A pokemon themed site that allows users to make quizzes and share them with friends. React was used for the frontend, and Flask was used for the backend.

Tech Stack: React, Redux Toolkit Sass, Flask, MySQL, Heroku, Pokemon API

- Made use of media queries and browser detection to ensure an excellent user experience on all desktop and mobile browser types.
- Utilized multi-threading to restart MySQL at a set interval and prevent connection timeouts.
- Leveraged the Redux Toolkit and local state to preserve newly created quiz data across all needed pages.
- Integrated Sass's features (such as Nesting and Mixins) to have reusable code and cleaner style sheets.
- Devised algorithms to retrieve and sort quiz data with minimal space and time complexity. These algorithms helped accomplish minimal load times for most of the app's pages.

TieBreaker - Full Stack Developer

[Live Site](#) | [Github](#)

A single-page app that contains useful tools for players to record scores and chat remotely. It uses Socket.io to let players make changes instantaneously.

Tech Stack: React, React Hooks, Node.js, Socket.io

- Used React Hooks to dynamically change state with real time data so that all users in a given room see the most updated game score.
- Collaborated with another engineer by using git workflow and an online storyboard (Mural). We pair programmed on a few tasks, and divided up the remaining features to finish the application.
- Configured an algorithm that checks every game room for active participants. Because of this, inactive rooms are deleted five minutes after the last participant exits a room.

360px - Full Stack Developer

[Live Site](#) | [Github](#)

A single-page app that allows users to upload and share photos, follow other users, and receive notifications

Tech Stack: React, Redux, Ruby on Rails, PostgreSQL, Webpack, Heroku, Cloudinary API

- Created multiple photo sorting algorithms that use follower and timestamp data to retrieve relevant photo information based on the current user and page.
- Provided a more streamlined user experience by decreasing picture load time through the use of image sizing in Cloudinary's API.
- Leveraged the local state of multiple components to feature adaptive user views that seamlessly change with follow toggling and notification updates
- Configured a user authentication with Bcrypt to make use of password hashing and promote better security for users in the database

EXPERIENCE

Technical Support Engineer - Developer Support

Twilio

January 2019 - July 2020

- Troubleshooted a variety of server-side and client-side issues in multiple environments and languages such as JavaScript, PHP, Swift, Ruby, C#, Java, and Python
- Assessed developers' use-cases and requirements in order to provide guidance on the most effective way of integrating Twilio's technologies in their applications
- Analyzed HTTP requests and data transfers to help developers pinpoint and solve problems that can occur in both their application and Twilio's SDKs/APIs
- Handled and maintains environments for most of Twilio's Generally Available products to reproduce customer issues when needed

EDUCATION

App Academy (July 2018)

Immersive software development course with a 3% acceptance rate that focuses on full stack web development.

University of Houston (May 2014)

BS - Mathematics

BS - Economics