Ajitesh Tiwari

I am a passionate computer engineer, full-stack developer with 2 years of professional experience. I have good problem-solving skills and want to solve complex real-world problems. I love to work on new technologies and enhance my skill set.

Bangalore, India.
(+91)-8939163843
ajiteshtiwari2011@gmail.com

ajitesh-tiwari.github.io
github.com/Ajitesh-Tiwari
linkedin.com/in/ajiteshtiwari

EXPERIENCE

Recrosoft, Bangalore — *full-stack developer*

APRIL 2019 - PRESENT

Working with a client (SpringRole) on building a React-Redux / ExpressJS application, which is used for verification of candidates.

Took ownership of entire end-to-end solution, including architecture decisions as well as code reviews.

Mr. Cooper, Chennai — *full-stack developer*

JUNE 2017 - APRIL 2019 (2 YEARS APPROX.)

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 system 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.

Mr. Cooper, Chennai — *software engineering intern*

DEC 2016 - MAY 2017 (6 MONTHS)

Worked on building an android application which has 100,000+ downloads on google play-store.

This application can be used to check loan-balance, FICO score and payment due date.

SKILLS

Java
Javascript
NodeJS / ExpressJS
ReactJS / Redux / SPA
SASS / CSS Modules
Spring Framework
Microservices Pattern / REST
Swagger / Open API
Git / Docker / AWS / Azure
Data structure and algorithm
Android SDK

EDUCATION

B-Tech (CSE)
SRM University,
Chennai.

2013 - 2017

CGPA - 9.55

Intermediate (+2)
City Montessori School,
Lucknow.

2012 - 2013

Percentage - 93.75%

High School
City Montessori School,
Lucknow.

2010 - 2011

Percentage - 92.40%

PROJECTS

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

- An identity verification platform, which allows companies to create account and sent 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.

Customer Pre-Qualification — *react / redux / express-JS / redis / redux-forms*

- Dynamic form which is used to collect user information for generating pre-qualification form. It has multiple steps and offers intermediate saving and pre-population of data.
- Functionality to show progress based on number of valid answers and dynamic scroll to show/hide questions, which made the form more intuitive.

Microservices — *spring-boot services*

- A gradle project with 20+ spring services using contract first approach.
- Contract first approach generated code from pre-defined RAML specification which helps client teams to work simultaneously, decreasing development time.
- All services are deployed separately with multiple instances with HA-Proxy load balancer in between.
- Used rabbit-MQ to make inter-service communication scalable and resilient.

Notification Framework — *spring-batch service*

- Worked on building real-time and batch notification system used to send email(s) and push-notification(s) to live customers.
- Used spring-batch and quartz scheduler to schedule jobs to read from database and send notifications to respective customers.
- Supports dynamic templating and allows customer to (un)subscribe to specific notifications. • Created a pagination endpoint which is used to get a list of all notifications send to a customer using various query parameters.

PUBLICATION

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 to 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.