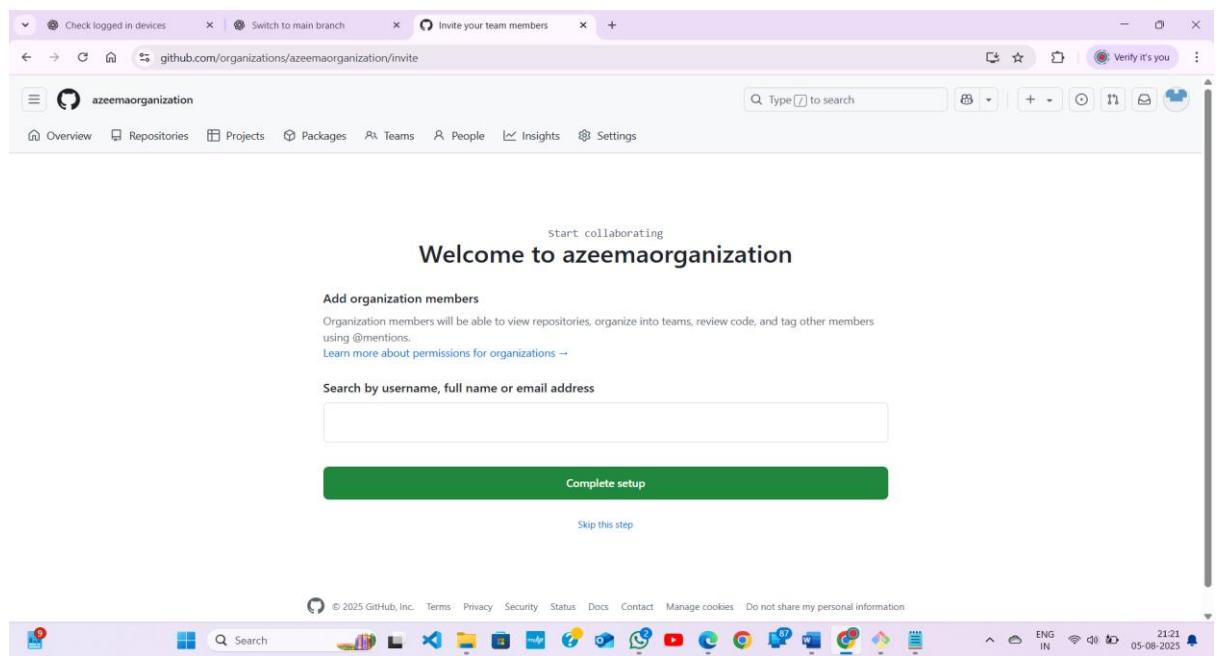


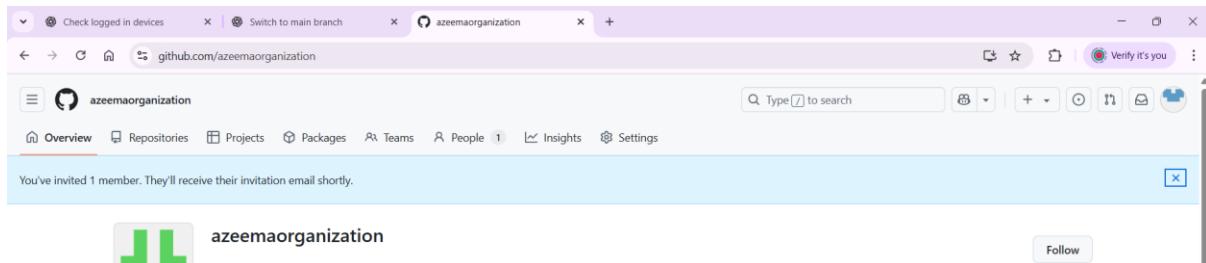
Week 3

Task 1

Collaborative coding.

1. Facilitating Collaborative Work by creating Organization





The screenshot shows the GitHub organization overview for 'azeemaorganization'. The page includes a header with tabs for Overview, Repositories, Projects, Packages, Teams, People (1), Insights, and Settings. A message at the top says 'You've invited 1 member. They'll receive their invitation email shortly.' Below this is a profile card for 'azeemaorganization' with a green logo icon. To the right is a 'Follow' button. A sidebar on the right contains sections for 'View as: Public', 'Discussions', and 'Repositories'.

We think you're gonna like it here.

We've suggested some tasks here in your organization's overview to help you get started.

Invite your people

Invite your first member
Find people by their GitHub username or email address.

Customize members' permissions
Set everyone's base permissions for your code.

Collaborative coding

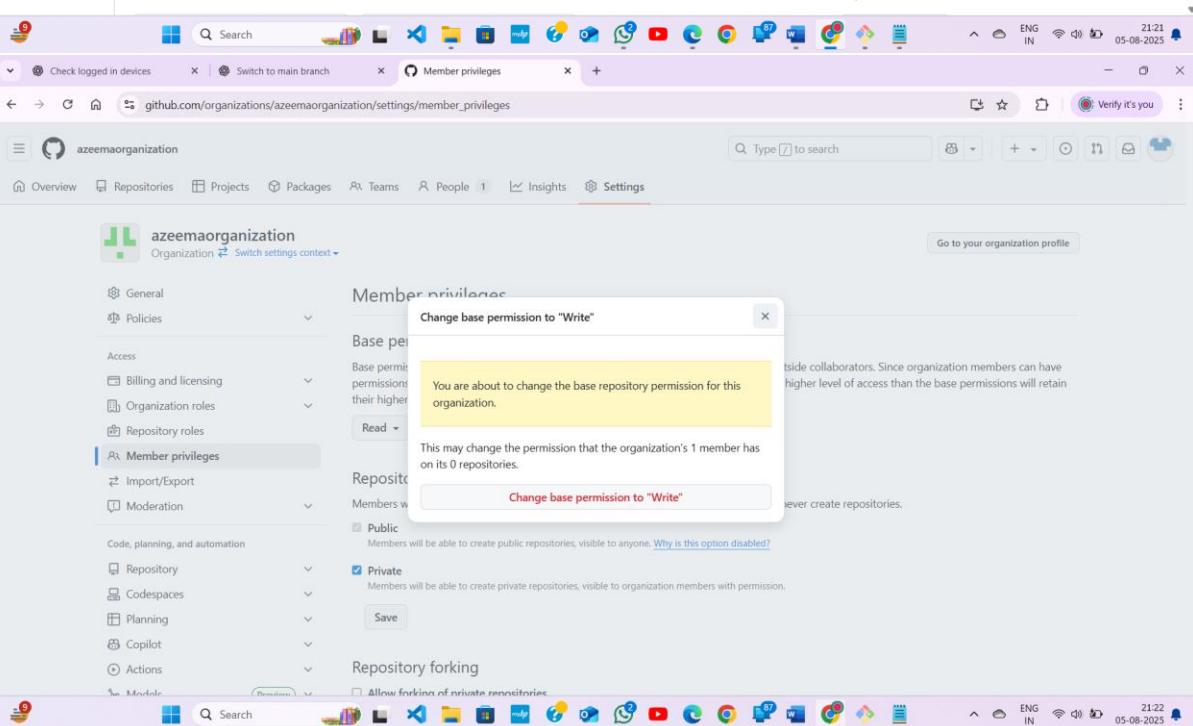
See more about collaborative coding →

Repositories

Discussions

Set up discussions to engage with your community!

[Turn on discussions](#)



The screenshot shows the 'Member privileges' section of the GitHub organization settings for 'azeemaorganization'. The left sidebar lists 'General', 'Policies', 'Access' (with 'Billing and licensing', 'Organization roles', and 'Repository roles'), 'Member privileges' (which is selected and highlighted in blue), 'Import/Export', 'Moderation', 'Code, planning, and automation' (with 'Repository', 'Codespaces', 'Planning', 'Copilot', and 'Actions'), and 'Repository forking'. The main content area is titled 'Member privileges' and shows a modal dialog titled 'Change base permission to "Write"'. The dialog contains a yellow warning box stating 'You are about to change the base repository permission for this organization.' It also notes 'This may change the permission that the organization's 1 member has on its 0 repositories.' At the bottom of the dialog is a red 'Change base permission to "Write"' button. Below the dialog, there are sections for 'Public' (disabled) and 'Private' (selected). A 'Save' button is at the bottom. A note at the bottom of the page says 'Allow creation of private repositories' with a checkbox. The status bar at the bottom shows '21:21 05-08-2025'.

The screenshot shows a Windows desktop environment with two main windows open:

- Top Window (GitHub Repository Settings):** A Microsoft Edge browser window displaying the GitHub settings for the repository "orgazeema". The "General" tab is selected. Key settings shown include:
 - Repository name:** orgazeema
 - Template repository:** A checkbox option is present.
 - Require contributors to sign off on web-based commits:** A checkbox option is present.
 - Default branch:** Set to "main".
- Bottom Window (Terminal):** A terminal window titled "MINGW64:/c/Users/azeem/Desktop/orgazeema" showing the command-line process of initializing a Git repository, committing files, adding a remote origin, and pushing changes to GitHub. The terminal output is as follows:

```
azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (main)
$ ls
textzeema.txt textkarunya.txt

azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (main)
$ git init
Initialized empty Git repository in C:/Users/azeem/OneDrive/Desktop/orgazeema/.git/
azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (master)
$ git add .

azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (master)
$ git commit -m "First commit"
[master (root-commit) 27641bc] first commit
 2 files changed, 2 insertions(+)
 create mode 100644 textzeema.txt
 create mode 100644 textkarunya.txt
azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (master)
$ git branch -M main
azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (main)
$ git remote add origin https://github.com/azeemaorganization/orgazeema.git
  * [new branch]  main > main
branch 'main' set up to track 'origin/main'.
azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (main)
$ git push -u origin main
Branch 'main' set up to track 'origin/main'.
Everything up-to-date

azeem@APTOP-MUV04INQ MINGW64 ~/OneDrive/Desktop/orgazeema (main)
$ |
```

Check logged in devices | Switch to main branch | Manage access

github.com/azeemaorganization/orgazeema/settings/access

azeemaorganization / orgazeema

Type / to search

Code Issues Pull requests Actions Projects Security Insights Settings

karunya1588 has been added as a collaborator with write permissions on the repository.

General Collaborators and teams

Access

Private repository Only those with access to this repository can view it

Manage visibility

Code and automation

- Branches
- Tags
- Rules
- Actions
- Models
- Webhooks
- Copilot
- Environments
- Pages
- Custom properties

Base role Write

All 2 members can access this repository.

Manage

Direct access Rx

1 entity has access to this repository. 1 member.

Organization access

0 users and 0 teams can access this repository through the organization.

Manage

Manage access

Create team Add people Add teams

Direct access Organization access

Select all Find people or a team... Type Role

karunya1588 Role: write

Search

ENG IN 21:26 05-08-2025

```
MINGW64/c/Users/azeem/Desktop/Desktop/orgazeema/new-organization
$ git init
Reinitialized existing Git repository in C:/Users/azeem/OneDrive/Desktop/orgazeema/.git/
$ git config --global user.name "azeemafirdous"
$ git config --global user.email "azeemafirdous48@gmail.com"
$ git remote -v
origin https://github.com/azeemaorganization/orgazeema.git (fetch)
origin https://github.com/azeemaorganization/orgazeema.git (push)
$ git clone https://github.com/azeemaorganization1588/new-organization.git
Cloning into 'new-organization...'...
remote: Enumerating objects: 1000 (4/4), done.
remote: Counting objects: 1000 (4/4), done.
remote: Compressing objects: 1000 (4/4), done.
remote: Total 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 1000 (4/4), done.
$ cd new-organization
$ git checkout -b feature/azemeafeature
Switched to a new branch 'feature/azemeafeature'
$ git add .
$ git commit -m "Changed the content in the text files"
[feature/azemeafeature 0430698] I changed the content in the text files
 2 files changed, 2 insertions(+), 2 deletions(-)
$ git push
Enumerating objects: 7, done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 409 bytes | 204.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Create a pull request for 'feature/azemeafeature' on GitHub by visiting:
remote: https://github.com/azeemaorganization1588/new-organization/pull/new/feature
remote: azemeafeature
remote: To https://github.com/azeemaorganization1588/new-organization.git
 * [new branch]  feature/azemeafeature > feature/azemeafeature
$ git push --set-upstream https://github.com/azeemaorganization1588/new-organization feature/azemeafeature

```

Search

ENG IN 21:55 05-08-2025

```

MINGW64:/c/Users/baman/OneDrive/Desktop/gitlab2
baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (main)
$ git init
Reinitialized existing Git repository in C:/Users/baman/OneDrive/Desktop/gitlab2/.git/
baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (main)
$ git fetch origin
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (4/4), 389 bytes | 10.00 KiB/s, done.
From https://github.com/new-organization1588/new-organization
 * [new branch]      feature/azeemafeature -> origin/feature/azeemafeature

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (main)
$ git checkout feature/azeemafeature
branch 'feature/azeemafeature' set up to track 'origin/feature/azeemafeature'.
Switched to a new branch 'feature/azeemafeature'

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (feature/azeemafeature)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (main)
$ git pull origin main
From https://github.com/new-organization1588/new-organization
 * branch            main      -> FETCH_HEAD
Already up to date.

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (main)
$ git merge feature/azeemafeature
Updating 6095626..0430698
Fast-forward
 text1.txt | 2 +-
 text2.txt | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (main)
$ git push origin main
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/new-organization1588/new-organization.git
 6095626..0430698 main -> main

baman@karunya MINGW64 ~/OneDrive/Desktop/gitlab2 (main)
$
```

The screenshot shows a GitHub repository page for 'new-organization1588 / new-organization'. The repository is private. It displays the following information:

- Branches:** main (1 Branch)
- Commits:** karunya1588 first commit (6095626 - 47 minutes ago, 1 Commit)
- Files:** text1.txt, text2.txt (both first commit, 47 minutes ago)
- Activity:** 0 stars, 0 forks, 0 watching
- Releases:** No releases published. Create a new release.
- Packages:** No packages published. Publish your first package.

Exercise on Fork

Screenshot of the GitHub fork creation process:

The user is creating a new fork of the repository `new-organization1588/new-organization`. The repository URL is `github.com/new-organization1588/new-organization/fork`.

Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

Required fields are marked with an asterisk (*).

Owner *: azeemafirdouss

Repository name *: karfork

Description (optional): (empty)

Copy the `main` branch only: (checkbox checked)

Contribute back to `new-organization1588/new-organization` by adding your own branch. [Learn more](#).

Note: You are creating a fork in your personal account.

Create fork button

Screenshot of the GitHub repository page for `karfork`:

The repository was created from `new-organization1588/new-organization`.

Code tab selected.

Branches: main (selected), 1 Branch, 0 Tags

Commits: 2 Commits (by azeemafirdouss)

Files: text1.txt, text2.txt (both updated 50 minutes ago)

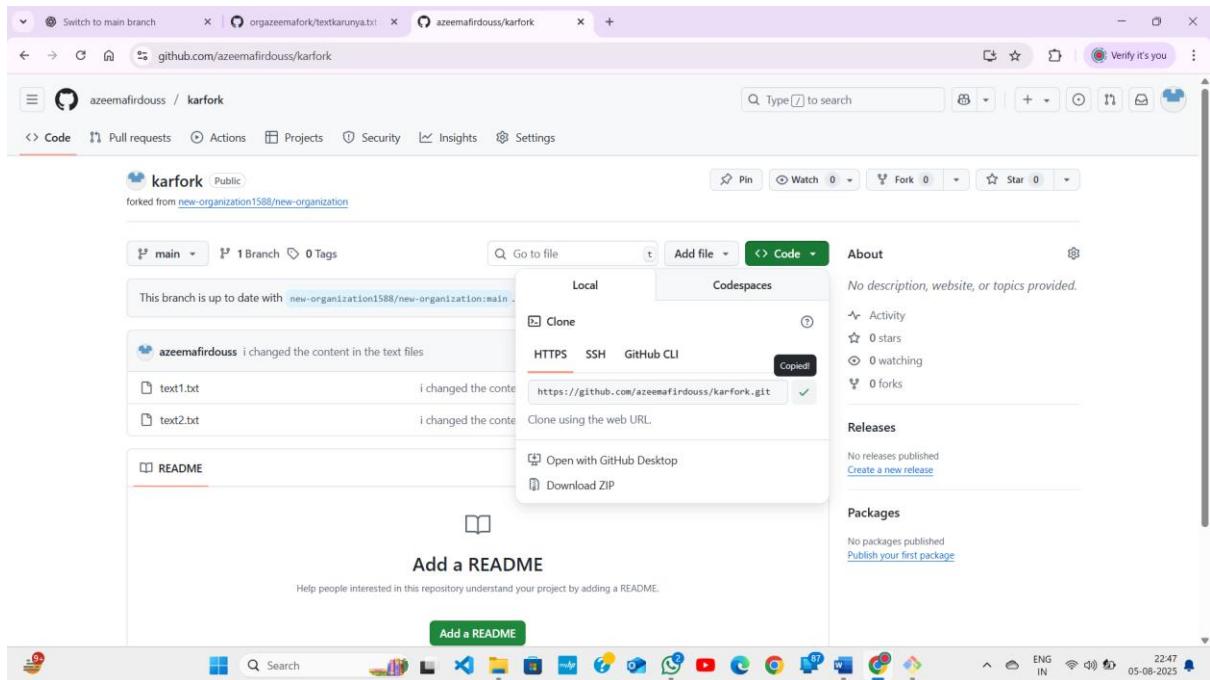
About: No description, website, or topics provided.

Activity: 0 stars, 0 watching, 0 forks

Releases: No releases published. [Create a new release](#)

Packages: No packages published. [Publish your first package](#)

README: Add a README



```
MINGW64:/c/Users/azeem/OneDrive/Desktop/orgazema/karfork
bash: warning: command substitution: ignored null byte in input
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema (main)
$ git init
Initialized existing Git repository in C:/Users/azeem/OneDrive/Desktop/orgazema/.git/
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema (main)
$ git clone https://github.com/azeemafirdous/karfork.git
Cloning into 'karfork'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 8, done.
remote: Compressing objects: 100%, done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0 (delta 0)
remote: 
remote: Receiving objects: 100% (8/8), done.
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema (main)
$ cd karfork
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema/karfork
$ (main)
$ git add .
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema/karfork
$ git commit -m "Fork created with name karfork"
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema/karfork
$ git add .
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema/karfork
$ (main)
$ git commit -m "Fork created with name karfork"
[main 0xdead0] Fork created with name karfork
2 files changed, 2 insertions(+), 0 deletions(-)
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema/karfork
$ (main)
$ git push origin main
Enumerating objects: 7, done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 43.00 KiB | 43.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object
To https://github.com/azeemafirdous/karfork.git
 043698...cdebd9  main > main
azeem@APTOP-MUVDAHQ: MINGW64 ~/OneDrive/Desktop/orgazema/karfork
$ (main)
```

The screenshot shows a Windows desktop environment with a taskbar at the bottom. The taskbar icons include Start, Search, File Explorer, Task View, Edge, and other system icons. The system tray shows the date as 05-08-2025 and the time as 22:50. The desktop background is white.

The main window is a web browser displaying a GitHub pull request page for the repository 'karfork' owned by 'azeemafirdous'. The page title is 'Pull requests · azeemafirdous/karfork'. The navigation bar includes links for Code, Pull requests (which is underlined), Actions, Projects, Security, Insights, and Settings. A search bar is located above the pull request list. Below the search bar, there are filters for 'is:pr is:open' and buttons for 'Labels' and 'Milestones'. A prominent button labeled 'New pull request' is visible.

The content area of the page features a 'Welcome to pull requests!' message with a small icon of two hands shaking. Below this, a note says: 'Pull requests help you collaborate on code with other people. As pull requests are created, they'll appear here in a searchable and filterable list. To get started, you should [create a pull request](#).'

At the bottom of the page, there is a footer with the GitHub logo and the text: '© 2025 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information'.



Switch to main branch orgazeemafork/textkarunya.txt Comparing new-organization1... +

github.com/new-organization1588/new-organization/compare/main...azeemafirdous:karforkmain

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. [Learn more about diff comparisons here.](#)

base repository: new-organization1588/new-organization | base: main | head repository: azeemafirdous/karfork | compare: main

Add a title

pullreqfork

Add a description

Write Preview

Add your description here...

Markdown is supported

Paste, drop, or click to add files

Allow edits by maintainers [?](#) 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

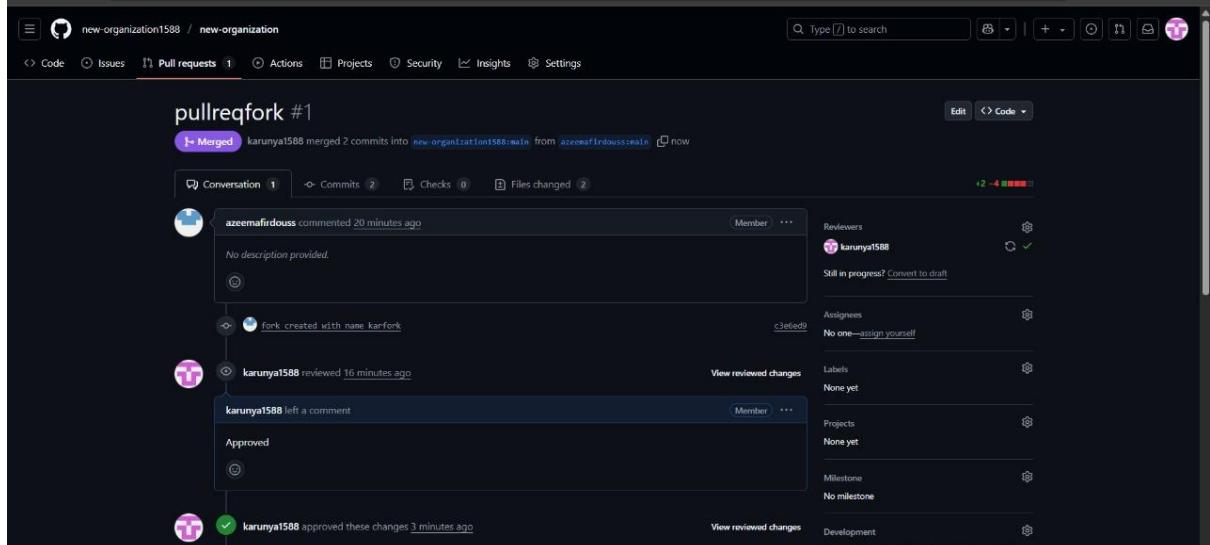
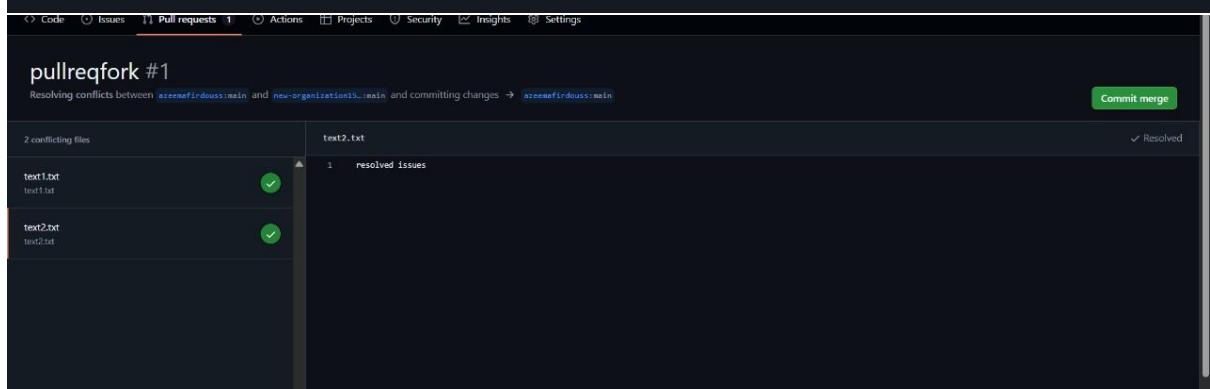
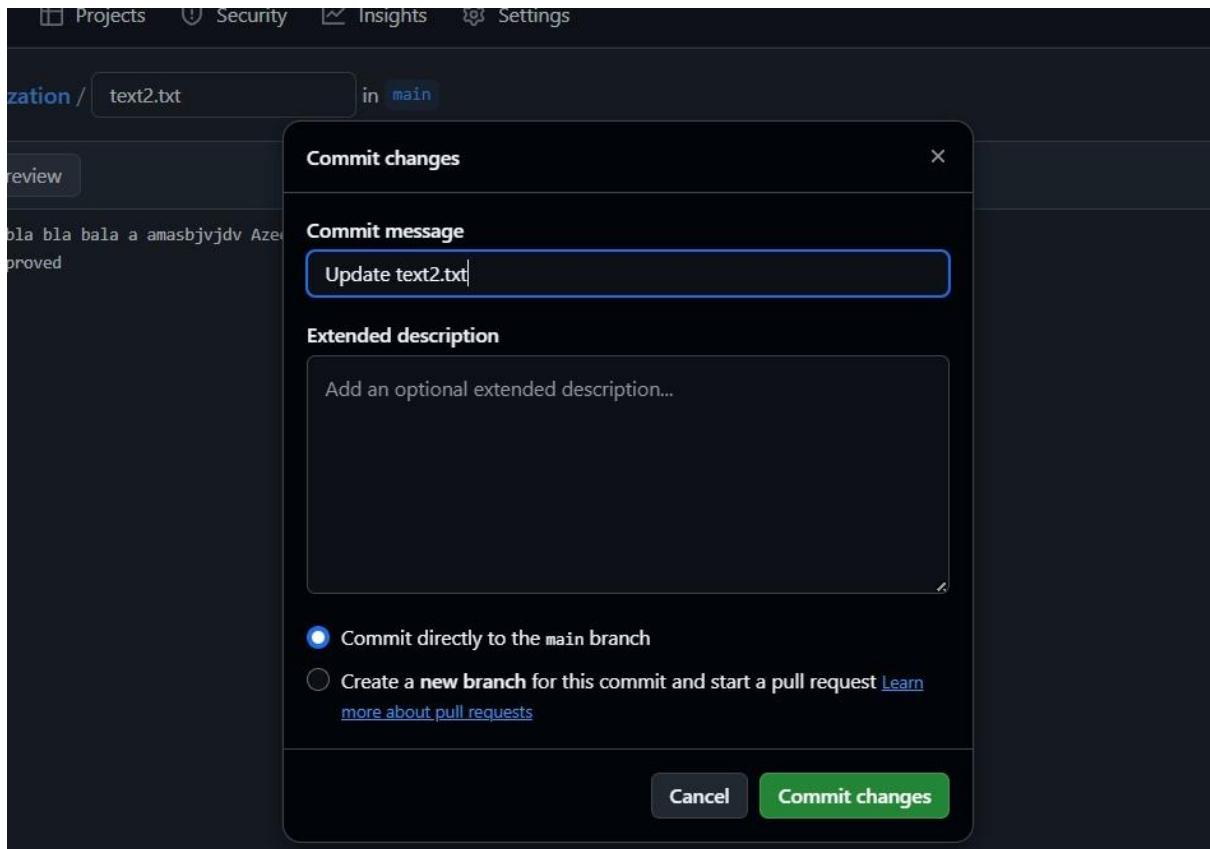
Development: Use [Closing keywords](#) in the description to automatically close issues

Helpful resources: GitHub Community Guidelines

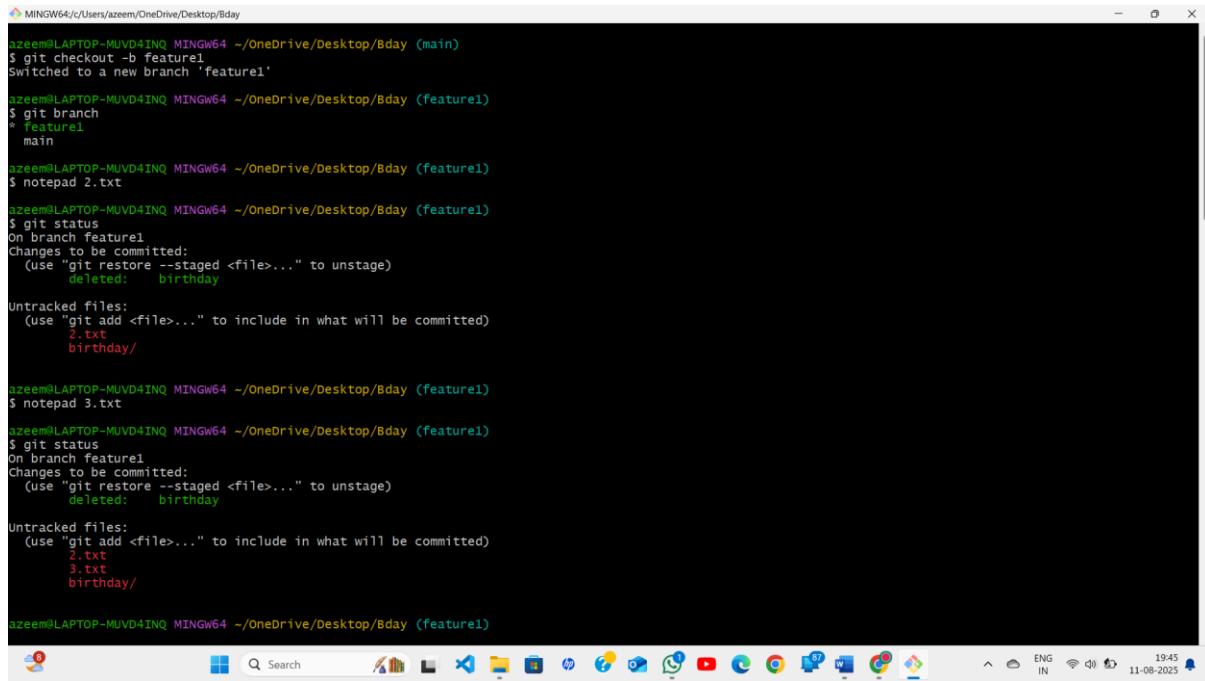
Screenshot of a GitHub repository page for 'new-organization'. The repository has 2 branches and 0 tags. A message says 'Your main branch isn't protected' with a 'Protect this branch' button. The 'Code' tab is selected. The 'main' branch shows two commits from 'azeemafirdous' changing 'text1.txt' and 'text2.txt'. A 'README' file is present but empty. The 'About' section shows no description, website, or topics. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published. The 'Contributors' section lists 'azeemafirdous' and 'azeemafirdous04'.

Screenshot of a GitHub pull request page for 'pullreqfork #1'. It shows a merge from 'azeemafirdous' into 'new-organization1588/main'. The 'Files changed' tab is selected, showing changes to 'text1.txt' and 'text2.txt'. The changes show 'azeemafirdous' changing the text in both files. The 'Conversation' tab shows a single message from 'azeemafirdous' asking for review. The bottom of the page includes standard GitHub footer links.

Screenshot of a GitHub file view for 'text1.txt' in the 'new-organization' repository. The file shows two lines of code: 'bla bla bla Azeema firdous I changed the text in this text1' and '//Approved by karunya'. The 'Code' tab is selected, showing the blame history with 'karunya1588' as the author. The 'Blame' tab shows the same two lines of code.



3. To create and apply patch



The screenshot shows a Windows terminal window with a black background and white text. It displays a series of git commands being run in a directory named 'Bday' under 'OneDrive/Desktop'. The session starts with checking out a new branch 'feature1' from the 'main' branch. It then lists the current branch as 'feature1'. The user runs 'git status' which shows changes to be committed, specifically a file named 'birthday' that has been deleted. The terminal then lists untracked files: '2.txt', '3.txt', and 'birthday/'. Finally, the user runs 'git add' on these files. The taskbar at the bottom of the screen shows various pinned icons, including Microsoft Edge, File Explorer, and the Start button.

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (main)
$ git checkout -b feature1
switched to a new branch 'feature1'

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git branch
* feature1
  main

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ notepad 2.txt

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   birthday

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    2.txt
    3.txt
    birthday/

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ notepad 3.txt

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:   birthday

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    2.txt
    3.txt
    birthday/

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
```

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git add .
warning: adding embedded git repository: birthday
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> birthday
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached birthday
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  2.txt
    new file:  3.txt

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git commit -m "update"
[feature1 2ed6d8] update
 2 files changed, 1 insertion(+)
 create mode 100644 2.txt
 create mode 100644 3.txt
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git log
commit 2ed6d8e8ccb6c68420b4692da7757cdc51e6c548 (HEAD -> feature1)
Author: azeemafirdous <azeemafirdous14@gmail.com>
Date:   Mon Aug 11 19:38:44 2025 +0530

    update

commit 1b085b98d01a4a34d641ab294d599573fc81df9c (bd/main, bd/HEAD, main)

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git log
commit 2ed6d8e8ccb6c68420b4692da7757cdc51e6c548 (HEAD -> feature1)
Author: azeemafirdous <azeemafirdous14@gmail.com>
Date:   Mon Aug 11 19:38:44 2025 +0530

    update

commit 1b085b98d01a4a34d641ab294d599573fc81df9c (bd/main, bd/HEAD, main)
Author: azeemafirdous <azeemafirdous14@gmail.com>
Date:   Sun Aug 10 14:05:02 2025 +0530

    first commit

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch -1 abc1234
fatal: ambiguous argument 'abc1234': unknown revision or path not in the working
tree.
Use '-' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch -1 abc1234
git: 'format--patch' is not a git command. See 'git --help'.

The most similar command is
  format-patch

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch -1 abc1234
git: 'format' is not a git command. See 'git --help'.

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch -1 abc1234
fatal: ambiguous argument 'abc1234': unknown revision or path not in the working
tree.
Use '-' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch -1 2.txt
0001-update.patch

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
```

```

MINGW64/c/Users/azeem/OneDrive/Desktop/Bday
$ git format--patch -1 abc1234
git: 'format--patch' is not a git command. See 'git --help'.

The most similar command is
    format-patch

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format--patch -1 abc1234
git: 'format' is not a git command. See 'git --help'.

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch -1 abc1234
fatal: ambiguous argument 'abc1234': unknown revision or path not in the working
tree.
Use '-' to separate paths from revisions, like this:
'git <commands> [<revision>...] -- [<file>...]''

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch -1 2.txt
0001-update.patch

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ ls -a
./ ... .git/ 0001-update.patch 2.txt 3.txt birthday/

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch <2.txt>..HEAD

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git format-patch <0001-update.patch>..HEAD

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)
$ git am 0001-update.patch
Applying: update
error: 2.txt: already exists in index
Patch failed at 0001 update
hint: Use 'git am --show-current-patch=diff' to see the failed patch
hint: When you have resolved this problem, run "git am --continue".
hint: If you prefer to skip this patch, run "git am --skip" instead.
hint: To restore the original branch and stop patching, run "git am --abort".
hint: Disable this message with "git config set advice.mergeConflict false"

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/Bday (feature1)AM 1/1
$ |

```

Scenario based questions on working with remote repository

1. You've cloned a repository and made some changes to a local branch. Now you want to push these changes to the remote repository, but you're getting an error saying "rejected - non-fast-forward." How would you resolve this?

```

MINGW64/c/Users/azeem/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema
no changes added to commit (use "git add" and/or "git commit -a")

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git add .
warning: in the working copy of 'OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/0001-update.patch'
h', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/secretKey.txt',
LF will be replaced by CRLF the next time Git touches it

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git commit -m "Local change causing non-fast-forward"
[Feature e01a418] Local change causing non-fast-forward
14 files changed, 84 insertions(+)
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/.gitignore
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/.gitrect
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/0001-update.patch
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/1.txt
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/README.md
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/file1.txt
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/footer.txt
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/new.txt
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/signup.py
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/signup1.py
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/secret.txt
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/secretKey.txt
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/see.txt
create mode 100644 OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema/signup.html

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git push origin feature
Enumerating objects: 24, done.
Counting objects: 100% (24/24), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (24/24), 2.22 Kib | 122.00 Kib/s, done.
To https://github.com/PROJECTGS49/C549KCH.git
   b599ccc..e01a418  feature -> feature

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git branch
* feature
  feature1/signup
  hasse-feature
  main
  rese
  searchFilter
  temp-test

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ |

```

2. You've been working on a feature branch, and now you need to push it to the remote repository. However, the remote repository already has a main branch. How do you push your feature branch without affecting the main branch?

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/  
git_seema (main)  
$ git branch  
  feature  
  feature1/signup  
* main  
  rese  
  searchFilter  
  
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/  
git_seema (main)  
$ git checkout feature  
Switched to branch 'feature'  
  
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/  
git_seema (feature)  
$ git push origin feature  
Everything up-to-date
```

3. You cloned a remote repository, but after a while, the repository's structure changed and new branches were added. How would you keep your local repository updated with the latest changes from the remote repository?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$ git fetch --all

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$ git branch -r
origin/HEAD -> origin/main
origin/feature
origin/hasse-feature
origin/main

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$ git checkout -b hasse-feature origin/hasse-feat
ure
branch 'hasse-feature' set up to track 'origin/ha
sse-feature'.
Switched to a new branch 'hasse-feature'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (hasse-feature)
$
```

4. A colleague has pushed some changes to the main branch, but you have local changes in the same branch. You want to pull their changes, but you want to avoid merge conflicts. What steps would you take?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$ git stash
No local changes to save

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$ git pull --rebase origin main
From https://github.com/PROJECTG549/G549KCH
 * branch           main        -> FETCH_HEAD
Already up to date.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$ git stash pop
No stash entries found.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$ git push origin main
Everything up-to-date

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/
git_seema (main)
$
```

5. You accidentally pushed a sensitive file (e.g., API keys) to the remote repository. How would you fix this situation?

```
MINGW64:/c/Users/azeem/OneDrive/Desktop/git_seema
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ nano secretKey.txt

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ git add .
warning: in the working copy of 'OneDrive/Desktop/git_seema
/secretKey.txt', LF will be replaced by CRLF the next time
Git touches it

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ git commit -m " api key"
[main cc06cf7] api key
 1 file changed, 1 insertion(+)
  create mode 100644 OneDrive/Desktop/git_seema/secretKey.tx
t

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ git push origin feature
Everything up-to-date

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ git rm --cached secretKey.txt
rm 'OneDrive/Desktop/git_seema/secretKey.txt'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ echo "secretKey.txt" >> .gitignore

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ git commit -m "Remove sensitive file and update .gitigno
re"
[main fe67664] Remove sensitive file and update .gitignore
 1 file changed, 1 deletion(-)
 delete mode 100644 onedrive/desktop/git_seema/secretKey.tx
t

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 657 bytes | 82.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from
0)
remote: Resolving deltas: 100% (1/1), completed with 1 loca
l object.
To https://github.com/PROJECTG549/G549KCH.git
 7823986..fe67664 main -> main

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$ git branch
  feature
  feature1/signup
  hasse-feature
* main
  rese
  searchFilter

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema
(main)
$
```

6. You're working on a feature branch, and your manager requests that you integrate the latest changes from main into your feature branch. What steps would you take?

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout feature1/signup
M OneDrive/Desktop/git_seema/.gitignore
Switched to branch 'feature1/signup'

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git fetch origin
$ git merge origin/main
Updating 02f3e72..fe67664
Fast-forward
OneDrive/Desktop/git_seema/.project | 11 ++++++----
OneDrive/Desktop/git_seema/l.txt | 1 +
OneDrive/Desktop/git_seema/README.md | 1 +
OneDrive/Desktop/git_seema/se.txt | 1 +
OneDrive/Desktop/git_seema/script.py | 2 ++
OneDrive/Desktop/git_seema/see.txt | 1 +
README.md | 3 +-+
7 files changed, 19 insertions(+), 1 deletion(-)
create mode 100644 OneDrive/Desktop/git_seema/.project
create mode 100644 OneDrive/Desktop/git_seema/l.txt
create mode 100644 OneDrive/Desktop/git_seema/README.md
create mode 100644 OneDrive/Desktop/git_seema/new.txt
create mode 100644 OneDrive/Desktop/git_seema/see.txt

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ nano se.txt

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git add .
warning: in the working copy of 'OneDrive/Desktop/git_seema/.gitignore', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'OneDrive/Desktop/git_seema/se.txt', LF will be replaced by CRLF the next time Git touches it

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git commit -m "update"
[feature1/signup 4e29640] update
2 files changed, 2 insertions(+)
create mode 100644 OneDrive/Desktop/git_seema/se.txt

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git merge origin/main
Already up-to-date.

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
```

7. You cloned a remote repository, but later you find that you need to push your changes to a different remote repository. How do you configure your local repository to push to this new remote?

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git remote -v
origin https://github.com/PROJECTG549/G549KCH.git (fetch)
origin https://github.com/PROJECTG549/G549KCH.git (push)

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git remote remove new-origin
error: No such remote: 'new-origin'

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git remote add origin1 https://github.com/PROJECTG549/G549KCH.git

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git push -u origin main
Branch 'main' set up to track 'origin/main'.
Everything up-to-date

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git push -u origin main
Branch 'main' set up to track 'origin/main'.
Everything up-to-date

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
```

8. After running git pull, you notice that your local branch is behind the remote branch. How would you proceed to bring your local branch up to date without losing your local changes?

```
...../...../...../Documents/  
...../...../...../Downloads - shortcut.lnk  
...../...../...../Home - shortcut.lnk  
...../...../...../Personal vault.lnk  
...../...../...../Pictures/  
...../...../...../desktop.ini  
...../...../...../Saved Games/  
...../...../...../Searches/  
...../...../...../Squaresquash/  
...../...../...../untitled.ipynb  
...../...../...../videos/  
...../...../...../anaconda3/  
...../...../...../anaconda_projects/  
...../...../...../eclipse-workspace/  
...../...../...../eclipse/  
...../...../...../ntuser.dat.LOG1  
...../...../...../ntuser.dat.LOG2  
...../...../...../ntuser.ini  
  
no changes added to commit (use "git add" and/or "git commit -a")  
Dropped refs/stash@{0} (caf7aef439f967cc24af4b4c926db9a1b3efce9f)  
  
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)  
$
```

9. You're working on a project with multiple collaborators, and you notice that your local changes conflict with changes that have been pushed by others. How would you resolve the conflicts?

10. You've pushed a feature branch to a remote repository, but now you need to delete the branch from the remote. How would you do that?

```
azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git checkout -b temp-test
Switched to a new branch 'temp-test'

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (temp-test)
$ git push origin temp-test
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (7/7), 545 bytes | 545.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: create a pull request for 'temp-test' on GitHub by visiting:
remote:     https://github.com/PROJECTG549/G549KCH/pull/new/temp-test
remote:
To https://github.com/PROJECTG549/G549KCH.git
 * [new branch]      temp-test -> temp-test

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (temp-test)
$ git push origin --delete temp-test
To https://github.com/PROJECTG549/G549KCH.git
 - [deleted]          temp-test

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (temp-test)
```

11. What command will you use to bring your local main branch up to date with the remote repository before merging it into your feature branch?

```
[deleted]          temp-test

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (temp-test)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git fetch origin

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git pull origin main
From https://github.com/PROJECTG549/G549KCH
 * branch            main      -> FETCH_HEAD
Already up to date.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git pull --rebase origin main
From https://github.com/PROJECTG549/G549KCH
 * branch            main      -> FETCH_HEAD
Already up to date.

azeem@LAPTOP-MUVD4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ |
```

12. How will you update your feature/ui-update branch to reflect the latest changes from main?

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (main)
$ git checkout feature1/signup
Switched to branch 'feature1/signup'

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git fetch origin

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$ git merge origin/main
Already up to date.

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/git_seema (feature1/signup)
$
```

13. Which command should you use to attempt pushing your local feature branch again, and what should you do if it fails due to conflicts?

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git add see.txt

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git commit -m " Local change causing non-fast-forward"
[feature 3b33cce] Local change causing non-fast-forward
 1 file changed, 1 insertion(+), 1 deletion(-)

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git stash
Saved working directory and index state WIP on feature: 3b33cce Local change causing non-fast-forward

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git pull --rebase origin feature
From https://github.com/PROJECTG549/G549KCH
 * branch      feature    -> FETCH_HEAD
warning: skipped previously applied commit 2ac981a
warning: skipped previously applied commit db4684a
hint: use --reapply-cherry-picks to include skipped commits
hint: Disable this message with "git config set advice.skippedCherryPicks false"
Deletion of directory 'OneDrive/desktop/git_seema' failed. Should I try again? (y/n) n
dropping 1c8e1ad5d28682dfd86079e72ca00487e33add64 Added SRS file -- patch contents already upstream
Successfully rebased and updated refs/heads/feature.

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$ git branch
* feature
  feature1/signup
  hasse-feature
  main
  rese
  searchFilter
  temp-test

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature)
$
```

14. How do you apply a **.patch** file provided by your teammate and include it in your commit history?

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (feature1)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 6 commits.
  (use "git push" to publish your local commits)

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (main)
$ git log --oneline
733395b (HEAD -> main) Merge branch 'feature1/signup'
3820739 see.txt
db4684a Local change causing non-fast-forward
1c8e1ad Added SRS file
2ac981a first commit
4e29640 (temp-test, feature1/signup) update
fe67664 (origin/main, origin/HEAD) Remove sensitive file and update .gitignore
cc06cf7 api key
7823986 Resolved merge conflicts
12f6899 Merge branch 'main' of https://github.com/PROJECTG549/G549KCH
02f3e72 Add signup page
4e7198f Added factorial and prime checker script
e5abceb added a change in feature-ui
39e91d4 footer.txt
55ed2ef accidentally committed API key
36d0993 (origin/hasse-feature, hasse-feature) Update README.md
ca00198 ignore log files
4d18bba accidentally committed API key
6bb4aa1 (searchFilter) seema
b35323b added srs document after reset
c837bf3 Added SRS document after reset
022fae2 Initial commit

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (main)
$ git format-patch -1 4e29640
0001-update.patch
```

15. After successful testing, describe the steps (with commands) to merge your feature branch into main and push it to GitHub.

```
azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (main)
$ git pull origin main
From https://github.com/PROJECTG549/G549KCH
 * branch            main      -> FETCH_HEAD
Already up to date.

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (main)
$ ls
0001-update.patch  1.txt  README.md  file1.txt  footer.txt  new.txt  script.py  script1.py

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (main)
$ git branch
  feature
  feature1/signup
  hasse-feature
* main
  rese
  searchFilter
  temp-test

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (main)
$ git merge feature1/signup
Already up to date.

azeem@LAPTOP-MUV4INQ MINGW64 ~/OneDrive/Desktop/ALL/SE LAB/WEEK 2/git_seema (main)
$ |
```