3260 San Bruno Ave, San Francisco, CA 94134
(412)-636-3968

ALABHYA FARKIYA

alabhya16@gmail.com, afarkiya@andrew.cmu.edu
Website: https://goo.gl/SrZE5G

https://www.linkedin.com/in/alabhyaf

EDUCATION

Carnegie Mellon University (Pittsburgh, PA) August 2017
School of Computer Science, MSIT eBusiness Technology, GPA:3.86/4.00

University of Pune (India) June 2015
Bachelor's in Computer Engineering

SKILLS

Programming Languages : Java, Python
Cloud & Data mining: AWS, Azure, GCP, Hadoop, Map-Reduce, Redis, Spark, Samza, Kafka, Zeppelin, Jupyter, sklearn
Database: MySQL, HBase, MongoDB
Web: HTML, CSS, JSON, SEO, J2EE, JSP, XML
Operating Systems: Linux, Mac OS, Windows, Unix, Android SDK
Others: Git, Eclipse, Android Studio, Maven, JDBC, RDS, EMR, S3, Servlets, Multithreading, Big Data, pandas

PROFESSIONAL EXPERIENCE

Daily Doc (**Link**: https://play.google.com/store/apps/details?id=com.chronic.wellbing) Jun '15 - Jun '16
Software/Product-Engineer (Full-time)

- Designed and implemented user-authentication, data logging workflows, chat application and user-profile management for the **Android app team** in an **agile start-up environment**.
- Initiated **market research studies** and analyzed findings to understand customer and market opportunities to streamline the product for customer thereby helping the product **pivot towards a better product-market fit**.
- Improved the performance of the app by 20% thereby reducing costly rendering of the UI.
- Designed, developed and maintained JSON based api's for communication in the **backend team**.
- Was **available at hand** during new version releases and **wrote technical specifications** of the development process.
- Conceptualized, designed, developed and deployed a **fully functional Android app** having more than **40,000 downloads**. (**Link**: https://play.google.com/store/apps/details?id=com.troika.Aptitude)

PROJECTS

Twitter Analytics Web Service (**Learn more:** https://goo.gl/nCql9J) Mar '17 - May '17

- Designed and implemented a **high performance**, **fault-tolerant** and **scalable cloud deployment strategy** responding to **live load** while meeting infrastructure and budgetary needs.
- Performed **ETL** on a **1 TB** dataset to load data into MySQL and HBase systems using **MapReduce** and **Spark** frameworks on **Amazon Web Service(s)**, **Google Cloud Platform**, and **Microsoft Azure**.
- Hiked the performance of service from 3000RPS to 10,000RPS by modelling effective schemas, **sharding the database** and optimizing server threads while utilizing the same resources. Also, implemented **distributed hash**.
- Configured the service to handle **data from all languages**, including emoji's and implemented **cache** using HashMap.
- Deployed the web service using **Docker images** on **Kubernetes** across multiple cloud service providers.

Stream Processing with Kafka and Samza (**Learn more:** https://goo.gl/hL1DPK) Apr '17 - May '17

- Generated a stream of data using **Kafka producer** and made it available for a **Samza consumer** on AWS.
- Designed and implemented a solution for a **driver matching service like Uber** by joining and processing multiple streams of GPS data and driver data using the Samza API.

Social Network with Heterogeneous Cloud Backends (**Learn more:** https://goo.gl/hAp5Md) Feb '17 - Mar '17

- Modelled, populated and deployed both SQL and NoSQL databases in a social network web service context.
- Employed a graph database in **HBase** to enable searching for friends, a **MongoDB** database for comments, likes and user wall and a **MySQL** database using RDS for authentication on AWS.
- Extended a distributed key-value store with **strong and eventually consistent** replication schemes.
- Implemented an **input text predictor** by building a probabilistic language model and later optimizing it with Elastic Cache for **Redis**.

Mutual fund trading application & Web service (**Link:** https://goo.gl/7kkq2W) Jan '17 - Feb '17

- Led the software design and implementation of the **MVC** components and deployment on the AWS Cloud.
- Identified and rectified **race conditions** occurring due to multiple users working on common data.
- Secured the application by using SSL, one-way hashing, and cleaning request parameters.
- Implemented a web service handling 10,000 RPS and created **load tests** using artillery.