

Karel The Robot Assignment

Atypon Java and DevOps Cohort / June 2024

Mahmoud J. Qudah

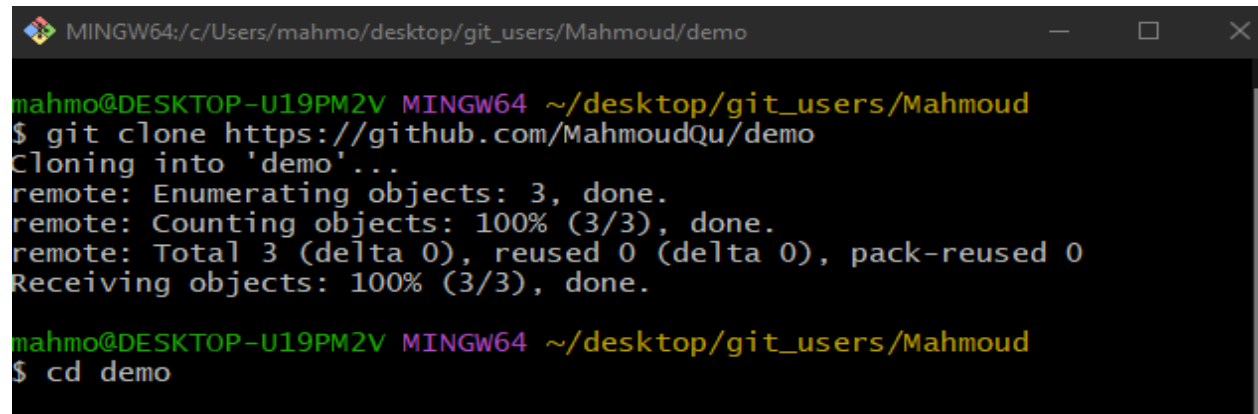
Introduction:

In this assignment, I was asked to simulate a collaborative development environment using Git and GitHub. By emulating two different users working on the same repository, I gained practical experience in using Git as a collaborative tool. This involved creating and managing branches, merging feature branches into a development branch, performing a release, and one hotfix, all while committing to the Gitflow workflow. This hands-on approach provided valuable insights into the processes and best practices of version control in a real-world development setting.

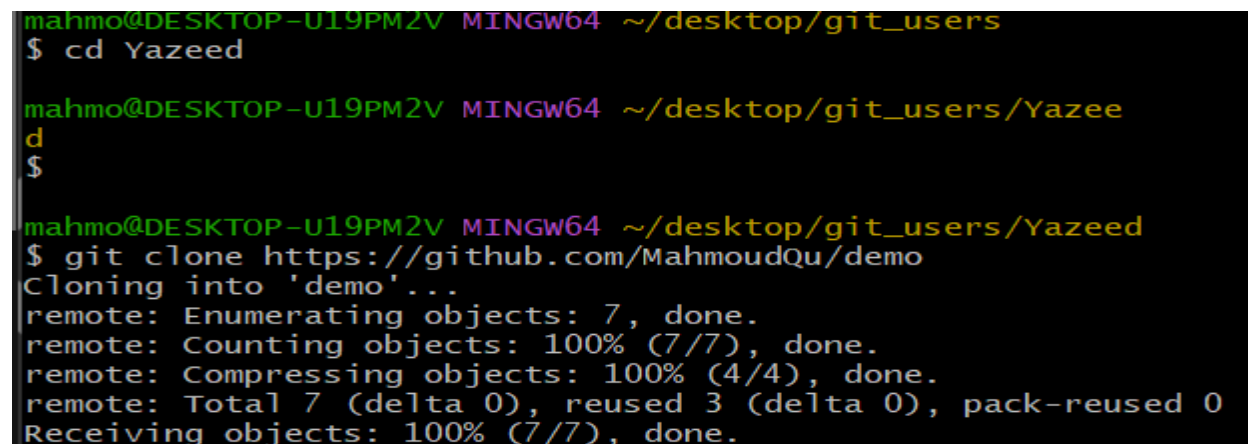
Project description:

First, I created two directories named “Mahmoud” and “Yazeed” to simulate two different users. I then cloned the Git repository into each directory, allowing each simulated user to work independently on their own feature branches.

- Mahmoud:

A terminal window titled 'MINGW64; c:/Users/mahmo/desktop/git_users/Mahmoud/demo'. The prompt is 'mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud'. The user runs 'git clone https://github.com/MahmoudQu/demo'. The output shows cloning progress: 'Cloning into 'demo'...', 'remote: Enumerating objects: 3, done.', 'remote: Counting objects: 100% (3/3), done.', 'remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0', and 'Receiving objects: 100% (3/3), done.'. The user then runs 'cd demo'.

- Yazeed:

A terminal window titled 'MINGW64; ~/desktop/git_users'. The prompt is 'mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users'. The user runs 'cd Yazeed'. The prompt changes to 'mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Yazeed'. The user runs 'git clone https://github.com/MahmoudQu/demo'. The output shows cloning progress: 'Cloning into 'demo'...', 'remote: Enumerating objects: 7, done.', 'remote: Counting objects: 100% (7/7), done.', 'remote: Compressing objects: 100% (4/4), done.', 'remote: Total 7 (delta 0), reused 3 (delta 0), pack-reused 0', and 'Receiving objects: 100% (7/7), done.'.

For Mahmoud:

After cloning the repository, I created a “develop” branch from the main branch and pushed it to the remote repository. This develop branch serves as the base for feature branches, ensuring that new features are integrated into develop before being merged into the main branch, this maintaining a clean and stable main branch.

```
mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (main)
$ git checkout -b develop
Switched to a new branch 'develop'

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ ls
README.md

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ git push origin main
Everything up-to-date

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ git push origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/MahmoudQu/demo/pull/new/develop
remote:
To https://github.com/MahmoudQu/demo
* [new branch]      develop -> develop
```

After ensuring that the develop branch was up to date on the server, I created a new branch named “feature/mah-create-file” from the develop branch. This branch was dedicated to the "adding file" feature, indicating that Mahmoud created a new file. I then pushed this feature branch to the remote develop branch, keeping the main branch clean and stable.

```
mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ git checkout -b feature/mah-create-file
Switched to a new branch 'feature/mah-create-file'

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (feature/mah-create-file)
$ vi fileM.txt

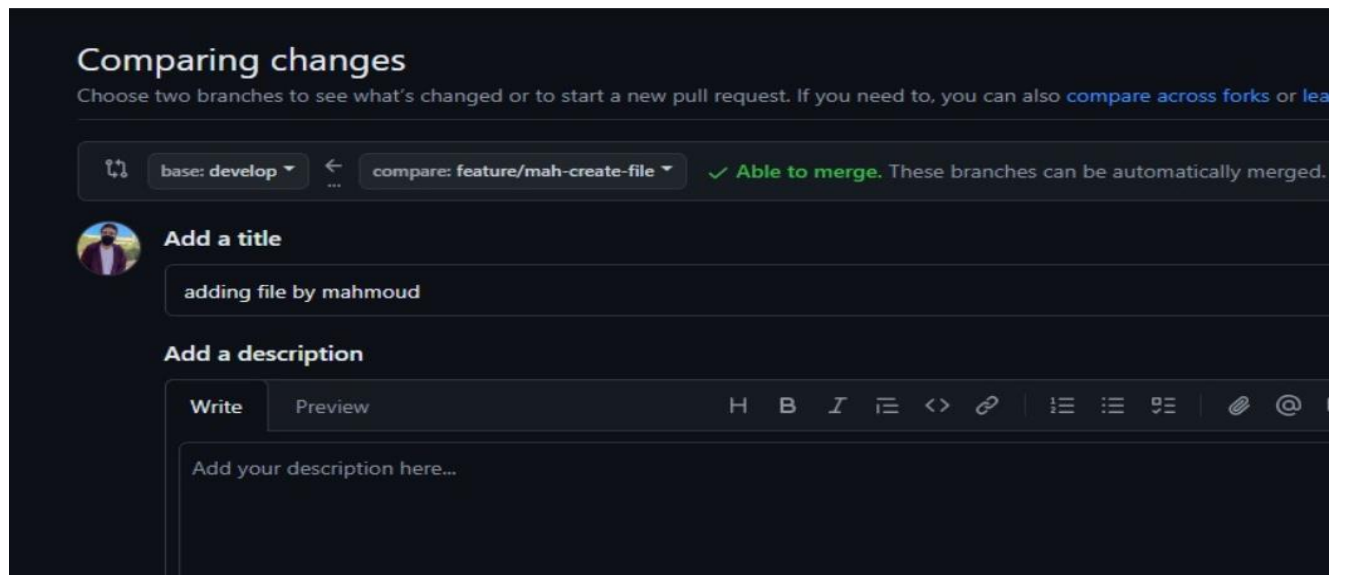
mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (feature/mah-create-file)
$ git add fileM.txt
warning: in the working copy of 'fileM.txt', LF will be replaced by CRLF the next time Git touches it

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (feature/mah-create-file)
$ git commit -m "adding file by mahmoud"
[feature/mah-create-file 0697d1a] adding file by mahmoud
1 file changed, 1 insertion(+)
create mode 100644 fileM.txt
```

```
mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (feature/mah-create-file)
$ git push origin develop feature/mah-create-file
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 320 bytes | 320.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: create a pull request for 'feature/mah-create-file' on GitHub by
visiting:
remote:      https://github.com/MahmoudQu/demo/pull/new/feature/mah-creat
e-file
remote:
To https://github.com/MahmoudQu/demo
* [new branch]      feature/mah-create-file -> feature/mah-create-file
```

After pushing, now the remote repo has 3 branches, main, develop and feature/mah-create-file, and there is a pull request here from the feature branch, after reviewing this pull request and deciding by the one who is responsible for this, let's assume it is confirmed

The screenshot shows the GitHub interface for a repository named 'demo' (Public). At the top, there are buttons for 'Pin' and 'Unwatch'. Below this, a notification bar states 'feature/mah-create-file had recent pushes 1 minute ago' with a green button 'Compare & pull request'. The main section shows the 'develop' branch selected, with '3 Branches' and '0 Tags'. A search bar 'Go to file' and buttons 'Add file' and 'Code' are present. A message indicates 'This branch is up to date with main'. Below this, a commit by 'MahmoudQu' is shown, labeled 'Initial commit', with commit hash 'c3bce94' and timestamp '7 minutes ago'. The commit details show a file 'README.md' added as an 'Initial commit' '7 minutes ago'. At the bottom, the 'README' file is listed with an edit icon.



The feature branch from Mahmoud is Successfully merged with develop branch!

Challenges and Solutions:

Challenge 1: Accidental Pushes to the Main Branch

I had trouble because I accidentally pushed changes to the main branch instead of the develop branch. This happened because I didn't specify the branch when pushing changes, which could make the main branch unstable.

Solution:

To fix this, I used this command to make sure the changes went to the right branch:

git push origin develop

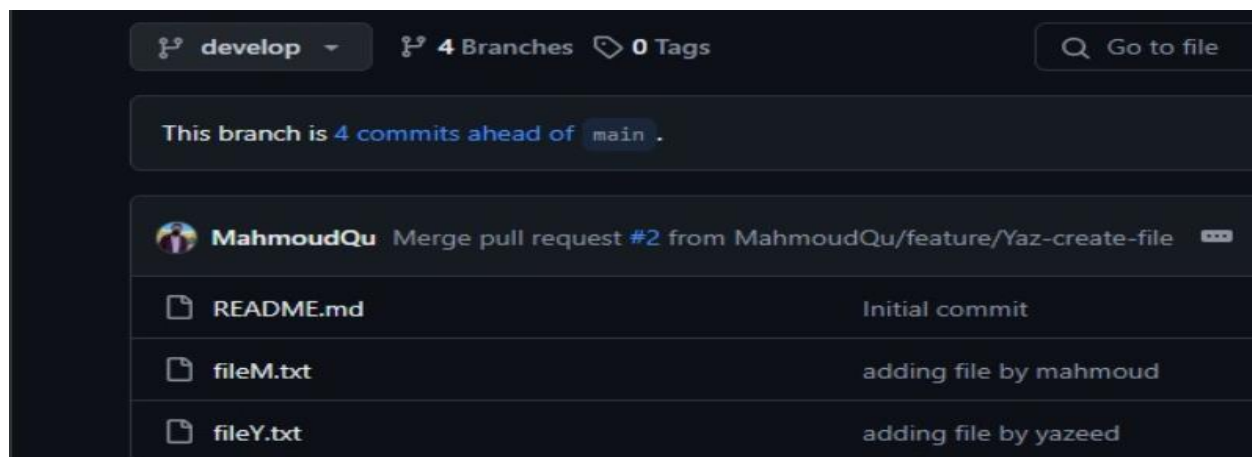
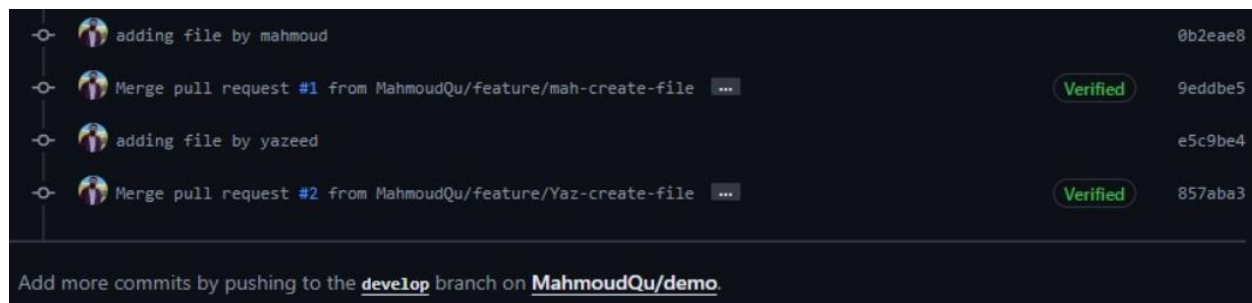
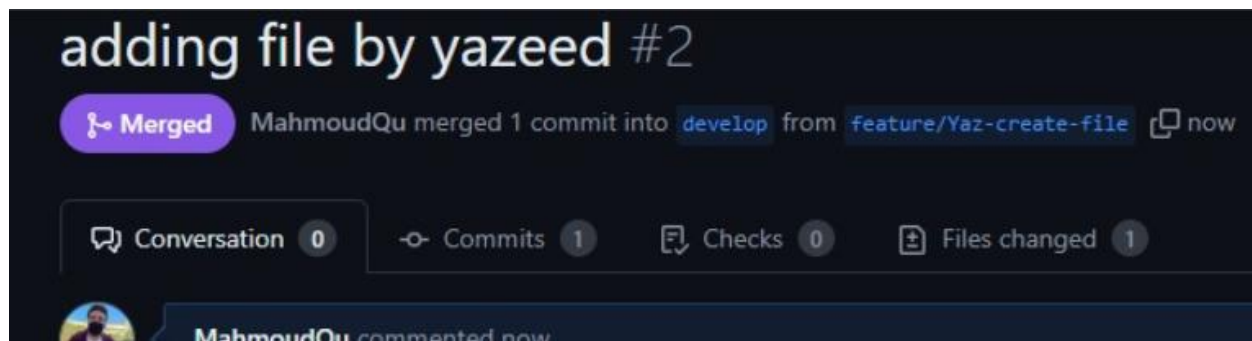
As shown in the images above, this command explicitly specifies both the remote repository (origin) and the target branch (develop), preventing accidental pushes to the main branch. This approach helped maintain the integrity of the main branch by ensuring that all new features and changes were first integrated into the develop branch.

Challenge 2: Keeping Branches Updated

Another challenge was keeping the local branches updated with the latest changes from the remote repository, especially when multiple collaborators were working on the same project. This required frequent synchronization to avoid conflicts and ensure smooth integration.

Solution: I regularly used the git pull command to fetch and merge changes from the remote repository into my local branches. For example: **git pull origin develop**, this command fetched the latest updates from the develop branch and merged them into my local develop branch, ensuring that I was working with the most recent codebase.

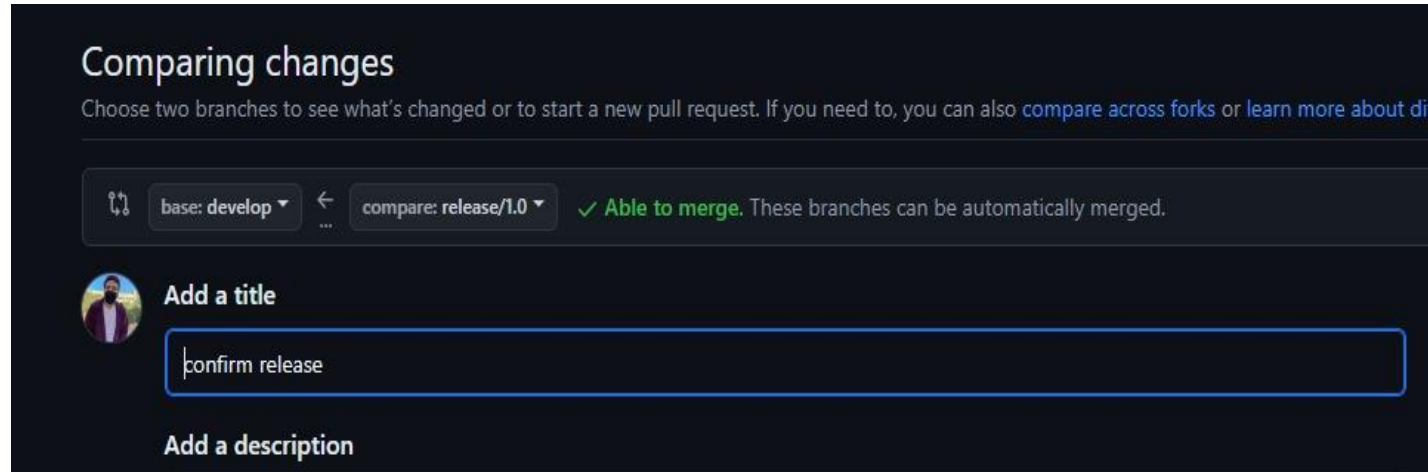
```
mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ git pull origin develop
From https://github.com/MahmoudQu/demo
* branch          develop    -> FETCH_HEAD
Updating 2380904..0a71826
Fast-forward
 fileM.txt | 1 +
 fileY.txt | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 fileM.txt
 create mode 100644 fileY.txt
```

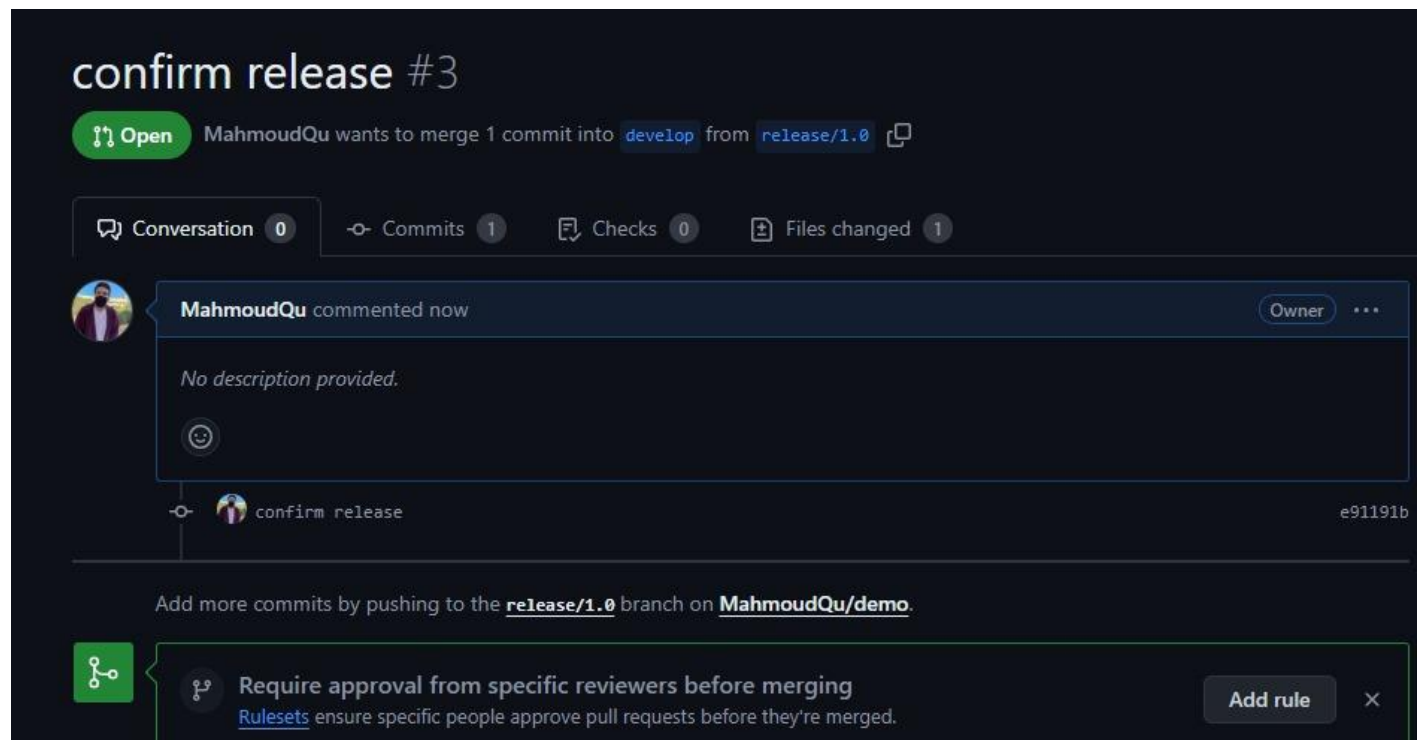
After confirming the pull request, we merged the feature branch into the development branch, and now all features have been pushed.

Release:

Mahmoud created the release to finalize the decision to push into production after reviewing the work done so far. Here is the process: from the develop branch, I created the release/1.0 branch to make the final changes. I added a file named 'finalFile.txt' to simulate the release work and then pushed it to the server. After that, I merged the release branch with the main branch, as this is the best practice to avoid directly merging with the main branch.




The screenshot shows the 'Comparing changes' page in GitHub. At the top, it says 'Comparing changes' and provides instructions: 'Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff](#).' Below this, there are two dropdown menus: 'base: develop' and 'compare: release/1.0'. To the right of these is a green checkmark and the text 'Able to merge. These branches can be automatically merged.' Below the dropdowns, there is a section for 'Add a title' with a text input field containing 'confirm release'. Below the title field is a section for 'Add a description'.



The screenshot shows the pull request page for 'confirm release #3'. At the top, it says 'confirm release #3' and 'MahmoudQu wants to merge 1 commit into develop from release/1.0'. Below this, there are tabs for 'Conversation' (0), 'Commits' (1), 'Checks' (0), and 'Files changed' (1). The 'Conversation' tab is selected, showing a comment from 'MahmoudQu' with the text 'No description provided.' and a smiley face emoji. Below the comment, there is a commit titled 'confirm release' with the hash 'e91191b'. At the bottom, there is a section for 'Add more commits by pushing to the release/1.0 branch on MahmoudQu/demo.' and a 'Require approval from specific reviewers before merging' rule set.







confirm release #3

 Merged

MahmoudQu merged 1 commit into `develop` from `release/1.0`  now



MahmoudQu and others added 6 commits 10 hours ago

-  adding file by mahmoud
-  Merge pull request #1 from MahmoudQu/feature/mah-create-file ...
-  adding file by yazeed
-  Merge pull request #2 from MahmoudQu/feature/Yaz-create-file ...
-  confirm release
-  Merge pull request #3 from MahmoudQu/release/1.0 ...

And then to make all consistent, I merge the develop branch from the release/1.0, doing this for all the features branching.

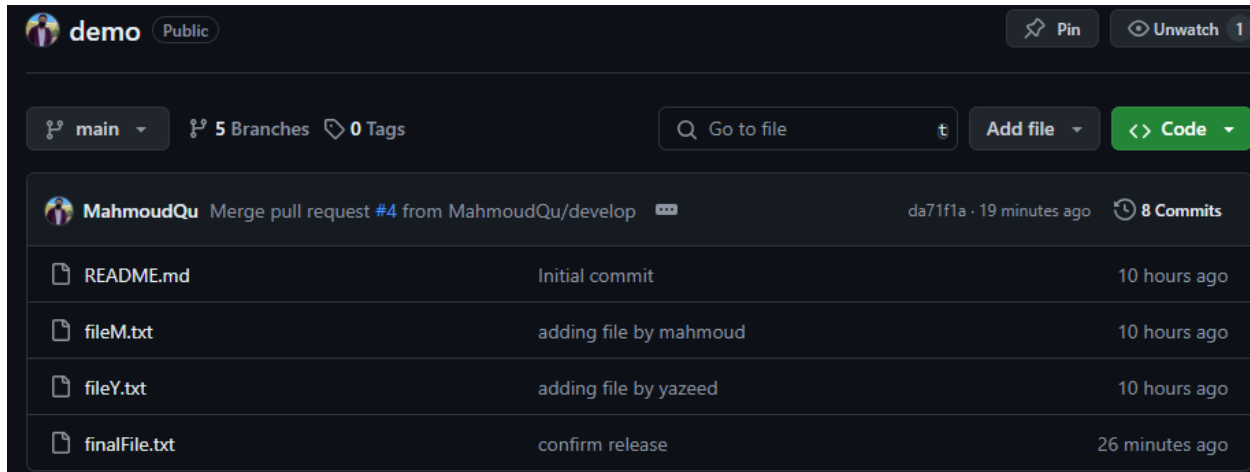
```
mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ ls
README.md  fileM.txt  fileY.txt

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ git merge release/1.0
Updating 0a71826..e91191b
Fast-forward
 finalFile.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 finalFile.txt

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ git push origin main
Everything up-to-date

mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Mahmoud/demo (develop)
$ ls
README.md  fileM.txt  fileY.txt  finalFile.txt
```

Main branch after merging release:



Hotfix, when to use? And why?

Hotfixes are crucial for addressing critical issues that need immediate attention in the production environment. A hotfix is typically added when a bug or issue is discovered in the main branch that impacts users and requires a swift resolution without waiting for the next release cycle.

Let's assume that the user "Yazeed" want to create one hotfix for this project, he want to add a file with the name "hotfix.txt", so he switched to the main branch, and created a hotfix/fix1 branch, the file is created and committed, and then he pushed the hotfix branch into the server to make pull request, after confirming the request, we got the fixing in the branch main,


```
mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Yazeed/demo (main)
$ git checkout -b hotfix/fix1
Switched to a new branch 'hotfix/fix1'



mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Yazeed/demo (hotfix/fix1)
$ vi hotfix.txt


mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Yazeed/demo (hotfix/fix1)
$ git add hotfix.txt
warning: in the working copy of 'hotfix.txt', LF will be replaced by CRLF the next time Git touches it




mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Yazeed/demo (hotfix/fix1)
$ git commit -m "add hotfix by yazeed"
[hotfix/fix1 6704e55] add hotfix by yazeed
1 file changed, 1 insertion(+)
 create mode 100644 hotfix.txt


mahmo@DESKTOP-U19PM2V MINGW64 ~/desktop/git_users/Yazeed/demo (hotfix/fix1)
$ git push origin hotfix/fix1
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 309.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
```





 demo Public

 Pin  Unwatch 1

 hotfix/fix1 had recent pushes 2 seconds ago Compare & pull request

 main  5 Branches  0 Tags t Add file <> Code

 MahmoudQu Merge pull request #4 from MahmoudQu/develop da71f1a · 37 minutes ago 8 Commits

 README.md	Initial commit	11 hours ago
 fileM.txt	adding file by mahmoud	10 hours ago
 fileY.txt	adding file by yazeed	10 hours ago
 finalFile.txt	confirm release	43 minutes ago

add hotfix by yazeed #5

Merged MahmoudQu merged 1 commit into main from hotfix/fix1  now

 Conversation 0  Commits 1  Checks 0  Files changed 1




 MahmoudQu commented now


No description provided.








  add hotfix by yazeed

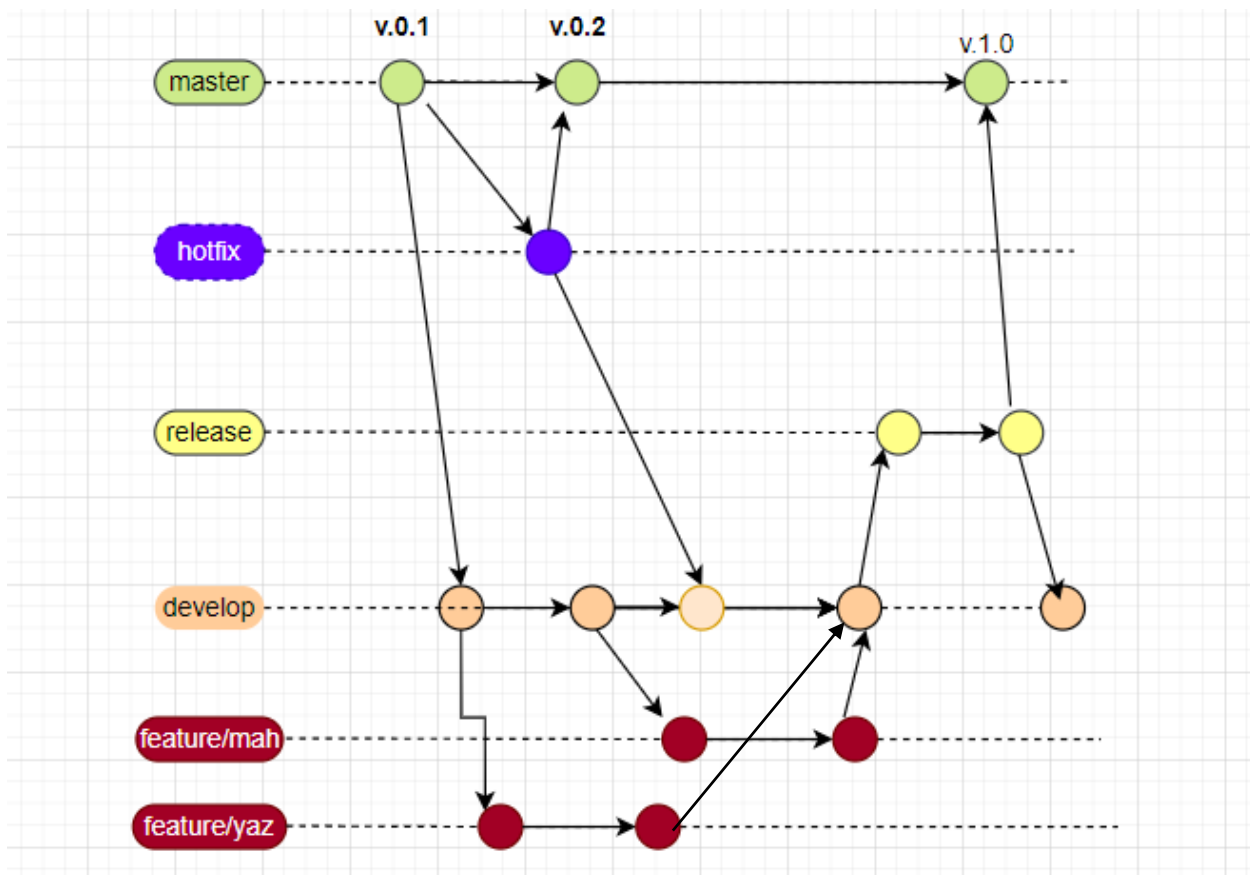
 MahmoudQu merged commit 0fc0884 into main now

 main  6 Branches  0 Tags

 MahmoudQu Merge pull request #5 from MahmoudQu/hotfix/fix1

 README.md	Initial commit
 fileM.txt	adding file by mahmoud
 fileY.txt	adding file by yazeed
 finalFile.txt	confirm release
 hotfix.txt	add hotfix by yazeed

Gitflow workflow chart:



So, we are done, and I would like to say that, through this assignment, I had the opportunity to learn about the real work environment, gaining practical experience with Git and the Gitflow workflow. Although it was a bit challenging, it was incredibly interesting and rewarding. The process helped me understand how collaborative development works and enhanced my skills in managing branches and integrating changes effectively.

More details can be found on the GitHub repository that I will share.

github repo link : <https://github.com/MahmoudQu/demo>

video link : https://youtu.be/e_zkXaScXWc

Thank You.