

Git Assignment

Done by: Lana Alnimreen

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 text file 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.

Implementation of GitFlow

Setup and Initialization

1. **Repository Initialization:** I create the project repository manually in GitHub, and an initial file was made in a `main` branch from server side.
2. **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 branch for features.

Feature Branch Development

1. **User 1 (Terminal 1):**
 - o Created a `feature1` branch.
 - o Implemented a new feature and committed the changes.
 - o Then merged into the `develop` branch using pull request from user 1 side in GitHub.

Release Branch

1. **User 2 (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 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 user 2 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 `hotfix` branch was merged into the `main` branch by pull request from user 1 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. Careful attention was to avoid conflicts.

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 on 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.