EXPERIMENT 2

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

Theory:-

Git:

1. Introduction:
- Version Control System (VCS): Git is a distributed version control system that tracks changes in source code during software development.

2. Key Concepts:
- Repository (Repo): A collection of files and the history of changes associated with them.
- Commit: A snapshot of changes made to the code. Each commit has a unique identifier (hash).
- Branch: A separate line of development within a repository, allowing for parallel work.
- Merge: Combining changes from one branch into another.
- Pull Request: A proposed set of changes submitted for review before merging into the main branch.

3. Basic Git Workflow:
- Initialize Repository: `git init` initializes a new Git repository.
- Add Changes: `git add` stages changes for commit.
- Commit Changes: `git commit -m "message"` records changes to the repository.
- Check Status: `git status` shows the status of changes as untracked, modified, or staged.
- Branching: `git branch` lists branches, and `git checkout` or `git switch` changes the active branch.
- Merging: `git merge` combines changes from one branch into another.

4. Collaboration:
- Clone: `git clone` creates a local copy of a remote repository.
- Pull: `git pull` fetches changes from a remote repository and merges them into the current branch.
- Push: `git push` uploads local changes to a remote repository.

5. Branching Strategies:
- Feature Branching: Create a branch for each new feature, making it easier to manage changes and collaborate.
- GitFlow: A branching model that defines specific branches for features, releases, and hotfixes.
- Pull Requests (PRs): Used for proposing changes, reviewing, and discussing before merging.

6. Git Configurations:
- User Configuration: `git config --global user.name "Your Name"` and `git config --global user.email "your@email.com"` set user identity.

- Aliases: Custom shorthand commands using `git config --global alias.<alias-name> <git-command>`.

GitHub:

1. Introduction:
- Web-Based Git Repository Hosting: GitHub is a web-based platform that provides hosting for Git repositories, collaboration features, and more.

2. Repository Management:
- Create a Repository: Allows you to create a new repository on GitHub.
- Fork: Creates a personal copy of someone else's project.
- Clone or Download: Provides the repository URL for cloning.

3. Collaboration:
- Issues: Used to track tasks, enhancements, bugs, and other kinds of questions.
- Pull Requests (PRs): Propose changes, review code, and discuss changes before merging.
- Actions: Automate workflows and tasks using GitHub Actions.

4. Code Review:
- Comments and Reviews: Allows collaborators to review code, leave comments, and suggest changes.
- Approvals: Required approvals before merging a PR can be configured.

5. Security:
- Branch Protection: Prevents direct pushes to specific branches, enforcing the use of PRs.
- Security Alerts: Notifies about vulnerable dependencies.

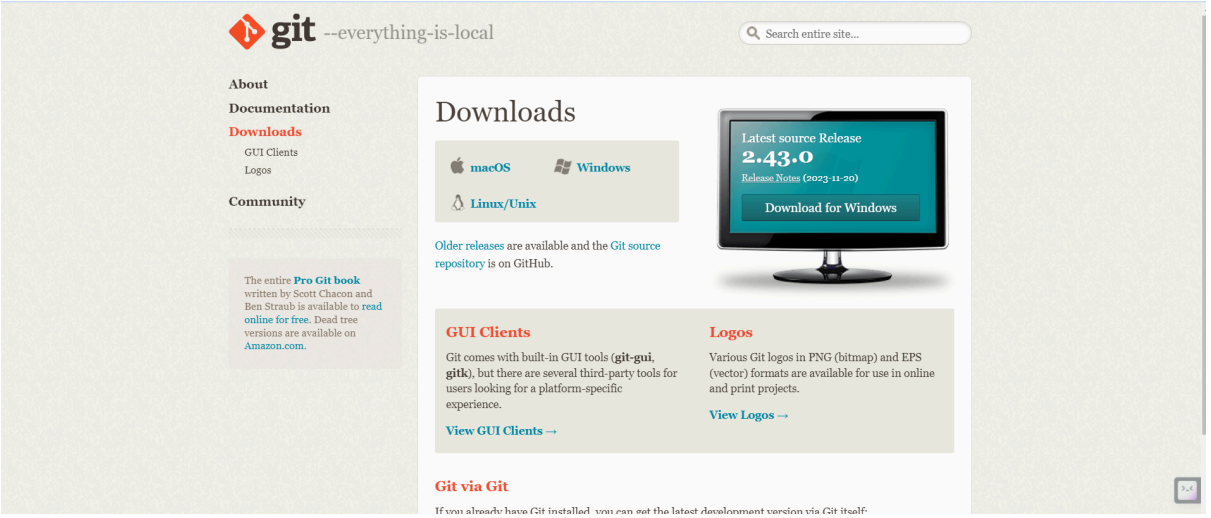
6. Integration:
- Webhooks: Enables external services to be notified about repository events.
- GitHub Pages: Allows publishing static web content directly from a repository.

7. Community and Social Features:
- Follow: Users can follow repositories and receive updates.
- Stars and Forks: Indicates popularity and allows users to contribute to projects.
- Discussions: Provides a platform for community discussions.

8. Organizations:
- Teams: Groups of collaborators with specific permissions.
- Billing: Provides paid plans for additional features and storage.

GitHub has become a central platform for collaborative software development, providing tools and features that enhance the Git workflow and support a wide range of development practices. It has played a crucial role in open-source development and has been widely adopted by individuals, teams, and organizations.

Installing Git:



Setting up a local repository:

```
ash123@sheesh:/mnt/c/Users/ASHIS/Desktop/ashish123$ git init
```

Making a file :-

```
ash123@sheesh:/mnt/c/Users/ASHIS/Desktop/ashish123$ touch ashish.txt
```

Staging files via git add:

```
ash123@sheesh:/mnt/c/Users/ASHIS/Desktop/ashish123$ git add ashish.txt
```

Checking repository status via git status:

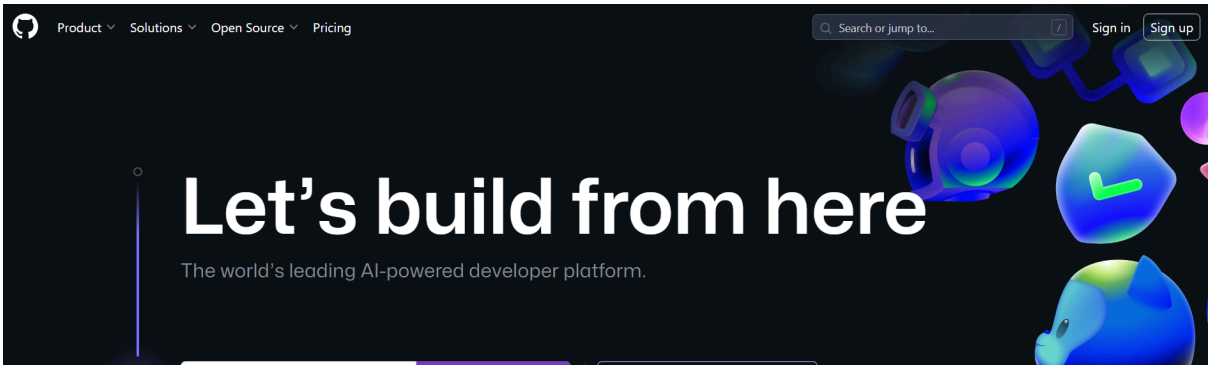
```
ash123@sheesh:/mnt/c/Users/ASHIS/Desktop/ashish123$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   ashish.txt
```

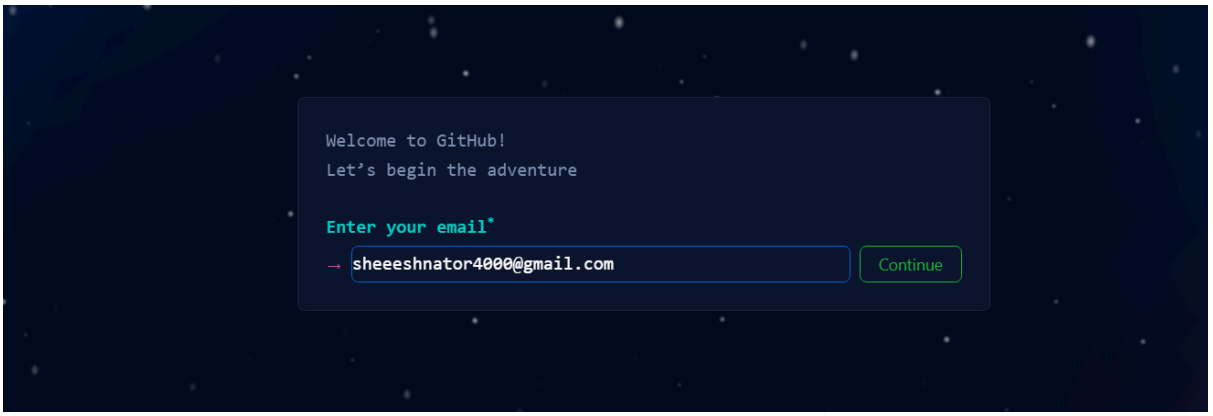
```
ash123@sheesh:/mnt/c/Users/ASHIS/Desktop/ashish123$ git commit -m 'this is the first commit'
```

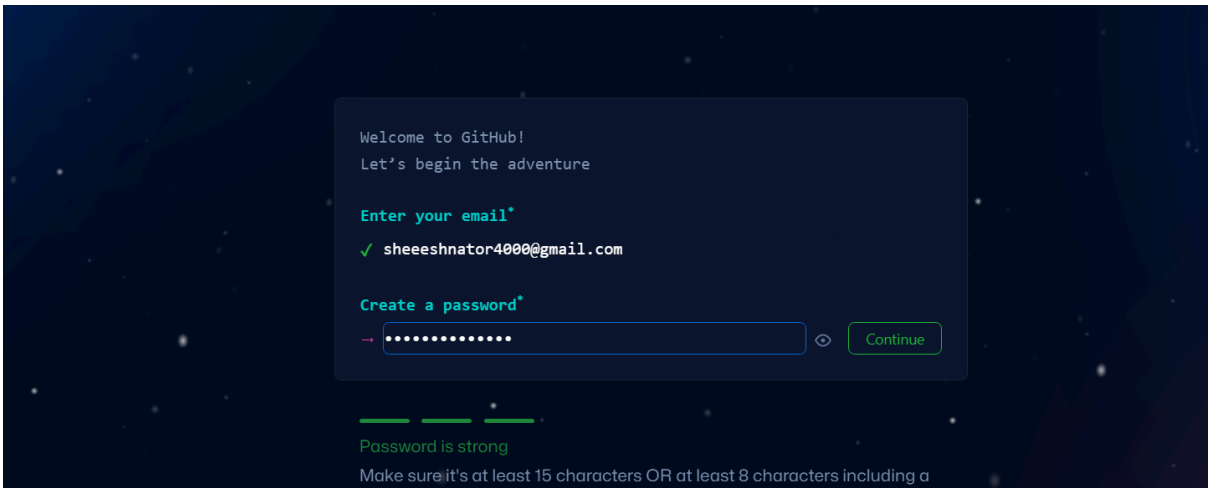
Making commits to the repository:

```
ash123@sheesh:/mnt/c/Users/ASHIS/Desktop/ashish123$ git commit -m "initial commit"
[master (root-commit) 9cfc267] initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 ashish.txt
```

Setting up a github account:-







Welcome to GitHub!
Let's begin the adventure

Enter your email*

✓ sheeeshnator4000@gmail.com

Create a password*

✓ ••••••••••••••

Enter a username*

✓ sheesh555

Email preferences

☑ Receive occasional product updates and announcements.

Continue

You're almost done!
We sent a launch code to sheeeshnator4000@gmail.com

→ Enter code*

Didn't get your email? Resend the code or update your email address.

Welcome to GitHub

We are glad you're here.

This will help us guide you to the tools that are best suited for your projects.

How many team members will be working with you?

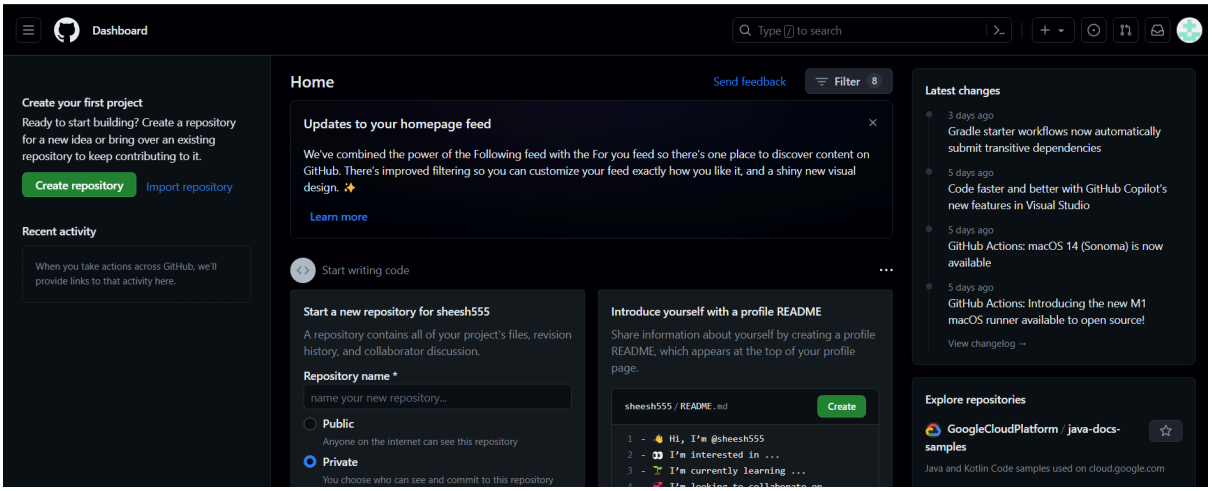
Just me | 2-5 | 5-10
10-20 | 20-50 | 50+

Are you a student or teacher?

N/A | Student | Teacher

Continue

Skip personalization



Conclusion: We have successfully installed git locally and created a github account.