

# Software Engineer

## Erick Melendez

**Contact Info:**  
(213) 308-7039  
[erick.cali@protonmail.com](mailto:erick.cali@protonmail.com)  
[github.com/mlndz-la](https://github.com/mlndz-la)  
[linkedin.com/erick-m](https://linkedin.com/erick-m)

### Experience

#### **Artemis** | Software Engineer | **OSLabs**

2020 ~ 2021

- Utilized **React** to create reusable components and leverage the reconciliation algorithm in order to re-render parts of the DOM that have changed
- Implemented **React Context API** to consume and provide a single source of truth, simplify the data flow between components and maintain a cleaner codebase by removing needless prop drilling from stateless components
- Outlined a modern **Neomorphic UI** with **CSS Grid** and leveraged **CSS Flexbox** to create a responsive, mobile-first UI regardless of screen size
- Selected **Aleph** to build a dynamic React Application and pre-render the application server-side at build time before displaying to the client for quicker initial load times as well as update Artemis, through Hot Module Reloading, during development without having to restart the server on every change
- Integrated **Chart.js** for its minimalistic design and small module size
- Architected a **Deno** server to produce a lightweight application with Deno's module system and safeguarded Artemis by utilizing its security measures to prevent unwanted script injections or file access from modules at runtime
- Applied **Typescript** across the stack to create a more predictable code base through explicit type definitions and greatly improve code maintainability by decreasing type casting errors

#### **Papyrist** | Software Engineer | **OSLabs**

2020 ~ 2021

- Incorporated **Styled-Components** to leverage Automatic Critical CSS benefits by having the client render only what is necessary and further separate CSS styling and Javascript logic within React Framework
- Deployed **React Router** to enhance UX by reducing unnecessary full-page DOM re-renders and create a smoother transition between pages
- Constructed a **Node.js** and **Express.js** backend to facilitate multiple requests with Node's non-blocking asynchronous request handling

#### **ChatterBox** | Software Engineer | **OSLabs**

2020 ~ 2021

- Adopted **SASS** to improve extensibility and readability of stylesheets and ensure that the design of each component remained modular and independent
- Overhauled the **PostgreSQL** database to be normalized to the 3rd form and remove unnecessary joins when querying
- Encrypted user data with **BCrypt** to keep sensitive user information secure and prevent brute force or rainbow table attacks via randomized salt rounds

#### **PawFinder** | Software Engineer | **Healthy Spot**

2018 ~ 2019

- Overhauled application with **React hooks** to share logic between multiple components using custom hooks and to reduce bloat and increase readability for future developers
- Connected a **MongoDB** to achieve high-speed query responses and lessen the need for complex queries by storing related data in single documents while using **Mongoose** to communicate with the database to ensure database schemas remained consistent when manipulating documents
- Configured **Webpack** to utilize hot module reloading for faster debugging and to enable the help of loaders, such as Babel, to transpile ES6 into ES5 Syntax

#### **PawAcquisition** | Data Engineer | **Healthy Spot**

2018 ~ 2019

- Accompanied brand ambassadors to collect data on people we meet on the streets in an **Excel Spreadsheet**
- Mapped out statistically relevant data within **Python** for brand ambassadors to better target customer acquisition
- Designed a layout with **Javascript**, **HTML**, and **CSS** to better visualize statistical data

### Technical Skills

#### **Languages ~**

JavaScript (ES6+)

TypeScript

Python

#### **Frontend ~**

React (Hooks, Context API)

React Router

Redux.js

CSS / SASS / Less / SCSS

Tailwind CSS

Styled Components

HTML

Electron

Aleph

#### **Backend ~**

Webpack

Node.js / Express

Deno / Oak

Websockets

OAuth 2.0

Auth (cookies, sessions)

#### **Databases ~**

PostgreSQL

MongoDB

GraphQL

#### **Testing ~**

Jest

Enzyme

Supertest

#### **Workflow ~**

Git / Github

VS Code

Agile

Jira

Scrum

Figma

### Education

**SMC ~ A.S. – C.P.**

Computer Science

**Codesmith**

Advanced Software

Engineer Residency

### Interest

**Museum Docent**

~ NHMLA

~ LBTP

**Bird Lover**

~ House Sparrow

~ Lyrebird

~ Blue-and-Yellow Macaw