

# Aditya Agrawal

Bengaluru, KA, India

---

[adiag1200@gmail.com](mailto:adiag1200@gmail.com) | [9785559273](https://www.linkedin.com/in/aditya-agrawal) | [GitHub](https://github.com/aditya-agrawal) | [LinkedIn](https://www.linkedin.com/in/aditya-agrawal) | [Instagram](https://www.instagram.com/aditya_agrawal) | [Medium](https://medium.com/@aditya_agrawal) | [CodeChef](https://www.codechef.com/users/aditya_agrawal)

---

## Work Experience

---

### Software Engineer - [Quicko](#)

June 2023 - December 2023

Ahmedabad, GJ

7 months

Quicko is a tax platform for individuals and businesses, offering D2C, B2B, integration-based, and service models to simplify taxes for everyone at scale.

- **MEET by Quicko:**

- Conducted **2,000+** meets for individuals tax filing in July month with **<40 tax experts** across India during the 2023 tax filing season.

- Developed features around **bookings, orders** and **plans** management.

- **Tax P&L APIs by Sandbox:**

- Developed **multi-threaded** Tax P&L APIs with **asynchronous** workflows to have resilient architecture for Domestic and Foreign assets trading data.

- These APIs provided calculation engines for integrations with **13+ brokers, SDK based, report based** and **B2B** integrations with **>60%** pitch success rate.

- Advocated for development and coding practices and helped in adopting and creating POCs for multiple technologies like **Terraform, Go, Code Bundler**, as better alternatives.

- **Tech stack:** Java - Spring Boot, Typescript, Flutter, NodeJS

### Software Engineer (Intern) - [Stockal](#)

January 2022 - July 2022

Remote - Bengaluru, KA

7 months

Stockal is a multi-tenant, US based trading provider and platform providing US investments across India, UAE, and other regions.

- Worked on systems for transaction data processing and calculations to OPs manual work to 0%, 3P integration for tax statements and providing the same to service partners and clients

- Worked on stock **prices** to provide ticker price data for multiple features and calculations and candlestick **charts for technical analysis** using [TradingView advance charting library](#)

- Worked on data migration, streaming and common code-base to reduce rewriting of code across multiple projects.

- **Technologies Used:** *DB* → TimescaleDB, AWS RDS, MongoDB, Redis | *Backend* → NodeJS, Python, Golang | *Others* → AWS, Pandas

## Technical Skills

---

### Languages

JavaScript, GoLang, Java, Python, C++, C, CUDA Programming

### Databases

Postgres, Mongo DB, Redis, TimescaleDB

### Frameworks

Next JS, Express JS, Spring Boot

### Systems

Linux, AWS, Digital Ocean, Heroku

### Tools

Postman, Terraform, Listmonk

## Education

---

### Bachelor of Technology - Computer Science and Engineering,

July 2019 — June 2023

Indian Institute of Information Technology, Raichur

CGPA (Current): 8.58/10.00

## Projects

---

### TNP Portal

Project | Gitlab

- Common portal for students and companies to manage placements in the institute
- **Rate limiting** (Authentication token based - fallback to IP) backed by **Redis**
- **Horizontally Scalable** application.
- Automatic expiry of incomplete user profiles using **Redis Pub/Sub**
- **Tech Stack:** Front-end → NextJS, MaterialUI | DB → Postgres, Redis | Backend → NodeJS, Firebase Storage | Others → Redis Pub/Sub, Helmet.js

### status-cron

Github

- A **multi-threaded cron** service to keep track of health of your services.
- Supports multiple service types (HTTP, Redis, SQL [Postgres, MySQL, Oracle], MongoDB)
- Helpful for tracking multiple services in a minimal form with flexibility to use the data acquired as wanted.
- **Tech Stack:** Golang

### Contrast Stretching Improvements

Github

- Implemented multiple methods to improve performance of Contrast Stretching Operation
- Compared CPU, and multiple GPU implementations with different approaches and shared memory usage to find a better way.
- **Tech Stack:** CUDA, C++

### Work Stealing Dequeues, Pull based work balancing **Sup.: Dr. Sathya Peri, Professor, CSE-IITH** Github

- Understand and Implement a work stealing dequeue to balance work among idle and loaded threads
- Implementation of the model in a multi-core environment
- Explore possible improvements and implementation in a distributed environment (balance multi-region load, etc.)
- **Tech Stack:** C++