Assignment no 05

Group Members:

Husnat Khalid Raja **BSE181010**

Syed Ammar Ali Shah **BSE181027**

Muhammad Ishaq **BSE181031**

Jibran Rafi **BSE181029**

Question no 1: What is Git?

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. So, what is Git in a nutshell? This is an important section to absorb, because if you understand what Git is and the fundamentals of how it works, then using Git effectively will probably be much easier for you. As you learn Git, try to clear your mind of the things you may know about other VCSs, such as CVS, Subversion or Perforce — doing so will help you avoid subtle confusion when using the tool. Even though Git's user interface is fairly similar to these other VCSs, Git stores and thinks about information in a very different way, and understanding these differences will help you avoid becoming confused while using it.

Question no 2: what is Git workflow?

A Git workflow is a recipe or recommendation for how to use Git to accomplish work in a consistent and productive manner. Git workflows encourage developers and DevOps teams to leverage Git effectively and consistently. Git offers a lot of flexibility in how users manage changes. Given Git's focus on flexibility, there is no standardized process on how to interact with Git. When working with a team on a Git-managed project, it's important to make sure the team is all in agreement on how the flow of changes will be applied. To ensure the team is on the same page, an agreed-upon Git workflow should be developed or selected. There are several publicized Git workflows that may be a good fit for your team. Here, we will discuss some of these Git workflow options.

Centralized workflow:

The Centralized Workflow is a great Git workflow for teams transitioning from SVN. Like Subversion, the Centralized Workflow uses a central repository to serve as the single point-of-entry for all changes to the project. Instead of trunk, the default development branch is called main and all changes are committed into this branch. This workflow doesn't require any other branches besides main.

Feature Branching:

Feature Branching is a logical extension of Centralized Workflow. The core idea behind the Feature Branch Workflow is that all feature development should take place in a dedicated branch instead of the main branch. This encapsulation makes it easy for multiple developers to work on a particular feature without disturbing the main codebase. It also means the main branch should never contain broken code, which is a huge advantage for continuous integration environments.