DENNIS LOPEZ

dnnis.lpz@gmail.com | (805) 760-6358 | github.com/DennisLpz | linkedin.com/in/dennis-lopezsb/

TECHNICAL SKILLS

STRONG: JavaScript (ES6+) | React (Router, Context API, Hooks)| Redux | Node.js | Express | Relational/Non-Relational Databases | Webpack | Babel | VCS (Git/Github) | OOP & Functional Programming | RESTful API | HTML/CSS/Sass | TDD (Jest, Enzyme, Cypress, SuperTest, Puppeteer) |Authentication/OAuth
EXPERIENCED: Angular|AWS |TypeScript| Docker | YAML | jQuery |D3 (Visx)| Websockets |Bash| Travis CI/CD | YAML |GraphQL |Agile

PROFESSIONAL EXPERIENCE

REACTIME (OSLABS) | REACT OPEN SOURCE AWARD 2020 | 1,200+ GITHUB STARS

Software Engineer 2020 - PRESENT

- Reengineered React Fiber's change detection algorithm to recursively travel through React Fiber's singly linked list to cache a copy of the React "tree" data structure thus adding functionality to store snapshots containing application's state change and rendering performance of each component
- Implemented a time travel feature using tree recursion algorithm, enabling developers to revert to previous states in the application to assist in debugging state
- Leveraged Visx to construct data-driven comparison bar graphs, enhancing the capabilities of the app to analyze performance differences between sessions
- Introduced metrics dashboard with vitals such as Time to First Byte and Largest Contentful Paint, leading to comprehensive insight into overall app performance
- Assembled React Hooks and Context API on new visualization tools to dispatch user's actions and update both the central state management store and Chrome local storage, streamlining the data to update target application's snapshot view with replicable and predictable updates in local state
- Facilitated React Router to consolidate multiple features into accessible navigational routes, creating a more responsive single page application and enabling opportunities for improved UI/UX effects such as CSS indicators for web metrics statistics and dendrogram and bar graph augmentation
- Harnessed Chrome API to scale application's analytical capabilities by storing performance times and state snapshots as objects in Chrome's local storage to provide users with access to historical performance data, subsequently minimizing application file size and complexity by removing a need for a database
- Enhanced compatibility with target React applications that utilize class and functional components by identifying target application's pattern of storing state data and injecting a filtering system that identifies different component's state management
- Incorporated TypeScript strict types on data visualization and web metric dashboard components to ensure consistency of data type usage across the codebase, reducing the risk of potential bugs and type-related errors during run-time on the client-side of the application
- Reinforced AGILE development methodology as a structured approach to collaborative team effort, developing concise solutions through 2-day sprints
- Led Test-Driven Development (TDD) across new application features using Jest and React Testing Library to conduct unit testing on bar graphs components

MOVABLE INK

Client Experience Associate 2019 - 2020

- Adapted client's data sources from REST APIs and SQL databases to generate custom data-driven application, streamlining client workflow and increasing ROI
- Applied JavaScript, HTML, and CSS to build customized frontend web apps that maintain customer behavior trackers to measure KPIs and client campaigns
- Developed distributed real-time data processing system, providing clients with an API for real-time access to customer profiles and campaign performance

AXIA MED

QA Engineer *(Provisioning Specialist)* 2017 - 2019

- Architected and managed 130+ monthly QA sessions that involved dev environments, to staging, to production
- Improved efficiency of development cycle by building automated QA testing suites using HTML/CSS/JS widgets for team to submit feature completions

OPEN SOURCE PRODUCTS

HELPDESK-PRO

Software Engineer 2020

- Bolstered React.js to create SPA with reusable UI components using a design pattern that enforces unidirectional flow of state and maximizes modularity by separating stateful vs. presentational components thus improving readability of the codebase and scalability due to the wide range of packages tailored for React
- Deployed React Router and Context API to scale application's UI/UX by accurately rendering subscribed components through routes and conditional rendering
- Implemented session-based authentication cookies to persist user's session in application, simplifying user authorization flow, minimizing bounce rate, and ensuring users have access to their pending tickets upon successful log-in
- Instituted Node.js and Express to architect a RESTful API service, allowing for the use of custom middleware that controls user credential in SQL database, user sessions, and ticketing system data through a non-relational database
- Enhanced password security using Bcrypt by incorporating salt rounds and one-way hashing prior to storage in SQL database and for password verification upon subsequent login attempts, boosting resistance against rainbow table and brute-forced search attacks, thereby complying with ACID standards
- Incorporated Sass to create consistency across style sheet files by using its variables feature to store reusable application brand colors and to nest styles for each selector improving code readability through the organization of style sheets
- Built Webpack to bundle JSX, Sass, and JS files on modularized app components and dependencies, reducing file size, optimizing application performance, and improving developer system with hot module replacement

PLAYERLYTICS

Software Engineer 2020

- Structured application's state management through React Hooks to streamline data across components, decreasing code redundancy and improving state logic
- Expanded app functionality using NBA API, enabling client-side queries to interface's endpoints, fetching and rendering data such as real-time player statistics
- Architected Node.js and Express to build RESTful API to set up server requests and data translations, allowing for server code modularity and flexible SQL queries
- Enforced Axios HTTP request library to improve the multiple calls made to each API endpoint to bring a pool of information for each player, improving the scalability of the application by giving developers the ability to easily make multiple requests concurrently and implement fetch request logic

PUBLIC TALKS

ALTERNATIVE FRAMEWORKS - **ANGULAR |** *SINGLESPROUT SPEAKER SERIES*

EDUCATION

B.A. | UNIVERSITY OF CALIFORNIA, SANTA BARBARA | DEC 2016

INTERESTS

Reenacting every new goal celebration in FIFA |Brazilian & Japanese Hip-Hop | Getting freaked out by Black Mirror episodes | Game nights with Risk & Catan|Go to one random Meet-Up event a month |Making memories with my camera one snapshot at a time