JEREMY ETZINE

JUNIOR WEB DEVELOPER

I'm a Junior Web Developer who loves the challenge of coding. In this field one can spend ages working on a difficult problem, but when one gets it right it is very fulfilling.
I am a team player, having played basketball for many years at a competitive level, I would describe myself as the supportive team member, ensuring I can fill the gaps to help my team succeed.
I have found that I love and thrive in back end development as I find it consistent and compelling. I have found great success using Ruby on Rails and it has become my favourite language to use.
What I like about coding is that it is logical and creative at the same time. When it all comes together it is very rewarding to see the final product.
I am so excited that I get to do something I am so passionate about as my career.

CONTACT DETAILS

- Mobile: 0400179807
- jeremyetzine@hotmail.com
- Sydney, NSW
- https://github.com/jeremyetzine

SPECIALIZATIONS

- Back end development
- Web coding
- API Development
I am proficient in HTML, CSS, JavaScript, Ruby, VueJS, React, NodeJS and jQuery.

PREVIOUS EDUCATION

General Assembly

Web Development Immersive

- Graduation: March 2019
- Coursework: HTML, CSS, JavaScript, jQuery, SQL, Ruby, Ruby on Rails, Axios requests, NodeJS, React

University of Queensland

- Studied: 2015 - 2018
- Coursework: Major in Criminology & Forensics, Minors in Mathematics and Psychology

Cranbrook School

High School Certificate

- Graduation: November 2014

PREVIOUS EXPERIENCE

The Island, Sydney

Bar-Back (September 2017 - Present)

- Maintained bathroom cleanliness
- Washed and cleaned glasses
- Re-stocked bar fridges and troughs

Mathematics Tutor

Freelance (March 2016 - March 2018)

- Helped high school students in grades 7-11
- Covered algebra, trigonometry, differentiation, integration
- Used proof-based teaching

Woolworths

Team Member (June 2014 - March 2015)

- Ran checkouts and self-checkout
- Opened and closed store
- Organised weekly rosters
- Ran service desk area

GA PROJECTS

Tic-Tac-Toe

Project 0 -
https://github.com/jeremyetzine/tic-tac-toe

- I created a functional in-browser game of tic-tac-toe using HTML, CSS and JavaScript. I used jQuery as a JavaScript library to simplify and shorten my code. The game allowed for two users with custom usernames to play, tracking turns and a winner/draw outcome. The game was built using a table in HTML and click functions to register the users' clicks inside the table which would then push either a 'X' or 'O' into the square.

NBA Players & Shoes Database

Project 1 -
https://github.com/jeremyetzine/Project-1

- I used Ruby on Rails to create a fully functional CRUD system that compared all NBA starters and their shoes with each other. I used CSS and HTML to show the views, with unique colour schemes for each team. My database had four models all with associations between each other. I compiled all the players and their shoes to compare which shoes had the highest scoring players. I built .a large seed file containing all my data. This was entirely a passion project for myself, fostered from a personal enthusiasm for basketball.

Opera House Website Remake

Project 2 -
https://github.com/jeremyetzine/operahouse_server

- This was a group project where my team and I attempted to redesign and rebuild the website for the Opera House. We split our team into two groups: front end and back end. I was the sole person working on the back end. I built the back end with Ruby on Rails, designing an API CRUD system to allow the front end to make collect the information. I built user-authentication on the back end as well as created custom JSON objects to help the front end when gathering the data.

Drafft - NBA Fantasy Draft

Project 3 -
https://github.com/jeremyetzine/Drafft_Server

- I built a basic NBA draft site using VueJS and Ruby on Rails. This project was very difficult and was mostly my first in-depth look into front end development but I thoroughly enjoyed it. I learnt and used VueJS within a week to build the draft site. I used a Rails API to send the data to the front end. I implemented user-authentication and custom JSON objects to minimise get and put requests used. I had asynchronous updating in the draft. I created blockers to only allow six users per draft in order to simplify the rounds process.