Adham Popal

Ashburn, VA | www.linkedin.com/in/adham-popal | (703) 505-4245 | apopal23@gwu.edu

EDUCATION

University **George Washington University**

Major: Computer Science *Graduation Date: May 2024*

- 3.95 GPA
- Applicable Coursework Completed Includes: Discrete Structures I & II, Data Structures & Algorithms, Systems Programming, Comp. Architecture I, Database Systems & Team Projects, Operating Systems, Comp. Networks
- Current Coursework: Machine Learning, Distributed Systems, Chemistry, Data Management Systems
- Skills: Java, Python, C,SQL, MongoDB, HTML+CSS, Git+Github, Docker, Debugging, Golang, Assembly, Development in Linux Environment, Networking Modeling and Understanding

WORK EXPERIENCE

REI Systems **Sterling, VA**

Software Developer Intern *May 2022-August 2022*

- Planning and development of a Real-Time Data Pipeline for evaluation of a ML based Search Engine
- Engaged in daily standups, weekly sprint planning, and weekly demonstrations of work completed during the week
- Researched and devised metrics for evaluating Search Engine accuracy to be displayed on a dashboard
- Management of AWS MSK Cluster and CloudWatch stream with use and exposure of AWS concepts
- Constructed Java Spring applications for continuously consuming logs from AWS CloudWatch logs and producing JSON formatted data to AWS MSK and for sending CSV formatted data to an S3 bucket
- Use of AWS Glue to extract and perform calculations on data and load into DynamoDB

Harris Teeter **Leesburg, VA**

Cashier/Bagger/Stocker *October 2019-December 2021*

- Provided customer assistance with finding items, bringing groceries to their vehicle, and checking out their purchases
- Ensured workplace cleanliness by organizing shelves, cleaning up workstations, sweeping the floors, cleaning restrooms, and emptying trash
- Restocking shelves, building displays, and online order shopping

PROJECTS

Microkernel Containers (C) (https://rb.gy/eaqaij): Implementation of containers in xv6 (microkernel operating system). Involved using shared memory with process mutex locks to communicate between a server and a client for starting a container. Pre-processing to start a container involved parsing, enforcing resource limitation, and setting a root in the filesystem. Priority scheduling was implemented for the containers to run above other processes. Involved modifications on both kernel and user-level with new system calls created to manage processes and sensitive actions.

Mini Shell (C) (https://biturl.top/JfEJJ3): A fully usable shell that makes use of system calls such as fork, dup, and exec. Parses user input at the command line and supports pipes by overwriting file descriptors of processes. Job control uses signals to support background tasks. File redirections can output stderr and stdout to a specified file.

GUI Client and Server Chatroom (Java) (https://biturl.top/MvmiU3, https://biturl.top/ZNjaiq): A threaded GUI client program utilizing socket objects in Java to connect to a server chatroom. Updates members as they join and leave the server and allows a user to output via direct messaging or to all other online users. The server adds and removes clients as they enter and leave the room with a Linked List being used to maintain direct messaging.

Encryption/Decryption of Messages (Assembly Prog.) (https://biturl.top/UneuQj): Use of LC3 assembly language to encrypt/decrypt a user inputted message using Vigenere Cipher, Caesar's Cipher, and Bit Shifting. It verifies a proper user inputted 5-character key for encryption/decryption and modifies the ASCII value of each character of a given message. Implements multiplication, division, modulo, and bit operations left shift, right shift, and XOR in assembly.

Shopping Cart (Python+Flask, HTML+CSS, SQLite) (https://biturl.top/ymayii): An online store that uses SQLite to store products and user information. Use of Python + Flask as backend to support proper insertion and updates to the database. Use of session variables to maintain user login and current products in the cart before checkout. Website is properly maintained by subtracting stock as orders are completed and by updating order history for customers.

University Registration-Advising (Python+Flask, HTML+CSS, mySQL) (https://rb.gy/wlassd): University System that maintains student records and assigned faculty advisors. Different permissions given based on user login which also includes faculty advisors and graduation secretaries who can view transcripts, approve applications to graduate, and assign grades to students who are registered for their specific course. Rate-my-professor functionality implemented and usable by students once a course has a grade confirmed in their self-created program of study.