Git Assignment

Done by: Rahaf Bassam Jazar

Supervisor: Dr. Motasem Aldiab

# Introduction

I developed a simple project for this assignment that uses GitHub and Git to simulate a real development environment. The project consists of a simple Java file that contain a source code (Student.java) to illustrates how to use GitFlow. The main goal was gaining a practical understanding of GitFlow and to simulate collaborative development by using two Git terminals representing different team members Which are Khaled and Ayman.

# Implementation of GitFlow

Setup and initialization

1. **Repository Initialization**: I created the project repository manually in GitHub
2. **Creating Collaboration environment**: I used Git Bash to create 2 developers. and clone the repository in each of them.
3. **Initialize the GitFlow**: using git flow init command, I initialize the git flow then by default a develop branch was created from the main branch to serve as the integration. I used a default prefixes (main, develop, feature, hotfix).
4. Go to your TOC and click anywhere in it. Then click **Update Table**, and click **OK** (**Update page numbers only** is selected by default).

Feature Branch Development

1. **Ayman (Terminal 1):**

* Created an adding-print Message-method feature branch.
* Implemented a new feature => create Student.java and modify on it and committed the changes.
* Then merged into the develop branch using pull request from side in GitHub
* Created a modify-print-statement feature branch. He modifies on the Student.java file while Khaled work in the same file.
* Push the updated file to the develop branch

1. **Khaled (Terminal 2):**

* Created an adding-get Age-method feature branch.
* Modify on Student.java file and merge feature with develop branch. automatically after merging, feature will be removed.
* Pull the origin develop from server to be with update, but conflict was faced, so I solve the conflict manually. then I push origin develop.

Release Branch

1. **Khaled (Terminal 2):** Firstly, I pull the work from the server using git pull command to be updated with every change from other collaborators in the repository.
2. **Creating the Release Branch**: A release branch was created from the develop branch using git flow Release start 0.3 to prepare for the first official release. Final adjustments and updates were made in this branch.
3. **Merging the Release**: The release branch was merged into the main branch by pull request from Khaled side in GitHub. This branch was also merged back into the develop branch to ensure that develop contained the latest changes.

Hotfix Branch

1. **Identifying and Fixing Bugs**: A hotfix branch named hotfix was created from the main branch for emergency bugs and urgent issues in the production code. Once the problem was fixed, the changes were committed.
2. **Merging the Hotfix**: The hot fix branch was merged into the main branch by pull request from Khaled side in GitHub. It was also merged back into the develop branch to keep it up- to-date.

Challenges and Solutions

Simulating Multiple Collaborators:

* Challenge: Simulating multiple collaborators on a single laptop required managing multiple terminals.
* Solution: Opened two separate Git terminals to represent different collaborators.

Conflict Problem:

* Challenge: Two collaborators working on same file at same time caused conflict
* Solution: I solved it manually.

Ensuring Branch Consistency:

* Challenge: Keeping the branches consistent with any changes from other users and keeping develop branch consistent with changes from release and hotfix branches
* Solution: Always pull from server before push to the server and ensure merging release and hotfix branches back into develop after merging them into main.

# Learning Experience and Insights

I gained hands-on experience with Git and GitHub, enhancing my understanding of version control in a team setting.

I understand the principles and commands of Gitflow and get a chance to be in practical situations in which different team members collaborate different parts of a project simultaneously so, team members can guarantee efficient environment for collaboration and development with an organized codebase by effectively using feature branches, a development branch, a release branch, and a hotfix branch.