­­

**DAVID LIAO**

415 860 1607 [| davidliao1438@gmail.com](mailto:|%20davidliao1438@gmail.com) | Fremont, CA

**SUMMARY**

* **5 years** of experience as a **Full stack** Javascript software engineer
* Created, adjusted, and maintained reusable UI Components in **React** using Javascript
* Experienced developing web user interface utilizing **JavaScript**, **HTML5**, **CSS3**
* Built fancy and responsive web pages with **Material UI** and **Bootstrap**
* Designed interactive web application using **One-way data flow**, **JSX**, and **Virtual DOM**
* Developed **Single Page Applications** (SPAs) with **React-router**
* Hand-on experience with React using **lifecycle** **Hooks** and **React**-**redux** for stateful management
* Used **Redux-thunk** as middleware to manipulate the data type and handle multiple action
* Experienced in using Middleware, **Redux-Promise** in application to retrieve data from back-end
* Designed and Developed with **Rest APIs** in Javascript for displaying **JSON** retrieved data
* Experienced in developing Restful web API using **Node.js**, **Express.js**
* Worked-on experience in using **NPM** packages like **MongoDB** in NoSQL and **Mongoose** to integrate MongoDB into web applications
* Designed complex schema using **SQL** in **MySQL** database to store and access data
* Worked on experience using Selectors in **JQuery** for updating content on **DOM** nodes
* Experienced in developing web application using **Spring MVC** framework in **Java** performing **CRUD** and aggregation operations on collections
* Designed **NoSQL** database and connect database with **Spring** **Hibernate** (ORM)
* Experienced on **Git** for version control to deploy whole code on **AWS** with continuous integration
* Experienced in DevOps such as integration testing and deploy the application to test servers using **Jenkins**
* Designed image processing system using **Optical Character Recognition** (OCR) in **C++**
* Experienced in image recognition using **CNN** and **GEI** in **Python**
* Worked on different operating systems such as Mac OS, Window and Linux
* Experienced working with various IDE’s like Eclipse, Visual Studio Code, and Notepad++
* Working knowledge of **Agile SCRUM** methodology
* Excellent communication, analytical, interpersonal and presentation skills

**TECHNICAL SKILLS**

* Web Developments: HTML5, CSS3, React, AJAX, JQuery, JSON, MVC
* Frameworks: React.js, Node.js, Express.js, Hooks, Redux, Thunk, Bootstrap, Material UI, Spring
* Databases: MySQL, MongoDB
* Application Servers: AWS, Apache Tomcat
* Programming Languages:Javascript, Java, C, C++, Python, SQL
* Tools: Git, Postman, Jenkins, NPM, OCR
* Operating Systems: Mac OS, Windows, Linux

**EXPERIENCE**

**Volvo Car Corporation - Sunnyvale, CA** Jan 2020 - now

**Full Stack Developer**

A full stack automotive parts inventory management system based on React and Node.js that allow employees to add, edit, update, remove different auto parts. Let employees easily track stock.

* Developed the web application base on **React**(front-end**), Node.js, Express.js**(back-end) and **MongoDB**(database)
* Designed web pages using **HTML5**, **CSS3** based on the **W3C** standards
* Designed home page with **CSS** **Flexbox, CSS grid** and **Material UI** to do **responsive web design**
* Used **lifecycle hooks** for stateful management
* Used **Redux-thunk** middleware and **Redux Sagas** to handle asynchronous event and side effects
* Worked on architectural separations for having a clear separation between reusable components and containers for connecting to **Redux** Models
* Implemented **React-router** to build **Single Page Application** to let user had a good user experience while switching pages
* Used **Ajax** framework for asynchronous data transfer between the browser and the server
* Prevented unauthorized users to hack data by using **lifecycle hook**, **redirect** and **Json web token** (**JWT)** method
* Adopted **redirect** method to let user easily choose the car model they need and link to website getting more information
* Implemented **Rest API** for accessing data from various data source **MySQL** and **MongoDB**
* Used **Redux** for maintaining the state and Dev tools for monitoring custom UI components.
* Used **Redux Form** for developing individual form components and connecting it to components with Store
* Implemented search feature with **Regular Expression** to allow user search without refreshing the page
* Worked on setting up user login to modify user information and update into **MySQL** database
* Designed **CRUD** operations with **Experess.js** framework to **MongoDB** database to process the data
* Parsed html request with **Expresss.js** server including body-parser, cookie-parser, cors, and morgan
* Implemented **NPM** and **WebPack setup** for building and generating production script files
* Developed and prepared test cases for unit testing with **Enzyme**
* Worked in an **Agile** methodology for application development and **Scrum** meeting
* Used **Git** for version control and deployed code on **Amazon EC2**

**Environment:** HTML5, CSS3, Material UI, Javascript, React, Redux, React-router, Ajax, Node.js, Express.js, Rest API, MySQL, MongoDB, NPM, WebPack, Enzyme, Git, Amazon EC2, Agile

**L’Oreal USA - Hudson Yards, NY** Aug 2018 – Dec 2019

**Full Stack Developer**

Designed and developed web application with different sections such as beauty research, latest news, and careers for customer to understand the brand core value proposition.

* Designed home page dashboard with different sections (products, article and menu) using **React** and **Javascript**
* Designed responsive layout for different devices by using **CSS** flex model and **Bootstrap**
* Implemented the **AJAX-based** frontend to reduce the network load and improve the user experience
* Implemented **pagination** features to divide all article and comments into different pages
* Designed front-end features like keyword search and filter article by using Javascript and **React Redux**
* Developed with **Rest APIs** in Javascript for displaying **JSON** retrieved data
* Handled different Http requests on different URLs for fetching user information, article, and comment by using **Fetch** API and **Express router**
* Implemented user authentication, encryption, and identification with **SHA256** and Salt
* Fetched article and comment using **Fetch** API in Javascript for accessing and manipulating requests and responses
* Implemented **Server-side Validation** to dynamically check user registration availability
* Adopted **Node.js** library and **Express.js** for the back-end framework to fetch **CRUD** operations with the **MySQL** database
* Built **MySQL** database to store user information, articles, comments, and product information
* Stored user profile picture on **Amazon S3** storage on AWS
* Test REST APIs through **Postman**
* Used **Bitbucket** with team and version control

Environment: React, React redux, Javascript, HTML, CSS, Bootstrap, AJAX, pagination, Express router, Node.js, MySQL, Postman, Bitbucket, Amazon S3

**Micron Technology - Taiwan** Apr 2014 – Jun 2017

**Full Stack Developer**

Develop a system to provide users easily get different semiconductor data and provide different chart for user to analysis semiconductor performance.

* Designed a web page for upload image and show result in **HTML** and **CSS**
* Sent data from client side to server through **Ajax** with **JSON** data
* Developed backend server by using **PHP** to connect frontend and database
* Wrote SQL statements to receive data from **MySQL** database
* Developed the server on **Apache Tomcat** as a Java servlet container
* Implemented **Optical Character Recognition** algorithm to analysis images
* Used **Postman** to mock HTTP requests and tested back-end function and reliability

Environment: HTML, CSS, Ajax, PHP, MySQL, Tomcat

**EDUCATION**

**Syracuse University** – Syracuse, NY, USA

* Master of Science in Computer Science

**Military** - Taiwan

**Chung Yuan Christian University** – Taiwan

* Bachelor of Science in Information & Computer Engineering