Ryan Sabouri

Irvine, CA • (714) 318 - 6848 • ryan.z.sabouri@gmail.com

EDUCATION

University of California, Irvine December 2024
Bachelor of Science in Computer Science (GPA: 3.40)

TECHNICAL SKILLS

- **Languages -** Java, Python, C, C++, HTML, CSS, JavaScript
- **Tools -** AWS, Jenkins, Git, Linux, Docker, Kubernetes, Apache Tomcat, Apache Spark, Postman
- **Databases -** MySQL, Apache Cassandra, MongoDB

EXPERIENCE

Software Engineer Intern June 2024 – Aug. 2024
Capital One | *Richmond, VA*

- Developed a standardized and automated data reconciliation process ensuring the reliability and integrity of customer core demographic data.
- Built the entire workflow using an AWS Step Function, incorporated Python and AWS Lambda for data validation.
- Utilized an EMR cluster with Java Spark to read and write to the customer core Apache Cassandra database.
- Logged all activities and transactions to an AWS S3 bucket, to enable seamless auditing and governance.
- Implemented pre-processing checks and created configurable logic for various data use-cases.

PROJECTS

Fabflix (https://github.com/sabouriryan/Fabflix)

- Developed a movie website to seamlessly search and navigate a MySQL database containing over 100,000 data points.
- Implemented robust movie search and browse features with full text search and autocomplete, including substring matching, pagination, and sorting, with detailed movie and star pages featuring hyperlink navigation.
- Created a shopping cart and checkout process allowing quantity modifications, secure payment processing, and transaction recording in the sales table.
- Ensured security with HTTPS, reCAPTCHA, and secure login using HTTP POST and encrypted password inputs, requiring user authentication for access to search and browse functionalities.
- Hosted website on an AWS EC2 instance using Apache Tomcat, implementing master-slave configuration and load balancing for optimized performance and reliability.

Search Engine (https://github.com/sabouriryan/SearchEngine)

- Built a high-performance search engine in Python to handle tens of thousands of JSON files under stringent operational constraints.
- Implemented an efficient inverted index, enabling rapid word lookup for improved search performance.
- Leveraged BeautifulSoup and regular expression matching (Regex) to parse JSON files accurately and extract relevant data.
- Utilized tf-idf scoring to rank the relevance of search results, enhancing user experience and query accuracy.
- Achieved an impressive query response time of under 300 milliseconds, optimizing the engine's performance.