**What happens when you initialize a repository? Why do you need to do it?**

**When you initialize a repository, a hidden git directory is created.**

**This make the directory a git repository.**

You need to initialize to create a repository that helps in the tracking of your files.

**How is the staging area different from the working directory and the repository?**

Staging area is a temporary location for adding files before committing them to the repository while working directory is the file system that stores the files and repository is the directory with the git hidden file which stores file with its different versions. Staging offers temporary file storage before committing the changes

**What value do you think it offers?**

Staging help with temporary storage which can prevent over writing of the repository with code one is not sure of

**How can you use the staging area to make sure you have one commit per logical change?**

Using git add command, add files to staging area, while adding file to staging area, add useful description and only when done with all files addition should one using the command git commit to commit all changes that has been added as unique file

**What are some situations when branches would be helpful in keeping your history**

**organized? How would branches help?**

Branches are useful when you want to try out an idea you are not ready to

Add to the working project or when you want a different feature outside of the working project

Gotten from the interview during lecture

\*\*Branching keeps you organised, helps compartmentalise

Example of branches you could have

Experimental branch

New feature branch

Development branch

Working/production branch

**How do the diagrams help you visualize the branch structure?**

**What is the result of merging two branches together? Why do we represent it in**

**the diagram the way we do?**

**What are the pros and cons of Git's automatic merging vs. always doing merges**

**manually?**