

Tommy Li

Fullstack Software Engineer | Los Angeles, CA
(415) 819-8043 | tommylidev@gmail.com | [LinkedIn](#) | [Github](#) | [Portfolio](#)

Technical Skills

JavaScript, React, Redux, Node.js, Express.js, SQL (PostgreSQL), NoSQL (MongoDB), GraphQL, Apollo, HTML/CSS/SASS, Authentication (bcrypt, JWT), Docker, TypeScript, AWS, Webpack, Firebase, Gatsby, Netlify, Git/Github, D3/Chart.js, Chakra-UI, Material-UI, jQuery, PHP, Bootstrap, TDD (Jest, Supertest), Github Actions, Railway, Heroku, WordPress

Experience

Software Engineer | Peconea Consultancy

Mar 2021 - Present

- Worked with clients to understand their needs and designed and built websites that align with their visions
- Leveraged **React**, **HTML/CSS**, and **Gatsby** to build responsive and performant websites that scale
- Handled RESTful and GraphQL API service requests with **Node.js/Express.js** and **GraphQL** backend
- Worked with modules like mongoose and node-postgres for Node.js to interact with **MongoDB** and **PostgreSQL**
- Set up continuous deployment (CD) pipeline and hosting for multiple websites using Netlify
- Implemented Mailchimp to handle the mailing lists for advertisement

Student Front End Developer | Cal Poly San Luis Obispo

Jan 2019 - Aug 2019

- Worked alongside the UI/UX developer for Cal Poly's Administration and Finance Department (AFD)
- Created dynamic forms with **JavaScript**, **jQuery**, and **PHP**

Web Developer | Canterbury Gardens & Gifts

Jan 2018 - Sep 2018

- Built website with HTML/CSS, JavaScript, and a NoSQL database that are ideal for large amount of store items
 - Managed a database of SKUs for eCommerce website for brick-and-mortar Christmas store
 - Used MailChimp to set up mailing list for advertising and marketing purposes
-

Projects

Kensa (Open Source) | [Site](#) | [Repo](#)

- Developed GraphQL API monitoring app that allows developers to observe and test custom GraphQL APIs
- Implemented a **GraphQL IDE** using Monaco Editor to allow users to test GraphQL queries, providing a one-stop testing and data-monitoring playground to speed up development process for GraphQL schemas and resolvers
- Created an npm package for app users to easily establish connection between Apollo server and a SQL database through an API key to collect and store query metrics of all requests that are sent to custom GraphQL API endpoints
- Used **Redux Toolkit** to allow React components to subscribe to a central store that holds a moderate amount of states
- Leveraged **TypeScript** auto type-checking functionality to minimize bugs and improving application scalability
- Integrated **Apollo Server** to an Express backend to communicate with SQL database and provide a self-documenting GraphQL API to the frontend for faster request queries with strongly typed schemas and fields
- Deployed application using **AWS** Elastic Beanstalk and setup Cloudfront distribution to direct all incoming traffic to registered domain name as well as sending all outgoing requests using a secure HTTPS port

MonoJStick | [Site](#) | [Repo](#)

- Created code snippets sharing app with **React-Router** to handle client-side routing
 - Used **Firebase** Authentication SDK to build secure multi-platform authentications and user creation and management
 - Utilized **Firestore** real time database to provide fast and reliable backend queries and mutations
-

Education

California Polytechnic State University, San Luis Obispo, CA | Bachelor of Architecture

Tech Talk

Jeeny and Bractlet Speakers Series - Serverless Architecture (Lambda)