

Randy Reyes

Software Engineer



rqreyes@gmail.com



linkedin.com/in/rqreyes



rqreyes.com



858-348-7017



github.com/rqreyes

Technical Skills

Strong | JavaScript (ES6+), React, Redux, jQuery, Node (Express), SQL (PostgreSQL), REST/GraphQL API, HTML, CSS, Sass
Experienced | NoSQL (MongoDB), Webpack, Docker, AWS (EC2, RDS, EBS), Travis CI, TDD (Jest/Enzyme), OAuth, YAML

Experience

Codesmith | Software Engineer

Nov 2019 - Feb 2020

- ⇒ Executed React and Redux testing in Jest and Enzyme to facilitate continuous integration and deployment, ensure product reliability for every future release, and minimize unexpected outcomes by evaluating critical edge cases
- ⇒ Updated content in YAML by uploading profile information and technical blog on GraphQL resolvers while capitalizing on its human-readable format, flexible translation into multiple language types, and resistance to delimiter collision
- ⇒ Mentored a team of engineers in building a GraphQL query testing node module to streamline test-driven development
- ⇒ Conducted interviews on fundamentals, communication, and best practices to uphold a cohesive and accelerated environment

React Blue | Software Engineer

Sep 2019 - Feb 2020

- ⇒ Leveraged React-Redux to construct an open source React prototyping and templating tool by centralizing state across multiple components to set the foundation for scalability and to simplify the testing process with Redux's time-travel debugger
- ⇒ Utilized React and D3 to engineer and visualize an interactive diagram of the component tree by separating control of the DOM to emphasize parent/child node relationships and to streamline the manipulation of component properties
- ⇒ Devised client-side routing by using React Router to mimic the feel of an MPA in an SPA, minimize server calls, and keep the UI in sync with the URL by dynamically matching endpoints and rendering their respective control panel component
- ⇒ Implemented React Hooks, used local storage, and deployed with Electron for easy access to the cached component tree data and to enable the option to deliver fast performance in a local and offline cross-platform compatible product
- ⇒ Employed Thunk to modularize functional features, enable asynchronous calls, making them accessible throughout the app

Granicus (acquired Vision Internet in 2019) | Front-End Developer

Apr 2017 - Aug 2019

- ⇒ Spearheaded numerous HTML, CSS, and jQuery conversion projects involving building, maintaining, and developing responsive features on government websites while complying with web accessibility standards
- ⇒ Refactored proprietary template constructed in Angular by meeting back-end requirements defined by the team

Ceatus Media Group | Web Developer

Mar 2015 - Jul 2016

- ⇒ Administered content management systems built with WordPress by manufacturing blog templates and applying robust security features, abstracting code, and granting customization of content and design for clients in the medical industry
- ⇒ Utilized HTML, CSS, and SEO practices to implement both visual hierarchy and responsive design, while boosting the traffic of the targeted audience through organic search results by configuring an internal linking system and highlighting LSI keywords

Open-Source Projects

Dinder | Location-based restaurant matchmaking application

- ⇒ Architected a React single page application by leveraging its reusable component features, simple state management capabilities, and tree reconciliation, along with asynchronously fetching data from Yelp's API
- ⇒ Stored persistent and relational data with SQL (PostgreSQL) to perform predictable queries for user information and other sensitive data while capitalizing on SQL's ability to maintain "all or nothing" transactions and strict data integrity

AppliCoder | Employment-related search engine for software engineering job listings

- ⇒ Created a GraphQL job posting interface by querying from GitHub and Authentic Jobs's API to separate concerns between client-side data requirements and server-side data structure and resolution and to abstain from over-fetching information
- ⇒ Employed noSQL (MongoDB) and OAuth to contain user information and preferences due to MongoDB's quick and efficient ability to manipulate document instances and OAuth's feature of decoupling authentication and authorization

Trail Mix | Hiking trail directory service and crowd-sourced review forum

- ⇒ Developed a robust Node server and leveraged its fast, flexible, and minimalist design to orchestrate routing of user data through the request handling pipeline between the client-side application and relational database
- ⇒ Integrated Sass functionality like variables, nesting, and partials to benefit from their reusability and maintainability features

Education

Codesmith | Software Engineering & Computer Science

2019

University of California San Diego | Web Publishing Certificate

2015

San Diego State University | Bachelor of Science in Bioengineering

2011

Interests

Producing electronic music, playing piano, breaking (breakdancing), snowboarding, and taking long walks to the fridge