**GIT Assignment**

**Assignment - 1**

1. Set up a Local Git Repository:
   * Create a new directory on your local machine.
   * Initialize a new Git repository in the directory using the command: git init.
2. Create Commits:
   * Create a new text file named "file1.txt" within the repository directory.
   * Add some content to "file1.txt".
   * Add the file to the staging area using the command: git add file1.txt.
   * Commit the changes with a descriptive message using the command: git commit -m "Add file1.txt".
3. Create and Switch Branches:
   * Create a new branch named "feature" using the command: git branch feature.
   * Switch to the "feature" branch using the command: git checkout feature.
   * Make changes to "file1.txt" by adding additional content.
   * Commit the changes on the "feature" branch.
4. Merge Branches:
   * Switch back to the master branch using the command: git checkout master.
   * Merge the changes from the "feature" branch into the master branch using the command: git merge feature.
   * Resolve any conflicts that may arise during the merge process.
5. Push to a Remote Repository:
   * Create a remote repository on a hosting platform (e.g., GitHub, GitLab).
   * Add the remote repository URL as a remote named "origin" using the command: git remote add origin <remote-url>.
   * Push the local repository to the remote repository using the command: git push -u origin master.

**Assignment – 2**

Using the above assignment Complete the below steps.

1. Collaborate with Others:

* Share your local repository with a partner or colleague.
* Have them clone the repository to their local machine.
* Have them create a new branch, make changes, and commit them.
* Fetch and merge their changes into your local repository.
* Resolve any conflicts that may arise during the merge process.
* Push the local repository to the remote repository using the command: git push -u origin master.

**Assignment - 3**

1. Initialize a new Git repository in a directory named "my\_project".
2. Create a new file named "README.md" and add some content to it.
3. Add "README.md" to the staging area.
4. Commit the changes with the message "Initial commit".
5. Create a new branch named "feature/add-login".
6. Switch to the "feature/add-login" branch.
7. Create a new file named "login.html" and add some content to it.
8. Add "login.html" to the staging area.
9. Commit the changes with the message "Add login page".
10. Switch back to the master branch.
11. Merge the "feature/add-login" branch into the master branch.
12. Resolve any conflicts that may arise during the merge.
13. Create a new branch named "bugfix/fix-typos".
14. Switch to the "bugfix/fix-typos" branch.
15. Edit some files to fix typos and make necessary changes.
16. Add the modified files to the staging area.
17. Commit the changes with the message "Fix typos and make changes".
18. Push the master branch to a remote repository named "origin".
19. Clone the remote repository "origin" to a new directory on your local machine.
20. Configure your global Git username and email for commits.

**Assignment - 4**

1. How do you initialize a new Git repository?
2. What command is used to add a file to the staging area?
3. How do you commit changes to the repository with a descriptive message?
4. What command is used to create a new branch?
5. How do you switch to a different branch?
6. How do you merge changes from one branch into another?
7. What is a Git conflict, and how do you resolve it?
8. How do you view the commit history of a Git repository?
9. What command is used to create a new remote repository?
10. How do you add a remote repository to your local repository?
11. How do you push local changes to a remote repository?
12. What is the purpose of the command git clone?
13. How do you view the differences between two commits?
14. How do you discard changes in a file and revert it to the last committed version?
15. What is the purpose of the command git stash?
16. How do you create a new tag in Git?
17. How do you view the branches in your Git repository?
18. How do you remove a branch from your local repository?
19. How do you update your local repository with the latest changes from the remote repository?
20. How do you configure a global username and email for Git commits?

Assignment -5 (Azure Repos)

1. Create an Azure DevOps Organization:
   * Sign in to the Azure DevOps portal (dev.azure.com) using your Azure account.
   * Create a new organization with a unique name.
   * Choose a project name and set it up within the organization.
2. Create a Git Repository:
   * Inside the project, create a new Git repository.
   * Name the repository "MyProject".
3. Clone the Repository:
   * Clone the "MyProject" repository to your local development environment using your preferred Git client.
4. Create a Branch:
   * Switch to a new branch named "feature/add-new-feature".
   * Make sure you are on the newly created branch before proceeding to the next steps.
5. Make Changes and Commit:
   * Create a new text file named "new-feature.txt".
   * Add some content to the file.
   * Commit your changes with a meaningful commit message.
6. Push the Changes:
   * Push the changes to the remote repository on the "feature/add-new-feature" branch.
7. Create a Pull Request:
   * Go back to the Azure DevOps portal and navigate to the "MyProject" repository.
   * Create a new pull request to merge your "feature/add-new-feature" branch into the "main" branch.
   * Assign a reviewer (can be yourself).
   * Add a description explaining the changes made.
   * Submit the pull request.
8. Review and Merge the Pull Request:
   * Review the changes made in the pull request.
   * If everything looks good, approve the pull request and merge it into the "main" branch.
9. Pull the Latest Changes:
   * Switch back to the "main" branch.
   * Pull the latest changes from the remote repository to your local development environment.
10. Verify Changes:
    * Check if the "new-feature.txt" file is now available in the "main" branch.

Submission:  
Take screenshots of the following steps and provide a brief explanation of each step:

1. Creating the Azure DevOps organization.
2. Creating the "MyProject" repository.
3. Cloning the repository to your local development environment.
4. Creating and switching to the "feature/add-new-feature" branch.
5. Making changes and committing them.
6. Pushing the changes to the remote repository.
7. Creating and submitting the pull request.
8. Reviewing and merging the pull request.
9. Pulling the latest changes to the "main" branch.
10. Verifying the changes in the "main" branch.