Kemal Ahmed

7 Model Avenue, Toronto, Ontario

(416) 333-4587

kemalahmed@protonmail.com

linkedin.com/in/kemalahmed

github.com/goatandsheep

Summary

> MEAN stack builder

> Frequent Hackathon Mentor

> Accessible tech advocate

> SAFe Agile certified

Recent Contracts

Front end engineer
Deloitte
[Nov 2019 to present]
> Built Vue.js app in a team of 15 for large scale project
> Internationalized components in English and French
> Made components accessible to WCAG 2.0 A

Freelance Software Engineer
BeatCaps
[Sept 2019 to Oct 2019]
> Built demo for deaf music accessibility tool in **Vue.js**
> Setup Continuous Integration on **Microsoft Azure**
> Generated sign language data for **PyTorch** translator
> Collaborated with Canadian Society for Deaf

Software Engineering Consultant
Vibrant New Media
[June 2018 to Sept 2019]
> Remote consultant for US video development company
> Periodically contracted time to do contracts for AWS
> Frequently built small projects to determine feasibility of various toolchains and emerging AWS services

Amazon Web Services
[Jun 2019 to Aug 2019]
> Built general purpose video-streaming app
> Designed and built iOS app using **React Native**
> Built web app mirror to mobile app using **Vue.js**
> Integrated app with Content Management System

CBS
[Jan 2019 to Jun 2019]
> Designed and built basic Content Management System for managing TV Series hierarchies in **Vue.js**
> Deployed to ECS using **Docker** scripts and buildspecs
> Experimented with Aurora (SQL) / Amplify (GraphQL)

Amazon Web Services
[Jun 2018 to Dec 2018]
> Built captions management system in **Vue.js**
> Designed components consistent to AWS styling
> Configured **AWS** services for hosting and deployment
> Published **open source** library to generated WebVTT
> Used Cognito, AppSync (GraphQL), Cloudfront, and S3

Front End Developer
HSBC
[Mar 2018 to May 2018]
> **React**, JavaScript, Test-driven development
> DIY Dev Ops / release environment
> Wrote tests: E2E, load, unit tests using Karma, **Selenium**

Employment

Full Stack Developer
AT&T / Quickplay
[June 2017 to Mar 2018]

> Rewrote Content Management System for videos in native **Polymer.js** and **Typescript** in team of 4
> Built and styled UI components in HTML and CSS
> Maintained legacy **Angular.js** app
> Started initiative to get direct feedback to developers from users to encourage user-centric design

Full Stack Developer
Communote
[April 2014 to May 2017]

> Educational resource-sharing website that rewards authors like YouTube for Class Notes
> Initially built in Tornado (**python**'s node.js equivalent) and **CouchBase**
> Revamped project in express / **node.js** (javascript), **Vue.js** (like **react**), and **MongoDB**
> First iteration over 900k page views, 4k documents
> Worked in incubator for 9 months

Contract Developer
Aiva Labs
[Oct 2016 to Dec 2016]

> Developed **responsive** HTML pages based on mockups
> Full **PHP** Stripe Suite integration with **Silex**
> Developed web scraper to generate colour scheme database to train **machine learning algorithm**

Software Developer
Freelance
[Mar 2017 to April 2017]

> designed name, branding, layouts using **Google's Material Design principles**
> Built static websites in **Angular**, **Bootstrap**, **PHP**
> Did SEO audits and fixed sites based on suggestions

Education

BEng, Software
McMaster University,
[Class of 2016]

> Support Vector Machines, OpenMP, Markov Chains
> Built notes repository reaching over 900k views
> Worked with Dean to bring hackathons to classes

Projects

Data Scientist
Airbnb Scraper
[Feb 2017 to June 2017]

> Developed a model to determine if an Airbnb listing is fake
> Tested it by scraping Airbnb using Tornado (Python)
> Categorized data to avoid false positives

Web Developer
Autopark
[March 2017]

> Built web interface as simulator for parking gate using **firebase**
> Integrated Square Payments reader into an **Android** app using native **Java**
> Won 2nd place at Ryerson hackathon

Webloops
[Sept 2015 to April 2016]

> Capstone: social YouTube playlist maker and player
> Full engineering documentation by group of 5
> **Python** (django) controlled **MySQL** on **docker**
> continuous integration using **circleci**