Syed Ali
07454 690 590
eskyerin@gmail.com

Notable Languages/Technologies used:

- PHP (including Symfony, and Laravel frameworks).
- JavaScript (including NodeJS, TypeScript, and React/Redux)
- Oracle / MySQL / AWS Redshift / DynamoDB / Cassandra / ScyllaDB
- Docker / Vagrant
- AWS (API Gateway, SNS, SQS, SES, Lambda, CDK, Kinesis)
- Redis

Work Experience:

04/03/2024 - Now Laka - Senior Full Stack Software Engineer

Laka is a bicycle insurance company that follows a collective insurance model rather than a traditional insurance model. When you are insured with Laka, you only pay for the claims made that month by the collective and payments are hard-capped so you will never pay more than you would for traditional bicycle insurance.

At Laka, I joined a squad which concentrates on internal tooling, working with a tech stack of Node (TypeScript), React, Go, Postgres, and GraphQL . Notable projects I have worked on so far include building our new claims management system, and amending our billing processes to be driven by the new claims management system (where before these were two separate processes handled outside of our admin tools).

07/02/2022 - 18/02/2024 Etika UK - Software Engineer

Etika is a finance company with an aim to keep finance fair and ethical. At Etika, The tech team is split between the UK and Australia, with most development work needing to take into account regional differences for Australia, UK, and New Zealand. The legacy platform was written in PHP (Laravel framework), and the new platform was written in NodeJS and React.

My role consisted of working on the new platform built in as we migrated away from the legacy platform (but still doing some changes on the legacy platform as required), and building / maintaining our official Shopify payment gateway app. The new platform is micro-service and micro-site based, with login areas for merchants, customers, and our ops team.

Notable projects:

- Official Shopify payment gateway app

  Etika received permission from Shopify to create an official payment gateway application. This required new development for both the app and on our global platform to process loan applications coming from Shopify. We built all of this in NodeJS, and React.

  The general process of this was the user would be sent to an endpoint on the plugin which would receive the basic checkout information of address, name, email, and cart amount. From there, we needed to acknowledge it with Shopify and then direct them to our platform to select the loan/deposit, and complete an application.

  Based on the outcome of that application, we would make a GraphQL query to specify if the application was accepted, declined, or needed further consideration (resolve, reject, and pending). That would provide us with a return url to send the user back to Shopify.

- Shopify companion app

  Shopify provides limited information to payment providers. This means that the merchant cannot configure finance product availability based on the products being purchased. There is also no way for a payment provider to be updated when an order has been fulfilled. The companion app was designed as an unofficial app that a merchant could choose to install, and this would allow them to configure financial products based on the cart, and also have the etika platform be notified of fulfilment based on it being done in Shopify.

For the financial product configuration, It would do this by using the cart id taken from the cancel payment link. This cart id could then be used in a query to the storefront GraphQL to get the products from the cart, and the metadata configured against the products. This would then be passed to etika along with the basic cart information..

This was built in NodeJS and not directly accessible. One app serves all of the merchants. With the cart id, we would also send through which merchant it was and would take the key/secret saved for that merchant to make the query.

- Accessibility review and improvements

  Our application and sign up process was not screen reader friendly due to it being a single page app. This required us to rethink the user's journey through the app and craft an experience not far different from a telephone tree. We leaned heavily on the ARIA Authoring Practices Guide to learn how best to achieve this.

14/09/2021 - 14/01/2022 Snug Shack Ltd - Software Engineer

Snug Shack operates the ecommerce site snugsofa.com in which they sell sofas in a box to simplify furniture delivery. The main application was written in PHP 7.4 with Laravel 8.

My role consisted of working on backend reports using our data warehouse (Redshift), working on data pipelines from our sales platform (Shopify) to our inventory sales system to Redshift, and helping on the frontend site as required. As the in-house technology is still in its early phases, there is emphasis on doing things right from the start, and implementing ideas that will benefit us in the long run.

17/12/2019 - 10/09/2021 Pitchero Ltd - Senior Full Stack Developer

Pitchero provides software solutions for grassroots to semi-professional sports clubs for managing and publicising themselves. The core application was written in PHP (Laravel framework), the club website application was React/NextJS, and the members area was written in React/Redux.

My most notable solo project was to replace an existing PHP application used for image resizing with one that runs on AWS Lambda, accessible via API gateway. In preparation for this task, I first had to analyse both performance and the external interface for the old PHP application to make sure that both performance was improved, but also so we could be sure it was a perfect drop in replacement. To prove this, both unit and acceptance tests were written using Jest. I created multiple pieces of documentation to allow anybody, regardless of experience with AWS products, to be able to re-deploy and configure the Lambda application, the API Gateway, and the VPCs as Lambda was not a technology used by the company already.

With the team, the biggest project I had been a part of is implementing a match fees module. This allows teams to charge their players match fees through our system, with the players being able to pay these via the mobile apps or the web app. It provides a simpler way for coaches to manage who needs to pay and how much, but also takes away the admin work from the clubs as this is handled through the Pitchero system instead. My contributions to this were mostly concentrated on the backend, but I assisted the frontend developer as required in both guiding them or taking on some of their workload to allow the work to progress.

15/09/2017 - 13/12/2019 Bytron Aviation Systems - Senior Full Stack Developer

During my time here, I architected large overarching systems to allow real time data updates, data processing with queuing systems and the use of Serverless (AWS Lambda) to both decrease cost, and increase response speed.

I was responsible for creating the brand new Flight Following module (to comply with GADSS regulations introduced in November 2018) built using React, Leaflet, Lambda for position data processing and SocketCluster to distribute the information as soon as it was processed. Alongside this, I created another application which would check flight data we were given and, based on a given rule set, would send out alerts if anything was amiss. It would also send out alerts for missing positions as well.

I also began rewriting the web application which was in a state of disrepair, introducing React and tools such as Webpack (with tree-shaking), Jest, and TypeScript to the team. This involved several training sessions where I explained the why of these tools, and then showed them how we use them.

16/02/2016 - 18/08/2017 Wren Kitchens - Full Stack Developer

Re-joined the kitchen planning tool team. However, as the majority of the developers for the planner team were overseas, my role was more working on major technical improvements in the system and work that spanned our various in-house applications, whilst most project work was given overseas.

Making performance and memory usage improvements was completely new to me at this level. So firstly, I had to learn how the memory profiling tools worked in depth in both firefox and chrome. I also then had to start looking into what we were doing in our application to see where memory was being used and why. This led me to make several improvements over the course of a few months, walking a fine line in many cases of CPU use vs memory usage, given a minimum spec that needed to be supported.

My role also consisted of supporting helpdesk and fielding issues as they came through, where I would do some investigation and pass the issue on with more information, or fix them as required depending on the severity of the issue and the cause.

I created, and was then responsible for an application which supported our kitchen planner in nodejs, utilising RabbitMQ and PhantomJS to create multiple images from the kitchen planner for use in our sales documentation. This solved several performance issues I discovered previously with our kitchen planner, and also gave us more flexibility as to how and when we'd generate these plan diagrams without any issue for the end user.

I created an offline version of the kitchen planner, which was then given to room surveyors so that on-site they could take full measurements and plan the room in our software. These would then be given to kitchen planners for them to build on.

08/09/2015 - 29/01/2016 Frog Education - Full Stack Developer

Placed onto a feature development team from day one, mid-project. Because of this, I had to learn the system pretty quickly from day one so that I could start contributing. I was tasked with writing new and modifying existing APIs, which facilitated creating new curriculums and recording assessment results. I was also working on a legacy SPA which used the JMVC framework to display data and also create a frontend to use the APIs. I was also given the responsibility of repairing the unit and acceptance test suites.

22/09/2014 - 04/09/2015 Wren Kitchens - PHP Developer/JS Developer

Originally hired to work on the internal sales system (written in Symfony 2.3), juggling between rapid response tasks and also working with project teams to help them implement new features. During this time, I also identified that bug replication for developers wasn't easy based on the given instructions. To combat this, I introduced the QA team to Selenium tests exported to PHPUnit which were given to developers to make bug fixing tasks easier.

After 3 months, I was moved to the kitchen planning tool team. The kitchen planner was built using HTML5 canvas and a custom rendering engine built in house on top of it. I was also tasked with extending the application to also design bathrooms. As part of the project, we introduced NodeJS with ElasticSearch. We also started to migrate from handlebars to React.

In later days, I oversaw the work done by an offshore development team that was brought in, addressing any issues they may have and also made decisions on how some features should be implemented. I also helped create a roadmap for upcoming projects.

15/11/2011 - 15/11/2013 Nexus Vehicle Management - Support & Web Developer

Started at Nexus as a support apprentice, but was also given the ability to learn PHP to also work on their web based IRIS software. In terms of development, the most notable projects I delivered were as follows;

- A web based CRM system designed for our field sales team designed for use on the ipad. I prototyped this after meetings with the directors and our champion field sales member, with guidance from the IT director. I also supported this dealing with both hardware issues and any software bugs that were found.

- Created a SOAP service to send bookings to Sixt, taking confirmations and updating the related rental booking.

- Software customisations for two of our rental broker customers who took their own version of IRIS to manage their own rental fleet bookings. For both go-live days, I was on-site to ensure the transition was as smooth as possible.