www.github.com/jj-28

Jeffrey Johnson

(845)-269-8744
jeffrey.johnsonswe@gmail.com

Education

Oswego, NY **State University of New York (SUNY) College at Oswego** **2016 – 2021**

- B.A in Computer Science, May 2021

Languages and Technologies

- Languages: Golang, Java, Python
- Frameworks: Java EE, Bootstrap 4, Spark Java, Dropwizard, Java Swing;
- Databases/Servers: MySQL, DynamoDb;
- Cloud Software: S3, EC2, ECS, CodeDeploy, CodeBuild, Docker;
- Version Control: Git, GitHub;

Employment

Software Engineer Intern **Splice** **Jun. 2021- Sep. 2021**

- Implemented RPC handlers using Twirp to simplify business logic and return JSON during HTTP calls.
- Deployed code to pre-production and production environments using Jenkins.
- Rewrote SQL statements to prevent database race conditions, resulting in improved query execution speed.
- Created unit tests to verify response payloads when making requests to the Merchandising API.
- Implemented Protocol Buffers to represent database models, requests and responses for the service.
- Utilized Buf to compile Protocol buffers and execute code generation.
- Published documentation and articles to facilitate the transition to a service oriented architecture.
- Worked with the Admin/Ingestion team in an Agile environment to ensure code was delivered on schedule.

Selected Coursework

Software Deployment (Fall 2020)

- Designed an API that polled subscriber data from a given YouTube Channel ID and implemented it in Go.
- Built and deployed Docker images to Docker containers on a hosted Amazon EC2 instance.
- Reduced the image footprint by **90 percent** (200 MB- 10 MB) by creating a Docker image build from scratch.
- Sanitized URL query parameters to ensure the system would be safe from XSS attacks.
- Utilized DynamoDb to create a NoSQL database that stored the data (subscribers, views, etc.) of a user.
- Utilized AWS S3 to host a static website.
- Used AWS CodeDeploy and CodeBuild to automatically build and deploy Go source code hosted on GitHub.

Web Services (Spring 2020)

- Designed a RESTful API for a beverage store, allowing an authenticated user CRUD permissions.
- Utilized design patterns such as Model View Controller and Data Access Object to minimize repetitive code.
- Used Postman to test API endpoints and token authentication.
- Configured the application to consume and produce JSON.
- Created SQL tables using MySQL to store user credentials, sneaker data and API tokens.

Algorithms (Fall 2020)

- Developed an application that compares all of the Wikipedia links on a page to derive the most relevant link.
- Implemented a separate-chaining HashTable to store word information.
- Implemented a term frequency–inverse document frequency (TD-IDF) statistic to compare links.
- Created an application that parsed customer grocery lists and returned the items that were strongly associated.
- Utilized the Apriori Algorithm to create frequent itemsets and association rules between frequent items.
- Implemented a image segmentation algorithm that isolated regions of an picture and colors similar regions.