# Mohit Chhaya

(919) 717-6538 · mchhaya2@illinois.edu · mohitchhaya.me · github.com/mrchhaya · linkedin.com/in/mrchhaya

## Education

**University of Illinois - Urbana Champaign** 2025

*Bachelors of Science in* **Computer Science + Linguistics**

*Bachelors of Science in* **Brain and Cognitive Science**

**Coursework:** Compiler Construction (Fall '23), Programming Language Semantics (Fall '23), Distributed Systems, Computer Systems, Programming Languages & Compilers, Algorithms, Data Structures, Software Engineering, Discrete Structures, Computational Morphology, Linear Algebra

## Skills and Interests

**Interests**: Programming Language Theory, Compilers, Fullstack Engineering, Large Language Models, Distributed Systems
**Languages**: Python, C++, Kotlin, C, Javascript, Haskell, Java, Typescript, HTML/CSS, SQL, SAS, C#
**Tools**: Git, AWS, GCP, Docker, Heroku, Pandas, Azure, MongoDB, React, Node.js, Flask, FastAPI, Kubernetes, Tensorflow

## Experience

**Balyasny Asset Management** June 2023 - August 2023

*Software Engineering Intern*

- Building Notebooks-as-a-Service software for on-demand container creation for Jupyter Notebooks utilizing **Kubernetes**.
- Developing lightweight process monitoring system for in-production application, leading to advanced optimizations.
- Developing API using **FastAPI**, **Redis** (caching, queuing) alongside Kubernetes for worker replication.

**Cisco** January 2023 - April 2023

*Software Engineering Intern*

- Architected custom React.js table rendering library to replace previous iterations. Used by **6+** team members. Implemented custom caching, filtering, pagination, and sorting.
- Rewrote Java API endpoints using the Spring framework to support **1000+** requests per hour.

**Amazon** August 2022 - November 2022

*Software Development Engineer Intern*

- Developed **in-production**, consumer-facing, features for the IMDb Android App.
- Engineered **GraphQL** queries for data fetching, optimized to reduce time to load by **5%**
- Built horizontal scrolling view pager with persistent user-based data caching using **Kotlin** and the Apollo Graph Client.
- Designed feature for the IMDb homepage to intercept **100,000** clicks that increases engagement by 2%.

**Addition Technologies Inc.** November 2021 - August 2022

*Software Engineer*

- Designed and deployed an end-to-end data pipeline utilizing Google's **Natural Language Processing** API's, **OpenAI's GPT-3**, and the Contrasive Language-Image Pre-Training (CLIP) **neural network** to generate question-answer scenarios and chatbots for various companies.
- Engineered web scraper on Google Cloud Run utilizing Selenium and Python to scrape **5000+** advertisements, exposed as an **REST API**, used to shorten marketers data wrangling time by ˜15%
- Created database utilizing both **Firestore** and GCP Cloud Storage to store **5000+** pieces of advertiser metadata.
- Decreased latency of **Google Cloud Run** jobs by **70%** and rendered data to a web app built using **React.js**
- Set up relational **MySQL** Database consisting of **12+** tables to streamline data pipelines, saving developers **˜100 hours**

**Mercury Signs Inc.** July 2020 - July 2021

*Software Engineer*

- Designed a financial dashboard using REST API's, Selenium Web Scraping, Python, and **Google Cloud Platform** to monitor financial data on **2000+** customers.
- Achieved **20%** increase in customer retention rate and handled **1000+** interactions by automating a customer pipeline using Google Cloud Functions, Cloud **Pub/Sub**, Python, and the Gmail REST API.

## Projects

**Search Engine** | *C++, Python, Flask, React.js, PyTest* January 2022

- Created a novel Search Engine that utilizes various string matching **algorithms** for efficacy. (Jaccard Index, TFIDF Ranking, Cosine Similarity, Ratcliff Obershelp)
- Programmed **Flask** backend in Python for a web application and **RESTful** API which was deployed on Heroku.
- Rewrote project in **C++** for performance and utilized **multithreading** for performance boost of ˜3s.

**COOL Compiler** | *C++, LLVM, yacc, bison, Make* August 2022

- Architected a compiler, including front, middle, and back-end systems for compiling the COOL language.
- Implemented compiler optimizations like Loop Invariant Code Motion, and Common Subexpression Elimination.
- Created three pass intermediate code generation system utilizing LLVM intermediate code.