Ajitesh Tiwari

Senior Software Engineer

Bengaluru, India
(+91) - 8939163843
ajiteshtiwari2011@gmail.com
ajitesh-tiwari.github.io | github.com/Ajitesh-Tiwari | linkedin.com/in/ajiteshtiwari

SKILLS

Java, Spring Framework, Microservices Pattern, Kafka, Domain Driven Design, System Design, Javascript, React JS, Redux, Single Page Applications, Node JS, Express JS, Kubernetes, Docker, Data structures and algorithms.

EXPERIENCE

Lowes India, Bengaluru - *senior software engineer*

NOVEMBER 2019 - PRESENT

- Working on Order Management System (OMS), which is a group of distributed microservices that work together and support order throughout its lifecycle (replacement of IBM Sterling). The orders being captured have various flavors and rules associated with it.
- Worked on seamlessly deploying and managing secrets, config, etc. for the applications. Learned how large-scale e-commerce works behind the scenes. Also learned, database design and code architecture in applications at scale.
- Mentored junior developers in our team to deliver high quality code on time. Also brainstormed with senior architects on various high and low level system design requirements.
- Helped on supporting the system during multiple holiday seasons, by actively participating in troubleshooting calls and debugging sessions.

Recrosoft Technologies, Bengaluru - *full-stack developer*

APRIL 2019 - NOVEMBER 2019

- Worked with a client (SpringRole) on building a React Redux / Express JS application, which is used for verification of candidates.
- Took ownership of the entire end-to-end solution, including architecture decisions as well as code reviews.

Mr. Cooper, Chennai - *software engineer*

JUNE 2017 - APRIL 2019

- Worked on building a solution which helps a customer get pre-qualified for a mortgage.
- Worked on building/supporting a gradle project with 20+ spring services using contract first approach. These services were used to empower most of the organization workflow.
- Worked on building real-time and batch notification systems used to send email(s) and push-notification(s) to live customers.
- Learned a lot of best practices for building services and on how to use them on various clients.

PROJECTS

Order Management System - *java / spring-framework / kafka / kubernetes / domain-driven-design (DDD)*

- An application which supports multi flavored (PICKUP-IN-STORE, PARCEL, TRUCK-DELIVERY, etc.) orders throughout their lifecycle (CREATED, SCHEDULED, RELEASED, SHIPPED, BACKORDERED, etc.)
- The application is a group of many microservices and kafka listeners that fit in the existing enterprise ecosystem. It has schedulers, error-re-processors, templating engines, etc.
- We have strictly followed Domain Driven Design (DDD) and Event Sourcing. Learned a lot about design patterns and clean / hexagonal architecture.
- Also worked with the DevOps team on managing infrastructure (Vault, Kubernetes, Spinnaker, Jenkins, etc.)

Spring Verify US - *react / redux / express-JS / formik*

- An identity verification platform, which allows companies to create accounts and send dynamic forms to its employees. These forms collect employee information, and validate him against education, employment, personal information, etc.
- Used Backend for Frontend (BFF) architecture, to scale the application to various platforms easily. The front-end application follows redux architecture.
- Used many third-party APIs like clearbit for data enrichment.

EDUCATION

SRM University, Chennai - *Bachelor of Technology (CSE)*

MAY 2013 - JUNE 2017

PUBLICATIONS

Surge Pricing Predictor in Taxi Market - *International Journal of Pure and Applied Mathematics*

https://acadpubl.eu/jsi/2017-115-6-7/articles/6/81.pdf

- The cab platforms adjust their prices using a specific algorithm which is real time and dynamic known as “Surge Pricing” or “Dynamic Pricing”.
- This algorithm automatically raises the price of a trip when the demand increases more than the supply. The surge algorithm generally outputs a multiplier which is adjusted along with the base fare, the price per mile and the price per minute to generate the final price.
- This price is communicated to the riders and the ride is initiated when they confirm the price shown. This surge multiplier is kept discrete and may range from 1.2 to the maximum allowed by the government based on geography.
- Our experiment helps in predicting surge pricing ahead of time, considering the previous trends.