

RIZZA MARZO

Rancho Cucamonga, CA 91739 | (909) 904-9029 | rizzacarla.12@gmail.com | [PORTFOLIO](#) | [GITHUB](#) | [LINKEDIN](#)

EXPERIENCE

Web Developer - *Foldscope Instruments*

Apr 2021 - Present

Web App #1 (NDA) - A **Single Page Application (SPA)** that allows users to view data collected from our measurement device

- Build an **Express** server and **API requests** that enable our devices to temporarily store their IP address and device information on **MongoDB** and so that the client and hardware can create a **WebSocket** connection
- Utilize **WebSockets** in the web app to retrieve **live data** and **saved data** from a **microcontroller**
- Develop reusable **functional React components** in conjunction with **React Hooks** such as **useState**, **useEffect**, and **useRef** to create **static** and **dynamic pages**

Web App #2 - Migration of the WordPress community website into a **React Single Page Application** to allow more complex features

- Build intuitive, **cross-browser compatible**, and **responsive pages** with consistent styling using **React-Bootstrap** and **Bootstrap** to create a positive user experience in any device
- Utilizing **React Hooks**, like **useState** for reusing stateful behavior between components and **useEffect** for executing side effects in a **functional React component**, **React Router DOM** for client-side routing, and **SVGs** to actualize the mockups designed by the graphic designer
- Integrate an infinitely scrolling page with **pagination** by performing short **simple** and **compound queries** to **Cloud Firestore** every time the user arrives at the bottom of the page to prevent overloading the client

PROJECTS

Satisfy (Javascript, React, Redux, AWS (S3), Ruby, Ruby on Rails)

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

Satisfy is a Spotify-inspired web app, where users can listen to music, like songs and albums, follow artists, and discover new songs.

- Integrated **user authentication** on the backend using **JWT**, **RESTFUL** routes, **BCRYPT** hashing, and custom **validations** to ensure that the user's login/signup inputs are valid and secure and these features are accessible in the frontend
- Utilized the state management system of **Redux** to create a **queue** for the user's liked songs, albums, and artists, which allows users to move to the next or previous song, shuffle their playlist or listen to a playlist continuously
- Built a search feature that calls multiple **APIs**, which parses the search query with **regular expressions**, to find partial or exact matches and used **SQL** commands to sort songs, artists, and playlists available within the **Amazon S3** storage service

Melting Pot (ReactJS, Express.js, Axios, Mongoose.js, Node.js, MongoDB)

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

Melting Pot is a recipe-sharing social media web app that mixes food lovers, culture, and history together.

- Designed a **wireframe** using Whimsical to provide structure to our **user flow**, elaborate on the **UI visual design** choices, and organize **UI** components necessary for our website
- Created the backend **routes**, **models**, and **validations** using **Express** and **Mongoose** for 4 features: liking, commenting, recipe creation, and event organization to allow users to interact or store data in the **MongoDB** cloud database
- Constructed custom **validations** for all the **RESTful** routes to ensure that the inputs received by the backend were in line with the **constraints** of each features' **model**

SKILLS

Programming Languages: JavaScript, Ruby

Development Frameworks: React, Express, Ruby on Rails

Structure & Styling: Bootstrap, React Bootstrap, HTML, CSS

Other Skills: Node, MongoDB, Heroku, Firebase, Axios, JBuilder, React-Redux, SQL, PostgreSQL, Amazon S3

EDUCATION

Software Engineering - App Academy | 2020 - 2021

BA Psychology - University of San Francisco | 2015 - 2018