**Part 3**

**What is GitHub?**

Developers create a program, app or browser extension and make constant changes to the code, releasing new versions up to and after the first stable release. It track the complete history of all changes to that code. So, Git is a Version-control systems that store these revisions in a central repository, which allows other developers to easily collaborate on the project and make and share their own changes.

It 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.  GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Repositories can have multiple collaborators and can be either public or private.

**When was it created?**

GitHub was created in February 2008. The company, GitHub, Inc., has existed since 2007 and is located in San Francisco.

**Why?**

Git was createdby Linus Torvalds in 2005 for development of the Linux kernel. It has multiple advantages over the other systems available. It stores file changes more efficiently and ensures file integrity better. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers.

**By who?**

GitHub was developed by Chris Wanstrath, PJ Hyett, Tom Preston-Werner and Scott Chacon using Ruby on Rails.

**What similar platforms exist?**

**Bitbucket** is a powerful, fully scalable and high-performance development platform designed to connect up to five users to the platform.

**GitLab** is an open source, powerful, secure, efficient and robust application for handling software development and operations (DevOps) lifecycle.

[**GitBucket**](https://gitbucket.github.io/)is an open source, with features such as a repository viewer, issues tracker, pull requests, pluggable Git platform that runs on **JVM** (**Java Virtual Machine**).

[**Launchpad**](https://launchpad.net/) is a fully free, well known platform for building, managing and collaborating on software projects.

**Why would you use such a platform?**

* Easy to contribute to the open source projects.
* Present your work
* It’s a repository
* Tracking changes in the code is easy
* Integration options
* collaborating with other developers

**Part 5**

* Repository - you'll need to create a repositoryto put your project up on GitHub, for it to live in. All the files for a particular project are stored in a repository. Each project has its own repository, and you can access it with a unique URL.
* Commit - A commit is some changes you make to your file, every time you save it, it creates a unique ID that allows you to keep record of what changes were made by whom and when.
* Push - if you make some changes in your project locally, you will have to push those changes to a repository on GitHub, so that others may access them.
* Branch - It is contained within the repository, but does not affect the primary or master branch allowing you to work freely without disrupting the "live" version. When you've made the changes you want to make, you can merge your branch back into the master branch to publish your changes
* Fork - forks are used to either propose changes without affecting the original project. A fork is a personal copy of another user's repository that lives on your account. Forks remain attached to the original, allowing you to submit a pull request to the original's author to update with your changes.
* Merge - Merging takes the changes from one branch and applies them into another. This can also be done automatically via a pull request/merge request or via the command line.
* Clone – Clone is a copy of the git repository that lives on your local machine .A clone is a copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the [remote](https://help.github.com/articles/github-glossary/#remote) to keep them synced when you're online.

When you create a repository on **GitHub**, it exists as a remote repository. You can **clone** your repository to create a local copy on your computer and sync between the two locations.

This procedure assumes you have already [created a repository on GitHub](https://help.github.com/articles/creating-a-new-repository), or have an existing repository owned by someone else you'd like to contribute to.

* Pull - Pull refers to when you are fetching in changes and merging them. For instance, if someone has edited the remote file you're both working on, you'll want to pull in those changes to your local copy so that it's up to date.

git pull is a convenient shortcut for completing both git fetch and git mergein the same command

* Pull request - Pull **requests** let you tell others about changes you've **pushed** to a **GitHub** repository. Once a pull **request** is sent, interested parties can review the set of changes, discuss potential modifications, and even **push** follow-up commits if necessary.

Pull requests let you tell others about changes you've pushed to a branch in a repository on GitHub. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add follow-up commits before your changes are merged into the base branch.

Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators. Like issues, pull requests each have their own discussion forum.