

Paul Olteanu

26 Willowbrook Rd, Markham, ON, L3T4W9

☎ 647-449-6597 | ✉ p.a.olteanu@gmail.com | 📄 github.com/PaulOlteanu

Work Experience

#paid

Toronto, ON

FULL STACK DEVELOPER

July - August 2017, January - April 2018

- Detailed full data collection project spec and timeline with CTO, specifying scope of versions 1 to 3, and user stories
- Architected parallelized job pipeline using **Elixir's GenStage** library for worker backpressure and rate limiting
- Scraped and processed **30,000+** users/day, discovering new influencers for creator relation to onboard
- Configured **AWS EC2** instances for staging and production use, and set up **Squid** proxy servers for request caching
- Led project updating multi-step form for clients to create campaign proposals, increasing amount of data collected from RFPs
- Key Technologies: **Ruby on Rails, Elixir, AWS EC2 & IAM, Postgres, Mongo**

Projects

Proplist

IMPLEMENTATION OF THE 'KEYWORD' API FOR BINARY KEYS

- Contributed to library adding binary (string) key support to keyword lists by re-implementing the Keyword API
- **Published** on Hex (Elixir's package manager), with **~1,000 installs**
- Implemented in **Elixir**

Shamrok

IMAGE SHARING AND VOTING SITE SIMILAR TO REDDIT AND INSTAGRAM

- Allows for hosting of static images as well as gifs
- Supports upvoting and downvoting, and a variety of sorting algorithms
- Unique URLs are created for every image to allow for image embedding on other sites
- Implemented in **Python** using **Flask, Postgres**, and hosted using **Heroku, AWS S3**

OMFPGA

LOGIC GATE SIMULATOR FOR THE ONION OMEGA2

- Allows for unlimited number of gates (AND, OR, XOR, and NAND) and connections
- Gate and net configuration done through CSV file
- User-configurable logging & debugging support
- Implemented in **C**

PVCS

VERSION CONTROL SYSTEM

- Supports commits with messages, reverting, and full logs
- Stores commits as diffs to reduce storage footprint, with complete backups every n commits to increase revert speed
- Implemented in **Python** using **Click**

CHIP-8

AN EMULATOR FOR THE 35 OPCODE CHIP-8 INTERPRETED LANGUAGE

- Support for all 35 opcodes + legacy mode for certain opcodes with different possible interpretations
- Interprets binary .ROM files (CHIP-8 machine code)
- Implemented in **C** using **SDL2**

Education

University of Waterloo

CANDIDATE FOR B.AS. IN COMPUTER ENGINEERING. CURRENTLY 1B

Expected Completion Apr. 2022