EXPERIMENT – 2

AIM – To understand Version Control System / Source Code Management, install git and create a GitHub account.

THEORY:

What Is Version Control?

A system called version control, sometimes referred to as source control or revision control, keeps track of changes made to a file or group of files over time so that you may retrieve particular versions at a later time. Although it can be applied to any circumstance where several versions of something are made and may need to be monitored and recalled, it is most frequently employed in software development.

What is GIT?

a. 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.
b. Git relies on the basis of distributed development of software where more than one developer may have access to the source code of a specific application and can modify changes to it that may be seen by other developers..
c. Initially designed and developed by Linus Torvalds for Linux kernel development in 2005.
d. Every git working directory is a full-fledged repository with complete history and full version tracking capabilities, independent of network access or a central server.
e. Git allows a team of people to work together, all using the same files. It helps the team cope with the confusion that tends to happen when multiple people are editing the same files.

What Is GitHub?

GitHub is an web based platform which hosts software development projects and uses Git for version management. Git is a distributed version control system that helps developers to work together on same software projects and keep track of changes made to their code by on another. GitHub offers a user-friendly interface, which is very collaborative tools, and more project management tools, GitHub will enhance the potential of the Git.

GitHub allows developers to create and manage the code in the repository in the remote location where others can aceses the code or Github is an collection repositories which contains the files of the project.

What is Use Of Version Control Software?

a. Version control software allows the user to have "versions" of a project, which show the changes that were made to the code over time, and allows the user to backtrack if necessary and undo those changes.
b. This **ability alone – of being able to compare two versions or reverse changes**, makes it fairly invaluable when working on larger projects.
c. In a version control system, **the changes would be saved just in time** – a patch file that could be applied to one version, in order to make it the same as the next version.
d. All **versions are stored on a central server**, and individual developers checkout and upload changes back to this server.

What Are the Uses Cases Of GitHub?

Following are some of the use cases of GitHub

a. **Version Control:** GitHub is also called an version control system because of it uses such has if the certain developers are working on the same project and if any developer make changes at it is effecting the entire code then they move back to previous version with immediate actions.
b. **Collaboration and Code Review:** GitHub allows group of developers work on same project where there can review the each others code and can work on the same project which will improve the productivity and where they can develop the complex application in faster manner.
c. **Issue Tracking:** Has an certain group of developers will work on the same project so when the issue arises in the GitHub then you can assign the issue to the other developer to whom you want.
d. **Open Source Development:** The most widely used platform for open source development is GitHub.

What Are Characteristics of Git?

A. Strong support for non-linear development

a. Git supports rapid branching and merging and includes specific tools for visualizing and navigating a non-linear development history.
b. A major assumption in Git is that a change will be merged more often than it is written.
c. **Branches** in Git **are very lightweight**.

B. Distributed development

a. Git **provides** each developer **a local copy** of the entire development history, and changes are copied from one such repository to another.
b. The changes can be merged in the same way as a locally developed branch very efficiently and effectively.

C. Compatibility with existing systems/protocol

a. Git has a CVS server emulation, which enables the use of existing CVS clients and IDE plugins to access Git repositories.

D. Efficient handling of large projects

a. Git is **very fast and scalable** compared to other version control systems.
b. The **fetching power from a local repository is much faster than** is possible with a **remote server**.

E. Data Assurance

a. The Git history is stored in such a way that **the ID** of a particular version **depends** upon the **complete development** history leading up to that commit.
b. Once published, it is not possible to change the old versions without them being noticed.

F. Automatic Garbage Collection

a. Git automatically performs **garbage collection** when enough loose objects have been created in the repository.
b. Garbage collection can be called explicitly using git ***gc –prune***.
c. **G.** Periodic explicit object packing
d. Git stores each newly created object as a separate file. It uses *packs* that store a large number of objects in a single file (or network byte stream) called packfile, delta-**compressed** among themselves.
e. A corresponding index file is created for each pack file, **specifying** the **offset** of each object in the packfile.
f. The process of **packing** can be very **expensive computationally.**

g. Git allows the expensive pack operation to be deferred until later when time does not matter.
h. Git **does periodic repacking automatically** but manual repacking can be done with the git gc command.

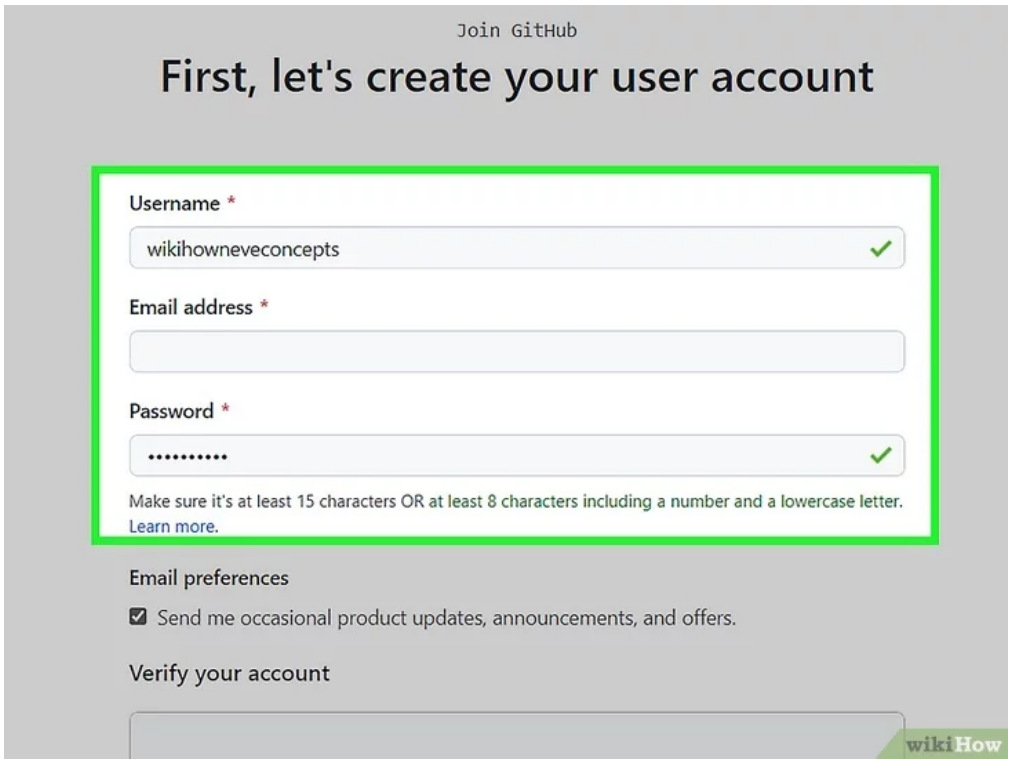
How does GIT work?

a. A Git repository is a **key-value** object store where all objects are indexed by their SHA-1 hash value.
b. All commits, files, tags, and filesystem tree nodes are different types of objects living in this repository.
c. A Git **repository is a large** [hash table](#) with **no provision** made for **hash collisions**.
d. Git specifically works by taking "**snapshots"** of files

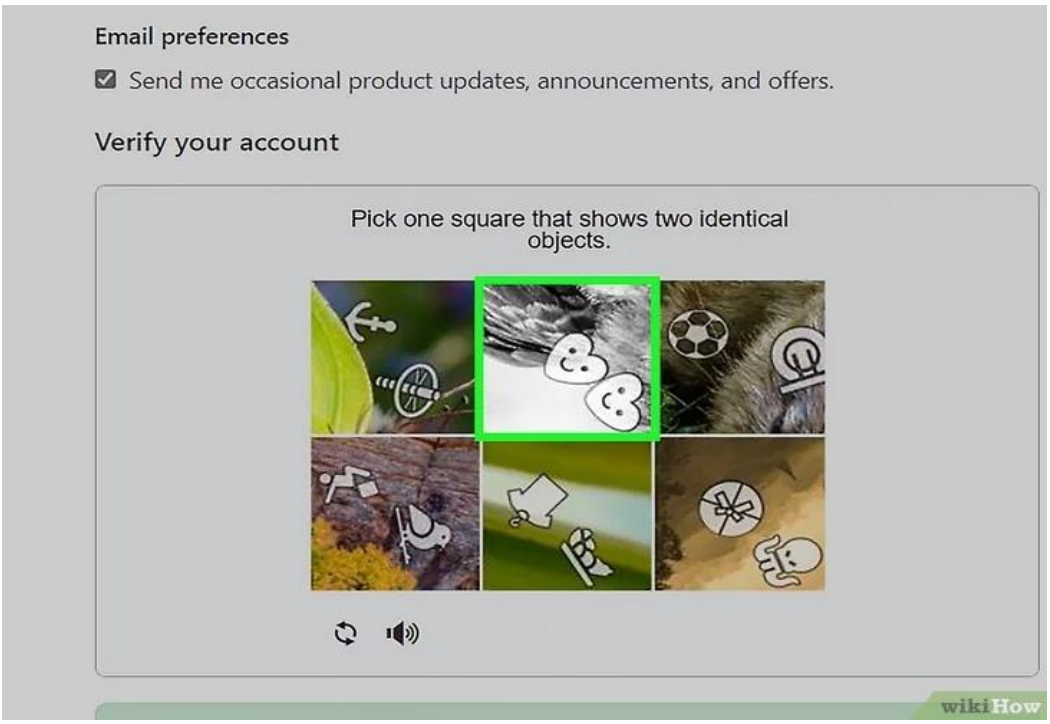
Steps to creating a Github account:

1. **Go to https://github.com/join in a web browser.**

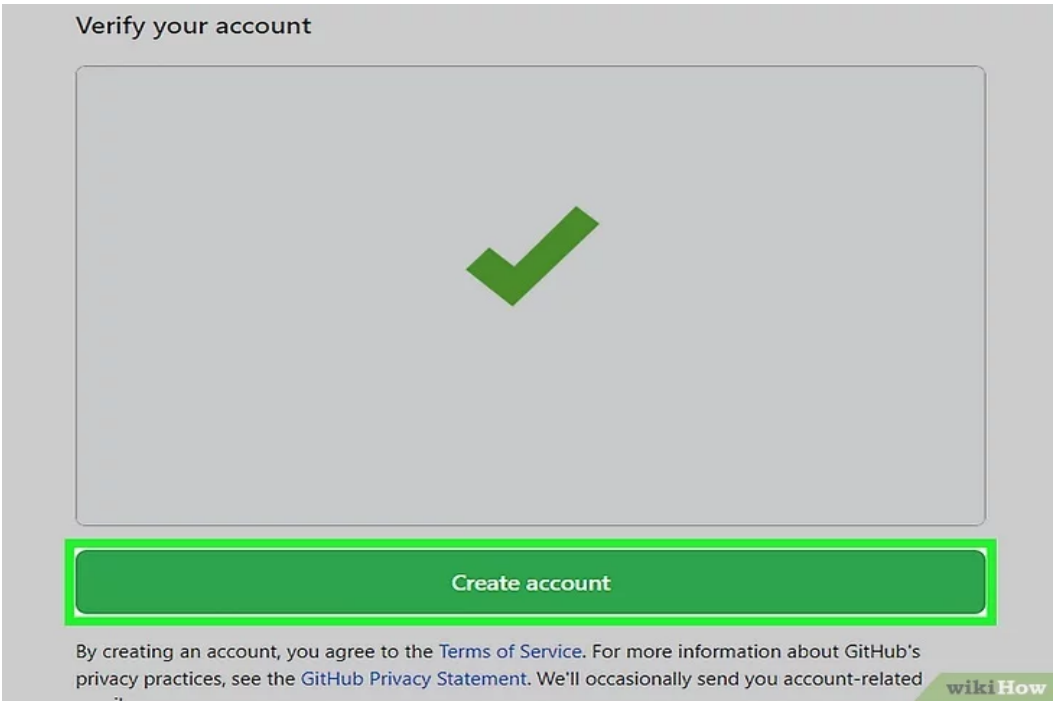
2. **Enter your personal details**



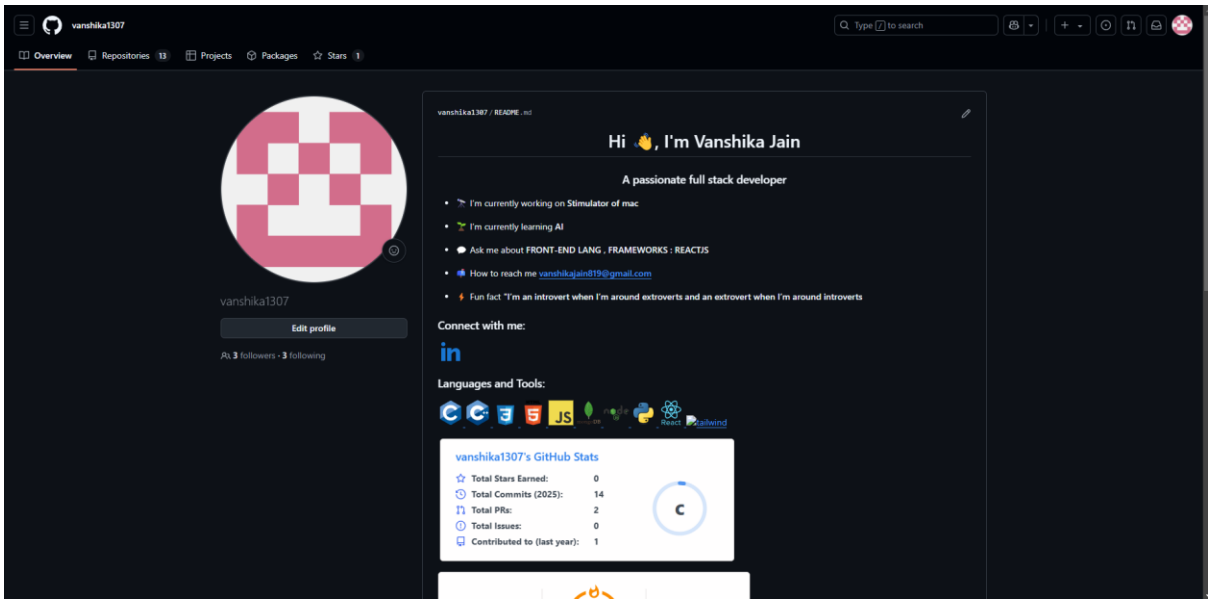
3. **Click Verify to start the verification puzzle**



4. **Click the green Create account button**



5. **Account Created**



6. **Repository Created**

CONCLUSION:

Thus, we understood Version Control System and learnt how to create a GitHub Account.