

1. Set your username and email in git config

```
Last login: Wed Jun 26 10:45:20 on ttys000
siddharthmalviya@practos-MacBook-Pro ~ % ls
Applications  Desktop      Downloads   Movies      Pictures
ChatCord     Documents   Library     Music       Public
siddharthmalviya@practos-MacBook-Pro ~ % git config --global user.name "siddharthmalviya123"
siddharthmalviya@practos-MacBook-Pro ~ % git config --global user.email "siddharth.malviya@practo.com"
siddharthmalviya@practos-MacBook-Pro ~ %
```

2. Create a new branch named "feature-branch" and switch to it.

```
siddharthmalviya@practos-MacBook-Pro assignment % ls
siddharthmalviya@practos-MacBook-Pro assignment % ls
Screenshot 2024-06-26 at 1.02.21 PM.png
siddharthmalviya@practos-MacBook-Pro assignment % git init
Initialized empty Git repository in /Users/siddharthmalviya/Desktop/assignment/.
git/
siddharthmalviya@practos-MacBook-Pro assignment % git checkout -b feature-branch

Switched to a new branch 'feature-branch'
siddharthmalviya@practos-MacBook-Pro assignment %
```

git checkout -b feature-branch

3. List all branches in the repository.

git branch

```
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
* main
siddharthmalviya@practos-MacBook-Pro assignment_p %
```

4. Delete the branch "feature-branch"

git branch -d feature-branch

```
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
* main
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch "feature-branch"
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
feature-branch
* main
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch -d feature-branch
Deleted branch feature-branch (was f43f034).
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
* main
siddharthmalviya@practos-MacBook-Pro assignment_p %
```

5. How do you undo the last commit

git revert HEAD

```
Last login: Wed Jun 26 13:27:00 on ttyss00
[siddharthmalviya@practos-MacBook-Pro ~ % ls
Applications  Documents      Movies          Public
ChatCord     Downloads     Music          assignment_p
Desktop      Library       Pictures
[siddharthmalviya@practos-MacBook-Pro ~ % cd assignment_p
[siddharthmalviya@practos-MacBook-Pro assignment_p % git init
Reinitialized existing Git repository in /Users/siddharthmalviya/assignment_p/.git/
[siddharthmalviya@practos-MacBook-Pro assignment_p % git add .
[siddharthmalviya@practos-MacBook-Pro assignment_p % git log
commit f43f03404e4544a5369269a6152ac23f0acc8960 (HEAD -> main)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date:   Wed Jun 26 13:43:00 2024 +0530
```

second

```
commit 7ae70140855a0866e08f5fcb1e79cb1a8e0d835
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date:   Wed Jun 26 13:37:45 2024 +0530
```

initial code

```
siddharthmalviya@practos-MacBook-Pro assignment_p % █
```

Revert "second"

This reverts commit f43f03404e4544a5369269a6152ac23f0acc8960.

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
```

```
#
# On branch main
# Changes to be committed:
#   deleted:    second.js
#
```

```
~
~
~
~
~
~
~
~
~
~
```

"~/assignment_p/.git/COMMIT_EDITMSG" 11L, 287B

6. Create a new branch names "conflict-branch"
git checkout -b conflict-branch

7. Create a another branch named "feature1"
git checkout -b feature1

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch "conflict-branch"
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
conflict-branch
* main
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch "feature 1"
fatal: 'feature 1' is not a valid branch name
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch "feature1"
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
conflict-branch
feature1
* main
[siddharthmalviya@practos-MacBook-Pro assignment_p % git log --oneline
f43f034 (HEAD -> main, feature1, conflict-branch) second
7ae7014 initial code
siddharthmalviya@practos-MacBook-Pro assignment_p % █
```

8. Make some changes in to feature1 branch

```
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
conflict-branch
* feature1
main
siddharthmalviya@practos-MacBook-Pro assignment_p % git log --oneline
f43f034 (HEAD -> feature1, main, conflict-branch) second
7ae7014 initial code
siddharthmalviya@practos-MacBook-Pro assignment_p % git add .
siddharthmalviya@practos-MacBook-Pro assignment_p % git commit -m "new file added"
[feature1 56ed483] new file added
1 file changed, 0 insertions(+), 0 deletions(-)
rename second.js -> newfileadded.js (100%)
siddharthmalviya@practos-MacBook-Pro assignment_p % git log
commit 56ed483ff5c9f17ba38dcb336ee1a2dfcd1764 (HEAD -> feature1)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 14:10:14 2024 +0530

    new file added

commit f43f03404e4544a5369269a6152ac23f0acc8960 (main, conflict-branch)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:43:00 2024 +0530

    second

commit 7ae70140855a086e08f5fcb1e79c1a8e0d835
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:37:45 2024 +0530

    initial code
siddharthmalviya@practos-MacBook-Pro assignment_p %
```

9. Merge “feature1” branch into main branch

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % git checkout main
Switched to branch 'main'
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
conflict-branch
feature1
* main
[siddharthmalviya@practos-MacBook-Pro assignment_p % git merge origin/feature1
merge: origin/feature1 - not something we can merge
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
conflict-branch
feature1
* main
[siddharthmalviya@practos-MacBook-Pro assignment_p % git log
commit f43f03404e4544a5369269a6152ac23f0acc8960 (HEAD -> main, conflict-branch)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:43:00 2024 +0530

    second
```

10. Make changes in “conflict-branch”, in the same file and line that you had made changes in feature1

Same above command : git checkout conflict-branch

Git add .

git commit -m “made conflicting changes”

11. Merge master into conflict-branch [Attach screenshot of terminal & file]

```
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
conflict-branch
feature1
* main
siddharthmalviya@practos-MacBook-Pro assignment_p % git checkout conflict-branch
Switched to branch 'conflict-branch'
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
* conflict-branch
feature1
main
siddharthmalviya@practos-MacBook-Pro assignment_p % git add .
siddharthmalviya@practos-MacBook-Pro assignment_p % git add .
siddharthmalviya@practos-MacBook-Pro assignment_p % git commit -m "made conflicting changes"
[conflict-branch 067db8a] made conflicting changes
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 newfileadded.js
siddharthmalviya@practos-MacBook-Pro assignment_p % git log
commit 067db8a2a9fd08e8fc5db72680e910d296c921be (HEAD -> conflict-branch)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 14:30:33 2024 +0530

made conflicting changes

commit f43f03404e4544a5369269a6152ac23f0acc8960
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:43:00 2024 +0530

second

commit 7ae70140855a0866e08f5fcb1a8e0d835
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:37:45 2024 +0530

initial code
siddharthmalviya@practos-MacBook-Pro assignment_p %
```

12. Resolve merge conflicts

It was done manually and then added

Git checkout conflict-branch

Git merge main

// conflict resolve

Git add .

Git commit -m "merger main into conflict-branch"

13. Add a remote named "origin" pointing to a GitHub repository.

```
Initial code
[siddharthmalviya@practos-MacBook-Pro assignment_p % git log --oneline
f43f034 (HEAD -> main) second
7ae7014 initial code
[siddharthmalviya@practos-MacBook-Pro assignment_p % git remote add origin https://github.com/siddharthmalviya123/Assignment-.git
[siddharthmalviya@practos-MacBook-Pro assignment_p % git remote -v
origin  https://github.com/siddharthmalviya123/Assignment-.git (fetch)
origin  https://github.com/siddharthmalviya123/Assignment-.git (push)
[siddharthmalviya@practos-MacBook-Pro assignment_p % █
```

14. Fork a repository on GitHub and clone it to your local machine.

```
Last login: Wed Jun 26 13:48:14 on ttys000
[siddharthmalviya@practos-MacBook-Pro ~ % ls
Applications  Desktop      Downloads    Movies       Pictures      assignment_p
ChatCord     Documents    Library      Music        Public
[siddharthmalviya@practos-MacBook-Pro ~ % cd assignment_p
[siddharthmalviya@practos-MacBook-Pro assignment_p % git init
Reinitialized existing Git repository in /Users/siddharthmalviya/assignment_p/.git/
[siddharthmalviya@practos-MacBook-Pro assignment_p % git clone https://github.com/siddharthmalviya123/wheather-tracker-WebApp.git
Cloning into 'wheather-tracker-WebApp'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
[siddharthmalviya@practos-MacBook-Pro assignment_p % ls
index.js          newfileadded.js  wheather-tracker-WebApp
[siddharthmalviya@practos-MacBook-Pro assignment_p % █
```


15. Create a new branch on your fork, make changes, and open a pull request to the original repository.

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % ls
index.js          newfileadded.js  wheather-tracker-WebApp
[siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
* main
[siddharthmalviya@practos-MacBook-Pro assignment_p % git checkout -b newbranch
Switched to a new branch 'newbranch'
[siddharthmalviya@practos-MacBook-Pro assignment_p % git add .
[siddharthmalviya@practos-MacBook-Pro assignment_p % git commit -m "made change in newbranch"
[newbranch 42cc4ff] made change in newbranch
2 files changed, 1 insertion(+)
 create mode 100644 .DS_Store
 create mode 160000 wheather-tracker-WebApp
[siddharthmalviya@practos-MacBook-Pro assignment_p % git push origin newbranch
..
..
..
```

16. Comment on a PR and suggest improvements

Compare changes

Compare changes across branches, commits, tags, and more below. If you need to, you can also [compare across forks](#).

 base repository: siddharthmalviya123/wheath... ▾

base: main ▾

←

head repository: siddharthmalviya123/wheath... ▾

compare: main ▾


Choose different branches or forks above to discuss and review changes. [Learn about pull requests](#)

Create pull request



Compare and review just about anything

Branches, tags, commit ranges, and time ranges. In the same repository and across forks.

Example comparisons	
 main@{1day}...main	24 hours ago



© 2024 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#) [Manage cookies](#) [Do not share my personal information](#)

17. Create a Git alias for the command `git log --oneline` named `gitlol`.

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % git config --global alias.lol "log --oneline"]
[siddharthmalviya@practos-MacBook-Pro assignment_p % git lol]
42cc4ff (HEAD -> newbranch) made change in newbranch
56ed483 (main) new file added
f43f034 second
7ae7014 initial code
[siddharthmalviya@practos-MacBook-Pro assignment_p %]
```

18. Create a pre-commit hook

19. You have made local changes in your branch, but you need to switch to another branch urgently without committing. How would you handle this situation?

We will use stash here stash temporarily stores

```
: git stash
git checkout otherbranch
//stashed changes
git stash app
```

20. You accidentally deleted a file in your local repository. How do you restore it using Git?

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % ls]
index.js          newfileadded.js    wheather-tracker-WebApp
[siddharthmalviya@practos-MacBook-Pro assignment_p % ls]
newfileadded.js    wheather-tracker-WebApp
[siddharthmalviya@practos-MacBook-Pro assignment_p % git restore index.js]
[siddharthmalviya@practos-MacBook-Pro assignment_p % ls]
index.js          newfileadded.js    wheather-tracker-WebApp
[siddharthmalviya@practos-MacBook-Pro assignment_p %]
```

21. You have committed changes to your branch but forgot to include a file. How do you add the file to the last commit without creating a new commit?

We use `--amend` here to changes in the last recent comment

For eg.

Git add filewhichforgot

Git commit `--amend`

22. You want to discard all changes in your working directory and revert to the last commit.

What Git command would you use?

`git reset - - hard`

This will discard all changes in working directory and resets to last commit

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % git log
commit 42cc4ffc1ef1f64be8f7b0f97d7600015d006fad (HEAD -> newbranch)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 18:41:47 2024 +0530
```

made change in newbranch

```
commit 56ed4833ff5c9f17ba38dcbb336ee1a2dfcd1754 (main)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 14:10:14 2024 +0530
```

new file added

```
commit f43f03404e4544a5369269a6152ac23f0acc8960
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:43:00 2024 +0530
```

second

```
commit 7ae70140855a0866e08f5fcb1a8e0d835
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:37:45 2024 +0530
```

initial code

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % git reset --hard
HEAD is now at 42cc4ff made change in newbranch
```

23. You need to view a specific commit's changes. What Git command can be used to show the changes introduced by a particular commit?

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % git show f43f03404e4544a5369269a6152ac23f0acc8960
commit f43f03404e4544a5369269a6152ac23f0acc8960
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:43:00 2024 +0530
```

second

```
diff --git a/second.js b/second.js
new file mode 100644
index 0000000..e69de29
```

I used `git show <commit id>` here

24. You want to change a commit message, after you have already committed, how do you do so?

We use `git commit —amend`

After that the text editor configure for git and edit commit msg

25. Your colleague has made changes in their branch, and you want to incorporate those changes into your branch without merging. How do you achieve this?

By using `stash`

Or

`git fetch origin`

`Git cherry-pick commit-hash`

26. You've made several commits on a branch, but you want to club them into a single commit before pushing to the remote repository. How would you do that?

27. You accidentally staged a file that you don't want to commit. How do you unstage it?

`Git reset <filename>`

When you stage a file using `git add`, Git takes a snapshot of the file's current state and adds it to the staging area. This means the file is marked to be included in the next commit. If you decide that you do not want to include this file in your next commit, you can unstage it. Unstaging a file removes it from the staging area, but the file itself remains unchanged in your working directory

28. You don't want to commit files that have `.yml` in the end, and also files inside folder `config`. How do you do that?

Add those file in `.gitignore` file this file will not commit afterthat

29. You want to see a list of all the files changed in the last commit. What Git command would you use?

`Git diff`

30. You realize that your local branch is outdated, and you want to fetch the latest changes from the remote repository. How do you do this without merging?

We use `git fetch` here

Which only updates remote-tracking branches and reflect changes without automatically merging anything into your current branch

31. You accidentally deleted a branch. How do you recover it?

Git maintains a `reflog` file from there we recover it using

`git checkout -b branchname branchid`


```
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
* main
siddharthmalviya@practos-MacBook-Pro assignment_p % git reflog
56ed483 (HEAD -> main) HEAD@{0}: checkout: moving from newbranch to main
599996f HEAD@{1}: commit (amend): dsafsd
42cc4ff HEAD@{2}: reset: moving to HEAD
42cc4ff HEAD@{3}: commit: made change in newbranch
56ed483 (HEAD -> main) HEAD@{4}: checkout: moving from main to newbranch
56ed483 (HEAD -> main) HEAD@{5}: checkout: moving from conflict-branch to main
067db8a HEAD@{6}: checkout: moving from conflict-branch to conflict-branch
067db8a HEAD@{7}: commit: made conflicting changes
f43f034 HEAD@{8}: checkout: moving from main to conflict-branch
56ed483 (HEAD -> main) HEAD@{9}: merge feature1: Fast-forward
f43f034 HEAD@{10}: checkout: moving from feature1 to main
56ed483 (HEAD -> main) HEAD@{11}: checkout: moving from main to feature1
f43f034 HEAD@{12}: checkout: moving from feature1 to main
56ed483 (HEAD -> main) HEAD@{13}: commit: new file added
f43f034 HEAD@{14}: checkout: moving from main to feature1
f43f034 HEAD@{15}: commit: second
7ae7014 HEAD@{16}: commit (initial): initial code
siddharthmalviya@practos-MacBook-Pro assignment_p % git checkout -b newbranch
Switched to a new branch 'newbranch'
siddharthmalviya@practos-MacBook-Pro assignment_p % git branch
main
* newbranch
siddharthmalviya@practos-MacBook-Pro assignment_p % █
```

32. You want to remove untracked files and directories from your working directory. What Git command would you use?

git clean -fd

Here f for force and d for directory

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % git clean -fd
[siddharthmalviya@practos-MacBook-Pro assignment_p % git log
commit 56ed4833ff5c9f17ba38dcbb336ee1a2dfcd1754 (HEAD -> newbranch, main)
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 14:10:14 2024 +0530
```

new file added

```
commit f43f03404e4544a5369269a6152ac23f0acc8960
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:43:00 2024 +0530
```

second

```
commit 7ae70140855a0866e08f5fcba1e79cb1a8e0d835
Author: siddharthmalviya123 <siddharth.malviya@practo.com>
Date: Wed Jun 26 13:37:45 2024 +0530
```

initial code

```
[siddharthmalviya@practos-MacBook-Pro assignment_p % ls
index.js          newfileadded.js      wheather-tracker-WebApp
siddharthmalviya@practos-MacBook-Pro assignment_p % █
```

33. You have a commit from a feature branch that you want to apply to the main branch without merging the entire feature branch.

git checkout main

Git cherry-pick commitid

34. You mistakenly committed a change to the wrong branch and need to apply that commit to the correct branch.

Git log

Git checkout correctbranch

Git cherry-pick hashofwrong

Conflict resolution if any

Git push origin correctbranch

35. There is a series of commits on a feature branch, but you only want to cherry-pick a specific range of commits.

Git cherry-pick starcommithash^..endcommithash

36. You want to clone a GitHub repository onto your local machine, but you only need a specific branch. How can you achieve this?

Git clone --branch branch-name --single-branch <https://github.com/siddharthmalviya123/chatcord.git>

37. You've made changes to your local repository and want to push them to your fork on GitHub. What Git commands would you use?

Git add .

Git commit -m "commit message"

Git push origin branchname

38. You want to create a new branch both locally and on GitHub to work on a new feature. What commands would you use?

Git checkout -b newfeaturebranch

Git push -u origin newfeaturebranch

here,

-u → upstream

39. You want to see the commit history of a GitHub repository. How can you do this using Git commands?

Git log

40. You've accidentally committed sensitive information and want to remove the commit from both your local and remote repositories on GitHub. What commands would you use?

Git log

Git rebase -i HEAD~n (where n is the number of commit to go back)

// After that command opens an interactive rebase editor where we can choose to edit , delete , // or squash commits change pick to edit and save file

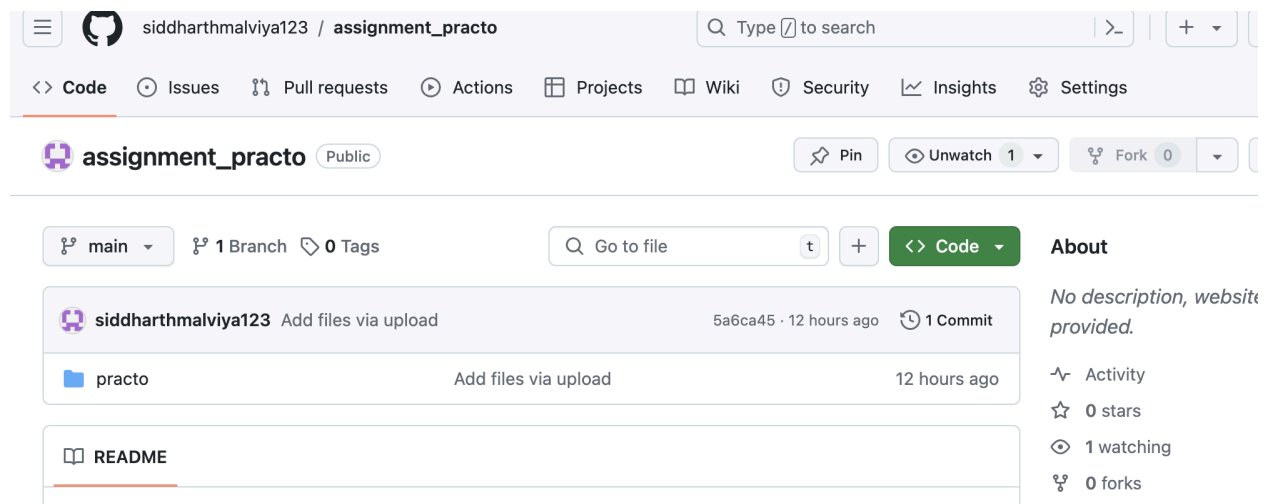
Pick commithash

41. You want to delete a remote branch on GitHub. What Git command would you use?

Git push origin -delete branchname

42. Create a git repository for all your assignments and upload them in it. Ask your peers to code review it, and you need to code review your peers assignments

```
[siddharthmalviya@practos-MacBook-Pro Desktop % cd practo
[siddharthmalviya@practos-MacBook-Pro practo % ls
github assinment.txt
[siddharthmalviya@practos-MacBook-Pro practo % git init
Initialized empty Git repository in /Users/siddharthmalviya/Desktop/practo/.git/
[siddharthmalviya@practos-MacBook-Pro practo % git add .
[siddharthmalviya@practos-MacBook-Pro practo % git commit -m "assignment added"
[main (root-commit) d4a9469] assignment added
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 github assinment.txt
[siddharthmalviya@practos-MacBook-Pro practo % git remote add origin https://github.com/siddharthmalviya123/assignment_practo.git
[siddharthmalviya@practos-MacBook-Pro practo % git push -u origin main
```



43. Create a pull request on any open source library on github, attach the pull request link to the readme file of this project's repository

First step to clone

Git clone <https://github.com/siddharth123/chatcord.git>

Cd chatcord

//make readme file

Git add .

Git commit -m "readme added"

Git push origin main

Now after that we will go to repository and click on pull requests -> new pull request ->select base repo (original repo) and base branch (main or master)

->create pull request