Kishan Wali
T23 123

EXPERIMENT 2

TO UNDERSTAND VERSION CONTROL SYSTEM / SOURCE CODE MANAGEMENT, INSTALL GIT AND CREATE A GITHUB ACCOUNT

THEORY:

What Is Version Control?

A version control system, also known as source control or revision control, tracks changes made to files over time, allowing users to retrieve specific versions when needed. While it can be applied in various scenarios, it is most commonly used in software development.

What is Git?

- Git is a free and open-source distributed version control system designed to handle projects of all sizes with speed and efficiency.
- It enables collaborative software development, allowing multiple developers to modify source code while tracking changes.
- Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005.
- Every Git working directory is a full-fledged repository with complete history and version tracking, independent of network access or a central server.
- Git enables teams to work on the same files simultaneously, resolving conflicts and maintaining version history.

What Is GitHub?

GitHub is a web-based platform that hosts software development projects and integrates with Git for version control. It provides a user-friendly interface and collaborative tools to enhance project management.

- It allows developers to create, manage, and access repositories remotely. GitHub is essentially a collection of repositories that store project files and enable seamless collaboration.

Use of Version Control Software

- Allows users to track different versions of a project and revert changes when necessary.

☏ Helps in comparing versions, debugging, and recovering lost data.

☏ Saves changes as patch files that can be applied to previous versions. ☏ Stores all versions on a central server, where developers can check out and update their work.

Use Cases of GitHub
1. Version Control – Allows developers to revert to previous versions when changes affect the project negatively.
2. Collaboration & Code Review – Enables teams to review each other's code, improving productivity and efficiency.
3. Issue Tracking – Developers can assign issues to team members and track progress.
4. Open-Source Development – GitHub is widely used for open-source projects, allowing public contributions.

Characteristics of Git
1. Strong Support for Non-Linear Development
   ☏ Supports rapid branching and merging.
   ☏ Changes are merged frequently.
   ☏ Lightweight branches make development faster.
2. Distributed Development
   ☏ Every developer gets a local copy of the repository.
   ☏ Changes can be merged efficiently between repositories.
3. Compatibility with Existing Systems
   ☏ Git supports CVS server emulation, allowing existing CVS clients to access Git repositories.
4. Efficient Handling of Large Projects
   ☏ Faster and more scalable than other version control systems.
   ☏ Fetching data from a local repository is quicker than from a remote server.
5. Data Assurance
   ☏ Git stores history in a way that ensures integrity—old versions cannot be changed unnoticed.
6. Automatic Garbage Collection
   ☏ Git performs automatic garbage collection when enough loose objects are created. ☏ Can be explicitly triggered using git gc --prune.
7. Periodic Explicit Object Packing
   ☏ Git stores objects as separate files, later compressed into a packfile.
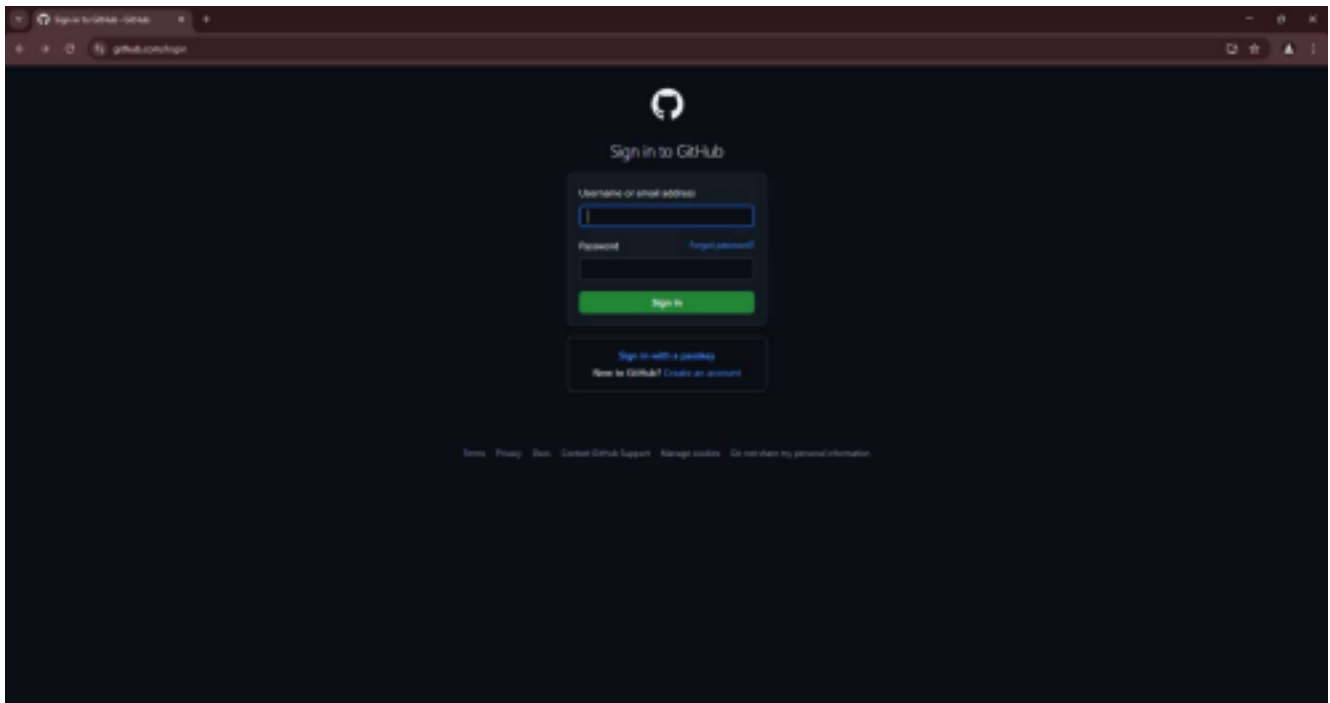   ☏ This improves performance but is computationally expensive.
How Does Git Work?
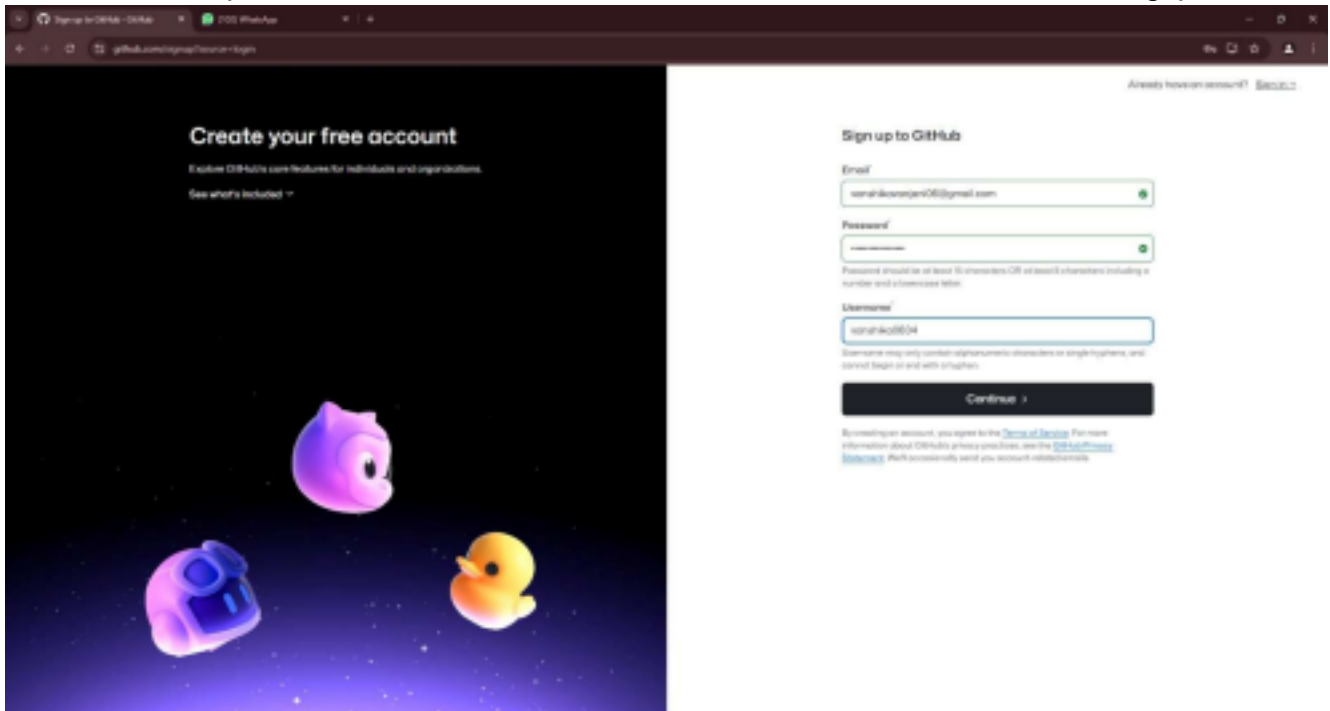   ☏ A Git repository is a key-value store where all objects are indexed by their SHA-1 hash.

📽 Objects include commits, files, tags, and tree nodes.

📽 Git works by taking snapshots of files at different points in time.

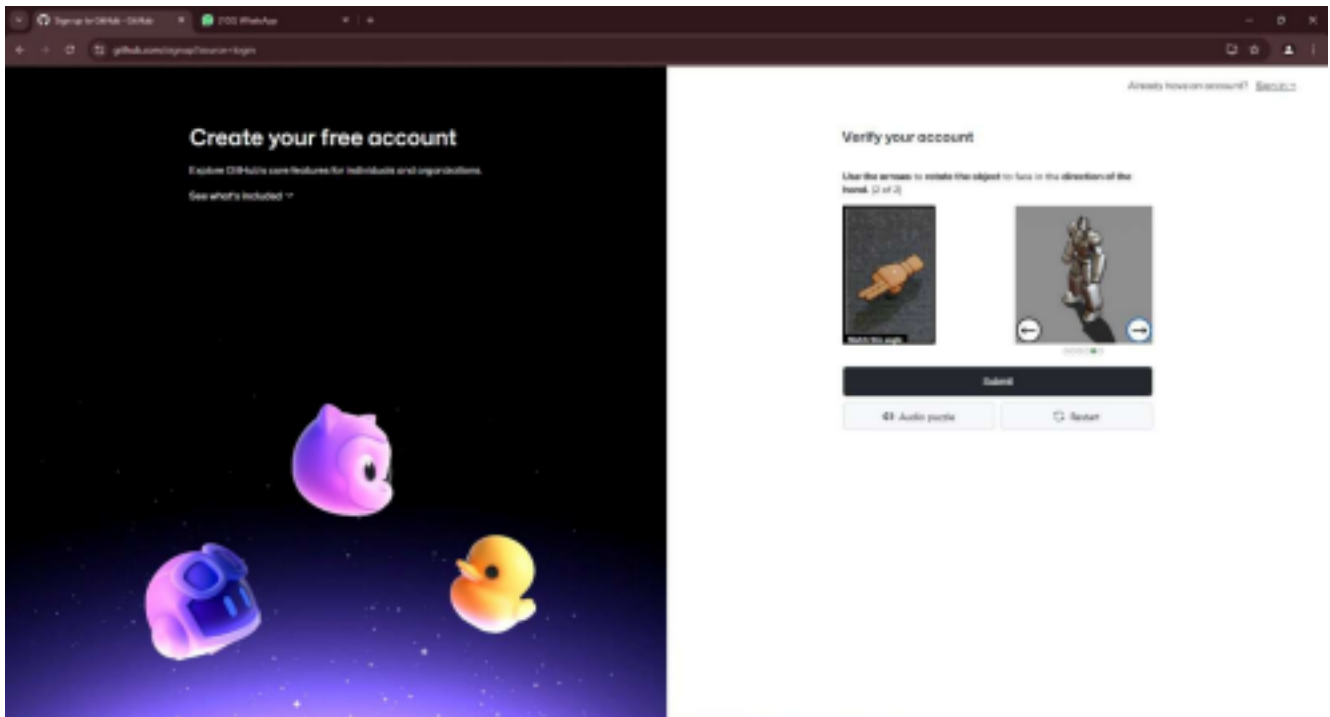Steps to Creating a GitHub Account

1. Open a web browser and go to [GitHub Signup](#).

2. Enter a unique username. Provide a valid email address. Create a strong password.

3. Click "Create account". Complete the CAPTCHA verification.

4. Choose your email preferences and click "Continue".
5. Verify your email by clicking the link sent to your registered email address.
6. Set up your profile (optional) and Select Free plan
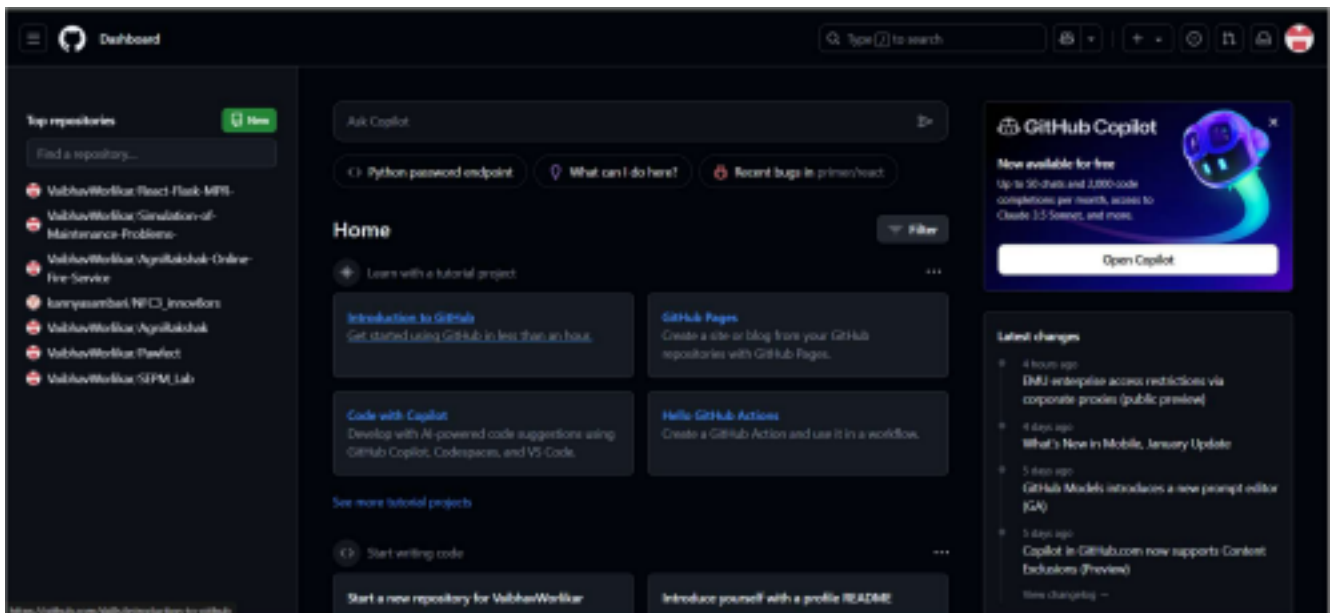
7. Your GitHub account is now ready to use!

Steps to Create a Repository on GitHub

1. Log in to GitHub:
   o Go to GitHub and sign in to your account.

2. Navigate to Repositories:
   o Click on your profile icon (top-right corner) and select "Your repositories" from the

dropdown.

    o Alternatively, click on the "+" icon in the top-right corner and select "New repository".
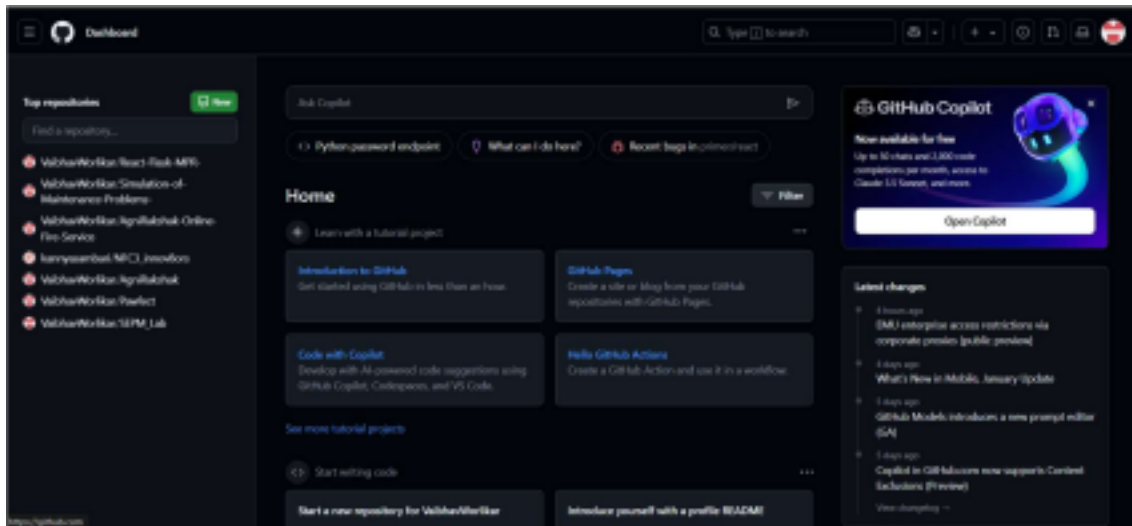

3. Create a New Repository:

    o Enter a repository name (e.g., my-project).

    o Optionally, add a description.

    o Choose the visibility:

        ⌨ Public (anyone can see your repository).

        ⌨ Private (only you and collaborators can access it).

4. Initialize the Repository (Optional):

    o You can check "Add a README file" to include a basic introduction. o Optionally, add a .gitignore file for excluding certain files from tracking. o Choose a license if needed.

5. Create the Repository:

    o Click the "Create repository" button.
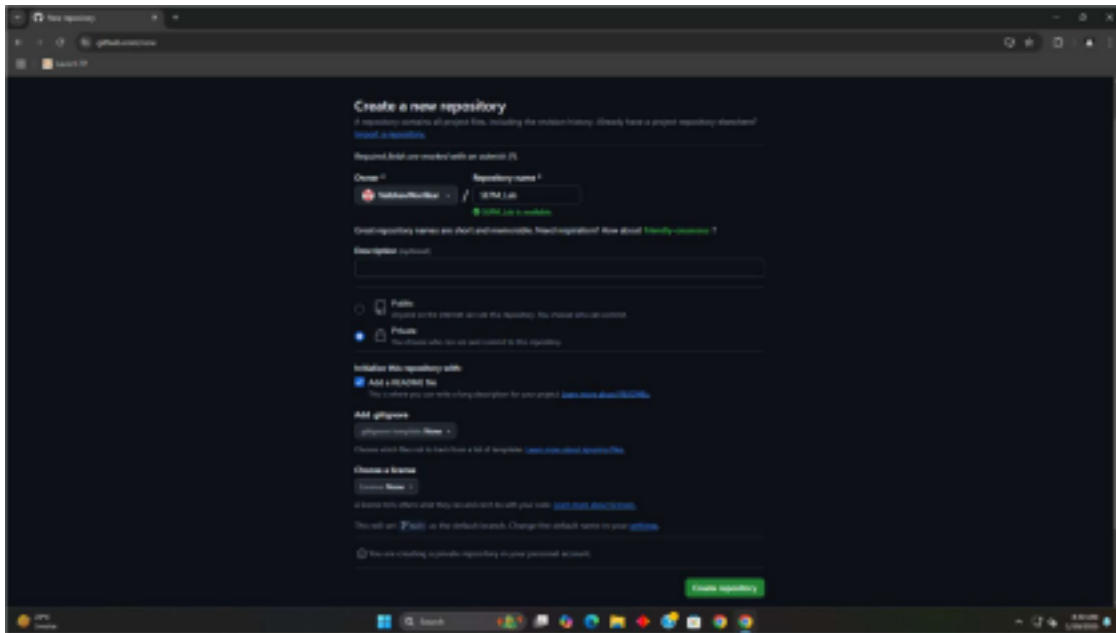
6. Set Up Locally (Optional):

   o Copy the repository URL and use Git to clone it:

   o git clone https://github.com/your-username/your-repository.git o

   Navigate to the folder and start working on your project!

Now your repository is ready to use! ��