New City, NY
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** **Aug. 2016 – Dec. 2020**

- B.A in Computer Science, Dec. 2020

LANGUAGES AND TECHNOLOGIES

- Languages: Java, HTML, CSS, JavaScript, Python, Golang;
- Frameworks: Java EE, Bootstrap 4, Spark Java, Dropwizard, Java Swing;
- Databases/Servers: MySQL, Tomcat 8;
- Cloud Software: AWS Docker, S3, DynamoDb, EC2, ECS, CodeDeploy, CodeBuild;
- Version Control: Git, GitHub;

PROJECTS

JBento (2019- Present)

- Founded a music brand for the sale of instrumental music licenses and sample packs.
- Designed and created a static website using Bootstrap 4 and JavaScript.
- Lowered the average page loading speed by **25 percent** through image compression and css minification.
- Utilized Git, GitHub and CPanel for version control and automatic deployment.
- Ran Google Ads campaigns which resulted in **42,700** impressions and **10,000** YouTube views to increase brand awareness and sales.

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.