**GITHUB**

GitHub is a code hosting platform for collaboration and version control.

GitHub lets you (and others) work together on projects.

A blue and white background with text and a logo

Description automatically generated

So let’s imagine you and your friends play together with lego blocks and create an awesome huge house.

Now you have created the house, it has lots of features, it is a multi-story house with many rooms, a play area, a kitchen, and washing area, and it has moving doors and windows.

Now just imagine you decide to keep this big Lego house in a common area and any of your friends or you can go there and make changes or add new blocks, remove old blocks, etc.

However, here is a problem, if people are making the changes independently and randomly without the presence of others, it can cause problems.

Let’s say I made a change in the doors to make them sliding vs hinged doors, and others will not know about it.

If some other friend added a block adjacent to the door, the door will not open and it will not work.

and this is what happens in Software Projects. Let’s say our Development team is working on a software application and they don’t communicate the changes made by them to the team in a proper way, these kinds of issues will come regularly

If one person makes a change in one of the modules, it may break the changes done by some other developer in the same of different module

Now let’s imagine, if we keep our Project at a common central repository and when any person wants to make any changes to the code, they will get a copy of the code from the central repository and work on it on their local systems.  so we are now not making changes directly to the main project but making a copy of it and working on our local systems

And after working, they merge the changes back to the code on the central repository.

We need a platform to keep our project/code, **GITHUB is that platform.**

we also need a tool that will help to pull, push, and commit changes which is **Git.**

GitHub essentials are:

* Repositories
* Branches
* Commits
* Pull Requests
* Git (the version control software GitHub is built on)

**Repository**

* A GitHub **repository** can be used to store a development **project**.
* It can contain **folders** and any type of **files** (HTML, CSS, JavaScript, Documents, Data, Images).
* A GitHub repository should also include a **licence** file and a **README** file about the project.
* A GitHub repository can also be used to store ideas, or any resources that you want to share.

## Branch

A GitHub branch is used to work with different **versions** of a repository at the same time.

By default a repository has a **master** branch (a production branch).

Any other branch is a **copy** of the master branch (as it was at a point in time).

New Branches are for bug fixes and feature work separate from the master branch. When changes are ready, they can be merged into the master branch. If you make changes to the master branch while working on a new branch, these updates can be pulled in.

**Commits**

At GitHub, changes are called commits.

Each commit (change) has a description explaining why a change was made.

**Pull Requests**

Pull Requests are the heart of GitHub **collaboration**.

With a pull request you are **proposing** that your changes should be **merged** (pulled in) with the master.

Pull requests show content **differences**, changes, additions, and subtractions in **colors** (green and red).

As soon as you have a commit, you can open a pull request and start a discussion, even before the code is finished.

**Some of its significant features are as follows.**

* Collaboration
* Integrated issue and bug tracking
* Graphical representation of branches
* Git repositories hosting
* Project management
* Team management
* Code hosting
* Track and assign tasks
* Conversations

**The key benefits of GitHub are as follows.**

* It is easy to contribute to open source projects via GitHub.
* It helps to create an excellent document.
* You can attract recruiter by showing off your work. If you have a profile on GitHub, you will have a higher chance of being recruited.
* It allows your work to get out there in front of the public.
* You can track changes in your code across versions.

**Important GIT commands**

* **Git add**

The git add command adds new or changed files in your working directory to the Git staging area. This gives you the opportunity to prepare a snapshot before committing it to the official history.

git add <file path>

git add .

* **Git branch**

The Git branch provides you the ability to create branches and perform risk-free development in isolated programming space.

The “branch” command helps you create, delete, and list branches.

git branch <new branch>

git branch -d <branch name>

* **Git commit**

Used to record the changes in the repository. Takes the staged snapshot and commits it to the project history. Combined with git add, this defines the basic workflow for all Git users.

git commit -m “<commit message>”

### **git pull**

It downloads a branch from a remote repository, and then immediately merges it into the current branch.

git pull <remote>

### **git push**

A git push command, when executed, pushes the changes that the user has made on the local machine to the remote repository.

Git push

Git push –set-upstream<remote branch> <branch name>

### **git merge**

A powerful way to integrate changes from divergent branches. After forking the project history with git branch, git merge lets you put it back together again.

You merge any changes into the master by clicking a "Merge pull request" button.

A screenshot of a computer

Description automatically generated

A blue and white background with text and a logo

Description automatically generated

A blue and white background with text and a logo

Description automatically generated