Zoravur Singh

Student, Coder, Hacker. Let's work on something awesome.

zoravur.singh@uwaterloo.ca
github.com/zorvyy
647.287.1303

EXPERIENCE

ROSS Intelligence, Inc. May 2019 – August 2019
Software Engineering Intern Toronto, ON

- Worked extensively with the subscription microservice and client web app, adding functionality to view, modify, and cancel subscriptions.
- **Prototyped a new Python microservice** which gave users access to the ROSS web app without making an account. Used browser fingerprinting to uniquely identify and authenticate users, giving them a consistent experience across multiple visits. Deployed on **Kubernetes**.
- Built reusable UI components in **React/Redux**, and assisted with codebase migration to **TypeScript**.
- Collected and analyzed usage metrics (e.g. button clicks, page navigations, etc.) throughout the ROSS web app and website in order to maximize user retention.

SKILLS

Languages | JavaScript, Python, C/C++, SASS/CSS, Java, Scheme
Web Development | TypeScript, React, Redux, Node, npm, Jest, Webpack, Express, Django
DevOps | AWS, Docker, Kubernetes, Flux, Travis CI
Tooling | Git, Vim, Bash, CMake, Postgres, MongoDB
Machine Learning | PyTorch, pandas, NumPy/SciPy, scikit-learn

PROJECTS

VM | Final Project, Object Oriented Programming (Enriched)

- Created a fully featured clone of the text editor Vim in C++, with support for normal and insert mode, common commands, and macros.
- Used the **Flux design pattern** and **Reactive Programming** to vastly simplify the program structure.
- Received a mark of **96%** (class average 70%).

Twitter-Emotion | Tweet Emotion Classification

- **Trained a CNN** on a dataset of tweets to classify the predominant emotion of a given tweet.
- Used GloVe **word embeddings** trained on a Twitter dataset to better exploit the semantics of tweets.
- Used NLTK, PyTorch, and scikit-learn.

Gravity.js | A highly accurate real-time gravity simulation in TypeScript

- Implemented particle path tracking, a number of **numerical integration methods**, and an intuitive UI for controlling simulation parameters.
- Used **multithreading** through Web Workers and HTML5 Canvas to achieve optimal performance.

Stream-transduce | Functional programming with Node streams

- Published a library designed to produce and consume Node.js streams in a functional paradigm.
- Ensured interoperability with other functional programming libraries, such as Ramda, by implementing the transducer protocol.
- Used best practices in documentation (JSDoc), testing (Jest), and semantic versioning (through npm) to ensure usability.

EDUCATION

Honors Bachelor of Computer Science Co-op (Expected) 2018 – 2023
University of Waterloo