775.434.8660  
[resume@snowskeleton.net](mailto:resume@snowskeleton.net)

Isaac Lyons

Durham, NC

[github.com/snowskeleton](https://github.com/snowskeleton)  
[linkedin.com/in/isaac-lyons](https://linkedin.com/in/isaac-lyons)

# Experience

**Desktop Support Technician** Net Friends, Inc – 10/2024 – Present

* Onboarding/offboarding users, including configuration of Active Directory, Azure AD, Office 365, and Exchange Online
* Manage SharePoint sites, including coordinating guest access and proper syncing between cloud and device
* Mitigate compromised user accounts and ensuring users can regain access in under 10 minutes
* Configure and manage Windows and macOS devices through Intune and Addigy, and virtual machines through Azure, AWS, and Hyper V
* Configure and troubleshooted Windows systems remotely through N-Central Take Control, and in-person
* Manage and updated DNS records, including A, CNAME, SPF, DKIM, and DMARC records through Cloudflare and others
* Update and reference knowledge base articles for common issues
* Surpass average performance metrics of the team by more than 300% on a regular basis

**Desktop Support Technician** Setton Consulting – 09/2023 – 12/2023

* Developed and deployed a Python Slackbot that improved security by finding user with incorrect MFA configurations
* Edited Powershell scripts for Intune deployment
* Remotely configured and troubleshooted Windows and macOS devices through Connectwise ScreenConnect
* Administered Office 365, Intune, and Active Directory
* Configured DNS records and domain name registration for clients

**Senior Tech Support Associate** Infosys Inc – 05/2022 – 09/2023

* Developed Javascript browser extension for assigning jobs to agents in queue
* Developed AutoHotkey scripts to automate lengthy tasks with IBM Mainframes
* Configured fiber routes to customer addresses through mainframe interface
* Coordinated process changes with hybrid team

**Desktop Support Technician** SureLock Technology – 03/2021 – 03/2022

* Designed and developed custom ticketing system in Python with Django and Docker to track device repair process, eliminating “lost devices” and accurately mapping incoming parts to necessary repairs
* Developed iFixit.com guides for device repair still in use by ~150 users per month
* Resolved tier 1 and 2 support requests for internal and external customers
* Created, updated, deleted Active Directory users, passwords, and permissions
* Imaged and repaired end-user devices (desktops, laptops, phones, tablets)
* Oversaw asset management and procurement
* Managed Linux and Windows VM through HyperV

**Repair Center Team Lead** Service Quick – 05/2019 – 09/2020

* Designed and developed AutoHotkey scripts to search customer records, automate call note templates, and search knowledge base for known issues
* Reduced customer record search time from >60 seconds down to <5 seconds by indexing the proper search field
* Handled highest call volume of all employees, with the highest customer satisfaction rate

# Skills

* **Homelab:** Maintain a Proxmox server hosting 4 Ubuntu virtual machines, 1 remote hardware Ubuntu server, and 1 cloud Debian server through Linode, and connect everything with Tailscale VPN to provide secure access while remote
* **Programming Languages:** Python, Swift, Bash, JavaScript, AutoHotkey, Golang, Ruby
* **Tools and Frameworks:** GitHub, Git, Docker, VSCode, Xcode, Django, Pytest, PyPi, SwiftUI, CoreData, SwiftData, HyperV, Linux, macOS, Windows, TailScale

# Projects

## iOS Apps

[**snowskeleton/acs-contacts**](https://github.com/snowskeleton/acs-contacts) 2024

* Written entirely in SwiftUI, ACS connects to a third-party contacts database for churches to facilitate members reaching out to each other. It connects to two different APIs and maintains an offline database kept in sync with the cloud.

[**snowskeleton/OSTRich**](https://github.com/snowskeleton/ostrich-ios) 2024

* Written entirely in SwiftUI with the new SwiftData database framework, OSTRich (Open-Source Tournament Reporter) is a tournament companion app that helps you receive pairings, submit matches, and scout players.

[**snowskeleton/Snowclock**](https://github.com/snowskeleton/Snowclock) 2023

* Developed in SwiftUI with CoreData, Snowclock is a declaritive alarm clock that simplifies having a consistent wakeup routine with a variable schedule.

[**snowskeleton/I-Got-Gas**](https://github.com/snowskeleton/I-Got-Gas) 2020

* My first iOS app, written entirely in SwiftUI with CoreData, I Got Gas helps track your car’s expenses and gives you a breakdown of how much it costs to drive per mile, and reminds you of upcoming service intervals either by calendar or mileage.

## Commandline utilities

[**snowskeleton/atomic-empire-cli**](https://github.com/snowskeleton/atomic-empire-cli) 2024

* Written in Python using the SQLAlchemy database framework, this tool automates the process of searching for and buying Magic: the Gathering cards. Just give it a list, and let it do the rest!

[**snowskeleton/kumad**](https://github.com/snowskeleton/kumad) 2024

* Written in Go, kumad (kuma daemon) is a lightweight client for the [louislam/uptime-kuma](https://github.com/louislam/uptime-kuma) project. Its memory footprint rests 6MB, as opposed to the tool it replaced which took more than 200MB.

[**snowskeleton/ynam**](https://github.com/snowskeleton/ynam) 2022

* ynam is written in Python with minimal imports (just 1). Now deprecated, it served the purpose of importing transactions from the Apple Card credit card into YNAB budgeting tool.

[**snowskeleton/snowcrypt**](https://github.com/snowskeleton/snowcrypt) 2022

* snowcrypt written in Python and decrypts Audible audiobook files. The purpose of this program is to remove FFMPEG as a dependency, which makes installation and maintenance easier.

[**snowskeleton/noblility**](https://github.com/snowskeleton/nobility) 2022

* Written in Python using Django as a frontend and Docker to deploy, nobility is a repair tracking tool aiming to be as simple as possible. More complicated systems, like ServiceNow (paid) or OSTicker (open source), were far too complicated for the simple process of entering a device, noting parts ordered, and tracking repair completion.

## Open Source Contributions

[**mkb79/audible-cli**](https://github.com/mkb79/audible-cli), [**mkb79/Audible**](https://github.com/mkb79/Audible) 2022, 2024

* Implemented retry logic if requested download format fails
* Added additional metadata information to download
* Created date-range filter for arbitrary file downloads
* Fixed typos in source code

[**tinytag/tinytag**](https://github.com/tinytag/tinytag) 2022, 2023

* Provided examples of new file metadata test cases
* Added recognition for Audible audio files

[**mintapi/mintapi**](https://github.com/mintapi/mintapi) 2022, 2023

* Reworked documentation, reviewed pull requests, and provided community support as a key member

Plus documentation fixes for [**thewca/tnoodle**](https://github.com/thewca/tnoodle), [**capcom6/pyapns\_client**](https://github.com/capcom6/pyapns_client), and [**EskoSalaka/mtgtools**](https://github.com/EskoSalaka/mtgtools)

**View the most recent version of this resume on GitHub:** [**runty.link/resume**](https://runty.link/resume)