Abhijeet Mohanty

(312)-468-1795 | amohan31@uic.edu
Personal Website

Break down problems into simpler ones.

EDUCATION

The University of Illinois at Chicago Aug '17 – May '19
Master of Science in Computer Science with thesis **GPA: 3.8/4.0**
Coursework: Computer Algorithms, Cloud Computing, Object-Oriented Languages, Software Development for Mobile Platforms, Advanced Software Engineering

National Institute of Technology, Karnataka Jun '13 – May '17
Bachelor of Technology in Computer Engineering

SKILLS

Languages: Java, Scala, Python, TypeScript, JavaScript
Databases: MySQL, PostgreSQL, MongoDB
Tools and Cloud: AWS (EC2, S3, EMR), Docker, Kubernetes, Maven, Gradle, sbt, Git
Web Technologies: Spring, Angular, React, Redux
Frameworks: Apache Spark, Hadoop, Junit, Mockito

WORK EXPERIENCE

The University of Illinois at Chicago | Graduate Student Researcher – Empirical Software Engineering May '19 – Present
Advisor: Dr Mark Grechanik

- **Thesis topic:** Pipeline for the generation of ontologies and research units from a corpus of research papers such as research questions.
- Built a tool **(Angular + Flask + Python + MongoDB)** to store meanings, categories, and relations of characteristics terms of the corpus.
- 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 an interactive user interface **(Angular)** for Orchestrator **(JBoss + JMS + ActiveMQ)** to record workflow configuration information.
- Migrated from **MyBatis** to a **Spring Data JPA with Hibernate** framework which allowed for generic DB connectivity.
- 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

Publication data analyzer (Deployment video)
Scala | Apache Hadoop | sbt | AWS EMR

- Leveraged **Apache Hadoop** and **Scala** to analyze 2 million publication records using the map-reduce framework.
- Calculated authorship score, co-author count and bucketing along with performing a sort using the shuffle and sort technique.
- Implemented a customized multi-tag record generator to accommodate for multiple publication types.

Stock trader (Deployment video)
Scala | Apache Spark | Alpha Vantage API | sbt | AWS EMR

- Utilized **Apache Spark** and **Scala** to iteratively select a portion of stock to predict possible losses/ profits from a portfolio.
- Made use of **Monte-Carlo** for random sampling of stocks and compared it with a greedy based sampling approach.

Design Pattern Verifier
Java | Annotation Processing | Google Guava

- Created a **customized annotation processor** to verify the correct/ incorrect implementations of the **memento design pattern**.
- Designed a **generic solution** which can handle an extended set of design patterns through a **rule set based architecture**.

Lightweight GraphQL client
Scala | GraphQL | Design patterns

- Created a **string-based GraphQL client** to fetch repository metadata from GitHub accounts.
- 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 a **profiler**.

Akka based cloud simulator
Scala | Akka | Akka-HTTP | Docker

- Developed a distributed and fault-tolerant network of nodes using the **Akka framework**.
- Implemented the **Chord algorithm** for optimized lookup and rearrangement of data across computing nodes through **web services**.
- Containerized and deployed the application using **Docker** and **DockerHub**, respectively.

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

- **Employee of the Month**, Societe Generale Global Solutions Centre: Rewarded for creating Orchestrator usage documentation. Jan '19
- **Special Mention**, SocGen New Joiner Hackathon: Developed a loan processing prediction engine using Weka. Sep '17