Alexander Werner

alexwerner9@gmail.com | github.com/alexwerner9 |
linkedin.com/in/alexander-a-werner/

Education

New York University New York, NY
Bachelor of Arts in Computer Science, Minor in Mathematics. *Class of 2024*

Experience

TDS Telecom Madison, WI (remote)
Software Engineer *Sep. 2022 -- Present*

- Led the project to rewrite our order intake Django backend in order to validate the migration of over 5 million customers from one billing system to another, including comprehensive data reporting, extensive automated tests and rollback.
- Decreased total time to deploy by 50% by streamlining our Jenkins CI/CD process by combining and caching OpenShift image builds.
- Maintained and upgraded the Django/Flask-based back-end of microservices responsible for translating, configuring, and processing internet service changes, including rebooting modems with the new configurations, tracking IP assignments and monitoring modem uptime.
- Designed and implemented a new set of Bash and Python scripts for the synchronization of our Oracle, MySQL and Couchbase datastores between development and production OpenShift containers.

Software Engineer Intern *Mar. 2022 -- Sep. 2022*

- Decreased on-site appointment times by redesigning and implementing our Django backend to allow configuration changes to be scheduled for a future date.
- Decreased the strain on datastore by creating system-wide cron jobs to delete stale documents.

Projects

ThePlaylistMaker.com | *NodeJS, Express, MongoDB, React* *Dec. 2023 - Present*

- A full-stack website to allow visitors to add a song to their friend’s playlist on Spotify, without ever having to interact directly with Spotify.
- On the day of release, got over 2,000 hits and songs added to my playlist alone, leading to over 150 hours of music.
- Interacted with the Spotify API from the backend, including a custom search on my website that queries Spotify search itself, so that the user does not have to redirect to Spotify.
- High focus on ease of use across mobile and desktop.

RPi Remote Control | *Bare-metal C* *Dec. 2021 -- Jan. 2022*

- Wrote a kernel for the Raspberry Pi that could work as a universal remote control.
- Kernel was written entirely in C with no external libraries. Kernel included using the built-in timers and interrupts in order to correctly use the universal infrared transmission protocol.
- Integrated the kernel with the GPIO pins in order to send signals to a circuit which would either output the infrared signals to an infrared LED or read signals from a remote in order to capture those signals to be used to control the devices.

Technical Skills

Languages: Java, Python, C/C++, JavaScript, TypeScript, C#, GLSL/WebGL, HTML/CSS, SQL/N1QL
Frameworks: Node.js, Express, Flask, Django, React, .NET
Developer Tools: Git, OpenShift, Jenkins, Docker, VS Code, Eclipse, Tox, Mocha
Technologies: Linux (CentOS), Selenium, AWS, MongoDB, RabbitMQ, Couchbase, Unity, Nginx