

TIMOTHY MILLER

FULL STACK DEVELOPER

PROFESSIONAL PROFILE

At Prepare Software, I am currently working on a government contract with CDC, Deloitte, & CACI for Full Stack Web Development & Data Science. In addition, I mentor students in Software Engineering & Web Development @ Thinkful & Springboard. I maintain a personal blog with my side projects, such as an iMessage scheduler . My blog can be found here:

timknowsbest.com

JAVASCRIPT

- Released a MERN stack template for making modern web apps (persistent data store, pretty text formatting, responsive components, optimized images, CDN integration, Progressive Web App support, and SEO where applicable) to open source. I added GraphQL, the Ant Design component library for React, JSX in Markdown, and TypeScript configs for webpack. In the documentation, I demonstrated usage of either PostgreSQL or MongoDB.

More information on the template can be found here: timknowsbest.com/deploy-mern-stack-to-production

- Created a personal blog based on my MERN stack template using Ghost CMS for hosting and editing the content. I composed GraphQL resolvers for accessing the Ghost CMS REST API to lower the number of round trips a client needs to make with the content server to improve load times when network latency is high. I wrote a Next.JS config to render dynamic URLs based on the article title. The landing page for the blog is here: timknowsbest.com

- Refactored the Python codebase for crypto-kill-switch to TypeScript. I sped up the runtime of algorithm which determines the optimal trading pairs to achieve a desired crypto to crypto trade by implementing Dijkstra's algorithm and weighting the edges based on exchange rates. The open source portions of the codebase can be found here:

github.com/timothymiller/crypto-kill-switch

- Designed a secure onboarding process as a single page app for Tendies Switch by using Ad-hoc wi-fi, Node.js, and React.

TIMOTHY MILLER

FULL STACK DEVELOPER

**TYPESCRIPT / D3.JS / DATA VISUALIZATION / CSS / REACT +
PYTHON3 / SPACY / NLTK / NLP / MONGODB / BIG DATA /
CONCURRENCY**

I created the web app and data pipeline for the Horizon Scanner project at the Centers for Disease Control as part of my company's government contract. Horizon Scanner is an experimental project to identify trending public health concerns.

An example question Horizon Scanner can answer is:

- Given a set of anchor terms, what are the top 100 most popular bi-grams from Reddit for July 2019.

Extracting this answer from a large data set is costly in terms of compute time. To allow real-time queries for arbitrary time periods & always changing anchor term sets inside the web app, I precomputed and cleaned the entire raw data source using natural language processing algorithms such as lemmatization & part of speech detection.

Horizon Scanner ingests all the Reddit posts & comments for a specified time period & applies a Natural Language Processing technique called lemmatization to extract a set of dictionary terms from a given post or comment. If this set overlaps with the anchor terms, then we consider that post relevant to public health. Using the set of dictionary terms, I then generate all combinations of n-grams (uni-grams, bi-grams, & tri-grams). An n-gram is a phrase containing n words. For example, the most popular bi-gram for the month of July 2019 is "get work". The runtime for lemmatizing a large data set is quite large, so I parallelized the lemmatization process by implementing the multiprocessing Python module. By doing so, I was able to make full use of all 16 cores & 32gb of RAM on the lab server I was provided. The runtime for processing one month worth of data went from over 17 hours to 90 minutes. My work demonstrates the latest Computer Science best practices in NLP, concurrency, and big data.

TIMOTHY MILLER

FULL STACK DEVELOPER

For the web app, I used an open source template, created by myself, which provides a React project using Next.js which is preconfigured to support Progressive Web App standards, Ant Design 4.0, scores 100 in all categories of the Google Lighthouse benchmark, includes examples for writing serverless Lambda functions as API endpoints, and supports server-side rendering. Further, by using Preact in production, I reduced the web app's bundle size to 20kb.

The data is visualized using D3.js. Popularity of a given time period is shown on a line graph with daily, weekly, and monthly granularity. The user can make changes to the anchor term set, as well as click the checkbox by a list of popular terms to add or remove data from the line graph. The website loads instantly, implementing the stale-while-revalidating strategy to make queries to our API server & return the requested set of data. CSS styling is done using styled-jsx to inline CSS styling & minimize round trips to the server.

This project is ongoing, & I have 2 full time, plus one part time data scientist working on top of the data pipeline I created, while I continue iterating on the web app.

CODE REFACTORING

Refactored the codebase for a cryptocurrency internet of things device for handling crypto trading from Python to TypeScript. I sped up the runtime of the algorithm which determines the optimal trading pairs to achieve a desired crypto to crypto trade by implementing Dijkstra's algorithm and weighing edges based on exchange rates.

PYTHON

- Wrote a dynamic DNS client for Cloudflare using Python 3. The source code for the project can be found here: github.com/timothymiller/cloudflare-ddns
- Created a Python script with event listeners for connecting to a Raspberry Pi GPIO based hardware switch to transfer between risky crypto assets to stable-coins at the flip of a switch. The script uses the Binance Python API.

TIMOTHY MILLER

FULL STACK DEVELOPER

JAVA

- Created the business logic for over 30 Android apps (including educational games, SMS + XMPP messaging apps, and cache cleaners).
- Created a trading algorithm using the Interactive Brokers Java API to implement a swing-trading strategy given to me by an advisor. I refined the strategy by back-testing and using statistical analysis.
- Implemented Java 7 and 8 on all of the above projects. Refactored numerous Android apps to Kotlin.

PYTHON

- Wrote a dynamic DNS client for Cloudflare using Python 3. The source code for the project can be found here: github.com/timothymiller/cloudflare-ddns
- Created a Python script with event listeners for connecting to a Raspberry Pi GPIO based hardware switch to transfer between risky crypto assets to stable-coins at the flip of a switch. The script uses the Binance Python API.

MOBILE

- End to end ownership of Clean My Android, released as one of the first 10,000 apps on the Android market. Maintained this app for over eight years, remodeling the codebase every few years to match the latest Android Design Guidelines. Currently, Clean My Android has been overhauled to support the MVVM (Model, View, ViewModel) pattern using Android Jetpack libraries, such as Room, ViewModel, LiveData, and AppCompatActivity. The app's website is written in Vue.js and deployed as a static site: cleanmyandroid.net

TIMOTHY MILLER

FULL STACK DEVELOPER

For the web app, I used an open source template, created by myself, which provides a React project using Next.js which is preconfigured to support Progressive Web App standards, Ant Design 4.0, scores 100 in all categories of the Google Lighthouse benchmark, includes examples for writing serverless Lambda functions as API endpoints, and supports server-side rendering. Further, by using Preact in production, I reduced the web app's bundle size to 20kb.

The data is visualized using D3.js. Popularity of a given time period is shown on a line graph with daily, weekly, and monthly granularity. The user can make changes to the anchor term set, as well as click the checkbox by a list of popular terms to add or remove data from the line graph. The website loads instantly, implementing the stale-while-revalidating strategy to make queries to our API server & return the requested set of data. CSS styling is done using styled-jsx to inline CSS styling & minimize round trips to the server.

This project is ongoing, & I have 2 full time, plus one part time data scientist working on top of the data pipeline I created, while I continue iterating on the web app.

CODE REFACTORING

Refactored the codebase for a cryptocurrency internet of things device for handling crypto trading from Python to TypeScript. I sped up the runtime of the algorithm which determines the optimal trading pairs to achieve a desired crypto to crypto trade by implementing Dijkstra's algorithm and weighing edges based on exchange rates.

CONTACT

(404) 225-9084

tim.miller@preparesoftware.com

timknowsbest.com

Atlanta, Georgia