Shreyas Seshadri

shreyasseshadri@gmail.com shreyasseshadri shreyas-seshadri ☎+1 (213) 569-2234

Education

University of Southern California **LA, CA**
Masters in Computer Science 2022–2024
National Institute Of Technology, Karntaka **Surathkal, Karnataka**
Bachelor of Technology in Information Technology 9.16/10 GPA 2016–2020

Skills

Frameworks Node JS, React.js, Typescript, Mocha, Cypress, Protractor, Tensorflow, Scikit Learn, Django
Languages C, C++, JS, Python, Java, Unix-Shell Scripting
Technologies Kubernetes, Docker, Git, Kafka, MySQL, Azure Functions, AzureSQL, LDAP, GraphQL

Work Experience

Walmart Global Tech **Bangalore**
Software Engineer 2 Aug 2020–Apr 2022

- Worked on a workflow management application used for compliance purposes in Walmart stores around the world
- Established CI-CD pipelines for services using **Docker**, and WCNP (Walmart's own pipeline orchestrator on top of **Kubernetes**). The services were built using **Angular**, **Node JS**, **GraphQL**, **MySQL**
- Co-ordinated with cloud engineering team to integrate **Azure Blob**, **Memcached**, **AzureSQL**, as part of the resource migration effort, cutting down cloud costs by 22%. Received the "**Bravo Award**" for the same
- Led the effort to implement scheduling for an email notification service, using **Azure Functions**, **Kafka**, **LDAP** servers
- Designed the high level architecture, DB schemas and LLDs for APIs for an automatic import flow. A scalable event-driven architecture using **Kafka** was built capable of handling transactions 1000 times more than the previous client side architecture

Walmart Global Tech **Bangalore**
Intern May 2019–Jul 2019

- Worked on modifying existing endpoints, developing and testing new endpoints on the proxy server used in Parts Fitment team, using **Hapi JS** and **Mocha**.
- Worked on unit and integration testing of the User Interface of Fitment modal using **Cypress**.

Projects

QuickByte Jun 2022–Jul 2022

- Implemented a Key Value store, built on **c++** where the underlying storage exists either in-memory using a BST or on-disk using file storage
- Optimised the BST with the fewest amount of simultaneous latches as possible, to allow multiple threads to mutate the BST simultaneously without blocking rest of the tree
- Tested it by spawning 10,000 threads which issue upserts, retrieves, and delete calls on the BST, taking an average time of about 257ms (averaged over 100 times)

Music Jump Apr 2020–May 2020

- Developed a full stack project aimed at integrating different music streaming platforms. It unifies users' current collection of albums, playlists at one place, and also migrates playlists and albums from one service to another
- Built a backend with **Express** framework, and **redis** for cache, *frontend* with **React.js** with **typescript**
- Integrated application with spotify's OAuth, playlist, Album APIs, and built a chrome browser *extension* for collecting data from Amazon music service

Task Scheduling Nov 2019–Jun 2020

- Designed a novel algorithm to classify tasks based on the intensity of the type (memory, compute, IO) of resources required. It also categorizes the virtual machines in the cloud environment based on their current workload using **python**, publishing a paper in *IJAST Journal*
- Broadened the algorithm to also schedule the task so as to optimally utilize the available resources as part of the academic **major project**

** All texts in italics, and project headings are hyperlinks*