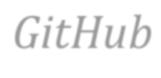
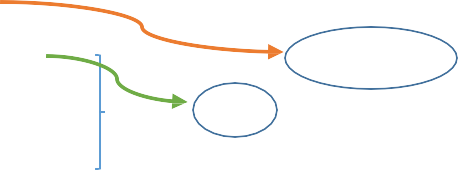
*GitHub*



# SMT (Source code management tool)

SVN

GitHub Intranet

Bitbucket Git Gitlab

* GitHub:
* It is a source code management tool which is available on internet.
  + Steps to create an account on [https://github.com](https://github.com/)
    1. Go to below url: [https://github.com](https://github.com/)
    2. Click on *sign up*.
    3. Provide *username*.
    4. Input valid *email-id*.
    5. Enter *password*.
    6. Verify account.
    7. Click on *create an account*.
    8. Select *free* and click on continue.
    9. Click on *skip this step*.

Automatically a verification mail is sent to Email-id.

* + - * Login to your email account and click on verify link.

# Steps to create Repository (shared folder)

1. Goto [https://github.com](https://github.com/)
2. Click on ‘+’ symbol on right hand side corner and select new repository.
3. Provide *repository name*.
4. Select *Public.*
5. Check **Initialize this repository with a README**
6. Click on *Create repository.*

Automatically repository is created and it is accessible using standard url: Example:

https://github.com/GAURAV1107/java.git

Client scm GitHub server internet Gitbash

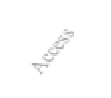
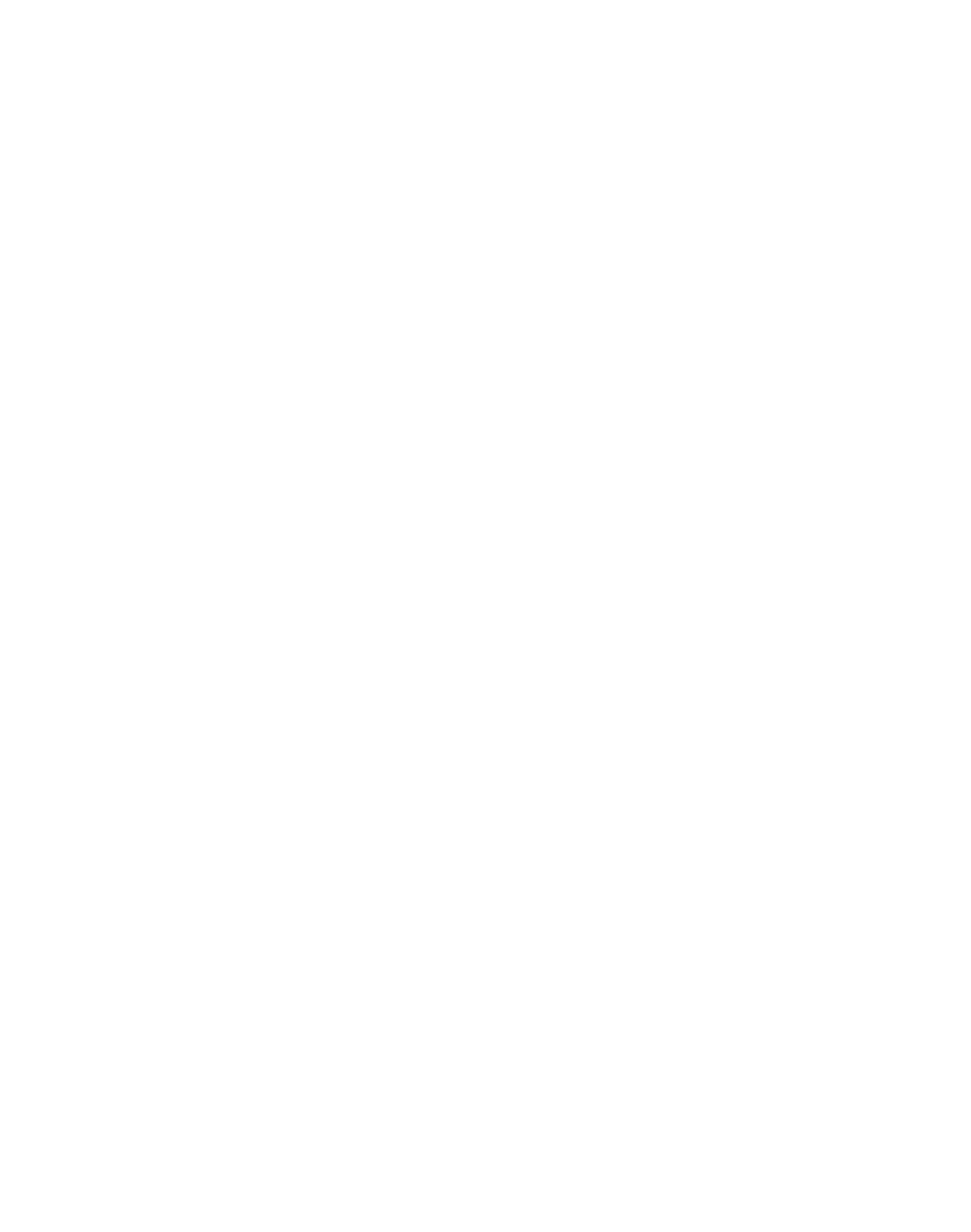
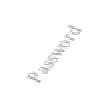
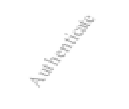
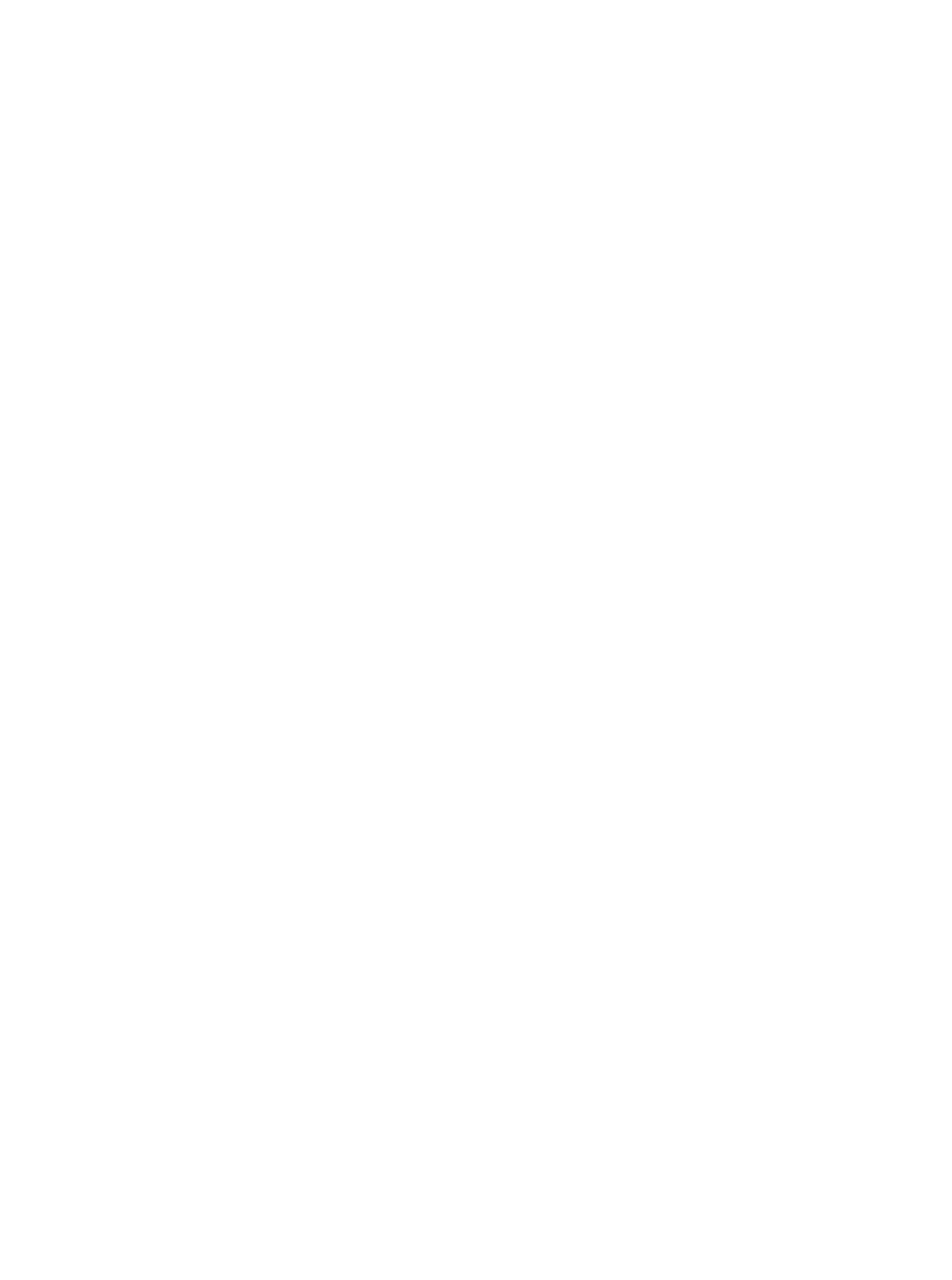
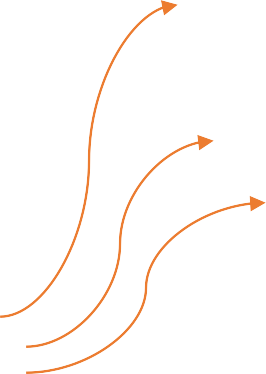
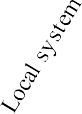
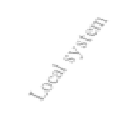
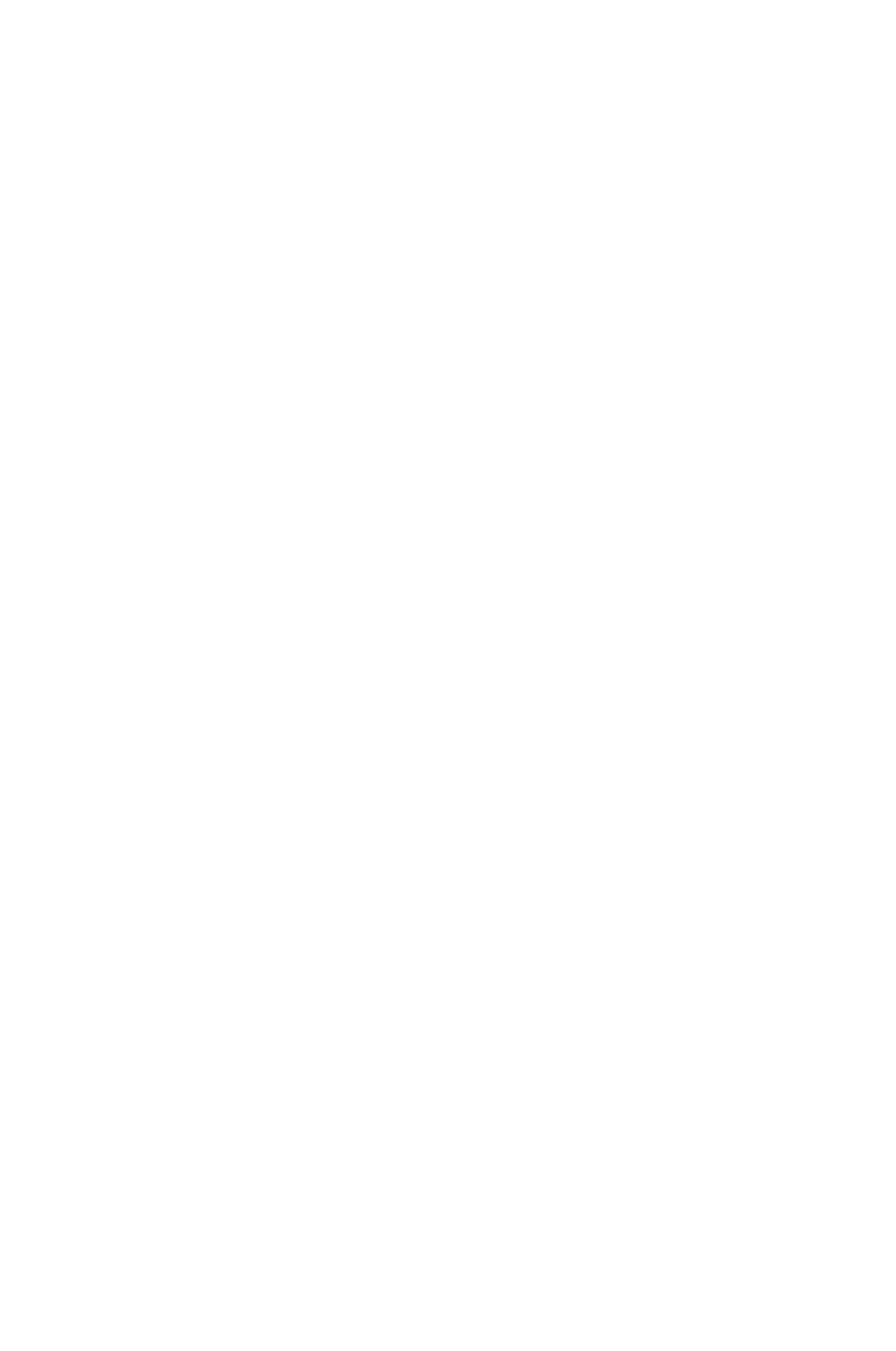
Install Account

This PC

Repository

Maintained by

IT Team



# Configuring local system to access GitHub repository:

Steps:

1. Go to url: [https://gitscm.com](https://gitscm.com/)
2. Click on ‘download for windows’.

It will start downloading an exe file automatically.

1. After successfuldownload, Install the .exe file.

**Note:**

* + To verify installation, go to any directory and perform right click.
  + If ‘Git Bash Here’ option is available, then the installation is successful.

# Steps to configure Git Bash:

1. Go to any directory, right click and choose ‘Git Bash Here’
2. Run following command:
   1. git config –global user.name (username of github account).
   2. git config --global user.email (email id of github account)

*Example:*

git config --global user.name GAURAV1107

git config -- global user.email [gauravsingh1107@gmail.com](mailto:%20gauravsingh1107@gmail.com)

# Steps to download repository to local system:

1. Go to [https://github.com](https://github.com/) and perform sign in.
2. Choose a repository link and click on it.
3. Click on ‘Clone or Download’.
4. Copy repository url.

*Example:*

[*https://github.com/GAURAV1107/java.git*](mailto:https://github.com/GAURAV1107/java.git)

1. Go to any directory, right click and choose ‘Git Bash Here’
2. Run below command to clone repository to local system.

*Example:*

git clone https://github.com/GAURAV1107/java.git

command path url

* + Steps to manage source code:

1. Create script
2. Get latest changes
3. Create a new branch in git bash
4. Add content (file)
5. Commit changes
6. Push changes from local to github server
7. Create PR (Pull request)
8. Ask for Review
9. Merge PR (Pull request)
10. Create Script:
    * Go to downloaded folder (test-sample)-Repository folder and create script using java or python.

test-sample repository folder

*javademofile.java pythondemofile.py*

java

python

file added

1. Get latest changes:
   * Go to repository directory (which contains folder) and right click, select ‘Git Bash Here’ and get the latest changes.

(In a Git Bash, It will highlight a branch)

* + To get latest changes run below *command*:

1. git checkout master
2. git pull

Note:

* *git checkout* will change current branch to master branch.
* *git pull* will fetch the latest from git hub directory.

1. Create a new branch in Git Bash:

*command-*

*Example:*

git checkout -b <branch name>

Note:

git checkout -b dev/java-python

* The above command will create a new branch and it will switch to new branch.

1. Add content (File):
   * In a git bash add changes by executing the following command-

*command-*

*Example:*

git add –A or

For adding single or multiple files use below command-

git add <File Path>

git add java/javademofile.java python/pythondemofile.py

Note:

* Separate files path with space as shown in above example.

1. Commit changes:
   * In a git bash commit the added changes.

*command*-

git commit -m “<commit message>”

*Example:*

git commit -m “Added java/python files”

1. Push changes from local to github server:
   * In a git bash push the local changes to github repository.

*command*-

git push origin <current\_branch\_name>

*Example:*

git push origin dev/java-python

1. Create PR (Pull request):
   * Steps:
2. Go to [https://github.com](https://github.com/) and perform sign in.
3. Choose repository link and click on it.
4. Click on new pull request.
5. Select ‘base branch’ as ‘Master’.
6. Select compare branch as newly created branch (dev/java- python)
7. Click on create pull request.

After creating a pull request the unique no is created for pull request and it is directly available using url.

*Example:*

[https://github.com/GAURAV1107/java.git/pull/1](https://github.com/Retheshks/test-sample/pull/1)

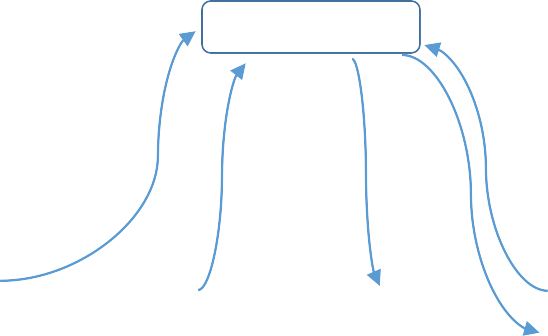
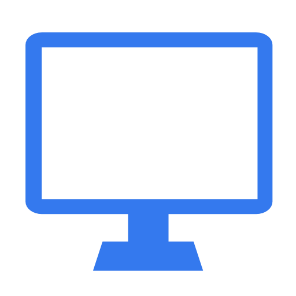
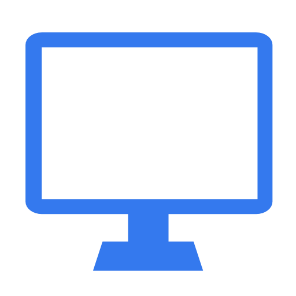
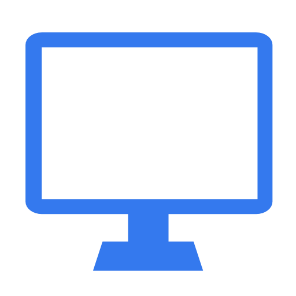
1. Ask for review:
   * After creating a pull request, share pull request no. with team member and ask for review.
2. Merge PR (Pull request):
   * After review if there are no comments, merge the pull request.

# Steps to merge pull request: -

1. Go to [https://github.com](https://github.com/) and perform sign in.
2. Select repository link and click on it.
3. Click on ‘Pull request’ tab.
4. Choose a pull request and click on it.
5. Click on ‘merge pull request’.
6. Click on ‘confirm merge’.

Automatically the content is committed to master branch.

# When we get merge conflicts:



test-sample

System 1

Engg. Python

System 2

Engg. Java

System 3

Engg. Python/Java

add

git pull

git pull

pythondemofile.py

* When engineer 1 updates a file and merged the changes to master branch.
* If engineer 3 updates the same file without fetching latest changes (without git pull).
* When engineer 3 creates a pull request, automatically it will

*highlight conflict.*

# How to resolve merge conflicts?

* Steps to resolve merge conflicts using command-

1. git checkout master
2. git pull
3. git checkout <current branch> (fix/python-1)
4. git merge master

* git merge master will list conflict files.

*Example:*

print(“hi”)

print(“hi welcome”)

<<<<<<<HEAD

print(“Java selenium”)

=======

print(“Python”) print(“Selenium”)

>>>>>>>MASTER

* HEAD- represents current changes.
* MASTER- represents existing changes.
* User has to decide to retain HEAD or MASTER or Both content and follow below:

*command*-

git add <Modified file path>

git commit -m “<commit message>” git push origin <current\_branch>