SHIVA REDDY KOKILATHOTA JAGIRDAR

shiva.reddy.kj@gmail.com | **+1 (312) 678-9776** | **github/shiva-reddy** | **linkedIn/shiva-reddy-kj** |

EDUCATION

University of Illinois at Chicago (UIC), Chicago, IL — Aug 2019 - Present **(Anticipated May-2021)**
Master of Science in **Computer Science.** — *GPA: 4/4*
Machine learning, *Android development*, *Web and network security*, *HCI*

Amrita school of engineering, Bangalore, India — Aug 2012 - Jun 2016
Bachelor of Technology in **Computer Science** — *GPA: 8.46/10*

TECHNICAL SKILLS

- **Programming Languages:** Java, Javascript, Python, C++ and Golang.
- **Platforms, web-development and databases:** Java EE,HTML/CSS, Android, Django, nodejs, Reactjs, MYSQL, MongoDB and DynamoDB
- **Devops and deployment:** Heroku, Google cloud platform, Firebase, AWS services (S3, API Gateway, Batch, Lambda, VPC and Cloud-Formation)

PROFESSIONAL EXPERIENCE

Amazon AWS, CodeGuru Reviewer — Seattle (Remote)
Software development Engineer intern - Full stack — June 2020 - August 2020

- Built an end-to-end service that allows users to tag Machine learning datapoints with properties, and generate custom ML datasets based on data point properties rules. Involved in the full development life cycle, from initial design and prototyping, to development, testing and release. Led discussions with stakeholders to discuss the overall product vision.
- The backend **REST API** service was built using **AWS Lambda (Java)** Compute, **API Gateway** and **DynamoDB**.
- The UI portal was built using **ReactJS** for page rendering with Polaris(internal) stylekit for styling, and **AWS Lambda** and **AWS S3** for serving UI content.

Chargebee — Chennai, India
Software engineer - Full stack — Jan 2016 - May 2019

- Built services for the *Integration* and *E-commerce* teams using **Java EE** and **MYSQL** databases, on top of internal web frameworks.
- Responsible for collaborating with product and design teams to build features and software modules for the subscription billing platform Chargebee. Designed, developed and maintained :-
  - The recurring delivery management module for the product. The module is a key market differentiator for the product and is currently being used by Chargebee's customers to service *thousands of deliveries* every day.
  - The revenue recognition reporting system in a team of 3, to automate the revenue recognition workflow for Chargebee's customers.
  - An internal **REST API** based integration framework to integrate Chargebee to other web apps. Also used the framework to integrate Chargebee to accounting platform Xero. The framework has been used extensively by multiple teams to support over 10 other core integrations.

PROJECTS

LoopedIn *github.com/LoopedIn/loopedin-web* — Feb 2020 - May 2020

- A chat-focused social networking platform built using **ReactJS** for UI, **Redux** for client state management, **MongoDB** as backing store, **Express + NodeJS** for backend server and **Firebase** for Authentication.
- Used **docker** containers with **Kubernetes** to setup *continuous development* and automatically deploy the service on the **Google cloud platform** stack. *Continuous integration* pipeline was implemented to run unit tests using **Github Actions**.

ARify *github.com/shiva-reddy-uic/arify-android-client* — November 2019

- Developed an AR **Android** mobile application which lets the user build dynamic **Augmented Reality** workflows. The app uses **Viro ARCore** for rendering 3D objects on image detection. The app's UI follows the latest design principles of material design.
- Developed a **Django (Python)** web server deployed on **Heroku** which lets the user upload a 3D object and allow them to link it to the image. The server also exposes a **REST** API, to be used by the android app. Used **sqlite-db** for data storage and **Amazon S3** web service for file storage.

Stackoverflow tag prediction — November 2019

- Worked in a team to build a machine learning classifier (**Python**, **Pytorch**) to predict tags for stackoverflow questions using datasets from Kaggle. Logistic regression, SVM and Neural network classifiers were used.

Mix network implementations *github.com/nkprince007/mix-networks* — Feb 2020 - April 2020

- Developed a mixed network proxy server implementation for **GoLang** which uses several mixing strategies to randomize the order of requests in a TOR network and defends the privacy of the network when under attack.