Executive Summary

1. Assignment Overview -

The purpose of this assignment is to enhance the functionality of the Key-Value Store developed in the previous projects by implementing replication and consistency mechanisms. Specifically, it aims to improve server bandwidth and ensure high availability by replicating the key-value store across five distinct server instances. The system will allow clients to interact with any replica to perform operations such as GET, PUT and DELETE while ensuring consistency and integrity of data across all replicas. To ensure that all replicas reflect consistent state changes during PUT and DELETE operations, a Two-Phase Commit (2PC) protocol will be used to synchronize updates. To verify successful updates, the 2PC protocol will include a Prepare phase (ensuring replica/participant agreement) and a Commit phase (finalizing the update). To ensure portability and efficient deployment, the assignment also uses Docker to containerize and deploy the client and server applications. This project emphasizes distributed systems concepts, including replication, consistency, fault tolerance and practical deployment using Docker to design scalable and reliable systems.

2. Technical Expression -

This assignment offered an enriching and hands-on experience with the complexities of designing and implementing distributed systems. Replicating the Key-Value Store across five server instances was a critical step in improving data availability, scalability, and fault tolerance, which are essential attributes of a distributed systems. The challenge of designing and implementing the Two-Phase Commit (2PC) protocol provided deep insights into distributed coordination and consistency. Understanding the complexities of the Prepare and Commit phases, as well as ensuring consistent message passing and synchronization across all replicas, required an organized approach and extensive testing.

The assignment became more challenging and practical when Docker was integrated for deployment. Containerizing the client and server applications significantly simplified deployment, enabling consistent environments and easy

replication across multiple nodes. However, configuring Docker networks for effective communication between containers proved challenging at first.

Overall, this assignment was both interesting and technically demanding, offering a deep insight into distributed systems development and implementation. In addition to providing useful skills in networking and containerization, it strengthened my understanding of fundamental ideas like consistency, replication and distributed coordination.