Assignment

**1. What is Git? Explain its key features and role in version control.**  
Git is a widely used distributed version control system (VCS) for tracking changes in source code. It allows developers to work collaboratively on projects without overwriting one another’s contributions. The key features of Git include:

* **Branching**: Developers can create independent branches to work on new features or bug fixes without affecting the main codebase.
* **Distributed Nature**: Each developer has a full copy of the repository, enabling fast performance and offline access.
* **Commit History**: Every change is recorded with detailed messages, which helps in tracking the project’s evolution.
* **Staging Area**: Files can be added to a staging area before committing them to the repository, allowing for better organization of changes. Git plays a crucial role in version control by enabling developers to work simultaneously on different aspects of a project, track revisions, and roll back to earlier states when needed.

**2. What is GitHub? How does it facilitate collaboration and project management?**  
GitHub is a web-based platform built around Git, designed for hosting repositories and supporting team collaboration. It provides a graphical interface and several additional features like:

* **Pull Requests**: Teams can review and discuss changes before they are merged into the main codebase.
* **Issue Tracking**: It helps teams manage tasks, bugs, and new feature requests in an organized manner.
* **Collaboration**: GitHub makes it easy for teams to collaborate, share code, and manage projects from one central platform. By enhancing Git with collaborative tools, GitHub enables centralized repository hosting, efficient teamwork, and streamlined project management.

**3. How do GitHub and Bitbucket compare?**

* **Features**: GitHub supports only Git, while Bitbucket (formerly) supported both Git and Mercurial. GitHub is more known for its public repositories, whereas Bitbucket offers extensive support for private repositories.
* **User Interface**: GitHub’s interface is simple and widely recognized, featuring pull requests and issues tracking as core elements. On the other hand, Bitbucket’s interface integrates closely with Jira, which is popular with teams using Atlassian’s suite of tools.
* **Integration**: GitHub is well-integrated with services like GitHub Actions, third-party CI/CD tools, and other platforms like Slack and Trello. Bitbucket, however, provides deeper integration with Jira and Confluence, which is advantageous for teams using Atlassian’s ecosystem.

**4. How do you initialize a Git repository in a project?**

* Command: git init
* **Purpose**: This command sets up a new Git repository in the current directory, initializing all necessary files and directories to start tracking changes.

**5. How do you stage a file in Git after making changes?**

* Command: git add <file>
* **Purpose**: The git add command adds the specified file to the staging area, preparing it for a future commit. This step allows you to select which changes will be included in the next commit.

**6. How do you commit staged changes?**

* Command: git commit -m "Commit message"
* **Purpose**: This command saves the staged changes to the repository, along with a commit message that describes the modifications. Commit messages are essential for keeping a clear history of changes.

**7. How do you check the status of a Git repository, and why is this necessary?**

* Command: git status
* **Purpose**: Running this command shows the current state of the working directory and the staging area, such as which files have been modified or staged. Checking the status is crucial for ensuring the correct changes are being prepared for commit.

**8. How do you create a new branch in Git, and what is branching used for?**

* Command: git branch <branch-name>
* **Purpose**: Branching allows developers to create a parallel line of development where they can experiment or work on features without affecting the main project. Once finished, branches can be merged back into the main branch.

**9. How do you switch to another branch in Git?**

* Command: git checkout <branch-name>
* **Purpose**: This command switches you to another branch, which is useful when you need to work on or review code changes in a different feature or version of the project.

**10. How do you merge one branch into another in Git?**

* Command: git merge <branch-name>
* **Purpose**: Merging takes the changes from one branch and incorporates them into another branch. This process is critical for bringing together different lines of development, particularly in collaborative projects.

**11. How do you push changes from your local repository to GitHub?**

* Command: git push origin <branch-name>
* **Purpose**: This command pushes the committed changes from your local branch to the corresponding branch on a remote repository like GitHub, updating it with your latest changes.

**12. How do you fetch and incorporate changes from a remote repository?**

* Command: git pull
* **Purpose**: The git pull command fetches changes from a remote repository and merges them with your local branch, ensuring your local repository stays synchronized with the remote one.

By: Azan Ijaz Butt, F2023-745