**GITHUB EXERCISE**

**What is GitHub? When was it created? Why? By who? What similar platforms exist? Why would you use such a platform?**

GIT HUB At a high level, GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. Executive chairman Chris Wanstrath, now 33, co-founded San Francisco-based GitHub in 2008 along with fellow software developers **Tom Preston-Werner**, 38, and PJ Hyett, 35. There are several competitor platforms available like GitHub. Some of them are GitLab, Microsoft Team Foundation Server, Bitbucket, Assembla, Phabricator, Gerrit, etc. Around the same time, you probably have several people working on projects, and all of them might be in different areas, maybe even in different nations. Collaborators can work together without getting on each other's toes by using a version management system like GitHub.

**Define the following terms in the context of Git.**

* **Repository** - GIT repositories include a set of files holding different copies of a project. These files are downloaded from the registry into the local server of the user for further updations and changes in the content of the file.
* **Commit** - "A commit is an individual update to a file (or set of files). It's like saving a file, but for Git, generating a special ID any time you save it, which helps you to keep track of what improvements were made when and by whom. Commits usually provide a commit note that is a short summary of what changes have been made.
* **Push** - To upload local repository material to a remote repository, the git push command is used. Pushing is how you move commits to a remote repository from your local repository.
* **Branch** - A Git branch is essentially a movable lightweight pointer to one of these commits. Master is the default name of the branch in Git. You're given a master branch when you start making commitments, which points to the last commit you made. The master branch pointer moves forward automatically every time you commit.
* **Fork** - Forking is a git cloning process executed on a project repo server replica. A Forking Workflow is mostly used in combination with a Bitbucket-like Git hosting program.
* **Merge** - Merging is Git's way of getting back together again a forked past. You may take the separate development lines generated by the git branch and combine them into a single branch using the git merge order.
* **Clone** - A local copy of the remote repo is generated by a GitHub repository. This helps you to make all your changes in the source files of the root repo locally rather than directly.
* **Pull and pull request** - Pull requests allow you to inform others about improvements you've made to a branch of a GitHub repository. If a pull request is opened, before the modifications are incorporated into the base branch, you will discuss and check the possible improvements with colleagues and add follow-up commits.