Yusef Napora

Experienced software engineer, technical writer, and system administrator.

1420 Preston Spring Ln.
Chapel Hill NC, 27516
USA

https://yusef.napora.org
yusef@napora.org
1-919-923-5109

Work history

- **April 2023 - present** **ShareFile** - *Senior software engineer (full stack)*
  - Began as contract position via **Revolution Technologies**. Converted to full-time in August 2023.
  - Developed a platform for integrating data from external applications with ShareFile's backend systems and web applications
  - Led development of React-based micro frontend to present a catalog of available integrations to users & display external data
  - Using Nest.js, developed microservice APIs for interfacing with external systems via an intermediary (iPaaS) service
  - Worked closely with iPaaS vendor to resolve issues and fast-track required features
  - Collaborated with internal teams to integrate external data with in-house asynchronous workflow engine
  - Led development of a "dynamic data service," which consisted of a multi-cluster router for MongoDB with event propagation via Kafka to trigger workflows based on data changes.
  - In November 2023, was tapped to join "kernel team" responsible for defining and owning the core services for ShareFile's internal microservice architecture, including authorization, resource storage, service-to-service API contract specification, service discovery, and data lifecycle management.
- **2019 - Feb. 2023** **Protocol Labs** - *Documentation engineer*
  - Wrote and maintained developer and user-facing documentation for **Web3.Storage**, **NFT.Storage**, **NFT School** and related projects
  - Dove deep into the internals of fast-moving projects to deliver tutorials, guides, and reference materials
  - Wrote many example applications and Solidity contracts
  - Presented about our APIs and products at developer conferences and workshops
  - Developed user-facing documentation for **libp2p**, the peer-to-peer protocol suite
  - Led specification and initial implementation of a new **channel encryption protocol** for libp2p
- **2017 - 2019** **Spotify** - *Senior backend engineer*
  - Wrote and maintained backend services and infrastructure for Spotify Analytics, a data platform which processed and segmented terabytes of listener data daily
  - Replaced critical python legacy services that everyone was scared to touch with modern, well-documented Java microservices and internal web applications
  - Re-architected data collection and processing practices to conform with GDPR regulations
- **2015 - 2017** **Mediachain Labs** - *Senior software engineer*
  - Wrote the **JavaScript implementation** of the **Mediachain protocol**
  - Was deeply involved in protocol design, research, and experimentation

  - Wrote much of the **user-facing documentation**
  - Led development of an iOS photo sharing & attribution app (unreleased due to strategy pivot)

Skills

Below are some of the skills that people have been happy to pay me for in the past. If you find yourself needing some of them, perhaps you'd like to **reach out**?

Programming languages

I have used many languages not listed below, but these are ones that I have taken especially seriously over the last several years.

- rust
  - My current favorite compiled language.
  - See **lumatone-rs** for an example of rust code I've written lately. Highlights include a custom MIDI driver for an esoteric keyboard, a "rust-native" GUI using **Dioxus**, and quite a bit of back and forth as I figure things out.
- JavaScript / TypeScript
  - Very comfortable with modern node.js & browser APIs. Deno-curious.
  - Strong with React, good with Svelte, can fake it with Vue.
  - See **lumachromatic** for a recent TypeScript React app. Highlights include WebMIDI, type-safe finite state machines, and a React UI made with `<svg>` elements and non-rectangular shapes.
- Nix / NixOS
  - It's weird, but powerful. I'm a fan.
  - I have a decent mental model for the Nix language, but still have much to learn about the standard lib & idioms.
  - See my **dotfiles**, which reflect my current understanding & preferences.
- python
  - An invaluable companion for data visualization, prototyping, experimenting, and many other things.
- Go
  - C with a garbage collector and coroutines is pretty good, actually.
  - Good grasp of the language and core idioms, but I have not yet used generics in a real-world capacity.
- Java / Kotlin
  - Strong grasp of the languages and core JVM platform.
  - Familiar with backend-focused library ecosystem.
  - Some outdated Android SDK experience, mentioned because I would enjoy a chance to renew it.
- Swift / Objective C
  - Strong grasp of both languages and the core Cocoa and UIKit SDKs.
  - Have made toy apps with SwiftUI, but I lack deep production experience.
  - Library / ecosystem knowledge is outdated, as I haven't focused on Apple platforms lately.

System and API design

I am experienced with REST and GraphQL API design and can offer thoughtful input on greenfield development, API evolution, schema design, and more.

I have built and maintained production systems that incorporate many loosely-coupled microservices, “big data” pipelines, message brokers, graph and document databases, and of course, good old SQL.

I will advocate for observability, instrumentation, and tracing for any project I have influence over, and I am happy to help set up dashboards, alerts, and other “visibility aids” that give confidence to operators of production deployments.

It’s a Unix system, I know this

I have decades of experience bossing Linux and macOS around. In the past few years I’ve also made it a point to run Windows at times, but I lean heavily on the Windows Subsystem for Linux and am not well versed in Windows internals.

Technical writing

I have worked for several years in a documentation-focused role, producing tutorials, example apps, how-to guides, API references, and other materials to make complex systems comprehensible.

Documentation is the human interface for code, and like all user interfaces, it’s difficult to do well. I think I have a knack for it, and I would be happy to help improve whatever sub-optimal documentation situation you may have found yourself in.

Education

Idiosyncratic.

After a frustrating elementary school experience, I began a self-directed homeschooling approach that helped kindle my enthusiasm for programming.

I started going to community college when I was fourteen as part of a bargain with my dad that allowed me to avoid high school. I later transferred to Warren Wilson College, a liberal arts school in the beautiful mountains of North Carolina. While this was a very pleasant experience and gave me an opportunity to read some excellent English literature, very little of my education there was relevant to my eventual career in technology.

After leaving Warren Wilson, I rediscovered my love of programming and eventually made a go at it professionally.

Links

If you are reading this resume in a format without native hyperlinks, you may find these URLs helpful:

Employers

- ShareFile: `https://sharefile.com`
- Protocol Labs: `https://protocol.ai`
- Spotify: `https://spotify.com`
- Mediachain: `https://mediachain.io`

Work products

- Web3.Storage: `https://web3.storage`
  - Documentation (stable): `https://web3.storage/docs`
  - Documentation (beta): `https://docs-beta.web3.storage`
- NFT.Storage: `https://nft.storage`

- NFT School: `https://nftschool.dev`
- Minty, an NFT minting example app: `https://github.com/yusefnapora/minty`
- Mediachain JavaScript implementation: `https://github.com/mediachain/aleph`
- Mediachain documentation: `https://docs.mediachain.io`

Personal repositories

- Lumachromatic - TypeScript tool for the Lumatone isomorphic MIDI keyboard
  - Live application: `https://app.lumachromatic.art`
  - Source code: `https://github.com/yusefnapora/lumachromatic`
- lumatone-rs - In-progress port of lumachromatic to rust
  - Source code: `https://github.com/yusefnapora/lumatone-rs`
- My Nix configuration - system definitions for NixOS and nix-darwin
  - `https://github.com/yusefnapora/nix-config`