Lee Lazarecky

Ann Arbor, MI

leelazarecky@gmail.com | https://github.com/Druage | leelazarecky.com

Work Experience

FordLabs, Senior Full-Stack Software Developer **Sep. 2017 - Present**

- Led development and maintenance of multiple open source and inner source software projects, designing the UI / UX and implementing the software code changes necessary for constant uptime and usage.
- Used test driven development as well as Agile development through the Extreme Programming methodology.
- Tested software features using Selenium, Cypress, and unit level testing frameworks.
- Wrote an inner-source Authentication and Authorization server based on the OAuth2 and OpenID specifications. Connected this auth server to various email relays in order to send verification codes to users.
- Worked on the software hiring board, interviewing potential candidates for hire, accessing their skills using paired programming.
- Built an open source distributed Retro tool for usage within the company and for users who want to stand up and host their own instance.
- Worked on the Ford Commercial Solutions project which utilizes MapBox to track vehicles across a geographic region and display on a map.
- Setup automated deployment environments for various applications using Jenkins, TravisCI, and Github Actions
- Worked in a variety of backend and frontend languages and tech stacks, Java, Kotlin, Javascript, SpringBoot, Express, React, Angular
- Integrated with software databases using Prisma, Spring JPA, and integrated viewers.
- Wrote Terraform scripts to build setup on-prem infrastructure such as databases, virtual machines, email relays.
- Built various static sites using Gatsby and Next.js while tying into NetlifyCMS for producing the page content and hosting on CDN's such as Vercel

Valeo, Research and Hardware Development Co-Op **Jul. 2016 – May 2017**

- Worked independently on developing internal software for automating the generation of camera calibration files.
- Created GUI software written with Qt and C++ that was used as a frontend application for the generation of calibration files.
- Worked independently on designing automated testing programs, using the Microsoft Windows C API and C++.
- Flashed calibration files over UART and CAN protocols, using the Vector Canoe software tool, to an embedded VPM which was used to test the vehicle cameras for various calibration errors, such as improperly rendered guidelines, camera bleeding, and image discoloration.
- Wrote applications to automate the testing of camera firmware files.

United State Air Force, Aircraft Armament Systems Journeyman **Aug. 2009 - Feb. 2016**

- Troubleshooted A-10 aircraft computer systems and diagnosed potentially damaging issues affecting the weapon systems, using wiring schematics and technical manuals.
- Repaired and reinstalled faulty computer systems such as wiring, connectors, and plugs, that were used on the A-10 weapon systems.

Passion Projects, Open Source & Inner Source Project Engineer / Maintainer

AuthQuest **Jan. 2019 - 2020**

- An open source authentication and authorization server, designed to serve as a free alternative to services like Auth0 and Okta.
- Uses Kotlin and SpringBoot on the backend and a Vue.js frontend.
- Provides users with a default and highly customizable login / sign up page.
- Follows OAuth2 and OpenID standards

RetroQuest **Jan. 2018 - 2022**

- Allows users to effectively facilitate agile retrospectives across distributed teams.
- Implemented websocket based CRUD operations so the application is constantly up to date.
- Designed a user interface that was put together with mobile usage in-mind.

Multi-System Video Game Emulator, *Phoenix* **Dec. 2013 - Present**

- Led a team of developers in implementing a multi-threaded and multi-process video game emulator.
- Utilizes a GUI which was implemented using QML and C++ and a SQLite database for library management.
- Designed an audio and video pipeline for rendering audio and video streams for consistent 60 fps.
- This program is split up into multiple processes, using local sockets and shared memory for IPC communication.
- Video data is passed between processes in a region of shared memory and constructed into an OpenGL texture for rendering.

Chip8 Interpreter

- Interprets and executes instruction sets from binary files compiled specifically for the Chip8 virtual machine, by re-implementing the Chip8 interpreter in C++.
- Handles all the Chip8 op-codes and displays the emulated video game using the Qt QML scene graph and OpenGL 2D textures.

Education

Wayne State University, Detroit B.S., Computer Science Graduated: **May 10th, 2017** GPA: **3.46** / 4.00

Skills

- JavaScript, Typescript, Kotlin, Python, SpringBoot, Next.js, Gatsby, TailwindCSS, React, Vue.js, Express, Koa, C++, QML, SQL, Heroku, PCF, Digital Ocean, NetlifyCMS, Jenkins, Github Actions
- Paired programming, Test Driven Development, Continuous Development & Integration