

Task 1.1: Create a normal folder in your local system.

Task 1.2: Convert it in the git working directory. - Command Screenshot

```
● dore@Reees-Macbook Devops % mkdir git-task
● dore@Reees-Macbook Devops % cd git-task
● dore@Reees-Macbook git-task % git init
  Initialized empty Git repository in /Users/dore/Desktop/Devops/git-task/.git/
○ dore@Reees-Macbook git-task % █
```

Task 1.3: Add any files in the working directory.

Task 1.4: Shift those files in the staging area - Command Screenshot

If required -Set Git Global name and email as required. Use email by which you can Login in your GitHub Account

```
● dore@Reees-Macbook git-task % git status
  On branch main

    No commits yet

    Untracked files:
      (use "git add <file>..." to include in what will be committed)
        notes.txt
        readme.txt

    nothing added to commit but untracked files present (use "git add" to track)
● dore@Reees-Macbook git-task % git add .
● dore@Reees-Macbook git-task % git status
  On branch main

    No commits yet

    Changes to be committed:
      (use "git rm --cached <file>..." to unstage)
        new file:   notes.txt
        new file:   readme.txt
```

Task 1.5: Shift those files to the Local repository - Command Screenshot

```
● dore@Reees-Macbook git-task % git commit -m "Initial commit"
[main (root-commit) 6439870] Initial commit
  2 files changed, 2 insertions(+)
  create mode 100644 notes.txt
  create mode 100644 readme.txt
○ dore@Reees-Macbook git-task % █
```

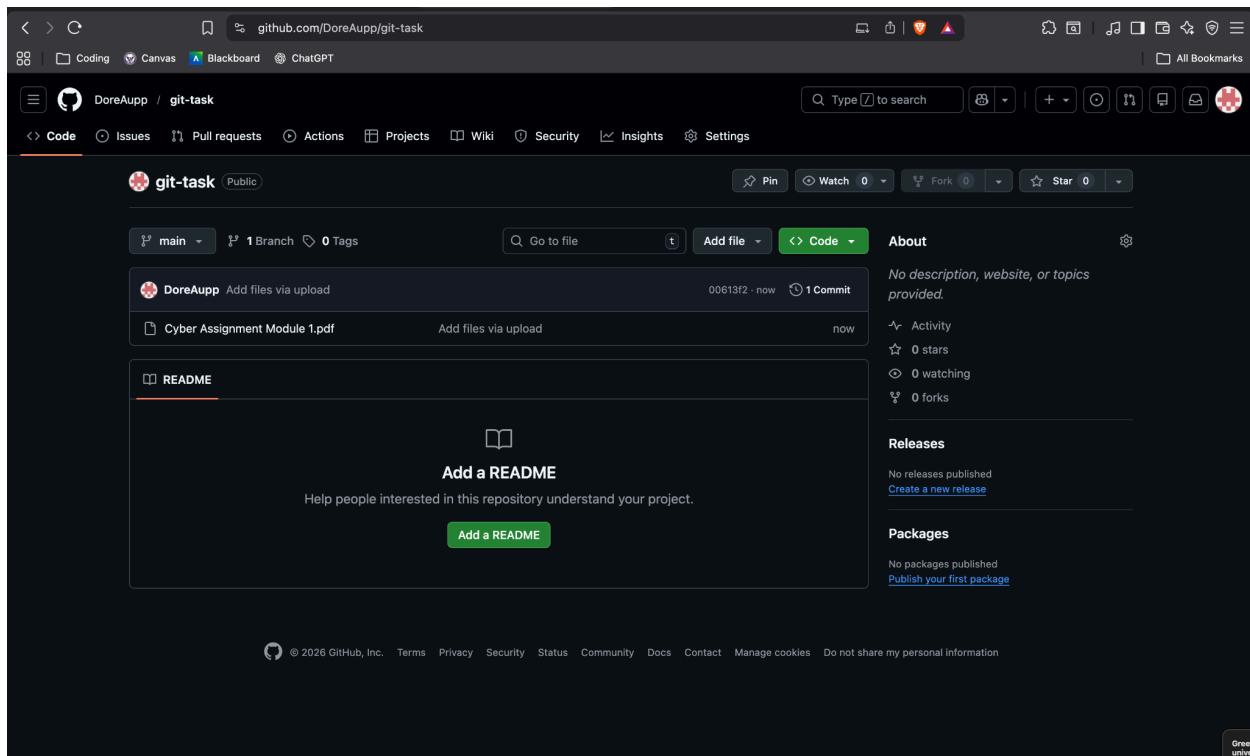
Task 1.6: Check and show by default in which branch files are added? Command / Output Screenshot

```
● dore@Reees-Macbook git-task % git branch
* main
```

Task 2.1: Create a repository in GitHub giving the same/different name as your local repository.

Task 2.2: Upload any different files in it

Task 2.3: Show by default in which branch files are added? Screenshot



Question 1: In this moment, without configuring anything is it possible to push local git repository to the GitHub repository? If no what things we need to do to resolve the issue - Answer / Command Screenshot

Ans Without any configuration, it is not possible to push from local to Github. We should first link the remote URL of the repository.

Solution

```
● dore@Reees-Macbook git-task % ssh -T git@github.com
Hi DoreAupp! You've successfully authenticated, but GitHub does not provide shell access.
● dore@Reees-Macbook git-task % git remote add origin git@github.com:DoreAupp/git-task.git
● dore@Reees-Macbook git-task % git remote -v
origin git@github.com:DoreAupp/git-task.git (fetch)
origin git@github.com:DoreAupp/git-task.git (push)
○ dore@Reees-Macbook git-task %
```

Question 2: After resolving the issue, if you push your repository in the GitHub, by default it will merge with main or will create another branch? Answer / Command Screenshot

Ans

Since I have changed the default name from master to main, it will merge.

Solution

```
● dore@Reees-Macbook git-task % git push -u origin main
  Enumerating objects: 7, done.
  Counting objects: 100% (7/7), done.
  Delta compression using up to 11 threads
  Compressing objects: 100% (4/4), done.
  Writing objects: 100% (6/6), 595 bytes | 595.00 KiB/s, done.
  Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
  To github.com:DoreAupp/git-task.git
    00613f2..f6b8ae5  main -> main
  branch 'main' set up to track 'origin/main'.
● dore@Reees-Macbook git-task % git branch
  * main
○ dore@Reees-Macbook git-task %
```

Question 3: Is it possible to change your local repository branch name from master to main? If yes, do it then check branch name. Screenshot

Ans

I have already changed the settings. So it will always be **main**.

Question 4: After change the branch name as main, is it possible to push the code in the GitHub and that files will merge in the main branch? Answer / Screenshot

Ans Yes, it will merge. However If GitHub main already has commits you don't have locally you must pull first.

Solution

Use **git pull origin main --allow-unrelated-histories**

And then push again.

Task 3.1: Clone your GitHub Repository Command / Output Screenshot

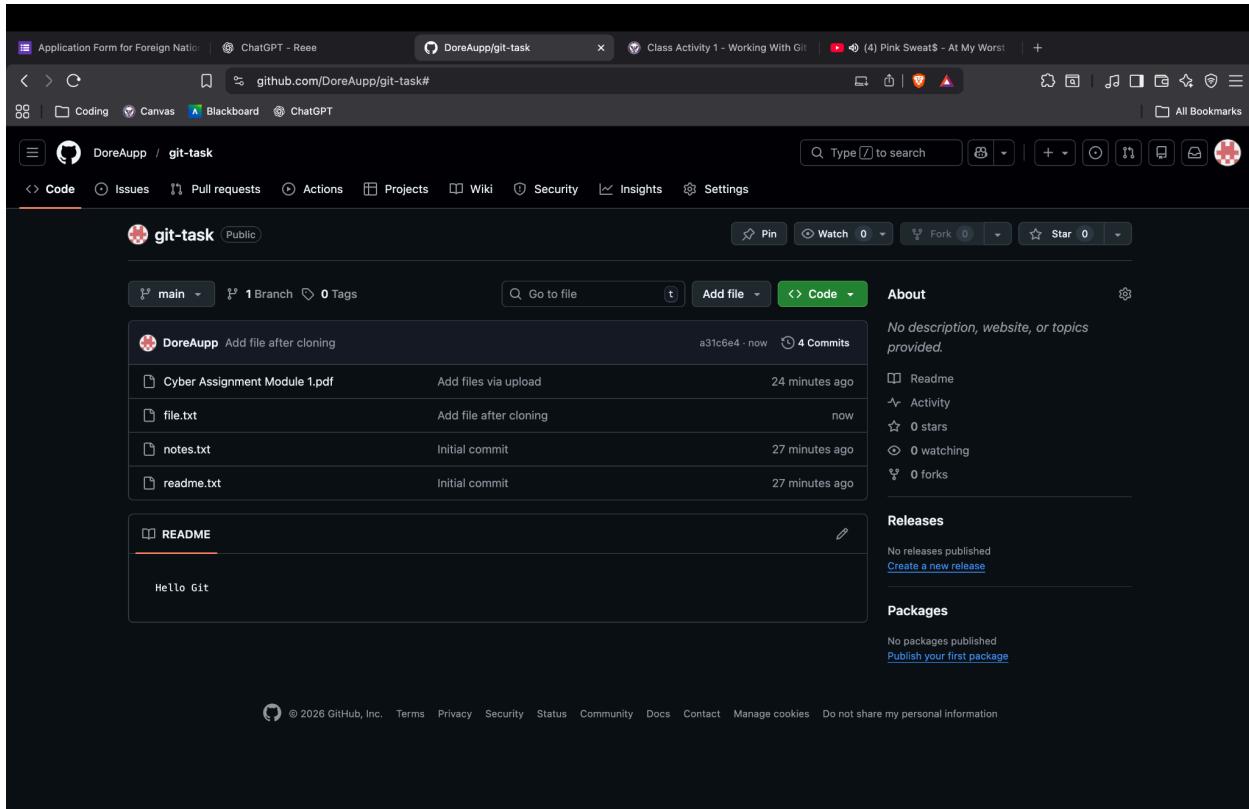
```
● dore@Reees-Macbook Devops2 % git clone git@github.com:DoreAupp/git-task.git
Cloning into 'git-task'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 1), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (9/9), 3.15 MiB | 2.25 MiB/s, done.
Resolving deltas: 100% (1/1), done.
○ dore@Reees-Macbook Devops2 % █
```

Task 3.2: Add new files in that Cloned working directory

Task 3.3. Do the needful to transfer your file from the working directory to the local repository.

Task 3.4: Push your local repository to the GitHub repository.

Task 3.5: Check in your GitHub repository whether the new files are added or not.
Screenshot



Ans Yes, I added a new file.txt and pushed it. It can be seen in the Github repository.

Task 4.1: Copy the same GitHub repository from One Account to the Other Account - Screenshot

The screenshot illustrates the process of forking a repository from one GitHub account to another.

Step 1: Creating a Fork

A user is on the GitHub interface, navigating to the repository `DoreAupp/git-task`. They click on the "Fork" button at the top right of the repository page. This leads to a "Create a new fork" form.

Form Fields:

- Owner:** DoreAupp2
- Repository name:** git-task-fork
- Description:** Cloning from another git account
- Copy the main branch only:** Checked

Step 2: Resulting Forked Repository

The user is now viewing the forked repository `DoreAupp2/git-task-fork`. The page shows the following details:

- Repository Name:** git-task-fork
- Forked From:** DoreAupp/git-task
- Branches:** main (selected), 1 Branch
- Commits:** 4 Commits
- Files:** Cyber Assignment Module 1.pdf, file.txt, notes.txt, readme.txt
- README:** Hello Git
- About:** Cloning from another git account
- Activity:** 0 stars, 0 watching, 0 forks
- Releases:** No releases published
- Packages:** No packages published

At the bottom of the page, there is a footer with links to GitHub's Terms, Privacy, Security, Status, Community, Docs, Contact, and Manage cookies, along with a "Do not share my personal information" checkbox.

Question 5: After copy if you change, update anything in your copied repository, Is there any effect on the original repository? Answer / Screenshot

Ans No any changes to the cloned repository will not affect the original repository.

The image contains two screenshots of GitHub repository pages. The top screenshot shows a forked repository named 'git-task-fork' at the 'main' branch. The commit history shows a deletion of a file named 'Cyber Assignment Module 1.pdf'. The bottom screenshot shows the original repository 'git-task' at the 'main' branch, which still contains the deleted file 'Cyber Assignment Module 1.pdf' in its history.

Screenshot 1: git-task-fork Repository

- Repository: git-task-fork at main · DoreAupp
- File successfully deleted: Cyber Assignment Module 1.pdf
- Branch status: This branch is 1 commit ahead of DoreAupp/git-task:main.
- Commit History:

 - file.txt: Add file after cloning (4 minutes ago)
 - notes.txt: Initial commit (30 minutes ago)
 - readme.txt: Initial commit (30 minutes ago)

- README content: Hello Git

Screenshot 2: git-task Repository

- Repository: git-task Public · DoreAupp
- Branch: main · 1 Branch · 0 Tags
- Commit History:

 - Cyber Assignment Module 1.pdf: Add files via upload (28 minutes ago)
 - file.txt: Add file after cloning (4 minutes ago)
 - notes.txt: initial commit (31 minutes ago)
 - readme.txt: initial commit (31 minutes ago)

- README content: Hello Git
- About: No description, website, or topics provided.
- Activity: 0 stars, 0 watching, 1 fork.
- Releases: No releases published. Create a new release.
- Packages: No packages published. Publish your first package.

Remark: As it can be seen, I've deleted the file named "Cyber Assignment Module 1.pdf" from the cloned repository, however that file still remains in the original repository.

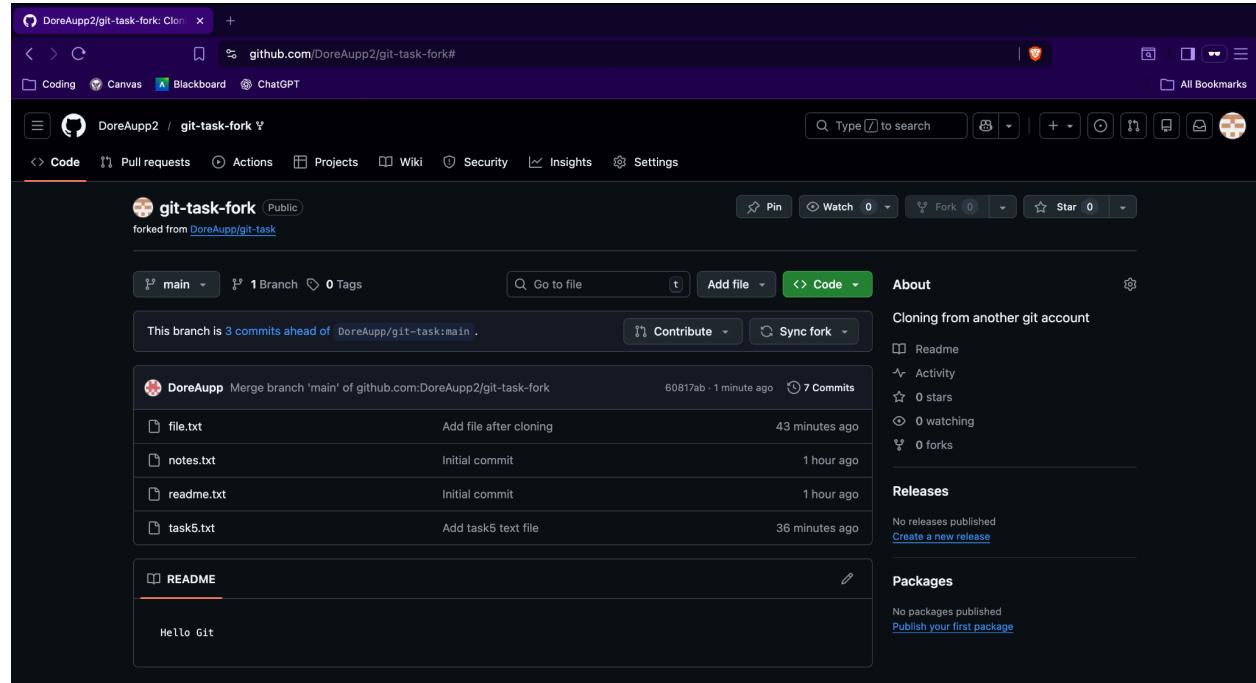
Task 5.1: Add 1 new file in your local working directory (previously cloned repository) and shift it to local repository.

Question 6: Without changing any configuration settings, is it possible to push the local repository to the GitHub copied repository? If yes, do it. If no resolve issue and then Push. Answer/Screenshot.

Ans No, that is not possible. Since the repository that your Git remote is connected to is still the previous repository (the original one). Thus, we have to re-route it to the new cloned repository and only then can we push it.

```
① dore@Reees-Macbook git-task % ssh -T git@github.com-doreaupp2
  Hi DoreAupp2! You've successfully authenticated, but GitHub does not provide shell access.

● dore@Reees-Macbook git-task % git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 11 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 517 bytes | 517.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To github.com:DoreAupp2/git-task-fork.git
  1e08f4f..60817ab  main -> main
branch 'main' set up to track 'origin/main'.
```



Note: After configuration, I now can push the new file named “task5.txt” to the cloned repository