CYBER SECURITY
LAB 11

Write the steps for Github installation, setup and basic local Git operations (creating repository, cloning, commit changes and uploading files and projects

Why Do We Need a Version Control System?

- A version control system (VCS) is a tool that helps manage changes to files over time.
- It keeps track of every modification made to the codebase, allows multiple people to collaborate on projects, and ensures that changes can be tracked, merged, and, if necessary, reverted.

The main reasons for having a version control system (VCS):

1. **Collaboration:** Allows multiple developers to work on the same project simultaneously without conflicts.
2. **History Tracking:** Records every change, making it easy to review, revert, and understand the evolution of the codebase.
3. **Branching and Merging:** Facilitates parallel development and safe experimentation by enabling isolated branches that can later be merged.
4. **Backup and Restore:** Ensures data integrity and provides backups that can be restored in case of data loss.
5. **Efficiency and Quality:** Supports continuous integration and deployment (CI/CD), automated testing, and project management, improving overall development efficiency and quality.

GIT AND FUNDAMENTALS OF GIT

Git is a powerful and flexible version control system (VCS). Specifically, it is a distributed version control system (DVCS) that supports efficient version tracking, branching, merging, and collaboration, making it a popular choice for software development projects.

FUNDAMENTALS OF GIT

1. **Repository (Repo)**

- 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 **license** 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.

2. **Commit**
   - A commit is a snapshot of your project at a specific point in time, including changes made to the files and a descriptive message.
   - It represents a complete version of the project that can be revisited later.
3. **Branch**
   - Branches allow you to work on different features or fixes separately from the main codebase.
   - The default branch in Git is usually called main or master.
4. **Merge**
   - Merging integrates changes from one branch into another, combining their histories.
   - It allows different lines of development to be brought together into a unified project.
5. **Clone**
   - Cloning a repository creates a complete copy of a remote repository on your local machine.
   - This allows you to work on the project locally and independently.
6. **Pull**
   - Pulling updates your local repository with changes from a remote repository. It is a combination of fetching changes from the remote and merging them into your local branch.
7. **Push**
   - Pushing sends your committed changes from your local repository to a remote repository.
   - This updates the remote repository with your latest commits.
8. **Status**
   - The git status command shows the state of the working directory andthe staging area.
   - It lets you see which changes have been staged, which haven't, and which files aren't being tracked by Git.

9. **Log**
   - The git log command displays the commit history for the repository, showing all the commits made along with their

messages and other details.
- It is useful for reviewing the project's development history.

10. Revert
- Reverting creates a new commit that undoes the changes made by a previous commit, without removing it from the history.
- This is useful for safely undoing changes in a way that is easy to track and understand.

What is Github ?

- It is a Platform that allows developers to create, store, manage, and share their code.

What is Version Conflict ?

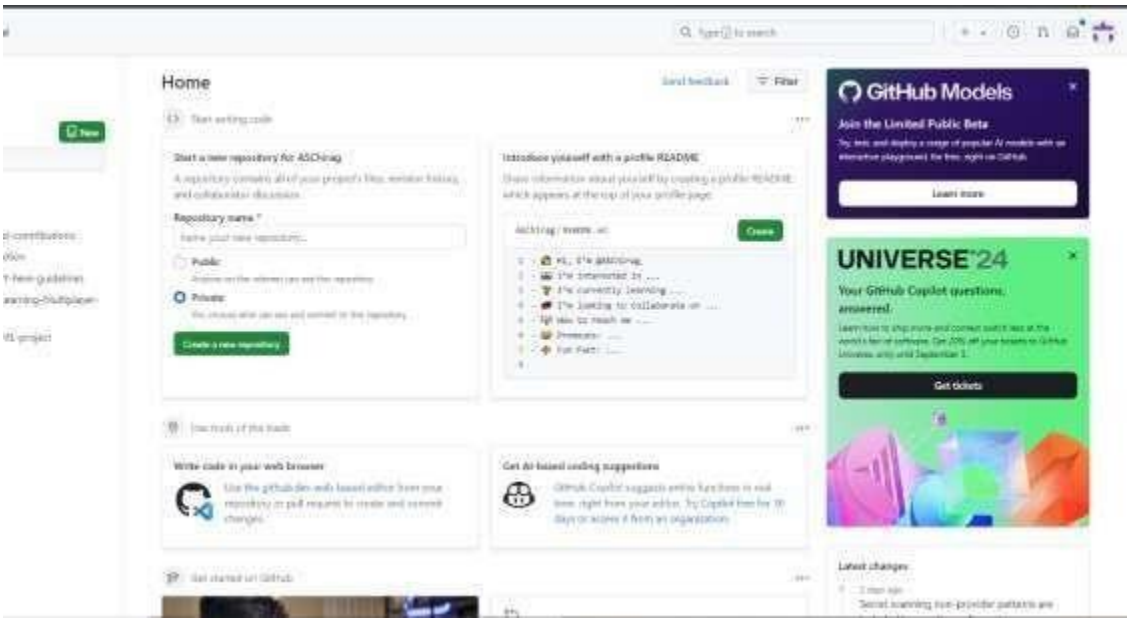
- Version control, also known as source control, is the practice of tracking and managing changes to software code. Version control systems are software tools that help software teams manage changes to source code over time.
- Version Control System that enables software development teams to have multiple local copies of the project's codebase in independent desktops.

What is Merge Conflict ?

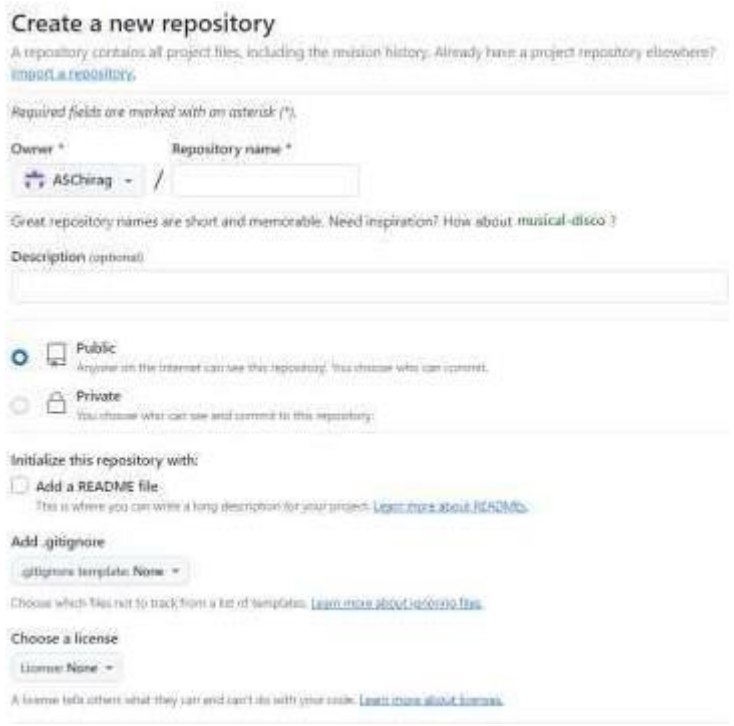
- A merge conflict happens when Git is unable to automatically reconcile differences in code between two commits. This typically occurs during a merge operation, where changes from different branches are combined.

Using Github to store, manage and update code individually on your localdesktop –

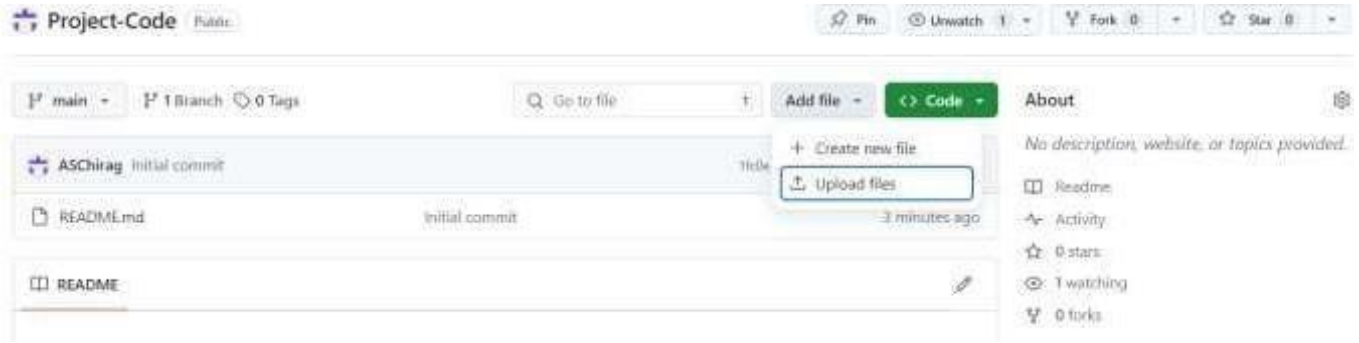
Step – 1 :– Open the Github website, After you create an account in Github or Signin to your account in Github. This dashboard will open.



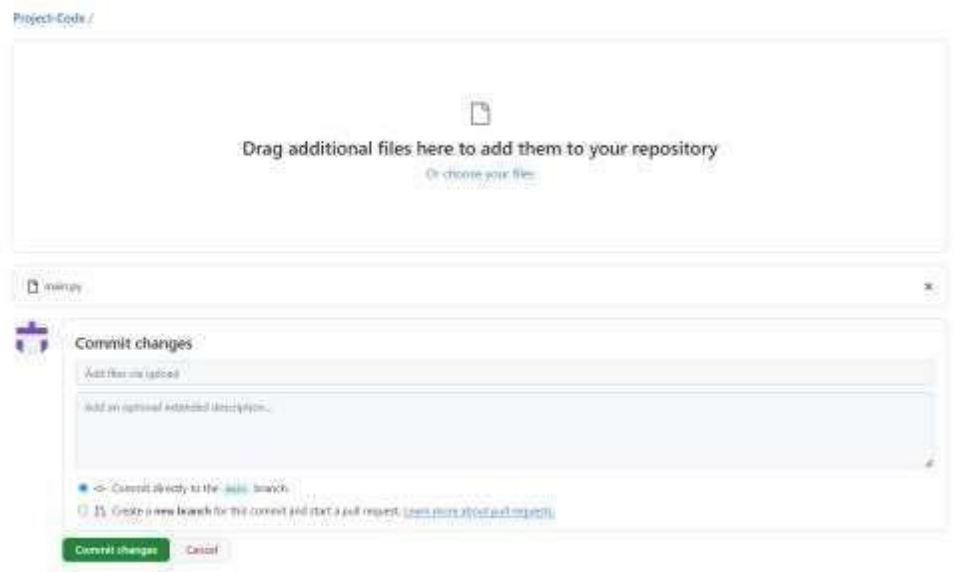
Step – 2 :- Click on "Create Repositry" button, and the create repository page will open up.



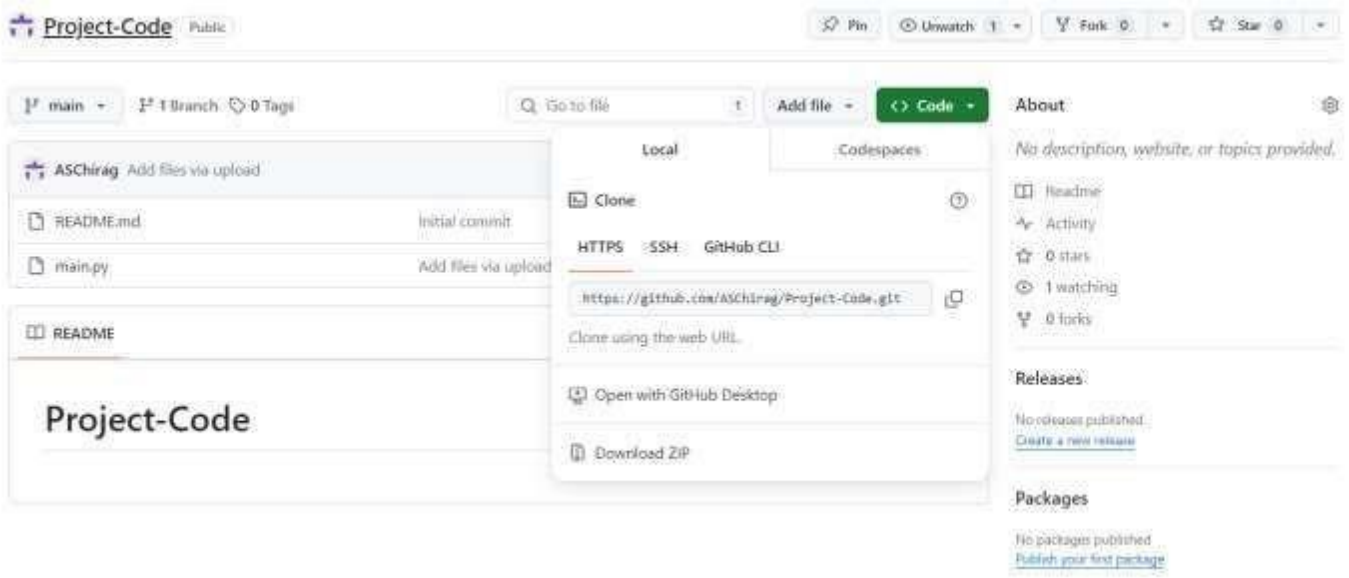
Step – 3 :- After Creating the repository, click on the "Add file" button.



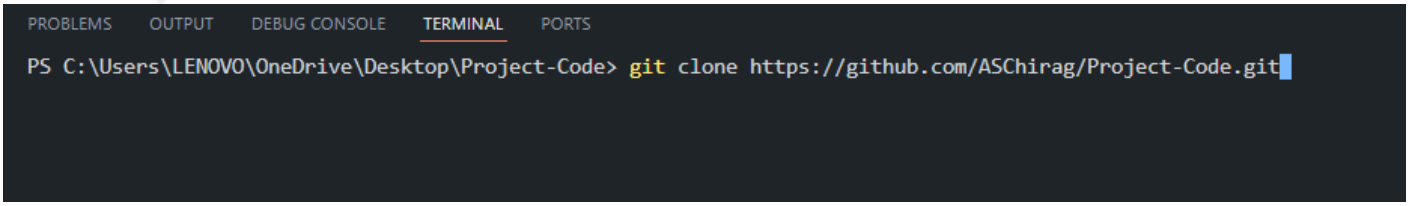
Step – 4:- The page to add files and folders will open up. Here you can drag and drop the files and folders or add them by choosing the respective files from your desktop. After you add the necessary files and folders, click on "Commit Changes".



Step – 5:- To make changes to the code file or work with the code file. Click on "Code" and copy the HTTPS Link.



Step – 6:- Open Visual Studio Code, and then open a new terminal, in the new terminal type "git clone [paste the link]" and press "Enter".



Step – 7:- After cloning the repository, a .git folder will be downloaded which will consist of all your files and folders saved in the repository. In the terminal, type "cd [repository name or folder name]".

```
PS C:\Users\LENOVO\OneDrive\Desktop\Project-Code> cd Project-Code
```

Step – 8:- Then make the necessary changes in the file, type "git add ." to addchanges of all the files. Or you can also do "git add [filename]".

```
PS C:\Users\LENOVO\OneDrive\Desktop\Project-Code\Project-Code> git add .
PS C:\Users\LENOVO\OneDrive\Desktop\Project-Code\Project-Code>
```

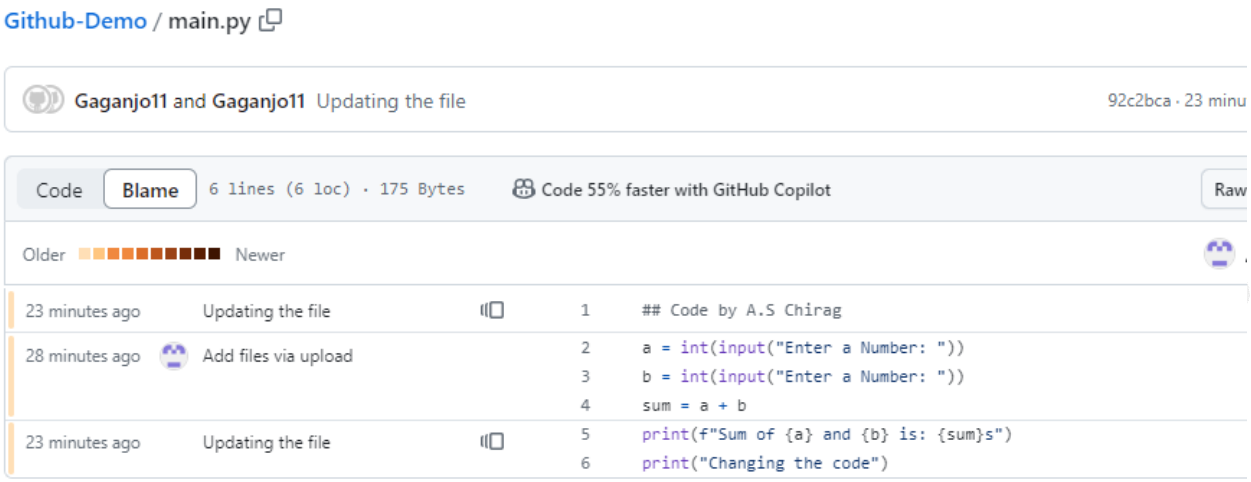
Step – 9:- After adding the changes, commit the changes, that is to confirm the changes with a message. You can do that by typing "git commit -m [message]".

```
PS C:\Users\LENOVO\OneDrive\Desktop\Project-Code\Project-Code> git commit -m "Making a Change"
```

Step-10 :- After confirming the changes, to finally make the changes, type "gitpush" in the terminal and click "Enter".

```
PS C:\Users\LENOVO\OneDrive\Desktop\Project-Code\Project-Code> git push
```

Step – 11:- After you push your changes, the changes made will be displayed in your Github repository online.



How Git works

Here is a basic overview of how Git works:

1. Create a "repository" (project) with a git hosting tool
2. Copy (or clone) the repository to your local machine
3. Add a file to your local repo and "commit" (save) the changes
4. "Push" your changes to your main branch
5. Make a change to your file with a git hosting tool and commit
6. "Pull" the changes to your local machine
7. Create a "branch" (version), make a change, commit the change
8. Open a "pull request" (propose changes to the main branch)
9. "Merge" your branch to the main branch

DOWNLOAD LINKS:

Visual Studio - https://code.visualstudio.com/

Git - https://git-scm.com/downloads

Git Hub : https://desktop.github.com/download/