

Download git from gitSCM.Configure git on git-bash terminal:(This required for self identity)

1. git config --global user.name "Neeraj Kumar"
2. git config --global user.email "nkurmbanshi@yahoo.com"
3. git config --list

Create project on our desktop:-----

1. git init #initialize git repository
2. git status
 - I. untracked (new file created.....)
 - II. modified (file already exist but new changes detected...)
 - III. staged (file is ready for committed.....)
 - IV. unmodified (file already exist but no change found.....)
3. git add file_name
4. git add . (add all file with single command)
5. git commit -m "remark according changes"
6. git remote add origin <-----github repo link----->(for linking our desktop project to github repo)
7. git remote -v (for checking remote repo link)
8. git branch -M main (Rename current branch name(Change our master branch into main branch as default))
9. git push origin main(Push our code to github)

Branching commands:-----

1. git branch (To check current working branch)
2. git branch -M <branch-name> (Rename current working branch)
3. git checkout <branch-name> (switch to given branch-name)
4. git checkout -b <new-branch-name> (create new branch)
5. git checkout -d <branch-name>

Merging code:---

1. git diff <---branch-name---->
2. git merge <---branch-name---->

Resetting files:-----

1. for stages changes:--
 1. git reset <file-name>
 2. git reset
2. for committed changes(for one commit)
 1. git reset HEAD-1
3. For committed changes (for many commits)
 1. git reset <----commit hash----->
2. git reset --hard <----commit hash----->

Open git-bash :---

1. git --version
2. ls (show all list)
3. clear
4. pwd (present working directory)

Configure git on git-bash terminal:----

1. git config --global user.name "Neeraj Kumar"
2. git config --global user.email "nkurmbanshi@yahoo.com"
3. git config --list

Open vs-code in your system:---

1. clone :---

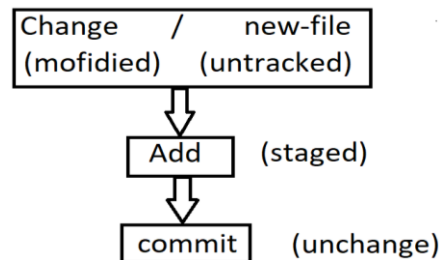
1. git clone <repo-----link>
2. git remote add origin <repo-----link>

remote(github) or local(laptop/desktop):--- it is used to clone our got repo to our laptop.

(git clone <https://github.com/neerajpatel2505/methods.git>)

2. status :--- (git status)

1. untracked (new file created....)
2. modified (file already exist but new changes detected)
3. staged (file is ready for committed)
4. unmodified (file already exist but no change found)



3. add :---

1. git add <file_name>
2. git add . -----(add all files)

4. commit :---

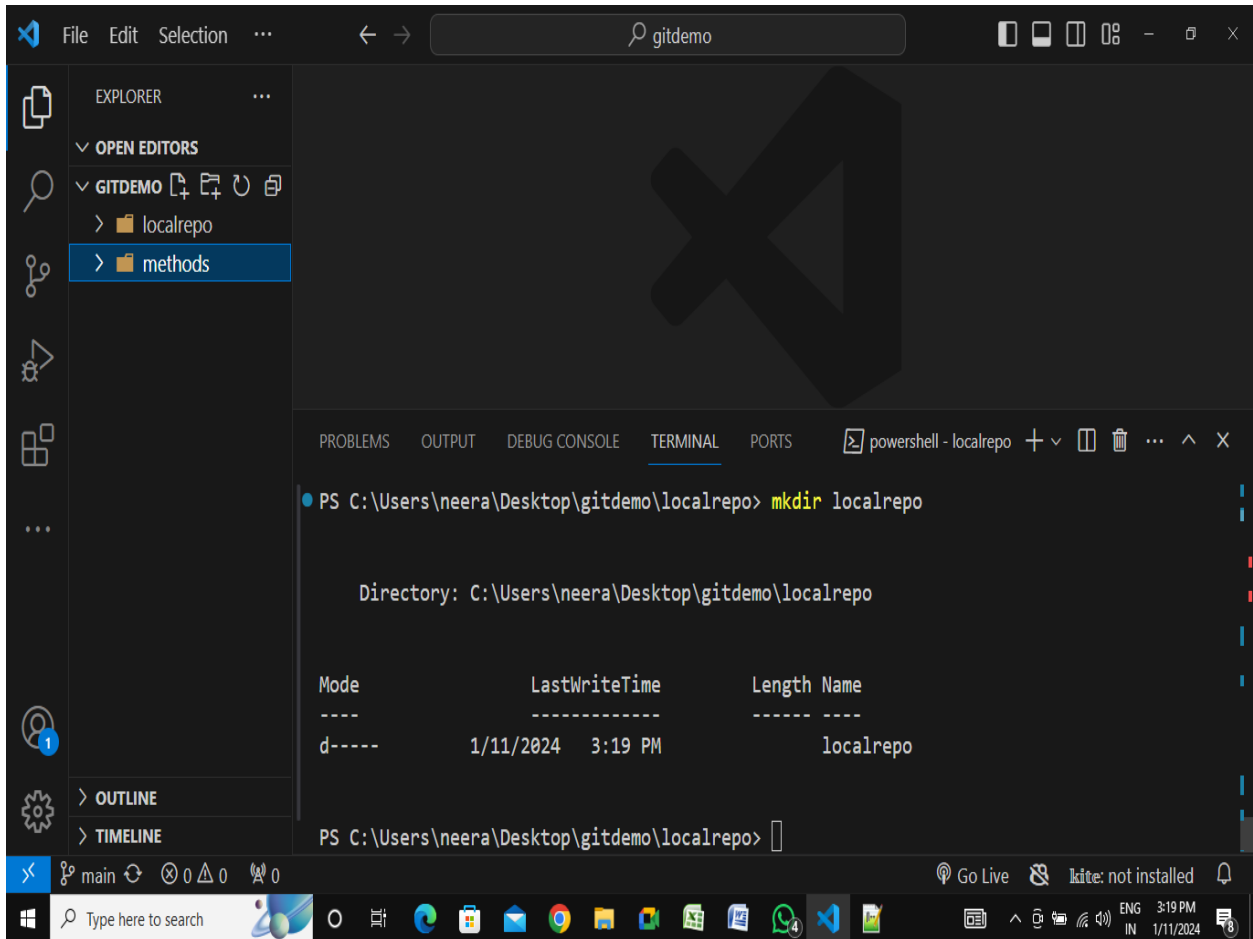
1. git commit -m "change description....."
2. git commit -am "Change description....."(add and commit both at a time)

5. Push :--

1. git branch -M main
2. git push origin main

Upload Project from laptop to Github:---

Create new repo in through laptop/our-system



The screenshot shows the Visual Studio Code interface with a PowerShell terminal open. The terminal is running the command `mkdir localrepo` in the directory `C:\Users\neera\Desktop\gitdemo\localrepo`. The output shows the directory was created successfully, listing its mode, last write time, and length.

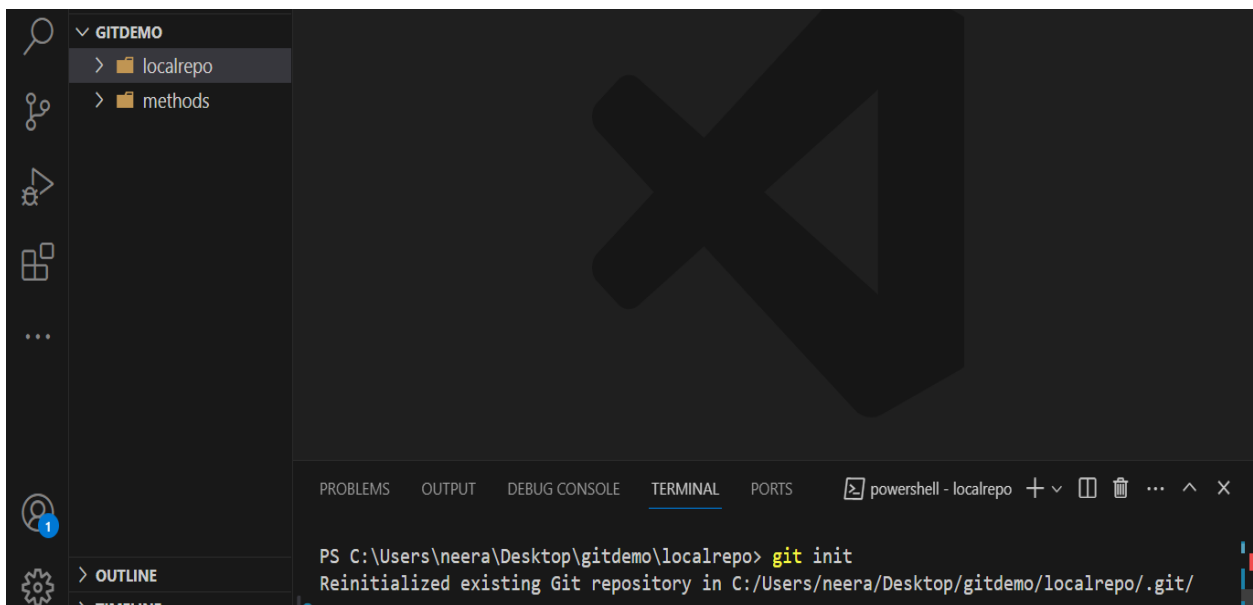
```
PS C:\Users\neera\Desktop\gitdemo\localrepo> mkdir localrepo

Directory: C:\Users\neera\Desktop\gitdemo\localrepo

Mode                LastWriteTime         Length Name
----                -
d-----            1/11/2024   3:19 PM             localrepo

PS C:\Users\neera\Desktop\gitdemo\localrepo>
```

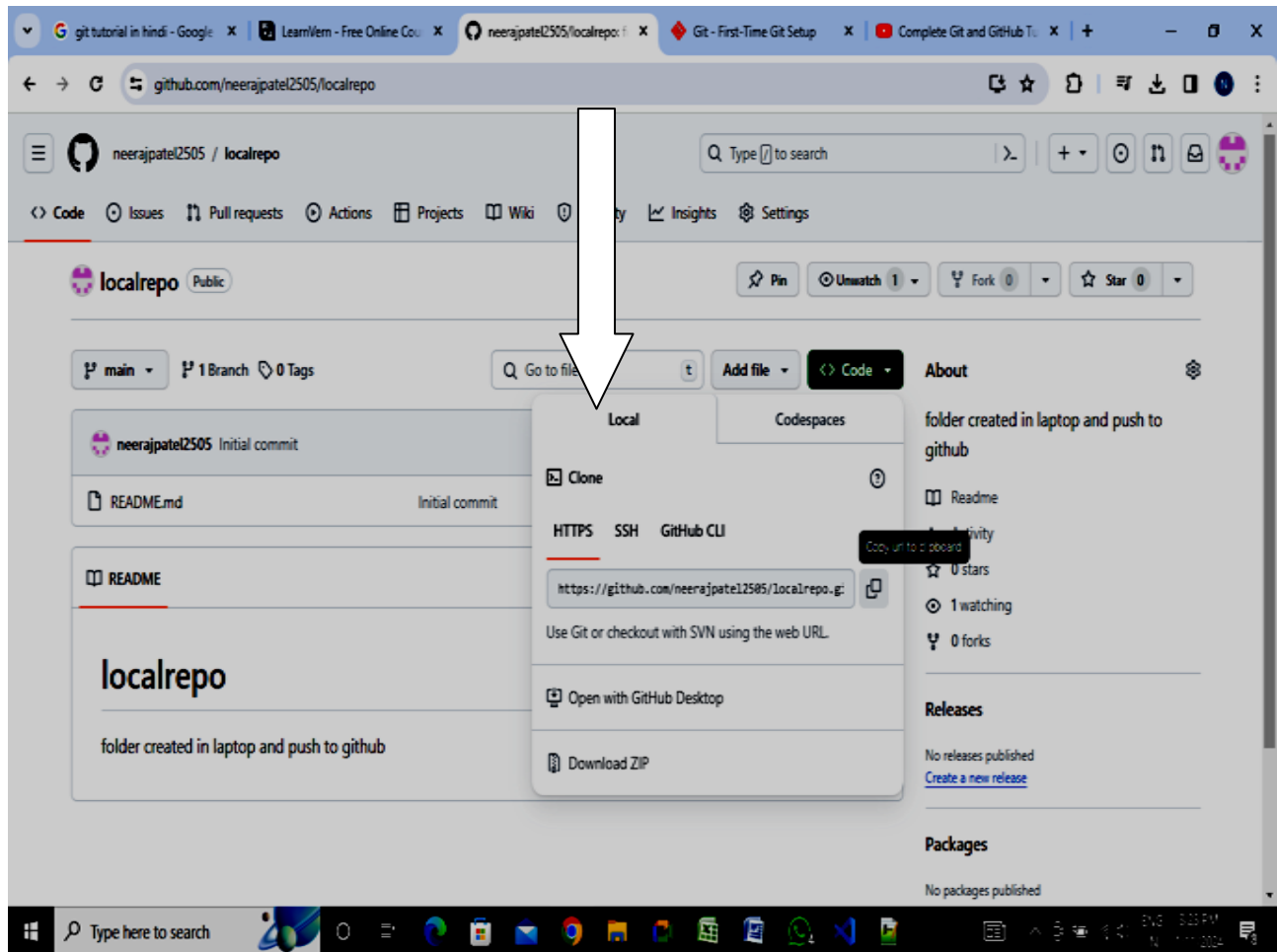
git init :-----



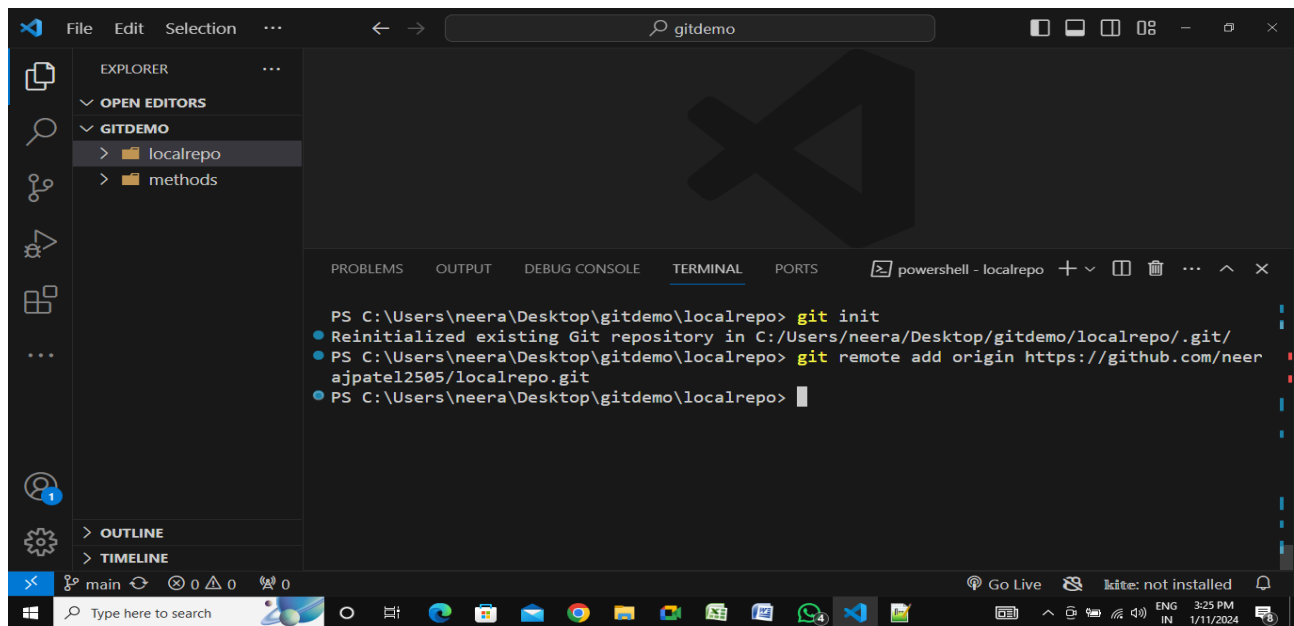
The screenshot shows the Visual Studio Code interface with a PowerShell terminal open. The terminal is running the command `git init` in the directory `C:\Users\neera\Desktop\gitdemo\localrepo`. The output shows that a new Git repository was initialized.

```
PS C:\Users\neera\Desktop\gitdemo\localrepo> git init
Reinitialized existing Git repository in C:/Users/neera/Desktop/gitdemo/localrepo/.git/
```

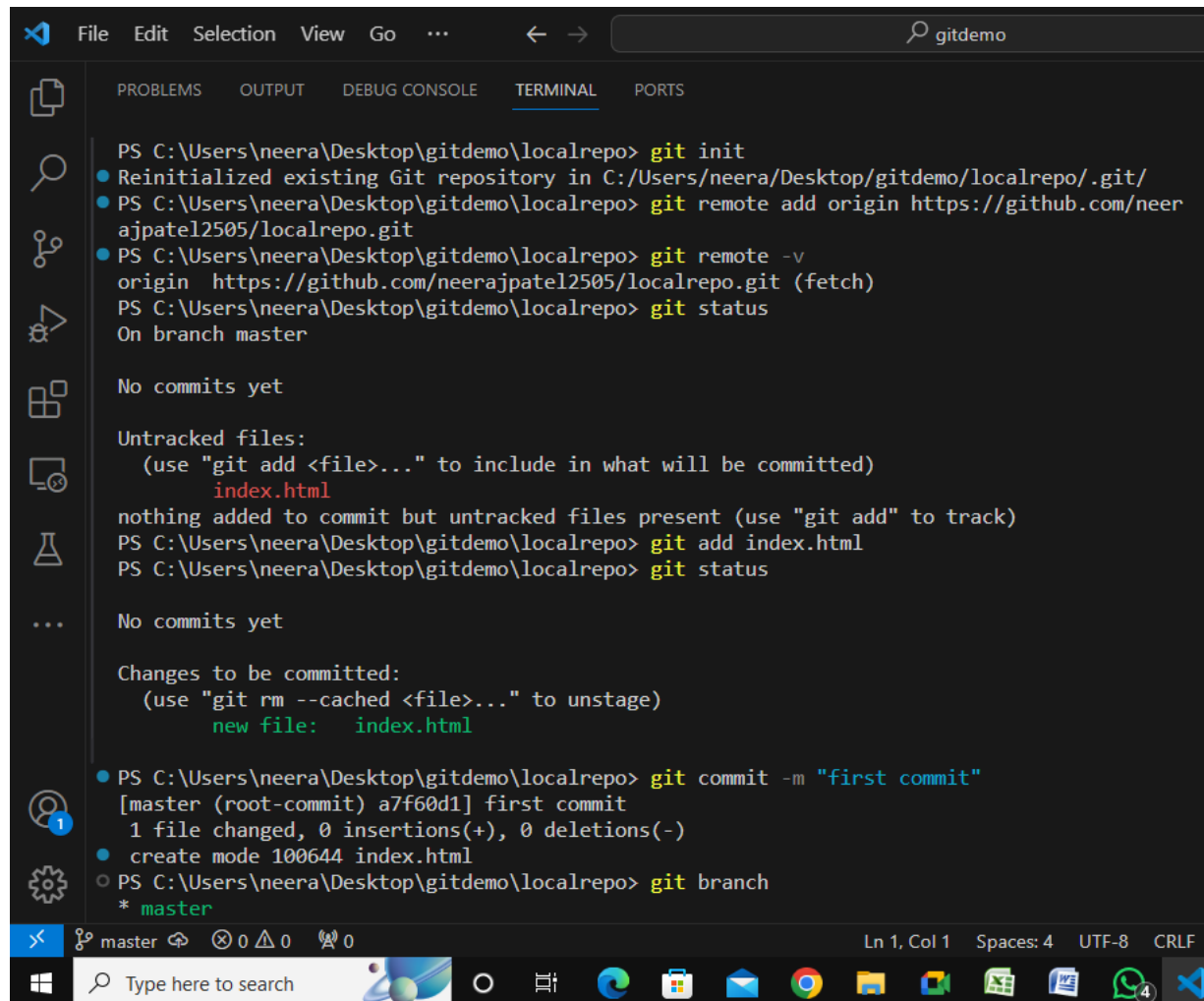
Create new repo in github and copy the link :--



git remote add origin < repo-link>



Check remote added or not:--- (git remote -v)



```
PS C:\Users\neera\Desktop\gitdemo\localrepo> git init
• Reinitialized existing Git repository in C:/Users/neera/Desktop/gitdemo/localrepo/.git/
• PS C:\Users\neera\Desktop\gitdemo\localrepo> git remote add origin https://github.com/neerajpatel2505/localrepo.git
• PS C:\Users\neera\Desktop\gitdemo\localrepo> git remote -v
origin https://github.com/neerajpatel2505/localrepo.git (fetch)
PS C:\Users\neera\Desktop\gitdemo\localrepo> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        index.html
nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\neera\Desktop\gitdemo\localrepo> git add index.html
PS C:\Users\neera\Desktop\gitdemo\localrepo> git status

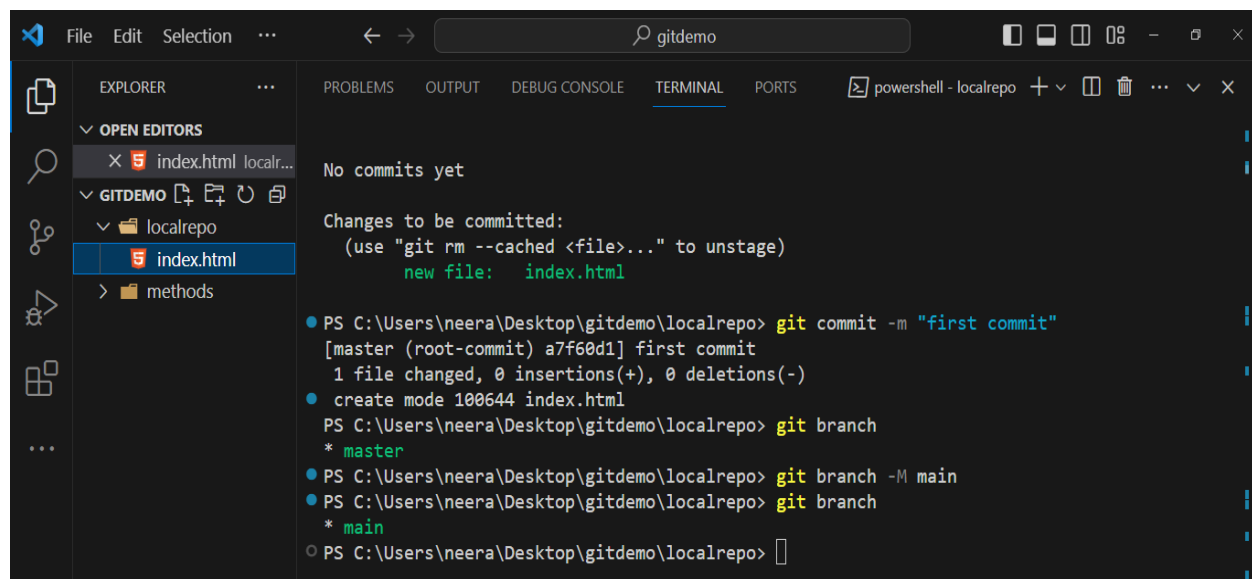
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

• PS C:\Users\neera\Desktop\gitdemo\localrepo> git commit -m "first commit"
[master (root-commit) a7f60d1] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
• create mode 100644 index.html
• PS C:\Users\neera\Desktop\gitdemo\localrepo> git branch
* master
```

Change master name to main :---

1. git branch (check branch name)
2. git branch -M main (Change into master to main)



```
PS C:\Users\neera\Desktop\gitdemo\localrepo> git commit -m "first commit"
[master (root-commit) a7f60d1] first commit
 1 file changed, 0 insertions(+), 0 deletions(-)
• create mode 100644 index.html
PS C:\Users\neera\Desktop\gitdemo\localrepo> git branch
* master

• PS C:\Users\neera\Desktop\gitdemo\localrepo> git branch -M main
• PS C:\Users\neera\Desktop\gitdemo\localrepo> git branch
* main

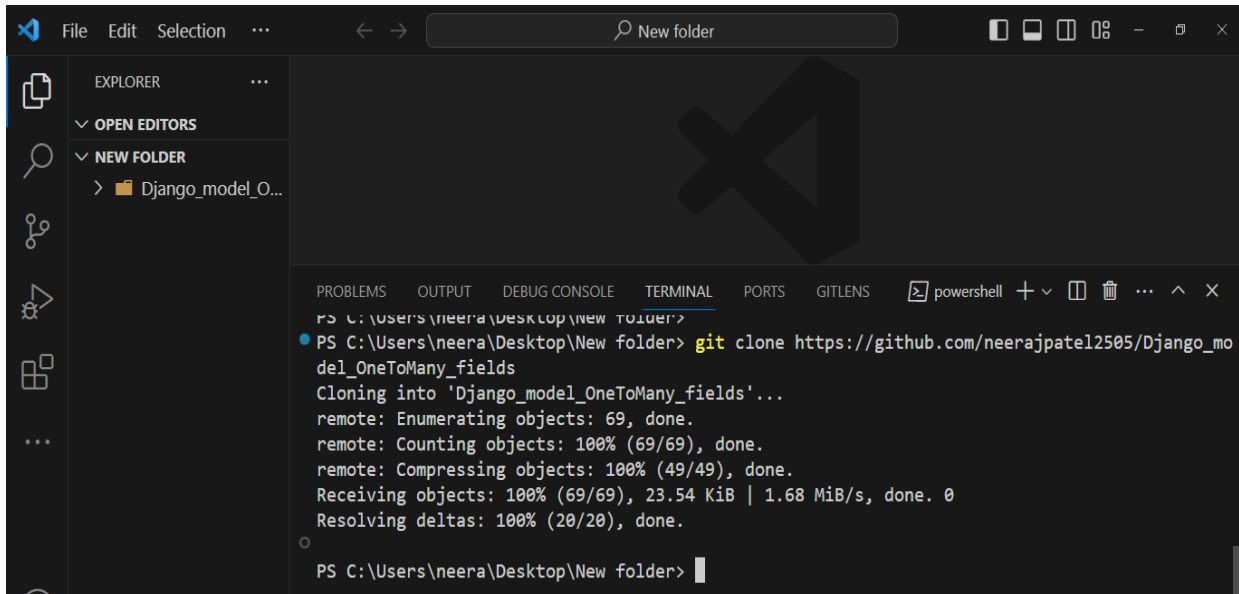
PS C:\Users\neera\Desktop\gitdemo\localrepo>
```

Now, finally push into github : -----

- 1. git push origin main / git push -u origin main-----**
- 2. git push origin master --force**

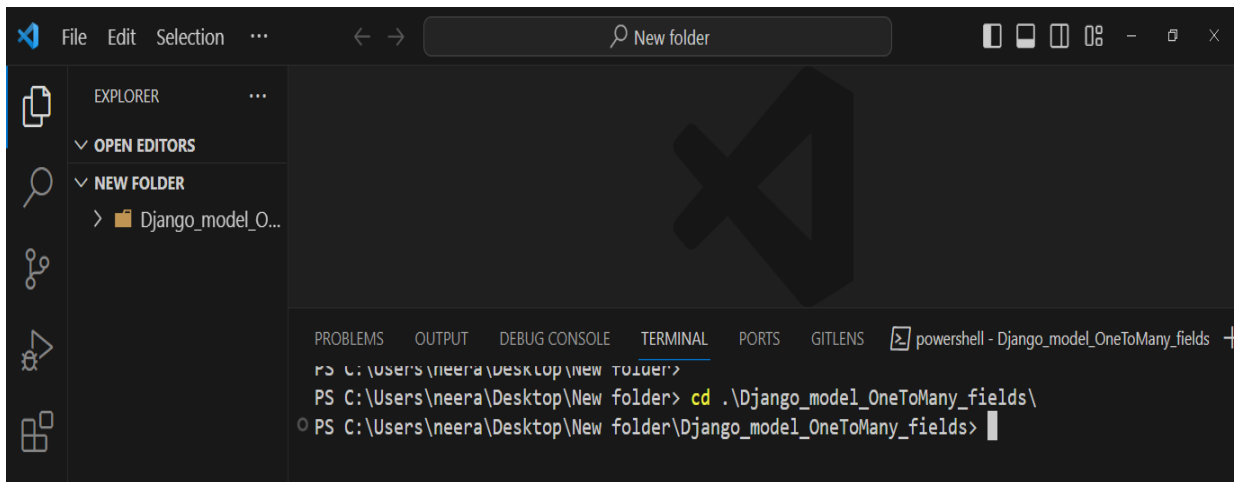
-----:Working with git repo by creating own branch:-----

1. git clone <repo link from github>



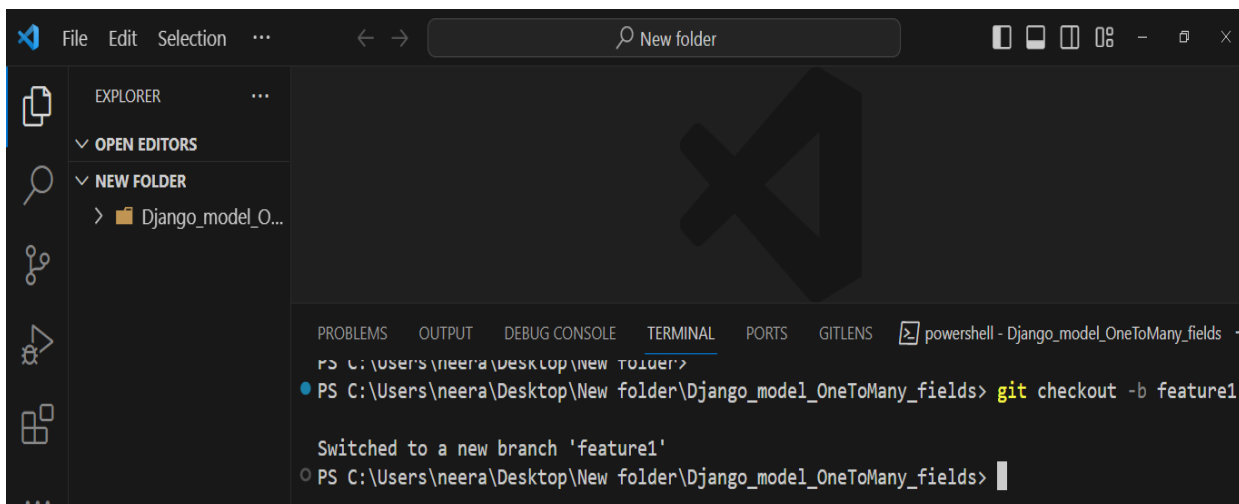
```
PS C:\Users\neera\Desktop\New folder> git clone https://github.com/neerajpate12505/Django_model_OneToMany_fields
Cloning into 'Django_model_OneToMany_fields'...
remote: Enumerating objects: 69, done.
remote: Counting objects: 100% (69/69), done.
remote: Compressing objects: 100% (49/49), done.
Receiving objects: 100% (69/69), 23.54 KiB | 1.68 MiB/s, done.
Resolving deltas: 100% (20/20), done.
```

2. enter clone folder:--- by using cd folder_name



```
PS C:\Users\neera\Desktop\New folder> cd .\Django_model_OneToMany_fields\
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields>
```

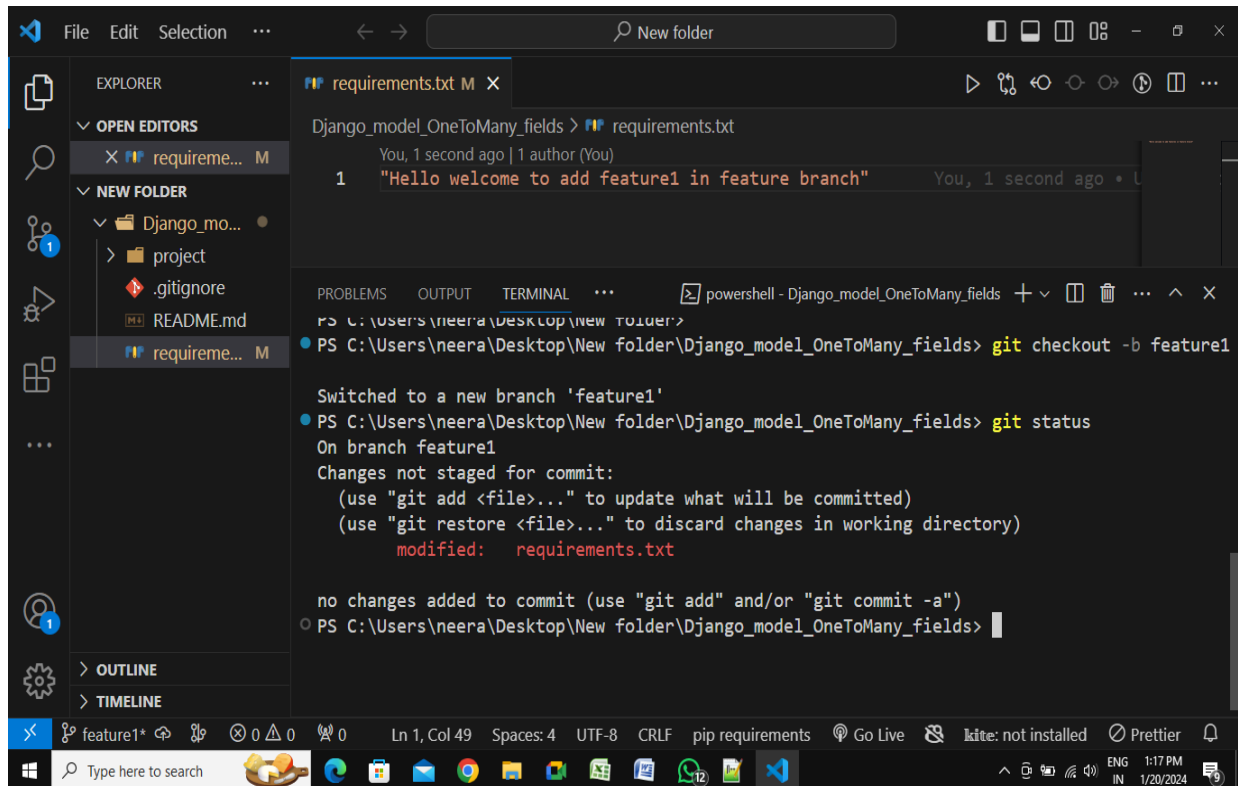
3. create own branch:----- (git checkout -b branchname)



```
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git checkout -b feature1

Switched to a new branch 'feature1'
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields>
```

4. Do own changes whatever we want on that branch and check status :-----



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project structure with a folder named 'project' containing '.gitignore', 'README.md', and 'requirements.txt'. The main editor window shows the 'requirements.txt' file with the following content:

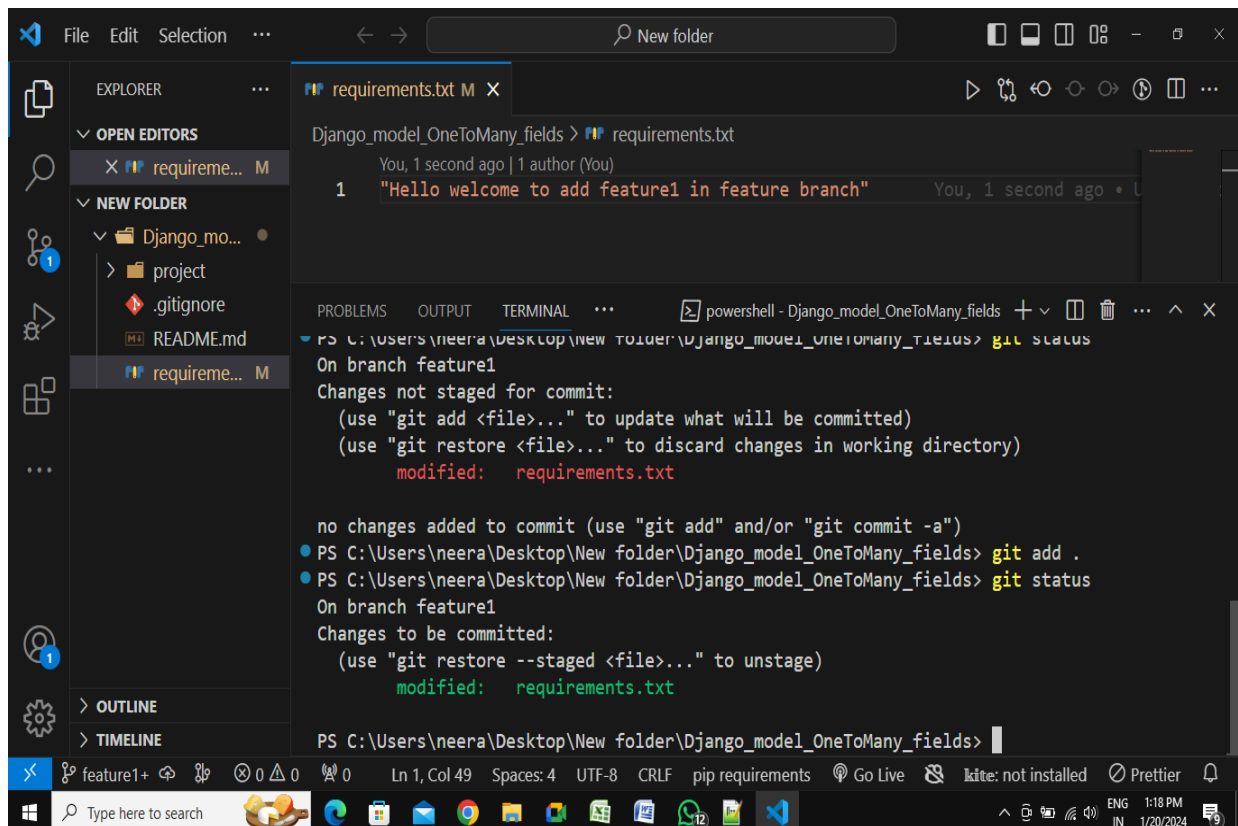
```
1 "Hello welcome to add feature1 in feature branch"
```

The terminal panel at the bottom shows the following commands and output:

```
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git checkout -b feature1
Switched to a new branch 'feature1'
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git status
On branch feature1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   requirements.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields>
```

5. git add file_name /git add . :-----to add all files and folders.



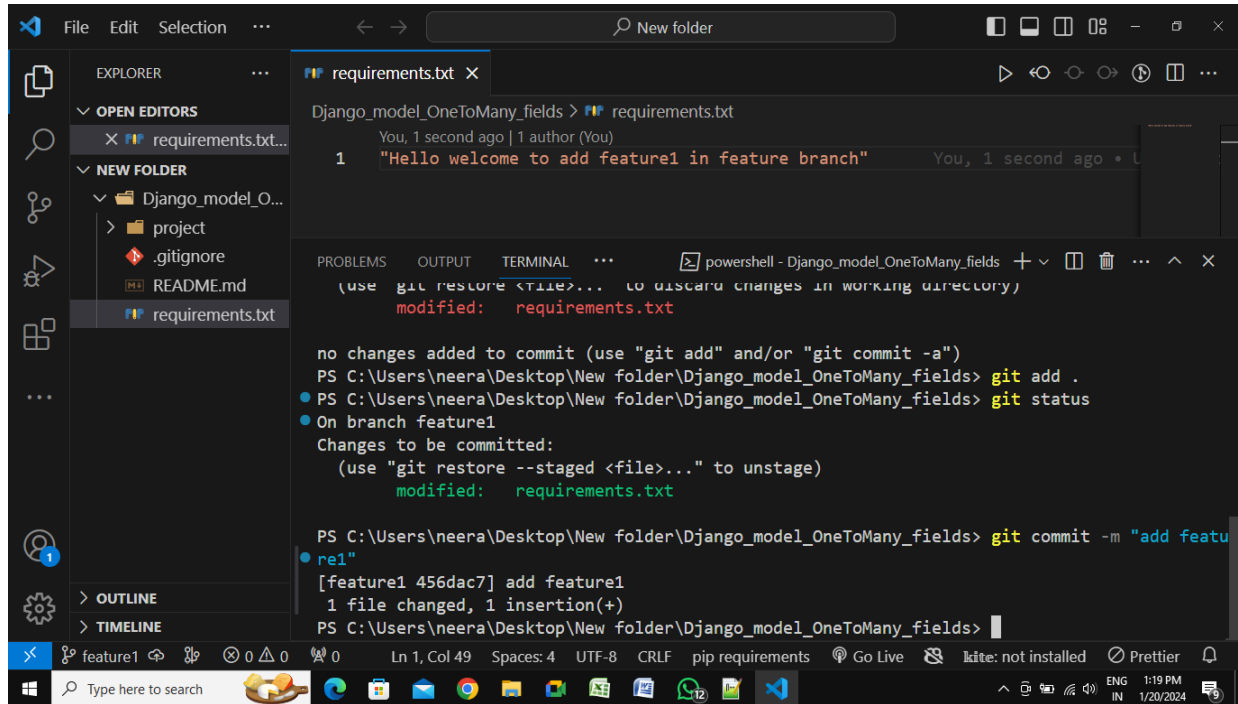
The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows the same project structure as before. The main editor window shows the 'requirements.txt' file with the same content. The terminal panel at the bottom shows the following commands and output:

```
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git status
On branch feature1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   requirements.txt

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git add .
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   requirements.txt

PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields>
```


6. Do commit on own branch:----

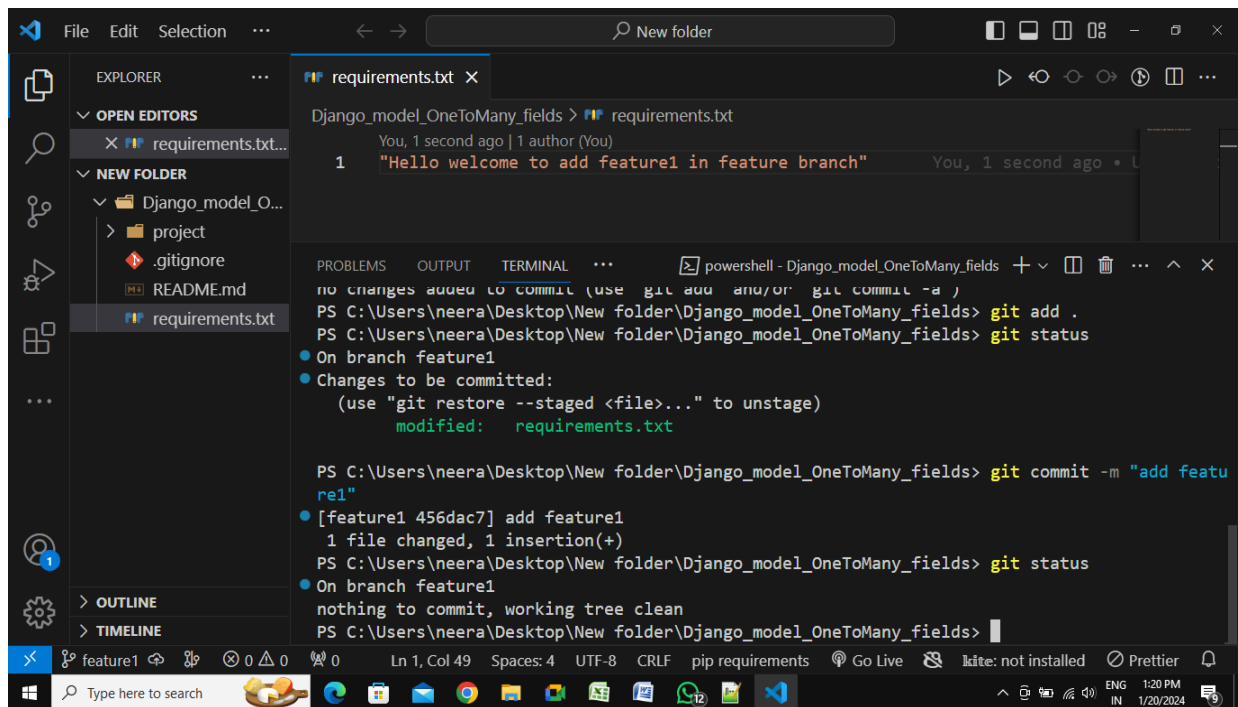


This screenshot shows the initial state of a Git repository in VS Code. The Explorer sidebar on the left shows a project folder containing `.gitignore`, `README.md`, and `requirements.txt`. The Editor window displays `requirements.txt` with the content: `"Hello welcome to add feature1 in feature branch"`. The integrated terminal at the bottom shows the following commands and output:

```
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git add .
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   requirements.txt

PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git commit -m "add feature1"
```

The status bar at the bottom indicates the current branch is `feature1` and the file `requirements.txt` is modified.



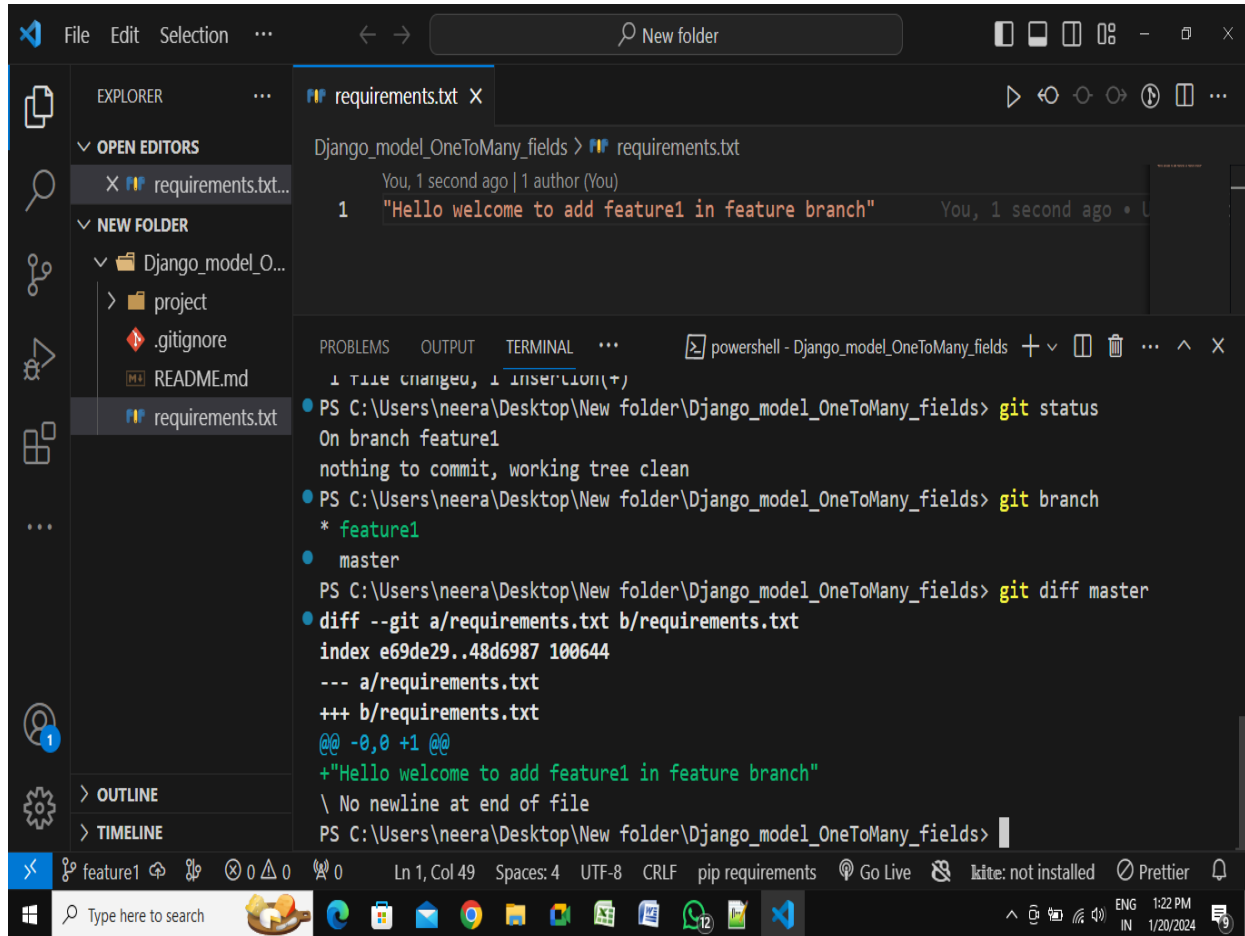
This screenshot shows the state of the repository after the commit. The Explorer sidebar remains the same. The integrated terminal shows the following commands and output:

```
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git add .
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   requirements.txt

PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git commit -m "add feature1"
[feature1 456dac7] add feature1
1 file changed, 1 insertion(+)
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git status
On branch feature1
nothing to commit, working tree clean
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields>
```

The status bar at the bottom now shows the repository is clean and the working tree is up-to-date.

7. Track all changes:----- git diff (branchname)



The screenshot shows the Visual Studio Code interface with a file explorer on the left showing a project structure. The main editor displays the content of `requirements.txt`, which contains a single line: `"Hello welcome to add feature1 in feature branch"`. The terminal at the bottom shows the output of several git commands:

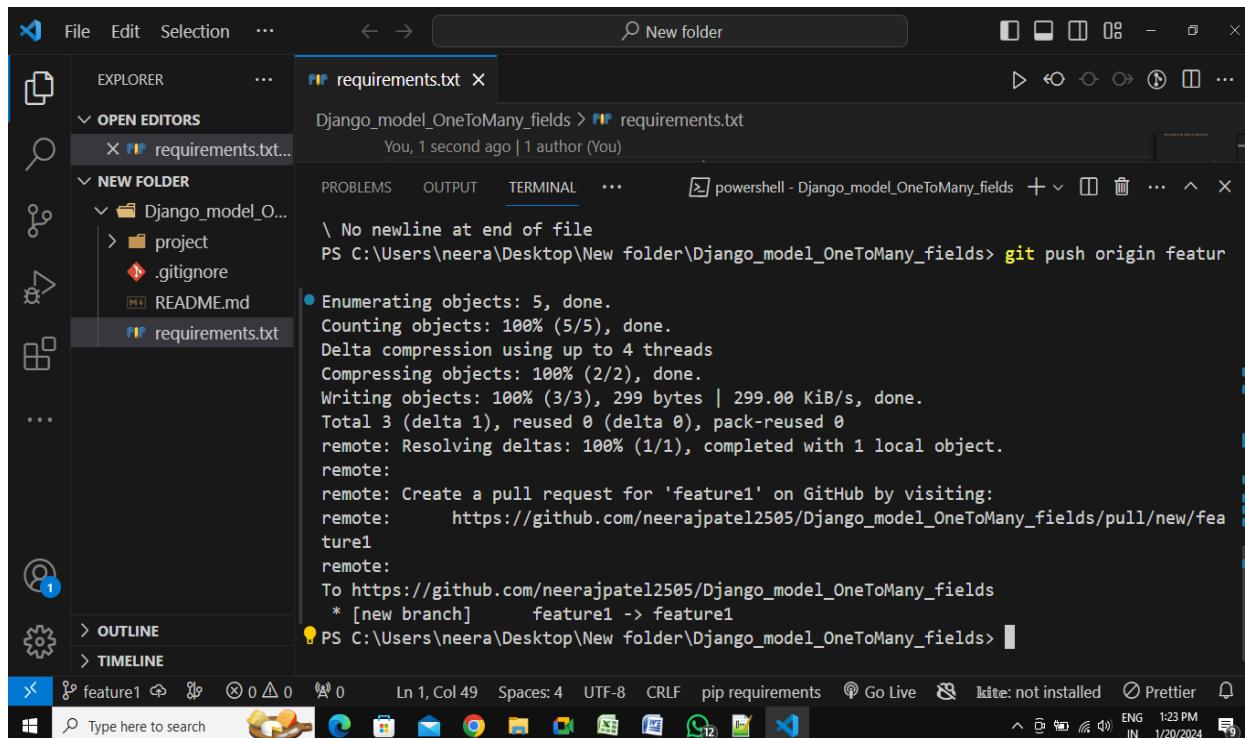
```
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git status
On branch feature1
nothing to commit, working tree clean

PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git branch
* feature1
master

PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git diff master
diff --git a/requirements.txt b/requirements.txt
index e69de29..48d6987 100644
--- a/requirements.txt
+++ b/requirements.txt
@@ -0,0 +1 @@
+"Hello welcome to add feature1 in feature branch"
\ No newline at end of file

PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields>
```

8. Push feature1 change code to github:----git push origin feature1



The screenshot shows the Visual Studio Code interface with the same project structure. The terminal at the bottom shows the output of the `git push` command:

```
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields> git push origin featur

Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature1' on GitHub by visiting:
remote:   https://github.com/neerajpatel2505/Django_model_OneToMany_fields/pull/new/fea
remote:   ture1
remote:
To https://github.com/neerajpatel2505/Django_model_OneToMany_fields
 * [new branch]   feature1 -> feature1
PS C:\Users\neera\Desktop\New folder\Django_model_OneToMany_fields>
```

9. Github account after push request by feature1 brach:---

The screenshot shows the GitHub repository page for 'Django_model_OneToMany_fields' (Public). The repository is owned by 'neerajpatel2505'. The 'feature1' branch is selected, and a yellow banner indicates 'feature1 had recent pushes 3 minutes ago' with a 'Compare & pull request' button. The repository has 2 branches and 0 tags. The commit history shows a 'template work add' commit by 'neerajpatel2505' 2 days ago, with 2 commits. The file list includes 'project', '.gitignore', 'README.md', and 'requirements.txt'. The right sidebar shows repository statistics: 0 stars, 1 watching, 0 forks, and no releases or packages published.

10. Click Compare & pull request:--

The screenshot shows the GitHub repository page for 'Django_model_OneToMany_fields' (Public). The repository is owned by 'neerajpatel2505'. The 'feature1' branch is selected, and a banner indicates 'This branch is 1 commit ahead of, 1 commit behind master'. A 'Compare & pull request' dialog box is open, showing the 'feature1' branch is 1 commit ahead of 'master'. The dialog box has a 'Compare' button and an 'Open pull request' button. The file list includes 'project', '.gitignore', 'README.md', and 'requirements.txt'. The right sidebar shows repository statistics: 0 stars, 1 watching, 0 forks, and no releases or packages published.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).

base: master ← compare: feature1 ✓ Able to merge. These branches can be automatically merged.

Add a title

add feature2

Add a description

Write Preview

add feature2

Markdown is supported Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Development

Use [Closing keywords](#) in the description to automatically close issues

Helpful resources

[GitHub Community Guidelines](#)

Create pull request(PR):-----

add feature2 by neerajpatel2505

neerajpatel2505 / Django_model_OneToMany_fields

Code Issues Pull requests 1 Actions Projects Wiki Security Insights Settings

add feature2 #2

Open neerajpatel2505 wants to merge 1 commit into master from feature1

Conversation 0 Commits 1 Checks 0 Files changed 1 +2 -1

neerajpatel2505 commented 1 minute ago

add feature2

add feature2 0b8535f

Add more commits by pushing to the [feature1](#) branch on [neerajpatel2505/Django_model_OneToMany_fields](#).

Require approval from specific reviewers before merging

Rulesets ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Add a comment

Write Preview

comment.....

Markdown is supported Paste, drop, or click to add files

Close with comment Comment

Reviewers: No reviews

Still in progress? [Convert to draft](#)

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant

Lock conversation

Click Merge pull request:----

The screenshot shows a web browser window with the URL `github.com/neerajpatel2505/Django_model_OneToMany_fields/pull/2`. The page displays a pull request titled "add feature2 #2" where "neerajpatel2505" wants to merge 1 commit into `master` from `feature1`. A green "Open" button is visible. Below the header, a message says "Add more commits by pushing to the `feature1` branch on `neerajpatel2505/Django_model_OneToMany_fields`." A merge confirmation dialog is open, showing the commit message "Merge pull request #2 from neerajpatel2505/feature1" and "add feature2". It notes the commit will be authored by `98191185+neerajpatel2505@users.noreply.github.com` and has "Confirm merge" and "Cancel" buttons. Below the dialog is a "Add a comment" section with a text area and a "Comment" button. On the right, there are settings for Labels, Projects, Milestone, Development, and Notifications. The Windows taskbar at the bottom shows the time as 3:05 PM on 1/20/2024.

Click conform merge:---

The screenshot shows the same GitHub pull request page, but now the pull request has been successfully merged. A message at the top states "neerajpatel2505 merged commit 39089fb into `master` now" with a "Revert" button. Below this, a purple box says "Pull request successfully merged and closed" and "You're all set—the `feature1` branch can be safely deleted." with a "Delete branch" button. The "Add a comment" section remains. The right-hand settings panel is still visible. The Windows taskbar at the bottom shows the time as 3:06 PM on 1/20/2024.

If you want to delete that particular branch then use delete branch button or add a comment on that particular section and press Comment button:----

The screenshot shows a GitHub pull request page for the repository 'neerajpatel2505/Django_model_OneToMany_fields/pull/2'. The pull request is titled 'add feature2 #2' and was merged 3 minutes ago. The merge message indicates that 'neerajpatel2505 merged 1 commit into master from feature1'. The pull request is marked as 'Merged' with a green checkmark icon. The main content area shows a comment from 'neerajpatel2505' stating 'Add new feature2' and 'ok'. The right sidebar shows the pull request details, including the milestone (No milestone), development status (Successfully merging this pull request may close these issues), and notifications (Unsubscribe button). At the bottom, a message states 'Pull request successfully merged and closed' and 'You're all set—the feature1 branch can be safely deleted.' with a 'Delete branch' button. The Windows taskbar is visible at the bottom.

add feature2 #2
neerajpatel2505 merged 1 commit into `master` from `feature1` 3 minutes ago

Milestone
No milestone

Development
Successfully merging this pull request may close these issues.

None yet

Notifications [Customize](#)
[Unsubscribe](#)

You're receiving notifications because you're watching this repository.

1 participant

[Lock conversation](#)

Pull request successfully merged and closed
You're all set—the `feature1` branch can be safely deleted. [Delete branch](#)

Add a comment
Type here to search

Pull request successfully merged and closed.