

# The Ultimate Git Cheat Sheet!

## Git Configuration

```
git config --global user.name "Your Name"
git config --global user.email "your@email.com"
git config --global init.defaultBranch main
git config --global core.editor code git config --list
```

## Starting a Git Project

```
git init
git clone <url>
```

## Tracking and Staging Files

```
git status
git add <file> git add .
git restore --staged <file> git rm <file>
```

## Comitting Changes

```
git commit -m "Message"
git commit -am "Msg"
```

## Working with Branches

```
git switch <branch>
git switch -c <new-branch> git switch -
git checkout <branch>
git checkout -b <new-branch> git branch
git branch -d <branch> git branch -D <branch>
```

# The Ultimate Git Cheat

## Merging & Rebasing

```
git merge <branch>  
git rebase <branch>  
git rebase -i HEAD~n
```

## Remote Repositories

```
git remote add origin <url>  
git remote -v  
git push -u origin main  
git push  
git pull  
git fetch
```

## File Restore Commands

```
git restore <file>  
git restore --source <commit> <file>  
git restore .
```

## Logs and Diffs

```
git log  
git log --oneline --graph  
git show <commit>  
git diff  
git diff --staged
```

## Stashing

```
git stash  
git stash list  
git stash pop  
git stash apply stash@{n}  
git stash drop stash@{n}
```

## Undoing Changes

```
git restore <file>
git reset <file>
git reset --hard
git reset --soft HEAD~1
git revert <commit>
```

## Tags

```
git tag
git tag v1.0
git tag -a v1.0 -m "Release v1.0"
git push origin --tags
```

## Collaboration Workflow

```
git checkout -b feature/cool-feature
# or:
git switch -c feature/cool-feature
git push origin feature/cool-feature

# Open PR on GitHub
git switch main
git pull origin main
```

## .gitignore Example

Writing these will simply ignore directories while staging for push

```
node_modules/
.env
*.log
.DS_Store
```

## Clean Up Untracked Files

```
git clean -f
git clean -fd
```

## Advanced Shortcuts

```
git log --oneline --graph --all
```

```
git diff HEAD~1
```

```
git shortlog -sne
```

---

## Pro Tips

- **Always git pull before starting new work.**
- Use git switch and git restore instead of checkout.
- Use branches for every new feature or bug fix.
- Keep commits small and meaningful.
- Make frequent commits; it's your safety net.



### Ettisaf Rup

CSE@KUET | Computer Science Aficionado Software Engineering | AI/ML

— **Aspiring Computer Engineer**, Undergraduate at **B.Sc, CSE | KUET** with a strong foundation in programming, problem-solving, and system-level understanding. Passionate about learning, exploring and understanding how things work under the hood. Enthusiastic about building efficient, scalable, meaningful and innovative solutions. Known for being a fast learner, team player, and someone who