

Akilesh Praveen

Email: akilesh.praveen@gmail.com | LinkedIn: <https://www.linkedin.com/in/akipraveen/> | GitHub: <https://github.com/AkiPraveen>

Education

University of Maryland, College Park (College Park, MD) - B.S. Computer Science

Dec 2021

Notable Coursework: Distributed Systems (Graduate), Parallel Programming, Advanced Data Structures, Algorithms

Experience

Founding Software Engineer @ Stealth Startup

Oct 2022 - Present

Fullstack engineer at a 5 person startup- rapidly building features to achieve product-market fit.

- Learned new stack: **NestJS**, **Prisma**, **PostgreSQL** & **Apollo GraphQL** & shipped multiple robust backend features with third-party integrations, while data model & specification changed rapidly prior to first deal closing.
- Rapidly shipped complex frontend features using **ReactJS**, **Redux** and **MaterialUI**, deployed on a **NextJS** frontend. Wrote first **Cypress** tests at the company to ensure quality of these deliverables.
- Projects include a **Twilio**-integrated appointment scheduling workflow, a **Mapbox**-integrated route optimization flow, an invoice reporting dashboard using **Recharts**, and a math-heavy trucking tariff assessment workflow.

Fullstack Software Engineer @ Athelas

Aug 2021 - Oct 2022

Fullstack engineer leading cross-functional engineering efforts across Enterprise, Growth, and CareML teams as the company transitioned to offering enterprise-level products.

- Redesigned package shipping automation flow using **Python** and **Flask**, increasing shipping throughput by ~1000%
- Designed & built out Athelas Enterprise API for first Enterprise-level deal in **Python/Flask** (<https://athelas.readme.io/docs>)
- Bespoke **Typescript** & **Vue** health data visualization suite project focused on meeting customer needs through rapid iteration. (**Vue** and **Typescript** frontend, **Flask**, **Python**, **SQLAlchemy** and **PostgreSQL** backend)

Software Engineer Intern @ Athelas

Summer 2021

Backend & firmware engineering intern, learning the role of a software engineer at a high-growth startup.

- Designed & implemented asynchronous, cadenced batch shipping of RPM related products for 25,000+ patients using **Python**, **Flask** and **Celery** deployed on **GCP** alongside exhaustive logging & error monitoring.
- Worked directly with multiple vendors in China to negotiate specifications for, then architect & implement backend and database changes using **Python**, **Flask** and **PostgreSQL** to support new Athelas RPM hardware.

Software Engineer Intern @ HP Enterprise

Summer 2019 & Summer 2020

Software engineering intern on various projects for QAOps, System Test & PerfOps teams.

- Won **HPE Best in Class Scholarship** & **1st Prize** at **HPE Intern Project Fair**.
 - Created tooling to manage orchestrated **docker** containers that would run different test suites on HPE Storage arrays
 - **~115% more efficient** in using system resources for testing than existing VM-based test workflows
-

Projects

Golang-FUSE Distributed Filesystem

Nov 2021

- Log-structured distributed filesystem written in **Golang** at the user level, thanks to **Bazil FUSE**.
- Uses **Merkle Trees** to efficiently determine inconsistencies between nodes, leverages **gRPC** & **Google Protobufs** to serialize and communicate this metadata.
- Implements **Rabin-Karp Fingerprinting** for content-defined chunk boundaries as in [A Low-bandwidth Network File System](#) (Muthitacharoen et. al).

CMSC389E: Digital Logic Design through Minecraft

Jan 2020 - Dec 2021

- Head lecturer for a **digital logic design & theory university course** at the University of Maryland, College Park.
- Guided students as they built a 2-bit computer in the hit sandbox game '*Minecraft*' complete with ALU, ROM, and RAM. Designed and built an online textbook to demonstrate concepts through online demos. (<https://www.cs.umd.edu/class/fall2021/cmssc389E/>)
- Executive Director of STICS (<https://stics.umd.edu/>), enabling university students to design and teach courses at UMD.
references available upon request