Carlos Botero-Vargas

SOFTWARE ENGINEER

cbotero-vargas@outlook.com
(517) 927-0518
github.com/Carlos-BoteroVargas
www.linkedin.com/in/carlosb-v

TECHNICAL SKILLS

Strong: JavaScript(ES6+), React(Router, Context API, Hooks), Redux (+ thunk), Node.js, Express.js, Axios, Webpack, SQL(Postgres), NoSQL(MongoDB/Mongoose), HTML, CSS/SCSS, jQuery, Version Control(Git, GitHub), OOP and Functional Programming, Material UI, Bootstrap 4, GraphQL, Electron Framework, Relay.

Experienced: Passport.js, TDD(Jest, Enzyme), RESTful Architecture, Authentication(OAuth, JWT, BCrypt, Auth0), D3.js, Recoil, Websocket

EXPERIENCE

Peach QE (OS Labs) | *Software Engineer* *2021 - PRESENT*

- Employed **React fiber** algorithm to create an interactive time travel debugging tool for React applications in parsing through the host fiber node and caching a copy of the 'tree' data structure using a recursive depth-first-search algorithm to store snapshots of incoming data each time that the state is altered allowing developers to better understand application performance in enabling dynamic re-rendering and performance data visualization.
- Implemented **React.js** container pattern by creating reusable components that utilized the diffing algorithm and tree reconciliation while maximizing maintainability and modularity by separating stateful and presentational components and enforcing unidirectional data flow.
- Utilized **D3** and **VisX** reusable low-level visualization components to generate intuitive visualization to display application architecture in a hierarchical tree structure, component mapping with dendrograms and performative stacked bar charts in order to improve developer debugging experience by providing a meaningful and dynamic data-driven representation of application state, cache, and user mutations.
- Leveraged **Chrome API** implementing Pub-Sub patterns updating components to generate and display state snapshots at each update.
- Used **Test Driven Development** (TDD) with **Jest** and **React Testing Library** that relied on React fiber nodes to both enforce accurate implementations of functions through unit testing and discern unwanted side effects making development experience more maintainable.
- Incorporated **Sass** to decrease redundancy with variables and write distinct stylesheets for different React components to be preprocessed to allow for only one HTTP request to load styles to ensure a consistency of style throughout the entire application.
- Utilized **Webpack** to run all assets in the application to bundle files and dependencies, minimizing file size to improve application load and render times as well as leverage hot module reloading to improve development workflow by instantly updating the extension when modifications are made in the source code and saving valuable development time by only updating what has been changed.

OPEN SOURCE PROJECT

OneNumber | *Software Engineer*

- Utilized **BCrypt** to protect user data in building a password security platform that ensures sufficient encryption in hashing every password with native salt round method to make the application resistant to brute-force attacks and enhancing application's user data protection.
- Implemented **React Context API** coupled with **React Hooks** providing an alternative to writing class-based components to easily handle state management within functional components and to optimize state management and improve the code readability reducing prop drilling and preserving data consistency and accessibility throughout the multi-layered React component tree.
- Employed **Google Material Design** using Material-UI React component library to effectively stylize the application and with the use of subtle skeuomorphism making it more intuitive for the user by using familiar semantic design patterns.
- Used **PostgreSQL** database to have customizable and relational data stored in order to persist data, support simple CRUD operations based on user input, employed one-to-many and many-to-many relations allowing efficient scalability while preventing data redundancy and being ACID compliant.

SleepRX | *Software Engineer*

- Set up **Google OAuth 2.0** with **Passport.js** in order to leverage a third party service for authentication security and create a convenient and secure login process to validate a user's identity minimizing interaction with user credentials and persisting user sessions between visits.
- Utilized **React Router routes** to enhance application's performance and user experience by providing a SPA handling dynamic routing and protecting routes depending on authentication status and reducing HTTP server calls.
- Used **PostgreSQL** database to have customizable and relational data stored in order to persist data, support simple CRUD operations based on user input, employed one-to-many and many-to-many relations allowing efficient scalability while preventing data redundancy and being ACID compliant.

- Built a **Node.js** server with **Express** framework to enable dynamic routing and facilitate real-time data transmission by customizing middleware to modularize and update relational database upon user interaction, handling concurrent requests and connecting the frontend and backend by leveraging the scalable and event-driven non-blocking I/O model of the Node.js runtime.

Adventure **|** *Software Engineer*

- Set up **Google OAuth 2.0** with **Passport.js** in o

EDUCATION

Escuela Reina sofia, Madrid | 2004 (Master with Honors) **Michigan State University |** 2014 (DMA, ABD)

PUBLIC TALKS

React Routers v5-v6-Alpha, React.js Framework | *SingleSprout Speaker Series* *2021*

INTERESTS

Latin American Literature | Creating interactive and educational concerts | Learning about new cultures via food, music, and games | Traveling and discovering new places | Watching Champions League Finals | Origami | Woodcarving | Hiking |