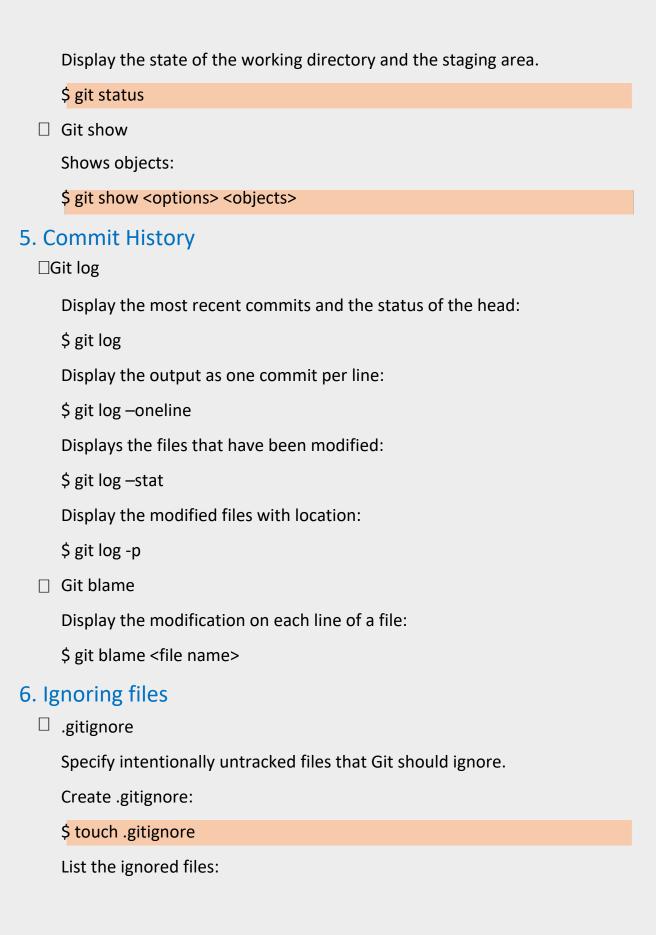
Git Cheat Sheet

. Git configuration		
	Git config	
	Get and set configuration variables that control all facets of how Git looks and operates.	
	Set the name:	
	\$ git configglobal user.name "User name"	
	Set the email:	
	\$ git configglobal user.email "himanshudubey481@gmail.com"	
	Set the default editor:	
	\$ git configglobal core.editor Vim	
	Check the setting:	
	\$ git config -list	
	Git alias	
	Set up an alias for each command:	
	\$ git configglobal alias.co checkout	
	\$ git configglobal alias.br branch	
	\$ git configglobal alias.ci commit	
	\$ git configglobal alias.st status	
. S	tarting a project	
	Git init	
	Create a local repository	
	\$ git init <repo name=""></repo>	
П	Git clana	

\$ git clone < remote Url> 3. Local changes ☐Git add Add a file to staging (Index) area \$ git add Filename Add all files of a repo to staging (Index) area \$ git add* ☐ Git commit Record or snapshots the file permanently in the version history with a message \$ git commit -m " Commit Message" 4. Track changes □Git diff Track the changes that have not been staged: \$ git diff Track the changes that have staged but not committed: \$ git diff --staged Track the changes after committing a file: \$ git diff HEAD Track the changes between two commits: \$ git diff <commit1-sha> <commit2-sha> Git Diff Branches: \$ git diff <branch 1> < branch 2>

Make a local copy of the server repository.

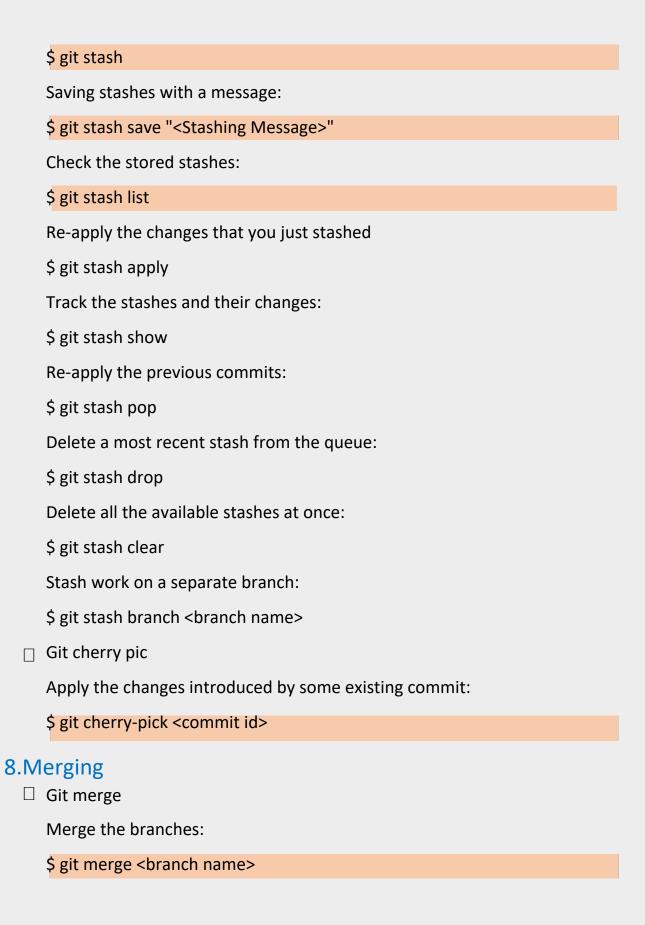
☐ Git status



\$ git Is-files -i --exclude-standard

7. Branching

□Git branch Create branch: \$ git branch <branch name> List Branch: \$ git branch -- list Delete Branch: \$ git branch -d<branch name> Delete a remote Branch: \$ git push origin -delete <branch name> Rename Branch: \$ git branch -m <old branch name><new branch name> ☐ Git checkout Switch between branches in a repository. Switch to a particular branch: \$ git checkout <branch name> Create a new branch and switch to it: \$ git checkout -b
branchname> Checkout a Remote branch: \$ git checkout < remotebranch> ☐ Git stash Switch branches without committing the current branch. Stash current work:



Merge the specified commit to currently active branch: \$ git merge <commit> ☐ Git rebase Apply a sequence of commits from distinct branches into a final commit. \$ git rebase < branch name > Continue the rebasing process: \$ git rebase —continue Abort the rebasing process: \$ git rebase --skip ☐ Git interactive rebase Allow various operations like edit, rewrite, reorder, and more on existing commits. \$ git rebase -i 9. Remote ☐Git remote Check the configuration of the remote server: \$ git remote -v Add a remote for the repository: \$ git remote add <short name><remote URL> Fetch the data from remote server \$ git fetch < Remote > Remove a remote connection from the repository: \$ git remote rm <destination> Rename remote server: \$ git remote rename <old name><new name>

	Show additional information about a particular remote:
	\$ git remote show <remote></remote>
	Change remote:
	\$ git remote set-url <remote name=""><newurl></newurl></remote>
	Git origin master
	Push data to remote server:
	\$ git push origin master
	Pull data from remote server:
	\$ git pull origin master
LO.F	Pushing Updates
	Git push
	Transfer the commits from your local repository to a remote server.
	Push data to remote server:
	\$ git push origin master
	Force push data:
	\$ git push <remote><branch> -f</branch></remote>
	Delete a remote branch by push command:
	\$ git push origin -delete edited
L1.F	Pulling updates
	Git pull
	Pull the data from the server:
	\$ git pull origin master
	Pull a remote branch:
	\$ git pull <remote branch="" url=""></remote>

	Git fetch
	Downloads branches and tags from one or more repositories.
	Fetch the remote repository:
	\$ git fetch< repository Url>
	Fetch a specific branch:
	\$ git fetch <branch url=""><branch name=""></branch></branch>
	Fetch all the branches simultaneously:
	\$ git fetch –all
	Synchronize the local repository:
	\$ git fetch origin
12.ل	Jndo changes
	Git revert
	Undo the changes
	\$ git revert
	Revert a particular commit:
	\$ git revert <commit-ish></commit-ish>
	Git reset
	Reset the changes:
	\$ git reset —hard
	\$ git reset –soft
	\$ git resetmixed
12 0	Removing files

13. Removing files

2 Git rm

Remove the files from the working tree and from the index:

\$ git rm <file Name>

Remove files from the Git But keep the files in your local repository:

\$ git rm --cached