



Karthik Subramanian

Karthiks3000@gmail.com | +16479099842 | WATERLOO, CANADA

LINKEDIN | PORTFOLIO | GITHUB | MEDIUM | STACKOVERFLOW

EDUCATION

MUMBAI UNIVERSITY

BACHELORS INFORMATION TECHNOLOGY

JUNE 2011

MUMBAI, INDIA

SKILLS

PROGRAMMING LANGUAGES	JAVA, PYTHON, JAVASCRIPT, TYPESCRIPT, PL/SQL
FRAMEWORKS	SPRINGBOOT, DJANGO, HIBERNATE, ENTITY FRAMEWORK, PETAPOCO
AWS	LAMBDA, EC2, KINESIS ANALYTICS, DATA STREAMS, FIREHOSE, S3, CLOUDFORMATION, SAM
TOOLS/ PLATFORMS	APACHE FLINK, APACHE AIRFLOW, APACHE KAFKA, RABBITMQ, POSTMAN, NEWRELIC, KIBANA, DATADOG, PAGERDUTY, DOCKER
DATABASES	MYSQL, POSTGRESQL, SQL SERVER, DYNAMODB, SNOWFLAKE, ELASTICSEARCH
FRONT-END FRAMEWORKS	ANGULARJS, ANGULAR, REACT

EXPERIENCE

RENAISSANCE LEARNING | TECHNICAL LEAD

TORONTO, ON | MAR 2022 – PRESENT

- Lead a team of highly motivated & driven data & software engineers in maintaining & enhancing Schoolzilla for Renaissance products
- Helped guide the successful migration from a single tenant DB setup in SQL Server to a multi-tenant DB in **Snowflake** which resulted in a **savings of around \$30K** per month
- Help tune & optimize the infrastructure to scale with new customer onboarding
- Work with multiple stakeholders to plan & prioritize business goals & balance tech-debt
- Translate technical concepts & implementation strategies into easy-to-understand language for dissemination among the larger organization
- Be a servant leader in empowering the team to deliver their sprint commitments by tackling cross-team collaboration & dependency management
- Hands-on contribution in debugging production bugs using **DataDog**, **Newrelic** & **CloudWatch** & incident management using **PagerDuty**
- Contribute to development of features in Angular 10, new DAG templates in **Apache Airflow 2.0** & a micro-service architecture comprising of services in **Groovy** using **SpringBoot 2.5** as well as **Python** using **Flask** with **PostgreSQL**
- Help transform data in Snowflake using custom queries in **DBT Cloud**
- Monitor infrastructure costs & utilization using **AWS Cost Explorer** & mitigate costs by scaling services accordingly in ECS
- Review & update the hiring process & backfill multiple open positions on the team

- Architected an ETL pipeline using **AWS Data Streams, Apache Flink & PostgreSQL** to support a custom analytics tool created for the internal research team.
- Implemented a highly reusable code base for Apache Flink supplemented with 3rd party plugins such as Lombok, podam, meanbean and gson, to accelerate development and unit testing
- Designed a highly scalable data model in **PostgreSQL** that supports **data aggregation** on a per user basis rolled up at various time intervals to allow for extremely dynamic & performant querying
- Developed highly scalable and optimized REST APIs in **Java** using the **Spring Boot & Python** using **Django Rest Framework**, working on both stacks simultaneously
- Leveraged the **K-Means** algorithm to automatically group students in a class based on their skills, interests and reading levels, providing valuable insight to teachers on students performing at the same level
- Implemented custom aggregation on streaming data using **AWS Kinesis Analytics & Firehose** with a **Data Lake** in **S3**
- Processed simple streaming data practically for **free** using Java functions in **AWS Lambda** where event streams were not likely to exceed 1 million requests per month
- Established a reusable query manager class that would allow clients to easily and quickly create custom **ElasticSearch** queries by leveraging a suite of helper methods
- Built intuitive reporting dashboards aggregating data from multiple sources like **ElasticSearch, DynamoDB & MySQL**
- Provided production support & debugged bugs using **Newrelic, Kibana, OpenShift & AWS CloudWatch** logs
- Helped optimize the CI/CD pipeline by automating deployment builds from **BitBucket** to **AWS & OpenShift** using **Jenkins**
- Developed a custom **Django middleware backend** for sending emails leveraging the organizations **Adobe Campaign Manager** implementation
- Maintained application SLAs by load testing the REST APIs using performance scripts written in **JMeter** & load tested in **BlazeMeter**
- Helped develop FE features in **AngularJS & Angular 8**
- Interfaced with multiple stakeholders & product owners to streamline requirements & translate them into actionable JIRA cards
- As part of the CTR (critical team review) group, conducted reviews of other teams to evaluate coding and testing approaches and recommend best practices

- Spearheaded the architecture, design & development of a Content Protection Platform that enabled Viacom to protect its digital assets via 3rd party vendors like YouTube, Vobile & Audible Magic.
- Delivered the Content Protection Platform on schedule & within the estimated budget
- Successfully mapped all business use cases & pain points into actionable work items ensuring 100% coverage & an end product that was intuitive, easy to use & a fully automated workflow for fingerprinting & watermarking to partners
- Leveraged **RabbitMQ** as the messaging framework to integrate with multiple internal systems, defining the exchanges, queues & routing patterns along with DLX queues with retry capabilities
- Created a highly durable & scalable RabbitMQ subscriber using **EasyNetQ**, that leveraged **.Net Async Tasks** to process multiple messages at once & implemented a re-try mechanism for failed messages using the message headers to keep track of the re-try count
- Designed & developed the FE dashboard in **Angular 4** using **PrimeNG** controls in **TypeScript**, including creating custom components for advanced filtering & state graph that can be leveraged by other angular projects
- Designed & developed the database schema in **PostgreSQL** & created the **REST APIs** in **python** using the **Django Rest Framework & Redis cache**
- Implemented a seamless authentication protocol for Active Directory authentication using ng2-adal, and Authorization using Django Tokens
- As the **Onsite Tech Lead**, simultaneously also managed an offshore team of 5 developers responsible for maintaining and enhancing a legacy windows forms application that was responsible for scheduling the broadcast of shows across more than 150 TV channels
- Refactored multiple key modules of the legacy application using **Infragistics** Controls and improving the application's look & feel

L&T Infotech | SOFTWARE ENGINEER

MUMBAI, INDIA | JUN 2011 – JUN 2012

- Responsible for development of critical & time sensitive modules using **VB.Net, C#.Net, LINQ, Crystal Reports & Infragistics**
- Effectively used **PetaPoco**, a micro-ORM, for a fast & seamless interaction with the SQL Server database
- Implemented server caching techniques, state management, forms authentication, and created Master Pages in ASP.NET using MVC Architecture
- Developed complex reports in SSRS leveraging stored procedures implemented in **SQL Server 2005**
- Worked extensively with database objects like Stored Procedures, Triggers, Views and Functions

L&T Infotech | SOFTWARE INTERN

MUMBAI, INDIA | JUN 2010 – JUN 2011

- Developed an algorithm in **VBScript** to crawl through websites to track piracy of specific audio and video content on the internet
- Generated analytical reports of Pirated Videos & extrapolated various patterns & usage across several demographics using SSRS
- Performed Unit & Integration Testing
- Documented & Maintained design documents & release notes

PERSONAL PROJECTS

REVIEWS SCRAPER | PYTHON, REACT, AWS LAMBDA, AWS SQS, AWS DYNAMODB, AWS API GATEWAY, AWS SAM

A simple website that allows users to download google map reviews for any business. The user provides the URL to the reviews page on google maps and is then able to download a csv of all the reviews.

[Web Page Link](#)

SAMPLE SERVERLESS APPLICATION | PYTHON, AWS LAMBDA, AWS SQS, AWS DYNAMODB, AWS API GATEWAY, AWS SAM

Based on my experience working on the Reviews Scraper, I wrote a series of posts on Medium detailing how to implement a simple serverless architecture using AWS SAM.

[GitHub Link](#)

POSTMAN-DOC-GEN | PYTHON, JINJA

A simple application that converts your postman json collection into an html documentation similar to the one Postman generates. This tool is handy for anyone that wishes to host the documentation themselves instead of on Postman's server.

[GitHub Link](#)

SPLITWISE DASHBOARD | REACT, PIZZLY

A simple dashboard for your Splitwise account. Allows you to view transactions by groups & provides 2 bar graphs that display the transactions by category or time for a selected group. The REACT app leverages Pizzly, an OAuth integration proxy to authenticate users with Splitwise directly and pulls their data using Splitwise APIs.

<https://splitwise-dash.herokuapp.com/>

POSTS/ARTICLES

- [AWS Serverless Architecture with SAM](#)
- [Hosting API Documentation with Postman & SpringBoot](#)