Git Task 1 Solution

Task 1 - Git Fundamentals

Theoretical Part:

1. Explain the purpose of the following Git commands:

a. git status:
Displays the current state of the working directory and the staging area. It shows which changes have been staged, which haven't, and which files aren't being tracked by Git.

b. git init:
Initializes a new Git repository in the current directory. It sets up the necessary Git structure so you can start version controlling files.

c. git commit:
Records changes to the repository with a message describing what has been changed. Commits are snapshots of your project at specific points in time.

d. git push:
Uploads local repository content to a remote repository. It transfers commits from your local branch to a branch on the remote.

2. Ahmed claims that there is no need for the git add command, and it is better to commit changes directly from the working directory.

Answer:
I disagree with Ahmed’s statement. The git add command is important because it allows you to choose which changes to include in the next commit. This gives you more control over what gets committed and when. Committing directly without staging can lead to mistakes or unintentional commits.

3. Answer the following questions about code hosting providers:

a. List two popular code hosting platforms besides GitHub:
- GitLab
- Bitbucket

b. What key features do code hosting providers offer to enhance version control for both individual developers and collaborative development environments?

- Pull requests and merge requests
- Issue tracking
- Continuous integration/continuous deployment (CI/CD)
- Access control and user management
- Code review tools
- Wiki and project documentation support

c. True or False: A team can collaborate on a software project without relying on popular code hosting providers by setting up their own self-hosted server for version control.
- True. A team can use a self-hosted Git server (e.g., GitLab CE, Gitea) for collaboration without relying on cloud-based platforms.

4. What does the -u flag in the git push -u origin main command signify, and how does it impact the behavior of the git push and git pull commands in subsequent usage?
- The -u flag sets the upstream (tracking) branch for the local branch. This allows future git push and git pull commands to automatically know which remote branch to sync with, so you don’t need to specify origin and branch name every time.

Practical Part:

To connect your local Git repository to GitHub using SSH:

Steps and Commands:

1. Generate SSH key (if not already created):

```
  ssh-keygen -t ed25519 -C "your_email@example.com"
```

2. Add SSH key to the ssh-agent:

```
  eval "$(ssh-agent -s)"
  ssh-add ~/.ssh/id_ed25519
```

3. Copy SSH public key and add it to your GitHub account:

```
  cat ~/.ssh/id_ed25519.pub
```

  (Paste it into GitHub > Settings > SSH and GPG keys > New SSH key)

4. Initialize a Git repository locally:

```
  git init
```

5. Add a file and make a commit:

```
  echo "Git practice task" > task1.txt
  git add task1.txt
  git commit -m "Add task1 answers"
```

6. Add remote repository using SSH:

```
git remote add origin git@github.com:your-username/your-repo-name.git
```

7. Push to remote repository and set upstream:

```
git push -u origin main
```