

SALIL NADKARNI

New York, NY | +1 (248) 949 9898 | salnad02@gmail.com | github.com/salnad | linkedin.com/in/salnad | US Citizen

EDUCATION

University of Michigan

Ann Arbor, MI

Bachelors of Science Engineering in Computer Science with a Math Minor

Expected Graduation May 2023

- GPA—3.90 / 4.00, *Honors*—Briggs Scholar (\$12,000 scholarship), Dare2Dream Mayleben Venture Fund Recipient
- *Relevant Coursework*—Distributed Systems (EECS 491), Operating Systems (EECS 482), Bayesian Data Analytics (STATS 451), Introduction to Algorithms (EECS 477), Data Structures and Algorithms (EECS 281), Theoretical Probability (MATH 425)

EXPERIENCE

Slack (Quip)

San Francisco, CA

Software Engineer

Aug 2023 - Present

- Reestablished Android development and mobile client release capabilities by updating outdated documentation, reducing non-fatal crashes by 80% through resolving longstanding issues, and addressing significant security vulnerabilities by mitigating intent and task mechanism risks using tools like Postman and an APK Decompiler.
- Prototyped and contributed to the design of cursor and selection presence within Quip and Canvas editors, using Typescript, React, and CRDT libraries to ensure best practices. Lead this feature with a team of 2 engineers to GA
- Improved client release processes by developing a risk assessment plan for API key permissions and collaborating with teams to update and document certificate refresh procedures.

Software Engineering Intern

May 2022 - Aug 2022

- Refactored Quip's Identity Management (SCIM) API using OpenAPI, adding request validation for 10K+ daily requests and updating REST endpoints for SCIM 2.0 compatibility with 3rd party apps like Okta.
- Developed a Python-based documentation pipeline for Quip's SCIM API, featuring full-text search, dynamic request/response samples, and versioning for current, deprecated, and internal methods.

Snorkel AI

Redwood City, CA (Remote)

Backend Software Engineering Intern

Jan 2022 - Apr 2022

- Designed an asynchronous job using Python, Dask, and Redis to automate the upload functionality of PDF documents into AI SaaS application, reducing upload time from 5+ minute Jupyter notebook script to <30s
- Enhanced data labeling module by batching data labeling functions and introducing retry-on-failure for unsuccessful batches

Google

Mountain View, CA (Remote)

Software Engineering Intern

May 2021 - Aug 2021

- Developed Ads Simulation tool for 10+ member Geotargeting team to reduce testing of ad models from >1 week to <7 hours
- Programmed a performant, multithreaded C++ binary with capabilities of processing 100K+ logged requests that transforms upstream data, calculates key metrics, randomly samples candidates, and highlights differences between two ad models
- Produced detailed design document, demoed tool to 60+ engineers and published documentation to team website

University of Michigan Computer Science Department

Ann Arbor, MI

Instructional Aide for Discrete Mathematics (EECS 203)

Aug 2021 - Dec 2021

- Selected as one of 30 Instructional Aides for an intro Computer Science course with <10% acceptance rate; Taught a weekly discussion of 20+ students reviewing concepts in Discrete Mathematics and created weekly homeworks for 800+ students

ProQuest

Ann Arbor, MI

MDP Student Research Engineer

Jan 2021 - Dec 2021

- Implemented deep-learning OCR correction system in Python that edits input XML files from ProQuest's database of New York Times 10 million articles from last ~150 years; created model reduces word error rate in validation set by 21.4%
- Applied transfer learning through hugging face library on BART sequence-2-sequence model with ~50k training samples
- Presented in weekly AGILE meetings and created creating in depth executive summaries for ProQuest senior executives

PROJECTS AND SKILLS

WikiRunner, code @ <https://github.com/salnad/wikirunner>

using HTML5, CSS, Bootstrap, SNAP, Flask

Created a chrome extension that automatically plays the "Wikipedia Game" (find the shortest path between two wikipedia articles by clicking links). Utilizes an API created with data from WikiDump, Stanfords SNAP library, and a custom implementation of A*.

Extension placed first for 30+ games against 300+ users after running one day.