Jason Cabrera

(626) 541 - 1335
Baldwin Park, CA

jrcabrer@ucsd.edu
jayrc7.github.io/personal-website/

github.com/jayrc7
linkedin.com/in/jasonrcabrera

Education

March 2020 – June 2020
University of California, San Diego | B.S. Computer Science

January 2022 – PRESENT (Part Time)
University of Texas at Austin | M.S. Computer Science

Technical Skills

Advanced | Java, Spring Boot, C#, Python, JavaScript, Node.js, ReactJS, Angular JS, SQL, Relational Databases

Familiar | C++, C, Kotlin, Haskell, Android, Pandas, Numpy, Android, JQuery, Amazon S3, ElasticSearch

Work Experience

July 2021 – PRESENT
General Motors | Software Engineer (Remote)
- Working on many Spring Boot Java microservices to support two of General Motors' internal tools used for vehicle validation.
- Created a new java service that helps with updating vehicle software over the air. Implemented coding logic by making use of the spring boot framework. Created controllers using the controller annotations, integrated a SQL server database by utilizing the spring boot data JPA annotations, added Kafka support using spring integration, created unit tests by implementing the spring boot test annotations and Mockito, and integrated many Spring Boot services and beans into the application. Assisted in the architectural design of the service.
- Updated Spring Boot framework in Java backend to the latest version in many applications to meet security standards. Required replacing deprecated classes, creating more spring boots beans, updating and creating more unit tests.
- Resolved bugs and worked on new features within the microservices.
- Utilized insomnia for testing existing and newly developed API from our services.
- Participating in meetings to clarify requirements and specifications for user stories.
- Made several services more cloud native by removing their reliance on file storage systems. Required integrating cloud storage with Amazon S3 to store certificates as well as HashiCorp Vault for storing application secrets.
- Configured applications so that their logs would route to the ELK (ElasticSearch, Logstash, Kibana) service to make application logs easier to navigate through. Created grok filters to parse logs and create new log fields that were used for data visualization.
- Deployed our applications to Kubernetes using Azure DevOps build and release pipelines.

September 2020 – December 2020
Bentley Systems | Software Engineer Intern
- Created unit tests using C# and the Moq testing framework, resulting in increased code coverage throughout the codebase.
- Converted Angular webapp pages such as User Acknowledgment and Custom Dashboard to its React equivalent. Integrated React Redux into the React pages to better maintain component states.

Projects

Kiwi Forum| Undergraduate Software Engineering Course, Project Manager and Full Stack Developer
- Led a team of ten developers in an Agile environment to develop a specialized forum web application for a software engineering course. Tech Stack: ReactJS, NodeJS, Express, Firebase
- Created the database schema, added navbar functionality that allowed employees to list/change the specialization by which their forum was filtered by, and created/tested RESTful API for CRUD operations on the database.
- Developed the authentication system using Express, Firebase, and cookies to allow employees and admin authenticated backend routes and pages. Created a secure employee invite system that emails a unique one-time-only invite link to a specified employee's email that would grant access to the admin's company forum.

Movie of the Week| Full Stack Software Development | *https://github.com/jayrc7/Movie-of-the-week*
- Developed a web app using React and NodeJS that allows friend groups to insert their movie suggestions and randomly picks a user's suggestion that hasn't been picked yet.
- Created a recent updates section using merge sort by sorting on multiple date fields in the database. Updates include someone recommending a new movie or a movie being marked as watched by the group.
- Implemented a user authentication system using server-side cookies and middleware functions as well as creating a react authentication component that checks if the user is logged in before showing a page.