Abdullah Riaz . Jarvis Consulting

I earned a Bachelor's in Electrical Engineering from Ryerson University. The coursework had allowed me to develop my understanding of key concepts in software that are fundamental, such as object-oriented programming, data structures and algorithms. From the numerous projects and assignments I have done, I developed and solidified these concepts alongside growing my communication skills by working in a teamwork environment. After working for an engineering consulting firm, I joined Jarvis as a Data Engineer, looking to apply my data engineering and software knowledge and experience, improve on data engineering/software development concepts, and learn new concepts and technologies that are used today. At Jarvis, I applied my software fundamentals to projects while working with others in an Agile/Scrum environment. Working in a software role fulfils a passion. A passion to work in a software environment while collaborating with others to achieve a common goal. This passion stems from my curiosity about how things work, always understanding the underlying mechanics that result in something meaningful.

Skills

Proficient: Java, Python, Linux/Bash, RDBMS/PostgreSQL, Git, Agile/Scrum

Competent: Docker, Maven, JUnit4, JDBC, RESTful API's, Spring/Spring Boot, Jupyter, GitFlow, GitHub

Familiar: SLF4J, MongoDB, Cassandra, Apache Hadoop, Apache Spark, Amazon Web Services/AWS

Jarvis Projects

Project source code: https://github.com/jarviscanada/jarvis_data_eng_AbdullahRiaz

Cluster Monitor [GitHub]: Developed a monitoring agent tool that allows the user to track host hardware specifications data and host resource usage data per node/server in a cluster of computers. The tool was implemented using Bash scripts to automate key processes, for example, starting up a Docker container running PostgreSQL, creating 'host_info' and 'host_usage' tables to store data, etc. To track host resource usage data in real-time, 'crontab' was used. The data is stored on a PostgreSQL database which then can be analyzed to answer business questions. A PostgreSQL instance was provisioned by running a Docker container. Git was used to providing version control of the code and then uploaded to a remote repository on GitHub.

Core Java Apps [GitHub]:

- Twitter App: Developed a Java-based Twitter application that allows the user to post a Tweet, show a Tweet and delete a Tweet through the use of Twitter's REST APIs. The application makes HTTP requests to version 1.1 of Twitter's REST APIs. The HTTP requests were made possible by using Apache's HTTP client. The HTTP client contained HTTP methods that allowed for making GET/POST requests to the Twitter API endpoints. Since Twitter APIs require authentication to make requests, an 'OAuthConsumer' was used to sign these requests. When an HTTP response object was received, the body of the response was parsed and serialized into a JSON string which was then deserialized and mapped into the Tweet object fields using Jackson. To handle dependencies the Spring Boot framework was utilized. The application was packaged using Maven and then dockerized by building an image through a docker file and subsequently uploaded on docker hub.
- JDBC App: Developed a Java-based application that allows the user to create a connection between a Java application and an RDBMS to run queries on a database for data analysis. The app was implemented using Java's Database Connectivity (JDBC) tool following the Data Access Object (DAO) design pattern. A 'CustomerDAO' class was implemented to enable CRUD operations on a PostgreSQL database, 'hplussport'. Thereafter, a Data Transfer Object (DTO), namely a 'Customer' class was created to represent the 'Customer' table in the database. Lastly, CRUD operations were performed in the 'JDBCExecutor' class to process the data.
- Grep App: Developed a Java-based application that allows the user to search for a specified string in a text file and returns instances of the string pattern as output onto a specified file. The application was implemented using a 'JavaGrep' interface that had essential grep app functionality which was then put into implementation via a Java grep implementation class 'JavaGrepImp'. Certain libraries were used to enable searching of a file, reading of a file, and writing to a file. The Java Regex API was used to search for a specified string pattern in a specified file. Clean and helpful logs were generated through 'SLF4J Logger' to understand what errors were being triggered and what information is being processed. All the dependencies were downloaded and installed with Maven. Lastly, the app was packaged into an Uber-jar and then dockerized by building an image through a docker file and subsequently uploaded on docker hub.

Highlighted Projects

Data Extraction: Exchange rate data of over 100 countries were extracted through an API call and stored the data into a CSV file. A large dataset of market capitalization data of over 50 banks was extracted from a webpage through a web scraping process and stored into a JSON file. An ETL pipeline (Extract, Transform and Load process) was implemented on Jupyter Notebook using Python to extract bank and market capitalization data from data files, transformed the market cap currency from USD to GBP using exchange rate data and loaded the transformed data into a CSV file, while logging each part of the ETL process into a separate logging file.

Data Modelling: Modelled a data set for improved operational efficiency on an RDBMS platform on Cloud. The data was reviewed in several different systems and modelled on a relational database on cloud-based PostgreSQL (RDBMS) by utilizing the pgAdmin framework. Modelled a data set schema on a relational database by creating an entity-relationship diagram (ERD) that generated table database objects. The data was imported into a relational database by running an SQL script through a script file. Views database objects were created according to the client's requirements and the data was stored as a CSV file. The data was exported into other cloud-based relational database management systems (RDBMS), namely IBM Db2 on cloud and MySQL on a cloud through the phpMyAdmin framework.

Professional Experiences

Data Engineer, Jarvis (2021-present): Worked in an Agile/Scrum environment to work on software projects with a team of Software Engineers. Worked as part of a team to design, implement, test and deploy features on a product. Daily scrum meetings were held to discuss issues and to set goals for the day. Weekly code reviews were done by senior developers to ensure code works as expected and to meet standards. Worked on features via tickets and uploaded code to GitHub for version control. To manage branches and features, the GitFlow methodology was followed on all projects.

Electrical Designer, Spectra Engineering (2019-2020): Worked in an engineering consulting environment as a part of an electrical engineering team to design electrical systems for buildings and write up reports. Coordinated with other disciplines such as Mechanical and Architectural team to develop designs in accordance to relevant codes and standards. Consulted with clients to determine best solutions that met clients requirements. Worked under senior engineer to ensure designs were in accordance to standards and were free from critical errors.

Education

Ryerson University (2013-2018), Bachelor of Engineering, Electrical Engineering

Miscellaneous

- Coursera - Data Engineering Professional Certificate by IBM
- Reading
- Sports