Rich Lee

Senior Software Engineer | Resume

Introduction

I’m a seasoned Senior Developer with:

1. Proven expertise in a diverse array of web development technologies, honed through extensive hands-on experience.

2. A solid grounding in advanced programming design patterns and architecture, equipping me with the ability to tackle complex technical challenges.

3. A track record of growing and nurturing teams, from selecting the right talent through comprehensive interviewing strategies, to fostering collaboration and unity via effective team-building exercises.

Objective

I'm focused on building disruptive technology and swiftly merging high-quality, rigorously tested code into the mainline. Looking forward to joining forces with top-notch developers, I'm committed to simplifying complex technical issues. Prioritizing team efficiency, my goal is to increase my colleagues' output, helping us collectively reach our targets. My ultimate objective is to leave a positive impact on the company’s development culture that lasts years into the future.

Noteworthy Technical Experience

1. Successfully transformed a significant portion of a monolithic application into a scalable service-based architecture, through the creation of a reusable component library using yarn workspaces and npm modules.

2. Steered a team of five in developing Credly's Canvas and Bridge integration, fostering effective collaboration and project execution.

3. Worked closely with technical teams at Credly to define coding standards for the transition of a large server-side rendered static app into a React single page application.

4. Authored and reviewed innumerable lines of code, while strategizing and implementing sophisticated application architecture.

5. Engaged in all stages of the development lifecycle, from individual contributions to collaborative efforts in both small and larger teams, delivering enterprise software suites.

6. Managed communications amongst diverse stakeholders, including clients, project managers, and third-party integration teams, ensuring the delivery of high-quality software.

7. Adapted to organizational growth by switching development stacks from PHP (LAMP) to Ruby on Rails, effectively supporting newly acquired applications.

Core Technical Expertise:

Front-End Development:
Proficient in React (both modern and legacy versions), Redux, and GraphQL. Skilled in using Webpack, Babel, and NPM for efficient development processes. Adept at crafting appealing UI with Styled Components, SASS, and Figma. Experienced in unit testing using Jest, including Jest-Axe for accessibility testing.

Back-End Development:
Skilled in Node, Git, Ruby, PHP, and SQL with ActiveRecord implementations. Proficient in both unit and end-to-end testing for reliable and high-performing applications.

Deployment Techniques:
Well-versed in using Codeship, AWS Code Build, and Docker for streamlined and robust application deployment.

Frameworks & Libraries:
Comprehensive experience with Ruby on Rails, Express.js, Symfony PHP, Laravel, WordPress, Angular 1 & 2, CodeIgniter, CoffeeScript, and Backbone.js, providing a versatile development capability.

Emerging Skills:
Continually expanding proficiency with Solidity, Hardhat, Ethers.js, TypeScript, and Go to stay at the cutting edge of technology.

Soft Skills

Demonstrated Leadership:
Displaying competence in guiding teams towards common objectives and success.

Stakeholder Engagement:
Efficient in contacting relevant stakeholders for updates or requests and managing expectations.

Project Management Collaboration:
Skilled in identifying and documenting project requirements in collaboration with Project Managers using JIRA. Capable of delegating tasks to team members based on priorities.

Scope Management:
Proficient in identifying and managing out-of-scope requests to guarantee timely delivery of primary tasks.

Strategic Partnership Management:
Proven ability in discussing feature development and orchestrating cooperative endeavors with third-party teams having shared clients.

Interdepartmental Communication:
Effective in translating requirements and issues between technical and non-technical parties, fostering clear and mutual understanding.

Career Experience

Credly (Software Engineer): website

2016-present

Starting from a small team of five developers and expanding to over a hundred, I honed my leadership skills on Credly's earner experience team. My contributions included conducting hundreds of code reviews, establishing and negotiating technical requirements, as well as writing, coding, and assigning JIRA-tracked features.

I spearheaded the development of Credly's Canvas and Bridge LMS integration, along with leading the charge to transform Credly's frontend design system architecture from a monolithic application to a distributed system across the Pearson ecosystem.

Moreover, I have built, documented, and deployed a host of Credly API features including badge expiration, elastic search, Single Sign-On (SSO), OAuth, among others.

Noteworthy Projects

1. Credly's Canvas integration

I initially developed Credly's Canvas integration project using PHP. Upon Credly's acquisition of Acclaim, I was given the responsibility of creating a second version of Credly's Canvas integration using Ruby on Rails in the Acclaim ecosystem.

The implemented feature allowed students to earn Credly badges based on their achievements in Canvas LMS and other third-party LMSes that implement LTI OAuth 1.0. For Canvas, it required scheduling data imports for various student data. For third parties supporting webhooks, we facilitated the push of data into Credly through user-generated JWT auth tokens.

Challenges and Solutions:

1. I delegated parts of the import and assignment evaluation process to jobs for better horizontal scalability.

2. I leveraged alternative API endpoints where possible to limit API throttling and to reduce unnecessary requests.

3. I designed a database table to monitor the status of certain jobs to prevent duplicate processes from running and to block jobs that couldn't succeed from initiating.

4. I developed several support scripts and tools, including a Chrome plugin, to allow developers to simulate the customer's experience in their respective Canvas instances for responding to support requests.

2. Modernization of Legacy Server-Rendered UI with React.

As Credly's team was experiencing rapid expansion, the maintenance of our extensive ERB/HAML Backbone.js-based frontend application started posing increasing challenges due to its susceptibility to errors.

I joined forces with the UI team with the objective of upgrading the frontend to a contemporary JavaScript library, React, with the aim of accelerating development speed and enhancing code reusability.

Solutions

1. Handled REST API requests using Redux and action managers.

2. Crafted form components that played nice with action managers, displaying and preserving API data models in a way we could easily test.

3. Spotted reusable elements and made sure to line up follow-up tickets to refashion them into standalone components.

4. Pulled the team together to hash out coding standards for CSS class names, setting up clear conventions for things like dynamic vertical spacing, color palettes, and padding.

5. Made sure that Accessibility wasn't an afterthought - it was front and center during every code review.

6. Upgraded our old React components, trading in the classes for sleek, functional components with the help of React Hooks.

3. Transformed Major Part of Ruby on Rails/React Monolith into Reusable npm Modules for Scalable Service Oriented Architecture.

The second phase of scaling our frontend design system entailed the separation of over 100 custom UI components and utilities into distinct npm packages. Several proposed solutions sparked intense discussions, including rewriting the entire library in a separate repository and direct replication of existing components. However, these approaches were deemed unsuitable due lack of developer resources and the high degree of interdependence among components and the application, respectively.

Ultimately, we opted for a strategy that leveraged yarn workspaces to transform significant portions of our existing libraries into independent npm modules. This allowed the code to remain in the existing repository while we incrementally disentangled dependencies to prepare the components for external use.

Decoupling these components and deploying them as npm packages paved the way for scaling Credly's - and subsequently Pearson's - system architecture, thereby ensuring a consistent user experience across multiple applications. Later, the decoupled code was transferred from the monolith into a separate repository, which was then managed by the design system team.

Challenges Encountered:

1. Developers exhibited strong convictions about their individual methods for creating a component library for the design system, necessitating numerous meetings, debates, and compromises.

2. The design system team's limited resources hindered the development of the component library. We encouraged developers from other teams to contribute, but their commitments to other projects often prevented their participation.

Solution Implemented:

1. We introduced yarn workspaces, which enabled us to create the package directly within the existing monolith repository. This provided a familiar environment and API for developers to contribute under the new guidance of the design system team, addressing the issue of limited developer participation.

2. Ensuring team acceptance of this solution involved two steps. First, I developed a PowerPoint presentation to demonstrate the benefits to the development team, highlighting how using yarn workspaces would maintain most of their current workflow without necessitating work in a separate repository. Secondly, I assured the design system team that the code would eventually be migrated to the design system repository once it had been successfully decoupled from dependencies in the monolith. This approach enabled the design system team to leverage existing components rather than completely rebuilding from scratch.

Quallsbenson (Director of Web Development): website

2014-2016

As the sole developer within Quallsbenson, a dynamic real estate advertising agency, I was responsible for creating web development services that integrated tracking features and property management tools. During my tenure, I accomplished the following:

1. Transformed PSD templates into responsive HTML5 templates, using the Symfony framework to infuse these templates with relevant database content.

2. Populated websites with data sourced from an array of REST APIs, including Online Residences (OLR), Multiple Listing Service (MLS), Google Maps and Geolocation, Twitter, among others.

3. Transmitted detailed tracking information such as user IDs, user behavior and events to Google Analytics, subsequently syncing them with our company database. This was done primarily for the purpose of facilitating market research and generating insightful reports.

4. Devised and structured Database Schemas, and manipulated active records classes and Database mapping implementations to interact with our stored data.

Noteworthy Projects:

1. Abacus

Abacus is a comprehensive property management system that provides clients with access to a unified database of properties and tenants. My responsibility involved building this platform from scratch, and integrating it with the suite of KRE property websites, Google Analytics API, and CallRail. This seamless integration was essential in tracking customer interactions from initial website visits to final leasing agreements.

2. Analytics Api Data Analyzer

This custom-built tool enabled marketers to leverage the power of Google Analytics API for validating certain hypotheses concerning user conversions. Its key function was to identify the source of user sessions and reconcile this information against tenant sign-ups in Abacus, thereby providing valuable insights into customer behavior and conversions.

Entrepreneur Experience

Love Amethyst Rose (Co-founder): website

2018-2020

Love Amethyst Rose is a boutique specializing in traditional Eastern Spiritual Bead Jewelry. As a co-founder, my responsibilities encompassed a broad spectrum. From conveying our unique story and leveraging Facebook insights to pinpoint our target audience, to securing funding, hiring the right talent, and scaling the team - my role was diverse and dynamic. Beyond the business end, I also held the technical reins, which involved maintaining the website and introducing new features to enhance our customer experience.

TechStack

Facebook pixel tracking, Wordpress, Linux, PHP

Noteworthy Achievements:

1. Developed a meme generator app that facilitated the rapid creation of ad content variations. These were then tested to deliver profitable Facebook/Instagram ads.

2. Established an email funnel and a customer portal to offer regular updates from third-party shipping providers, effectively easing the load on our customer support amidst supply-chain disruption.
3. Authored compelling sales copy that played a significant role in our top-converting Facebook ads.
4. Adapted our website to ensure ad tracking accuracy following the iOS14 app tracking update.
5. Successfully drove $30k in sales during our peak month as a direct outcome of the above-mentioned efforts.

Personal Projects:

Web3Social

A groundbreaking social network that establishes ownership of user profiles utilizing NFTs minted on a decentralized EVM network. This innovation ensures the unique ownership and seamless transferability of usernames, enhancing authenticity and reputation in the digital sphere. The platform is structured around a robust tokenomics system, which encourages the creation of high-quality content while actively discouraging spam for an optimal user experience.

View Source code.

Discern php.

The Discern Parameter State Library is an innovative PHP library engineered to facilitate the creation of human-readable, complex, reusable, and strictly-typed parameter structures for your functions and class methods. This practical tool extends the static typing capabilities of PHP 7+ and introduces an intuitive way to define the type of each array element, enhancing code readability and maintainability. Moreover, it provides extensive error handling for easier debugging. The Discern Parameter State Library thus serves as an incredibly powerful tool, bridging the gap between stringent type validation and user-friendly coding in PHP.

View Source code.

Social Media meme generator

This tool functions as an image generator, specifically designed to create diverse, testable variations of ad content. It retrieves images and captions from Airtable, then generates meme images complete with captions. The size of these images can be adjusted to suit the specific social media network where they will be posted.

View Source code.