

# Abhijeet Mohanty

(312)-468-1795 | 1062 West Polk, Chicago, IL, 60607 | [mabhijeet1995@gmail.com](mailto:mabhijeet1995@gmail.com)

Break down problems into simpler ones.

[Project Portfolio](#)

## EDUCATION

### University of Illinois at Chicago

Master of Science in Computer Science with thesis

Coursework: Cloud Computing, Object Oriented Languages, Software Development for Mobile Platforms, Advanced Software Engineering, Advanced Security & Online Privacy

Aug '19 – May '21

GPA: 3.83/4.0

### National Institute of Technology, Karnataka

Bachelor of Technology in Computer Engineering

Jun '13 – May '17

## SKILLS

**Languages:** Java, Scala, Python, C, C++, TypeScript, JavaScript (ES6), Golang, bash

**Databases:** MySQL, PostgreSQL, MongoDB, Cassandra, Redis

**Web Technologies:** Angular, React, Redux, GraphQL, Kafka

**Frameworks:** Spring, Apache Spark, Hadoop, JUnit, Mockito, Akka

**Tools and Cloud:** AWS (EC2, S3, EMR, BeanStalk, CodeDeploy), GKE, Docker, Kubernetes, Maven, Gradle, sbt, Git, Jenkins (CI/CD), XLDeploy, Puppet

## WORK EXPERIENCE

### University of Illinois at Chicago | Graduate Research Assistant – Empirical Software Engineering

Jan '21 – Present

Advisor: Dr. Mark Grechanik

- **Thesis topic:** Automating meta-research through ontology and research question generation from a corpus of MSR conference papers.
- Built a tool (**Angular + flask + marshmallow + MongoDB + AWS Beanstalk**) to construct a concept graph pertaining to 600 odd papers and terms in them, evaluated **latent semantic analysis** to group related terms and **RNN based architectures** as **LSTM** and **GRU** for research question generation.
- Developed an entropy loss-based algorithm (**Pandas + NumPy + scikit-learn**) for the categorization of characteristic terms from the corpus.

### Societe Generale Global Solutions Centre | Specialist Software Engineer – Investment Banking Technology

Aug '17 – Jul '19

Java | Spring Framework | RESTful Web Services | Angular

- Developed REST APIs (**Spring**) with a user interface (**Angular**) for Orchestrator (**JBPM + JMS + ActiveMQ**) to record workflow configuration information.
- Migrated **MyBatis** to a **Spring Data JPA with Hibernate** framework which allowed for generic DB connectivity for **Oracle SQL, PostgreSQL** servers.
- Wrote the data acquisition layer with **OAuth** integration eliminating load of files thereby reducing SLA by 4 hours/day and code debt by 5%.
- Implemented the purge module (**Spring Data JPA**) to remove resources (DB entries/ files) saving support team manhours by 2 hours/month.
- Developed a widget using **React** and **Redux** to automate daily morning checks requiring manual DB querying saving manhours by 20 mins/ day.
- Led a team of 4 software engineers and 2 business analysts for the integration of **Spring WebFlux** into web services.

## PROJECTS

### Fill the form

Python | Selenium | BeautifulSoup | Google Vision API

- Built a fully **automated tool** for form filling of registration forms with a **filling accuracy of 72.9%** and **error message detection accuracy of 97.3%**.
- Automated **validation error message detection** and **refilling of affected input element** using **Google Vision API, Selenium** and **BeautifulSoup**
- Developed a **rule-based module** leveraging **sentiment analysis, PoS tagging, regex filtering** to fill input elements with apt text based on input and label attributes and the validation error message.

### Publication data analyzer

Scala | Apache Hadoop | sbt | AWS EMR

- Leveraged **Apache Hadoop** and **Scala** to analyze **2 million publication records** using the map-reduce framework deployed on an AWS EMR cluster.
- Calculated authorship score, co-author count and publication-based bucketing along with performing a sort leveraging the shuffle and sort technique.
- Implemented a customized multi-tag record generator to generate input splits for multiple publication types.

### Lightweight GraphQL client

Scala | GraphQL | Design patterns

- Created a **string based GraphQL client** to fetch metadata from GitHub accounts such as viewer and repository related information.
- Implemented **abstract factory pattern, observer pattern** and the **façade pattern** to implement response parsers and query loggers.
- Recorded memory usage, process load and threading information using **JProfiler**.

### Distributed Hash Table

Scala | Akka | Akka-HTTP | Docker

- Implemented a distributed hash table using the **Chord** and **CAN** algorithm on an overlay network with the Akka actor abstraction for nodes.
- Persisted actor states in memory for fault tolerance by using **Akka persistence with a Cassandra datastore**.
- Leveraged **Akka-HTTP** to expose pertinent hash table functions as REST APIs such as putting or accessing a key-value pair.
- Containerized the application and its runtime dependencies using **Docker** and deployed it to **DockerHub**.

### Functional command executor

Scala | Functional Programming | Linux commands

- Developed a composable Linux command execution framework using functional programming constructs.
- Leveraged concepts such as **sealed traits, monads, implicit type checking and type aliases** to enable options and piping for commands.

## ACHIEVEMENTS & MENTORSHIP

- **Employee of the Month**, Societe Generale Global Solutions Centre: Rewarded for creating Orchestrator usage documentation. Jan '19
- **Teaching Assistant**, The University of Illinois at Chicago: Engineering Distributed Objects for Cloud Computing Aug '20 – Dec '20