

## ASSIGNMENT 1

Name : Gunjan Kadu

Matriculation Number : 11012377

### TASK 1 - GIT CLI ASSIGNMENT

1. An SSH key is a way to identify yourself rather than typing your username and password for each time you commit to a repository. It works on Secure Shell network protocol which is an authenticated and encrypted secure network protocol used for remote communications between machines on an unsecured open network.

SSH keys generally come in pairs of public key that get shared with services like version controlling system like Git Hub and a private key that is only stored on your computer.

To Generate a SSH Key the following steps need to be followed.

#### Generate Your SSH Key (1)-

To generate an SSH key use the following command.

***ssh-keygen -t rsa -b 4096 -C "your mail address"***

After this we get the address of the key stored on the local machine.

#### Use the Key (2)-

After the key is generated we need to use it by starting the ssh agent by the command

***eval \$(ssh-agent -s)***

Now we need to add the key generated to by using the command.

***ssh-add ~/.ssh/id\_rsa***

#### Add the Ssh Key on Github (3)-

Go to your GitHub account and in the menu click SSH and GPG keys. Click New SSH key and give it a title. Then paste the key into the box or drag the file directly from your local machine to the box area.

```
Last login: Sun Feb 16 23:18:31 on ttys001

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Gunjans-MacBook-Air:~ gunjankadu$ ssh-keygen -t rsa -b 4096 -C "agunjan.kadu@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/gunjankadu/.ssh/id_rsa):
Created directory '/Users/gunjankadu/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /Users/gunjankadu/.ssh/id_rsa.
Your public key has been saved in /Users/gunjankadu/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:Qm0CqQ01jwVLNRRQnnbv4oCPUuTo03qVKVndjB7L5bA agunjan.kadu@gmail.com
The key's randomart image is:
+----[RSA 4096]-----+
|  =*O.                |
|  . X =               |
|  + %..+              |
|  .B.O=..+            |
|  +.oo=SB.            |
|  . =+.E..            |
|  . o.O. . .          |
|  + oo O .            |
|  . =. . .            |
+----[SHA256]-----+
Gunjans-MacBook-Air:~ gunjankadu$
```

### RESULTS OF STEP 1 & 2

- Personal settings
- Profile
- Account
- Security
- Security log
- Notifications
- Linking
- SSH and GPG keys
- Connected users
- Repositories
- Organizations
- Unread replies
- Applications
- Developer settings

## SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

### RESULT OF STEP 3 -1

- Personal settings
- Profile
- Account
- Security
- Security log
- Notifications
- Linking
- SSH and GPG keys
- Connected users
- Repositories
- Organizations
- Unread replies
- Applications
- Developer settings

## SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



SSH

Home - Mac

a4:f1:17:d2:d6:a5:eb:88:91:3a:89:c0:ea:d2:d9:03

Added on 19 Feb 2020

Never used — Read/write

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

## GPG keys

New GPG key

There are no GPG keys associated with your account.

Learn how to [generate a GPG key and add it to your account](#).

### RESULT OF STEP 3 -2

## 2. To create a local repository we need to install git in our system.

- To create a local repository navigate to folder using CLI and then type the command
  - ***git init***

```
The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT208050.
Gunjans-MacBook-Air:GIT TEST gunjankadu$ git init
Initialized empty Git repository in /Users/gunjankadu/Desktop/Study/Coding/GIT TEST/.git/
Gunjans-MacBook-Air:GIT TEST gunjankadu$
```

### STEP 1

- Create some files in the folder and we can see that the git has started tracking the files in the folder.
- To check the status of the folder type the command
  - ***git -status***

```
The default interactive shell is now zsh.
To update your account to use zsh, please run 'chsh -s /bin/zsh'.
For more details, please visit https://support.apple.com/kb/HT208050.
Gunjans-MacBook-Air:GIT TEST gunjankadu$ git init
Initialized empty Git repository in /Users/gunjankadu/Desktop/Study/Coding/GIT TEST/.git/
Gunjans-MacBook-Air:GIT TEST gunjankadu$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    test.js
    test.ts

nothing added to commit but untracked files present (use "git add" to track)
Gunjans-MacBook-Air:GIT TEST gunjankadu$
```

### STEP - 2

- Now we need to add the files to the staging area which is done by the following command
    - ***git add test.js && git add test.ts***
- Or**                      ***git add .***   - To add all the files in the workspace.
- After adding the files we need to commit the files which is done by using the command.
    - ***git commit -m "Initial Commit"***

- To push the local folder to the remote git server we need to create a repository on the git and follow the following commands.
  - **git remote add origin git@github.com:GunjanKadu/Test.git**
  - **git push -u origin master**

```
Gunjans-MacBook-Air:Test gunjankadu$ git add index.html
Gunjans-MacBook-Air:Test gunjankadu$ git commit -m "Change in index"
[master 0455733] Change in index
Committer: Gunjan Kadu <gunjankadu@Gunjans-MacBook-Air.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 2 insertions(+), 2 deletions(-)
Gunjans-MacBook-Air:Test gunjankadu$ git push
Enter passphrase for key '/Users/gunjankadu/.ssh/id_rsa':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 317 bytes | 158.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:GunjanKadu/Test.git
8bf58e6..0455733 master -> master
```

Then we can see that our local files and the git is now available on the git server.

The screenshot shows the GitHub interface for a repository named 'GunjanKadu / Test'. The repository is private and has 1 commit, 1 branch, 0 packages, 0 releases, and 0 contributors. The 'Code' tab is selected, showing a list of files: .vscode, index.html, styles.css, test.js, test.ts, test1.js, and test1.ts. The 'index.html' file is highlighted, showing its commit history. The 'Initial Commit' is listed for all files, with the commit message 'Change in index' for 'index.html'. The 'Latest commit' is 0455733, committed 4 minutes ago. At the bottom, there is a button to 'Add a README'.

**OUTPUT**

- We now will create a new branch in our repository and edit our code there. For that first we used
  - ***git branch test-branch***
- No we will checkout on this branch using
  - ***git checkout test-branch***

```
Gunjans-MacBook-Air:Test gunjankadu$ git branch
* master
  test-branch
Gunjans-MacBook-Air:Test gunjankadu$ git checkout test-branch
Switched to branch 'test-branch'
Gunjans-MacBook-Air:Test gunjankadu$ git branch
  master
* test-branch
Gunjans-MacBook-Air:Test gunjankadu$
```

#### OUTPUT

```
Gunjans-MacBook-Air:Test gunjankadu$ git branch test-branch1
Gunjans-MacBook-Air:Test gunjankadu$ git branch
  master
* test-branch
  test-branch1
Gunjans-MacBook-Air:Test gunjankadu$
```

#### CREATE OTHER BRANCH

- On test branch now we edit a file and commit it to git. And then checkout to master and merge the branch using.
  - ***git checkout master***
  - ***git merge test-branch***
- Now we checkout to test-branch1 and make edits in the same file and on the same line and checkout to master and merge the branch
  - ***git checkout master***
  - ***git merge test-branch1***
- Here we get a merge conflicts. Which means that if 2 developers are working on 2 different branch and they edit the same code then a merge conflict could occur.
- Conflicts generally arise when two people have changed the same lines in a file, or if one developer deleted a file while another developer was modifying it. In these cases, Git cannot automatically determine what is correct.
- Conflicts only affect the developer conducting the merge, the rest of the team is unaware of the conflict. Git will mark the file as being conflicted and halt the merging process. It is then the developers' responsibility to resolve the conflict.

```

<button
  className='btn btn-success'
  Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
  onMouseOver={this.handleHoverChange}
  onMouseOver={this.handleHoverOnChange}
>>>>>>> test-branch1 (Incoming Change)
  Hover Me
</button>
</div>
);
}
}

```

## OUTPUT 1

```
Gunjans-MacBook-Air:Test gunjankadu$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Gunjans-MacBook-Air:Test gunjankadu$ git merge test-branch1
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

## OUTPUT 2

- To Resolve this merge conflict we will accept the incoming changes as we had deliberately made those changes and resolve that conflict.
- After solving the merge conflicts we can now push the changed file to the master branch.

GunjanKadu / Test
 

Private

Unwatch

1

Star

0

Fork

0

[Code](#)
[Issues 0](#)
[Pull requests 0](#)
[Actions](#)
[Projects 0](#)
[Wiki](#)
[Security](#)
[Insights](#)
[Settings](#)

Branch: master

Commits on Feb 24, 2020

Merge branch 'test-branch1'

Gunjan Kadu authored and Gunjan Kadu committed 29 seconds ago

efb6a2

test-branch1 commit

Gunjan Kadu authored and Gunjan Kadu committed 15 minutes ago

d156a3e

test-branch commit

Gunjan Kadu authored and Gunjan Kadu committed 20 minutes ago

992505a

Change in index

Gunjan Kadu authored and Gunjan Kadu committed 1 hour ago

0455733

Initial Commit

Gunjan Kadu authored and Gunjan Kadu committed 1 hour ago

8bf58e6

## OUTPUT

