

SETH BARRIE

P. 360-540-8629

sethbarrie@gmail.com

[LinkedIn](#)

[Github](#)

Seattle, WA

SKILLS

JavaScript, React, Redux, HTML, CSS, Ruby, Ruby on Rails, Mongoose, MongoDB, Node.js, Express.js, SQL, PostgreSQL, Webpack, jQuery, Git, Heroku, Amazon Web Service S3

PROJECTS

Dogflix

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

A pixel perfect clone of Netflix with React front end, Ruby on Rails back end, and AWS DB

- Pop out individual video players for each movie on the browse page on hover, built with React hooks and State Management from Redux for quick reaction, with querying to AWS for media storage.
- Large selection of movies to choose by using a carousel to browse different genres. Carousel built using React to dynamically display select titles from a genre list.
- Integrated movie controls over main browse movie player to mute or turn off Auto-play. Settings for Auto-play are saved to individual users across multiple browsing sessions, with React/Redux sending AJAX requests to Ruby on Rails controllers updating user preferences.

Job-plz

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

A website to aggregate searches for jobs using multiple API's. Built on the MERN stack

- Users are able to search for jobs and save individual postings to their profile, and retrieve them later to fill out. Job postings are presented in React components and sent to MongoDB using Express.js routing.
- Search component queries several APIs and parses the data into an organized format, with the user being able to save jobs from the display page or the search list. User searches are sent to MongoDB from the Mongoose router.
- The search component queries several public APIs, and sends the data back to the React component, not to the Database. This way you only query traffic when you want a posting saved, in addition to being faster not having to wait for the Database. Queries are sent from the Front end to the Mongoose router and then directly back to the React component.

Tower Climb

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

A short Javascript game that allows fluid movement inside HTML5 using vanilla Javascript

- Player has natural and intuitive interactions with the environment due to a simple but powerful collision system, with being able to jump through platforms but also walk on them. Data is sent to a collision class with maps to determine type of collision to expect from tile.
- Environment has simple but very appealing pixel art along with many styles to build from sprite templates. A castle class determines the tile in a certain spot based off of a map and key.
- User has control over the character with the keyboard, either with wasd or the arrow keys and the space bar. Players can't jump indefinitely by holding the jump button down due to a dynamic key press reader that updates only after the key was released, either with the space bar, or the arrow up.

EXPERIENCE

Lead Craftsman

AAA Kartak, Jan 2015 - June 2020

- Trained and led a team of 4 people to install \$50,000 of cabinet sales monthly, an increase by 50% over 5 years.
- Exceeded company expectations by receiving 94% positive reviews in 5+ years of 200+ customer interactions.
- Recipient of multiple awards, including Craftsman of the Quarter, and have had personal work featured in 2 home design magazines.

EDUCATION

- **App Academy** - Immersive software development course with focus on full stack web development (July 2020)