### **Module 10: Converting Local Discoveries into Global Improvements**

**Course:** CSD380-H326 DevOps  
**Assignment:** Module 10 - **Shared Code Repository - Git Exercise**  
**Gitlab GitHub Repository:** <https://github.com/DrPanglo55/GitLab>   
**Name:** Arun Sharma  
**GitHub :** <https://github.com/SharmaArun017/csd-380>   
**Date:** 03/02/2025

### **Shared Code Repository in GitHub: A Collaborative Experience**

In this assignment, our team worked together to simulate a **real-world development environment** by utilizing **GitHub** as a shared repository. This exercise was designed to help us practice **branching, merging, handling pull requests, and resolving merge conflicts** while working collaboratively. Our team consisted of three members: **Aaron Camp (Release Manager & Developer 1), myself, Arun Sharma (Developer 2), and Shayna Solomon (Developer 3).** Each of us had specific responsibilities and tasks to complete within the GitHub repository, contributing to the overall workflow.

As **Developer 2**, my primary responsibility was to modify the index.html file by adding a **title (<h1>) and a quote (<blockquote>)**, and later, to ensure that the **author’s name was correctly formatted in uppercase**. This required creating multiple branches, pushing changes to GitHub, and working within an organized **pull request (PR) and merge** structure. The structured workflow allowed our team to mimic how software development teams manage shared codebases in a real-world DevOps environment.

The first task I completed was **creating the quotes branch** in which I added a **title and a quote from Winston Churchill** inside the index.html file. After pulling the latest changes from GitHub to ensure my local copy was updated, I switched to a new branch and made the necessary modifications. Once the changes were implemented, I staged and committed them, ensuring that my commit message clearly described my contribution. After pushing my branch to GitHub, I created a **pull request (PR)**, allowing my changes to be reviewed and merged by the **Release Manager (Aaron)**. Once the PR was approved and merged into the main branch, the next stage of development began.

Later in the exercise, a **hotfix (hotfix1) was created by Developer 1 (Aaron)** to correct the author’s name. My next task was to **ensure that the author’s name was in uppercase** as per the requirements. To accomplish this, I first **pulled the latest changes from main**, ensuring that my local repository was updated with all merged changes. Then, I **created a new branch named authorupdate**, where I updated the author’s name to **WINSTON CHURCHILL** instead of the previous mixed-case format. At this point, I was instructed **not to commit the changes yet** until the hotfix1 branch was merged.

Once the **hotfix1** changes were merged into the main branch, I proceeded to **merge main into authorupdate**. This resulted in a **merge conflict**, which was expected since both hotfix1 and my authorupdate branch modified the same part of index.html. Resolving the conflict required manually editing the file by keeping the correct **uppercase format for the author’s name**. After resolving the conflict, I staged and committed my changes before pushing them to GitHub. The final step was creating a **pull request for authorupdate**, which was reviewed and merged by the Release Manager, completing my assigned responsibilities.

This exercise was a **valuable learning experience in Git collaboration**. It reinforced best practices such as **pulling the latest changes before making modifications, properly using branches, writing meaningful commit messages, and handling merge conflicts efficiently**. Working within a structured GitHub workflow demonstrated how **DevOps teams manage version control, prevent code conflicts, and ensure smooth code integration in a shared repository**.

By the end of this exercise, I had successfully completed all assigned tasks, and the final merged repository reflected all updates from our team members. **The collaborative process of creating pull requests, merging branches, and handling conflicts simulated a real-world development environment where multiple contributors work on the same codebase.** The structured approach helped me understand the importance of **clear communication, version control strategies, and collaborative coding practices**.

With all the tasks completed, the final step is to include **screenshots of the pull requests and merges from GitHub**, showcasing the workflow and confirming successful execution of the assignment. These screenshots will provide a visual representation of the **branching, merging, and conflict resolution process** that we followed throughout the assignment.

### **Screenshots**



