***SOFTWARE CONSTRUCTION AND DEVELOPMENT***

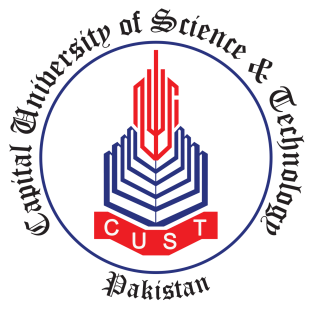
***PROJECT***

***Submitted by: Muhammad Abdullah Warsi***

***Moazzam ali***

***Jibran tariq***

***Zain-ul-Abidin***

****

**CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY**

**Question no 1:**

**What is Git?**

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. A Git repository is a virtual storage of your project. It allows you to save versions of your code, which you can access when needed.

Git helps you in following

* Creating/initializing new git repository
* Versioning project with new git repository
* Cloning existing git repository
* Saving changes to the repository

**Question no 2:**

**What are git workflows?**

Some common Git workflow are following

* Centralized workflow
* Feature Branching
* Gitflow workflow
* Forking workflow

**1)Centralized Workflow**

There is one central repository. Each developer clones the repo, works locally on the code, creates a commit with changes, and pushes it to the central repository for other developers to pull and use in their work.Centralized Workflow uses a central repository to serve as the single point-of-entry for all changes to the project.

**How it works?**

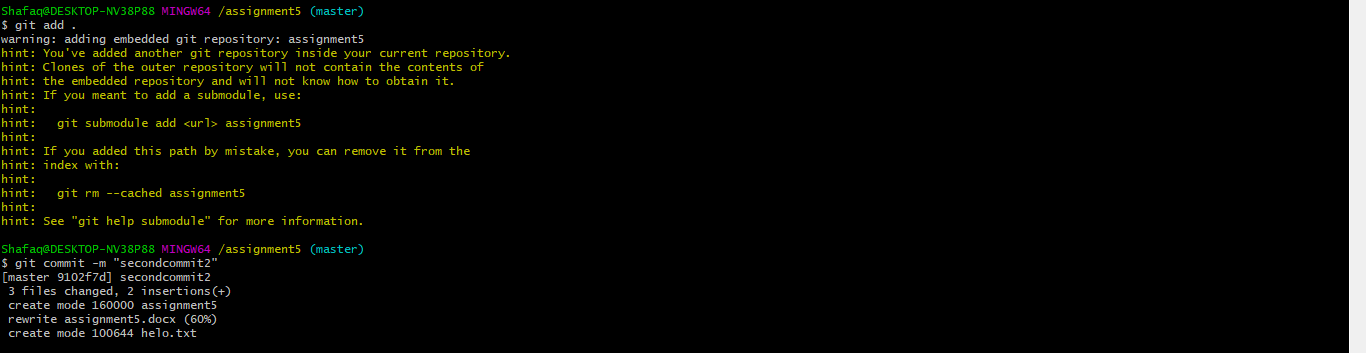
Developers start by cloning the central repository. In their own local copies of the project, they edit files and commit changes, however, these new commits are stored locally. To publish changes to the official project, developers "push" their local repository to the central repository.

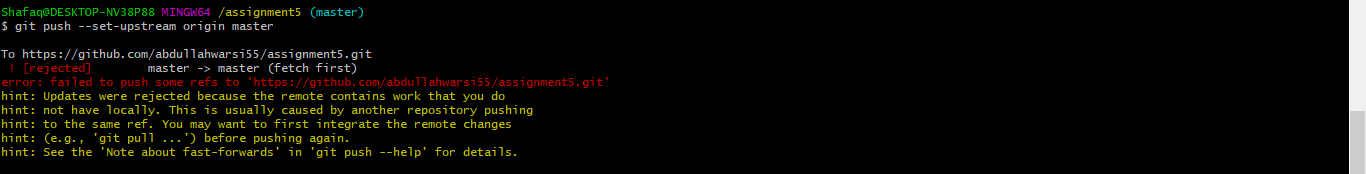
The central repository represents the official project, so its commit history should be treated as sacred and immutable. If a developer’s local commits diverge from the central repository, Git will refuse to push their changes because this would overwrite official commits.

**Example**

In his local repository, Ali can develop features using the standard Git commit process: edit, stage, and commit. Remember that since these commands create local commits, Ali can repeat this process as many times as he wants without worrying about what’s going on in the central repository. Meanwhile, Ahmed is working on her own feature in her own local repository using the same edit/stage/commit process. Like ali, he doesn’t care what’s going on in the central repository, and he really doesn’t care what ali is doing in his local repository, since all local repositories are private. Once ali finishes his feature, he should publish his local commits to the central repository so other team members can access it. He can do this with the git push command, like so if Ahmed tries to push her feature after ali has successfully published his changes to the central repository. but, since her local history has diverged from the central repository, Git will refuse the request with a rather verbose error message This prevents ahmed from overwriting official commits. he needs to pull ali’s updates into her repository, integrate them with her local changes, and then try again.





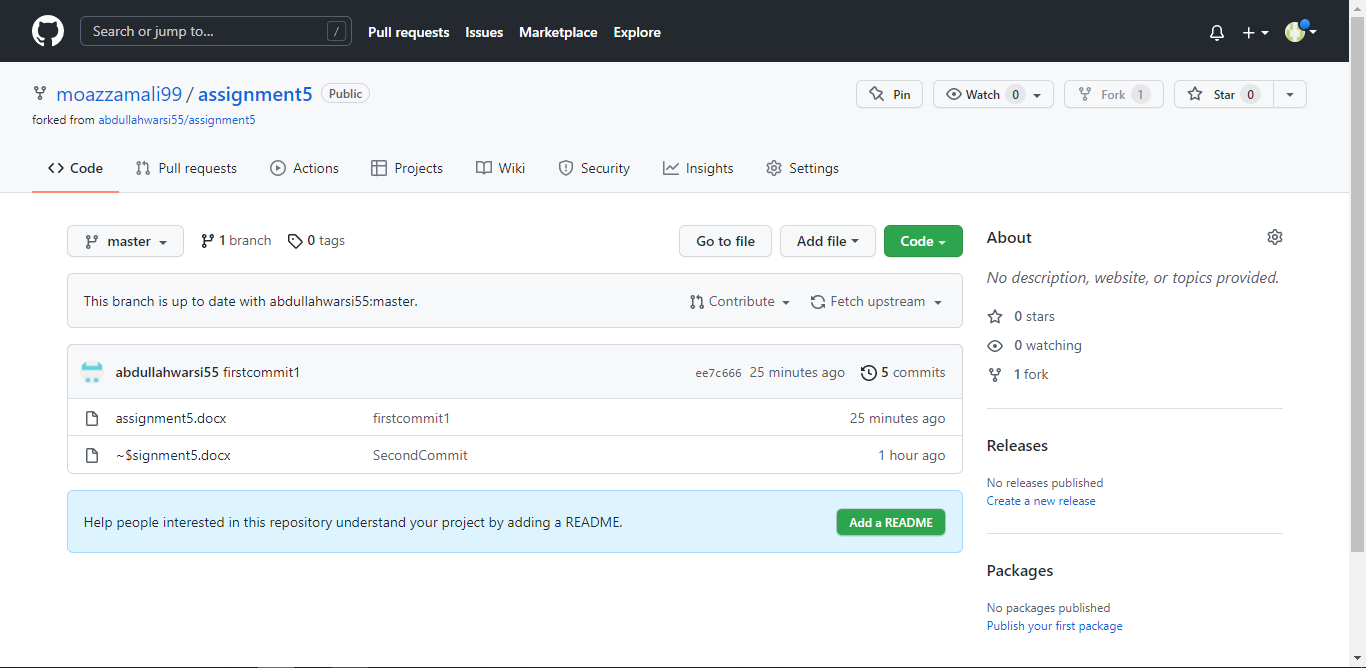


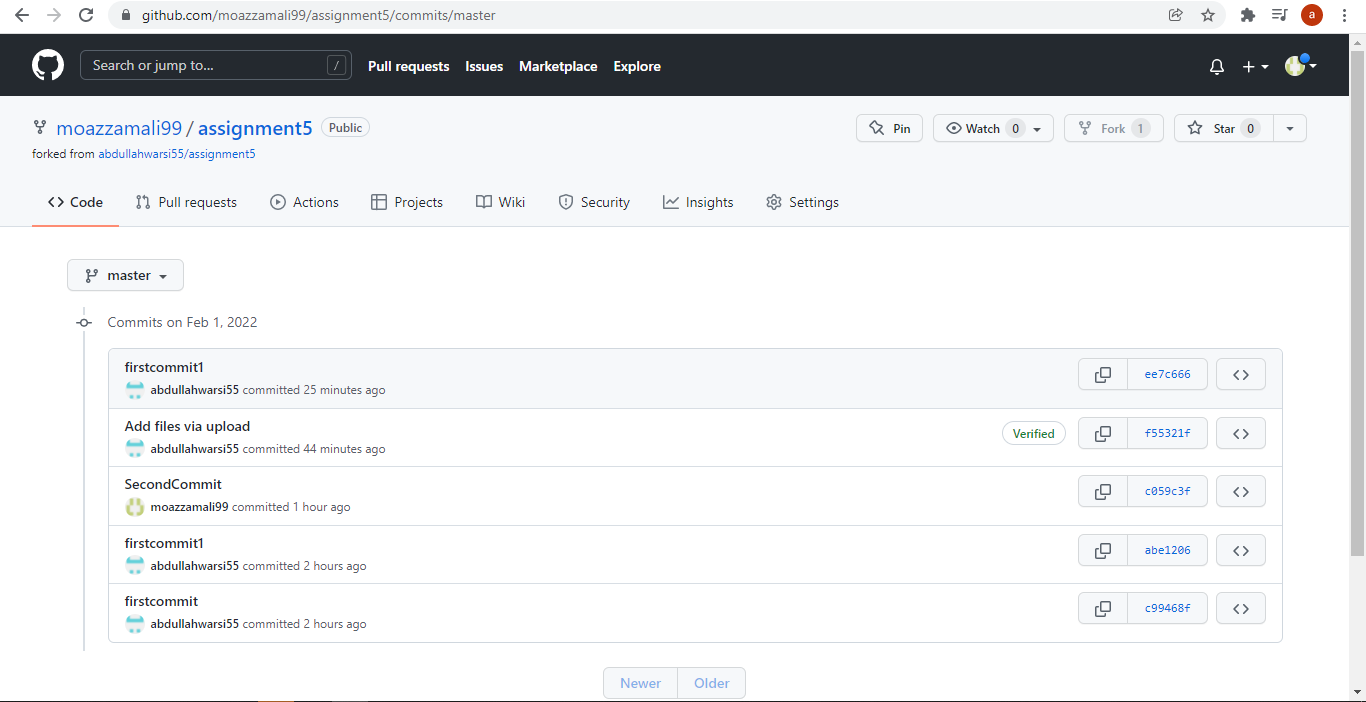
**2) Forking workflow**

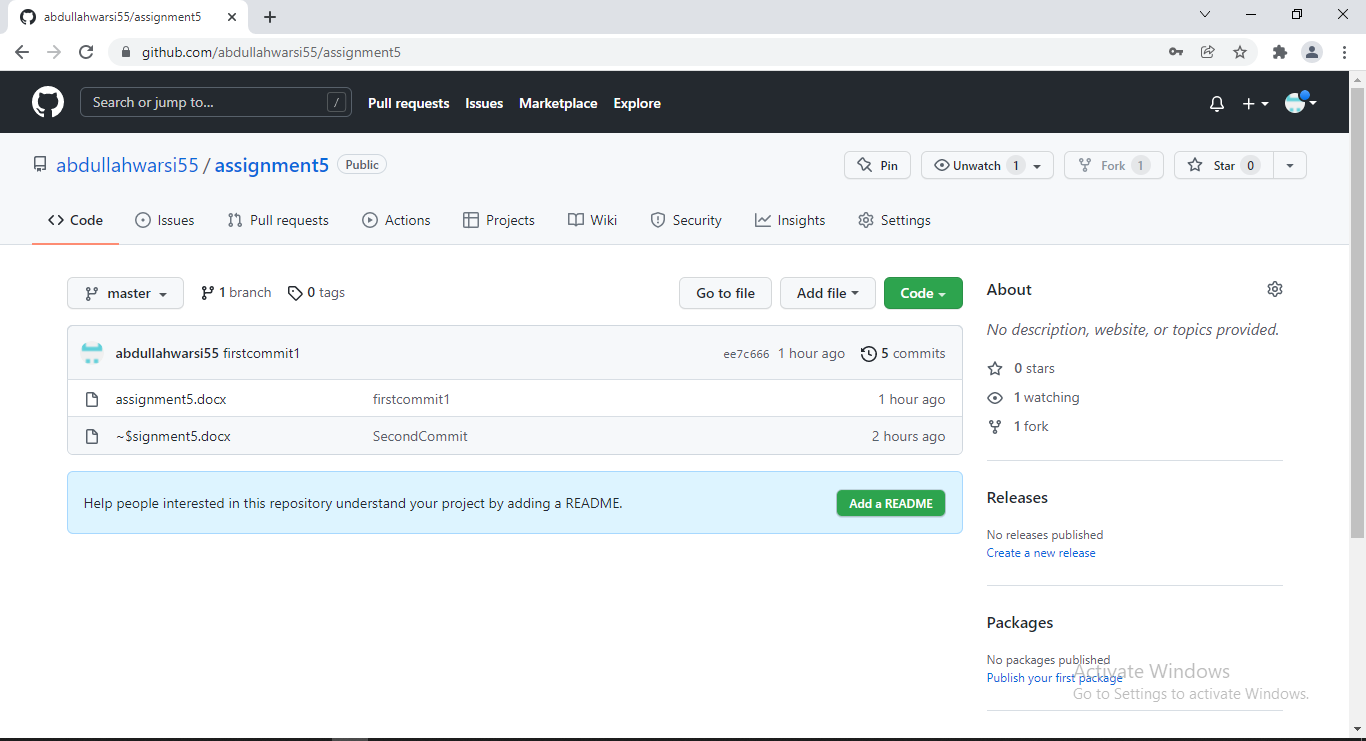
A fork is a copy of a repository that you manage. Forks let you make changes to a project without affecting the original repository. You can fetch updates from or submit changes to the original repository with pull requests. Forking a repository is similar to copying a repository, with two major differences:

* You can use a pull request to suggest changes from your user-owned fork to the original repository, also known as the upstream repository.
* You can bring changes from the upstream repository to your local fork by synchronizing your fork with the upstream repository

.







**Simple push, clone and pull**

