

RICHARD LAM

SOFTWARE ENGINEER

(347) 707 - 8189 | richardlam96@gmail.com
[Github](#) | [LinkedIn](#)

SKILLS

JS, React, Redux, Express, MongoDB, HTML, CSS, Selenium, C#, SQL, Git, NodeJS, Webpack, ThreeJS, Linux

EXPERIENCE

QA ENGINEER

GREENSLATE | OCT 2018 - OCT 2021

- Refactored .NET codebase, leading to improved performance, cleaner code, and more reusable modules
- Utilized Selenium in C# to build and maintain thorough tests in a .NET Automation Framework
- Reduced testing time by 20% through writing scripts in JavaScript, SQL, or Bash to gather relevant data
- Entrusted with testing high priority features directly related to tax calculations and company revenue
- Investigated and addressed urgent issues relayed from the App Support Team related to user authorization and data validation
- Collaborated with the Product Design and Dev teams to ensure proper communication of feature requirements

WEB DEVELOPER INTERN

LINCHPIN INC. | AUG 2018 - NOV 2018

- Created effective Data Visualization charts using React and Redux by communicating with project managers and backend developers
- Implemented animations with JavaScript and CSS to provide immediate feedback to users
- Improved the vision of the web application by designing interactive wireframes with Figma

PROJECTS

SINGULARITY

([Live](#) | [Repo](#)) A fast-paced jet fighter game with basic RPG elements and responsive enemies

- Leveraged SOLID principles and patterns, such as the Component Pattern, to increase extensibility for entity Behaviors and Attributes
- Implemented Collision Detection, multiple Leveling Options, and Player-reactive Enemy logic utilizing JavaScript and the ThreeJS library

CHATBOX

([Live](#) | [Frontend Repo](#) | [Backend Repo](#)) A Discord-clone made from scratch with features such as user authentication, instant messaging, and channel management

- Built a React, React-Router and Redux frontend and a RESTful backend created with Express and NodeJS, and paired with a NoSQL MongoDB database
- Implemented a live chat feature using the socket.io API to provide bidirectional communication between clients and the server
- Leveraged password hashing and middleware to enforce authentication and authorization to defend against improper access of message channels