Je1234GIT Commands

Login gitHub

1.username:

2.password :

1. Git init
2. ssh-keygen –t rsa
3. passphrase 1234
4. once again 1234
5. cd ~/.ssh
6. ls
7. cat id-rsa.pub
8. key copy

**incase already ssh key created means**

1. $ ls -al ~/.ssh
2. $ ssh-keygen -t rsa -b 4096 -C “[rajesh9951179@gmail.com](mailto:rajesh9951179@gmail.com)” ->your\_email@example.com
3. $ eval "$(ssh-agent -s)"
4. $ ssh-add ~/.ssh/id\_rsa
5. $ cat ~/.ssh/id\_rsa.pub
6. $ ssh -T [git@github.com](mailto:git@github.com)

Git Account

SSH key is used in git hub

1. Settings
2. SSH and GPG keys
3. New SSH key
4. generate

Incase not Working

1. Settings
2. Developer setting
3. Person acess token
4. Token classic
5. Generate new token
6. Generate new token classic
7. Note
8. Custom and date select
9. Select all
10. Generate token
11. That key copy after TsE9bdfXUo71o3yKUUwyumarLItvuWA98pd836eMsQw

git remote set-url origin https://TsE9bdfXUo71o3yKUUwyumarLItvuWA98pd836eMsQw@github.com/rajesh9951179/MyMechanic

**MAIN COMMANDS**

1. git bash -> path added after
2. git clone <https://github.com/rajesh9951179/MyMechanic.git>
3. modification that project
4. git status
5. git add .
6. git restore –staged filename take back
7. git commit –m “haii”

SOME TIMES git config –global user.email “rajesh9951179@gmail.com”

git config –global user.name “958848484854”

1. git branch -b “customer”
2. git checkout customer
3. git branch
4. git push origin main
5. INCASE NOT WORK August 31 authentication problem
6. git remote set-url origin <https://TsE9bdfXUo71o3yKUUwyumarLItvuWA98pd836eMsQw@github.com/rajesh9951179/MyMechanic>1
7. git push origin main
8. git push origin main –force
9. git remote –v
10. git push origin main
11. project click repositories
12. settings
13. collaborators
14. add people
15. another notification
16. invite accept

EXTRA : git pull origin main

Git merge main

Commit take back

1. git log
2. git reset --soft HEAD~1 **commit take back**
3. git restore --staged **fileName**
4. git reset --hard <commit-hash>
5. git reset --hard 89abcdef987654324567890jhdgfv
6. git push origin main –force

important: git remote set-url origin [git@github.com:gandhamdurgesh/testing.git](mailto:git@github.com:gandhamdurgesh/testing.git)

git remote set-url origin git@github.com:rajesh9951179/MERN-API-BASIC.git



To set up branch protection for the main branch with the typical rules to ensure code quality and safety, here are the recommended checkboxes to tick:

1. **Branch name pattern**:
   * Enter main to apply the rules specifically to the main branch.
2. **Protect matching branches**:
   * **Require a pull request before merging**: ✔️ (Ensures that all changes to the main branch are reviewed via a pull request before being merged.)
   * **Require status checks to pass before merging**: ✔️ (Ensures that any automated tests or checks you have set up must pass before the code can be merged.)
   * **Require conversation resolution before merging**: ✔️ (Ensures that all conversations on a pull request must be resolved before merging, promoting better collaboration and code quality.)
   * **Require signed commits**: Optional (Tick this if you want to ensure that all commits are signed and verified. This is more security-focused.)
   * **Require linear history**: Optional (Tick this if you want to enforce a linear commit history, which can make the commit history cleaner but might restrict some workflows.)
   * **Require deployments to succeed before merging**: Optional (Tick this if you have a deployment pipeline set up and want to require successful deployment before merging.)
   * **Lock branch**: Optional (Tick this if you want to make the branch read-only, where no one can push directly to it. This is highly restrictive.)
   * **Do not allow bypassing the above settings**: ✔️ (Tick this to ensure that even administrators cannot bypass these protection rules.)
3. **Rules applied to everyone including administrators**: ✔️ (Ensures that all rules apply to everyone, including admins.)
4. **Allow force pushes**: Leave unchecked (Do not allow force pushes as they can overwrite commit history, which can lead to loss of work.)
5. **Allow deletions**: Leave unchecked (Do not allow the deletion of the main branch to protect it from accidental deletion.)

After ticking these checkboxes, click **"Save changes"** to apply the protection rules.

These settings will help protect the integrity of your main branch, requiring reviews and checks before any changes are merged.

4o