

A large, light blue wireframe sphere is positioned on the left side of the slide, partially overlapping the title text.

SEZG514

INTRODUCTION OF DEVOPS

ASSIGNMENT 1

From,

SUPRIYA P

2023TM93755

INDEX

2

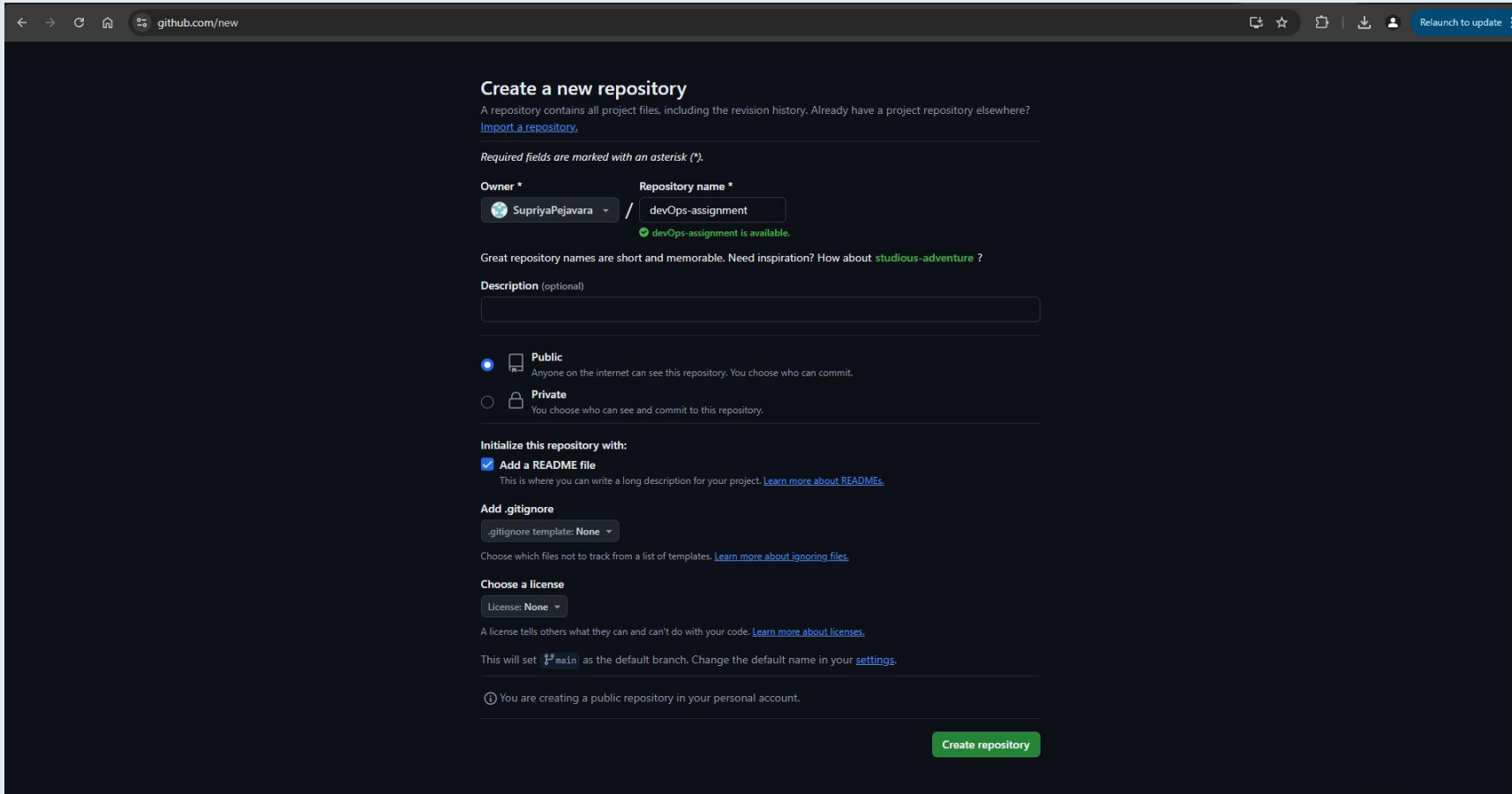
TITLE	SLIDE NUMBER
Create a repository	3,4
Add team members as collaborators and assign them appropriate rights	5
Create a branch(development/production/feature)	6
Edit files or create new files followed by commit	6
Clone the repo and Create pull-request	7,8
While collaborating your work, showcase how conflicts are resolved	9 - 16
Create tag such as open issue, or feature-added	17
Do a force push/commit and then later reset the changes	18,19
Stage “development branch to production branch”	20
Showcase how features are released in versions (merging production to master branch)	21
State importance of Readme and gitignore files and their usage while working in a distributed environment.	22,23
Pull Requests to be linked with e-mail to the manager who finally approves the changes.	24

GIT WORKFLOW ASSIGNMENT

3

1. SET UP REPOSITORY AND TEAM MEMBERS

1.1 Create or Fork a Repository



The screenshot shows the GitHub 'Create a new repository' page. The browser address bar shows 'github.com/new'. The page title is 'Create a new repository'. Below the title, it says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. A note states 'Required fields are marked with an asterisk (*)'. The 'Owner' field is set to 'SupriyaPejavara' and the 'Repository name' field is 'devOps-assignment'. A green checkmark indicates 'devOps-assignment is available.'. Below this, it says 'Great repository names are short and memorable. Need inspiration? How about [studious-adventure](#) ?'. The 'Description (optional)' field is empty. The 'Public' radio button is selected, with the text 'Anyone on the internet can see this repository. You choose who can commit.'. The 'Private' radio button is unselected, with the text 'You choose who can see and commit to this repository.'. Under 'Initialize this repository with:', the 'Add a README file' checkbox is checked, with the text 'This is where you can write a long description for your project. [Learn more about READMEs.](#)'. The 'Add .gitignore' section shows '.gitignore template: None'. Below this, it says 'Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)'. The 'Choose a license' section shows 'License: None'. Below this, it says 'A license tells others what they can and can't do with your code. [Learn more about licenses.](#)'. It also states 'This will set `main` as the default branch. Change the default name in your [settings](#).'. A note at the bottom says 'You are creating a public repository in your personal account.'. A green 'Create repository' button is at the bottom right.

GIT WORKFLOW ASSIGNMENT

4

1. SET UP REPOSITORY AND TEAM MEMBERS

1.2 Clone the repository to local

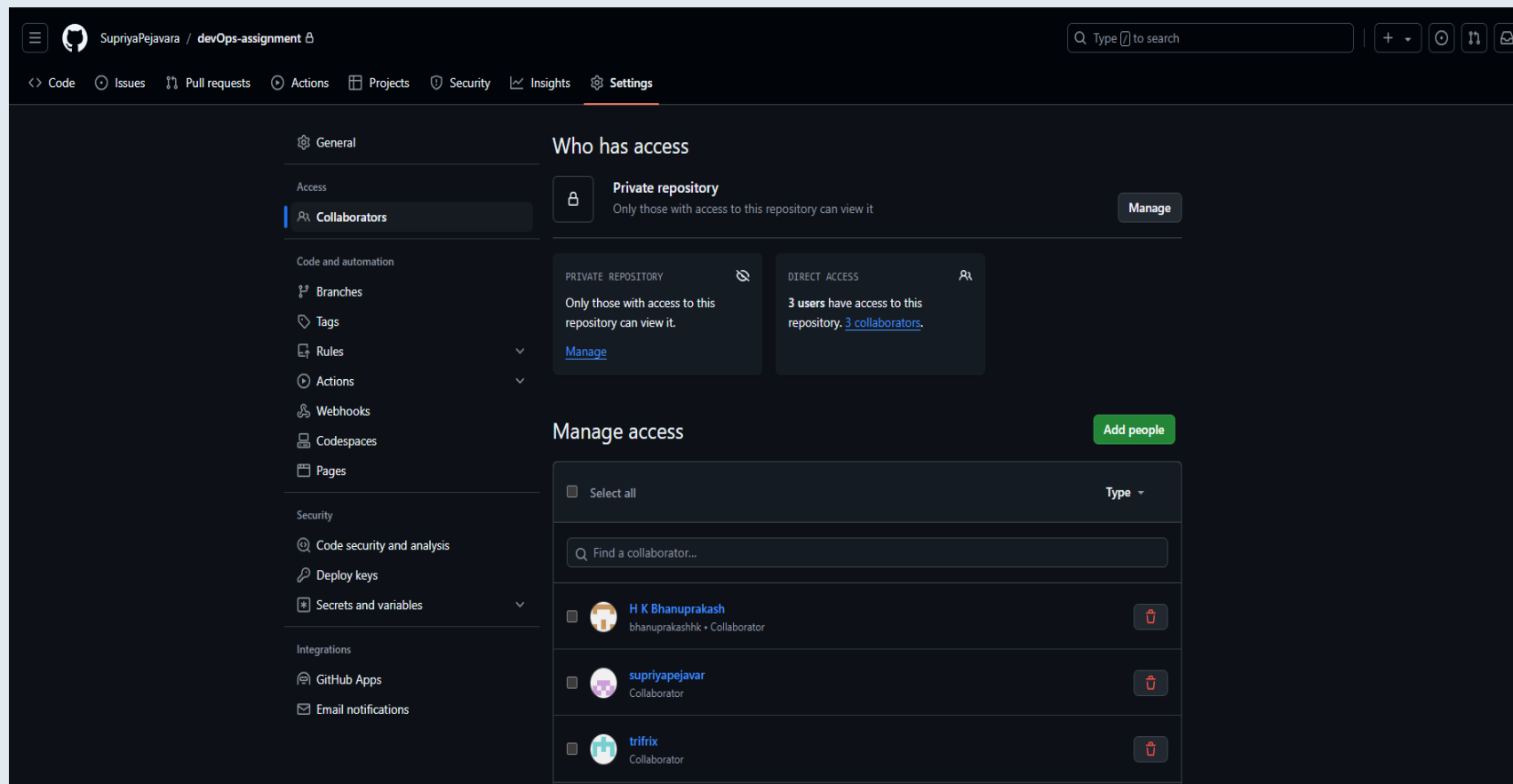
```
sunag@Lenovo-PC MINGW64 ~  
$ git config --global user.name "SupriyaPejavara"  
  
sunag@Lenovo-PC MINGW64 ~  
$ git config --global user.email "supriyapejavar@gmail.com"  
  
sunag@Lenovo-PC MINGW64 ~  
$ git clone https://github.com/SupriyaPejavara/devOps-assignment.git  
Cloning into 'devOps-assignment'...  
remote: Enumerating objects: 21, done.  
remote: Counting objects: 100% (21/21), done.  
remote: Compressing objects: 100% (14/14), done.  
remote: Total 21 (delta 2), reused 11 (delta 1), pack-reused 0 (from 0)  
Receiving objects: 100% (21/21), 7.12 KiB | 137.00 KiB/s, done.  
Resolving deltas: 100% (2/2), done.
```

GIT WORKFLOW ASSIGNMENT

5

1. SET UP REPOSITORY AND TEAM MEMBERS

1.3 Add Team Members as Collaborators



GIT WORKFLOW ASSIGNMENT

6

2. CREATE AND MANAGE BRANCHES

2.1 Create a branch and create new file followed by commit

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (main)
$ git checkout -b feature
Switched to a new branch 'feature'

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ echo "Feature content"> feature.txt

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git add feature.txt
warning: in the working copy of 'feature.txt', LF will be replaced by CRLF the n
ext time Git touches it

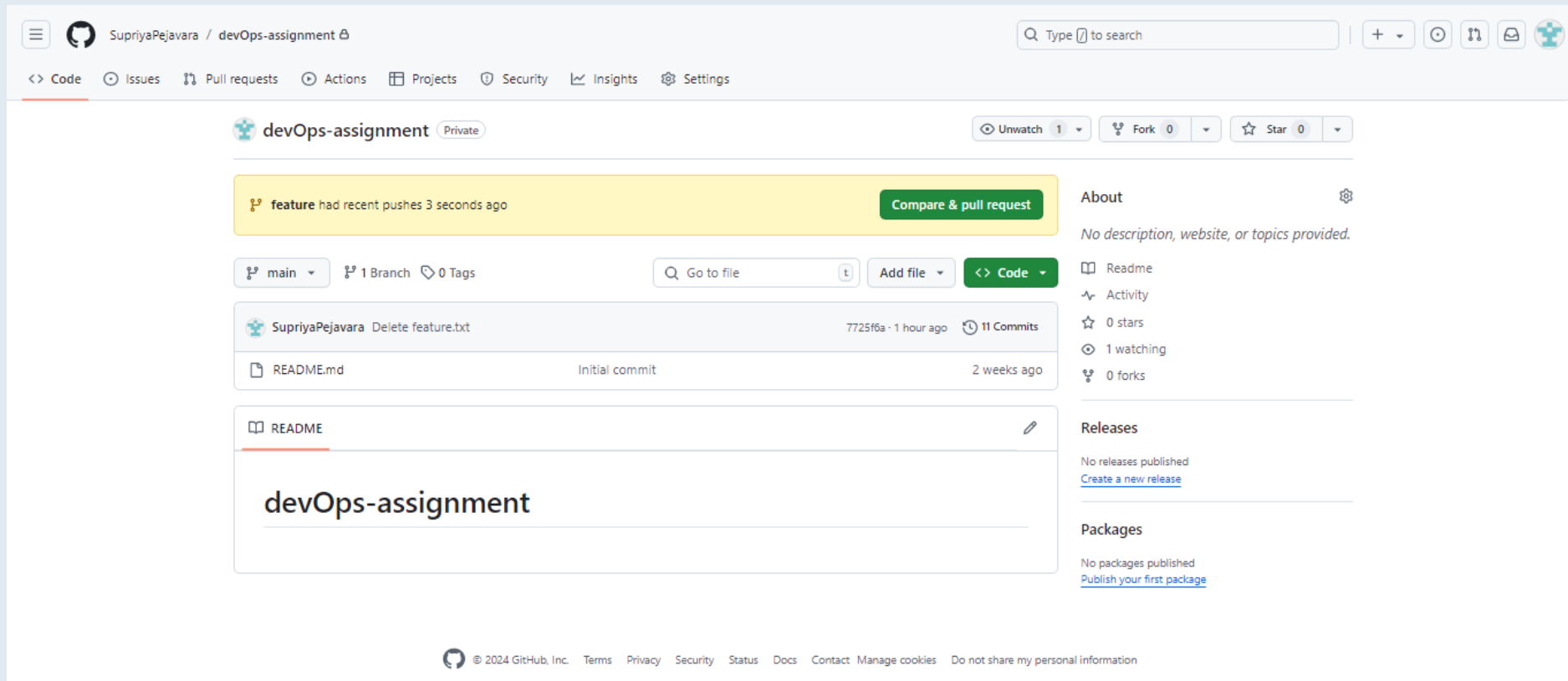
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git commit -m "Add feature"
[feature 3533e18] Add feature
 1 file changed, 1 insertion(+)
 create mode 100644 feature.txt

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git push origin feature
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 100.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:   https://github.com/SupriyaPejavara/devOps-assignment/pull/new/featu
re
remote:
To https://github.com/SupriyaPejavara/devOps-assignment.git
 * [new branch]      feature -> feature
```

GIT WORKFLOW ASSIGNMENT

2. CREATE AND MANAGE BRANCHES

2.2 Compare & pull request



GIT WORKFLOW ASSIGNMENT

8

2. CREATE AND MANAGE BRANCHES

2.3 Create pull request

The screenshot shows the GitHub 'Open a pull request' page. At the top, the repository name 'SupriyaPejavara / devOps-assignment' is displayed. Below the navigation bar, the 'Pull requests' tab is selected. The main heading is 'Open a pull request', followed by a subtext: '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](#).' Below this, a comparison bar shows 'base: main' and 'compare: feature' with a green checkmark and the text 'Able to merge. These branches can be automatically merged.' The 'Add a title' section has a text input field with the placeholder 'Add feature'. The 'Add a description' section has a rich text editor with a 'Write' tab selected and a placeholder 'Add your description here...'. The right sidebar contains settings for 'Reviewers' (No reviews), 'Assignees' (SupriyaPejavara), 'Labels' (None yet), 'Projects' (None yet), 'Milestone' (No milestone), and 'Development' (Use [Closing keywords](#) in the description to automatically close issues). At the bottom right, there is a green 'Create pull request' button.

GIT WORKFLOW ASSIGNMENT

9

3. RESOLVE THE CONFLICTS

3.1 Introduce a conflict

Created a new branch 'development' > Modified the same the file 'feature.txt' > Pushed the changes

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git checkout -b development
Switched to a new branch 'development'

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ echo "This is a change in the development branch" > feature.txt

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git add feature.txt
git commit -m "Edit feature.txt in development branch"
warning: in the working copy of 'feature.txt', LF will be replaced by CRLF the next time Git touches it
[development f37e747] Edit feature.txt in development branch
1 file changed, 1 insertion(+), 1 deletion(-)

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git push origin development
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), 348 bytes | 116.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'development' on GitHub by visiting:
remote:   https://github.com/SupriyaPejavara/devOps-assignment/pull/new/development
remote:
To https://github.com/SupriyaPejavara/devOps-assignment.git
 * [new branch]      development -> development
```

GIT WORKFLOW ASSIGNMENT

10

3. RESOLVE THE CONFLICTS

3.1 Introduce a conflict

Edit the same file on 'feature' branch

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git checkout feature
Switched to branch 'feature'

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ ^[[200~echo "This is a conflicting change in the feature-xyz branch" > feature.txt
bash: $'\E[200~echo': command not found

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ echo "This is a conflicting change in the feature branch" > feature.txt

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git add feature.txt
warning: in the working copy of 'feature.txt', LF will be replaced by CRLF the next time Git touches it

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git commit -m "Edit feature.txt in feature branch"
[feature d026fa7] Edit feature.txt in feature branch
1 file changed, 1 insertion(+), 1 deletion(-)

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git push origin feature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 348 bytes | 116.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SupriyaPejavara/devOps-assignment.git
3533e18..d026fa7 feature -> feature
```

GIT WORKFLOW ASSIGNMENT

11

3. RESOLVE THE CONFLICTS

3.2 Experience the conflict

Whenever there is a conflict, one can see the message 'Can't automatically merge' as below

The screenshot shows the GitHub web interface for a repository named 'devOps-assignment' by user 'SupriyaPejavara'. The 'Pull requests' tab is active. The main heading is 'Comparing changes', with a subtext: '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](#).' Below this, a comparison bar shows 'base: feature' and 'compare: development'. A red error message states: 'Can't automatically merge. Don't worry, you can still create the pull request.' The interface includes fields for 'Add a title' (with placeholder text 'Edit feature.txt in development branch') and 'Add a description' (with a rich text editor). On the right, a sidebar lists various settings: Reviewers (No reviews), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development (Use [Closing keywords](#) in the description to automatically close issues). At the bottom right, there is a green 'Create pull request' button. A footer note mentions: 'Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).'

GIT WORKFLOW ASSIGNMENT

12

3. RESOLVE THE CONFLICTS

3.2 Experience the conflict

When we try to merge the pull request, GitHub will detect a conflict as the same lines of the files we modified differently in both branches.

The screenshot displays a GitHub pull request interface for a pull request titled "Edit feature.txt in development branch #7". At the top, it indicates that SupriyaPejavara wants to merge 1 commit into the feature branch from the development branch. Below this, there are tabs for Conversation (0), Commits (1), Checks (0), and Files changed (1). A comment from SupriyaPejavara is visible, stating "No description provided." Below the comment, there is a section for "This branch has conflicts that must be resolved", which lists "feature.txt" as a conflicting file. A "Resolve conflicts" button is present next to this message. At the bottom, there is a "Merge pull request" button and a note that says "You can also open this in GitHub Desktop or view command line instructions." On the right side of the interface, there are sections for Reviewers (No reviews), Assignees (SupriyaPejavara), Labels (None yet), Projects (None yet), Milestone (No milestone), and Development (Successfully merging this pull request may close these issues).

GIT WORKFLOW ASSIGNMENT

13

3. RESOLVE THE CONFLICTS

3.3 Resolving the conflict

Fetches the latest changes and attempted to merge locally

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (feature)
$ git checkout development
Switched to branch 'development'

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git pull origin development
From https://github.com/SupriyaPejavara/devOps-assignment
 * branch                development -> FETCH_HEAD
Already up to date.

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git merge feature
Auto-merging feature.txt
CONFLICT (content): Merge conflict in feature.txt
Automatic merge failed; fix conflicts and then commit the result.
```

GIT WORKFLOW ASSIGNMENT

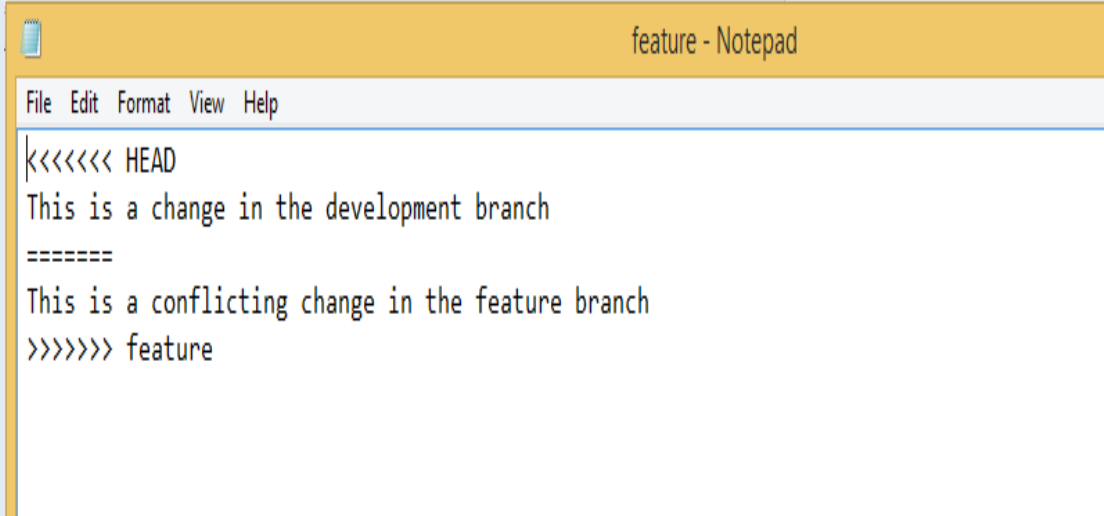
14

3. RESOLVE THE CONFLICTS

3.3 Resolving the conflict

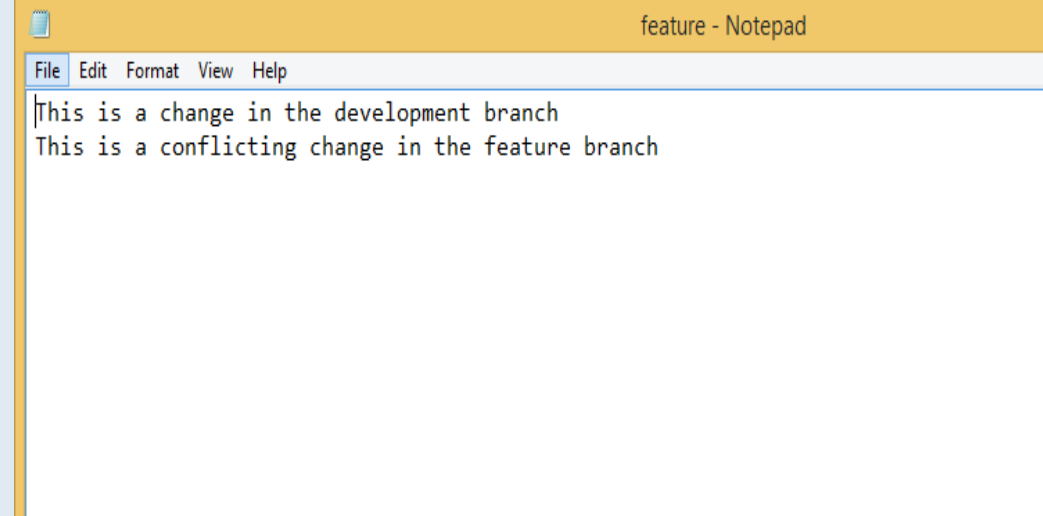
Opened 'feature.txt' and could see the conflict markers. Removed the conflict markers and saved the file.

Before removing the conflict markers



```
feature - Notepad
File Edit Format View Help
|<<<<<< HEAD
This is a change in the development branch
=====
This is a conflicting change in the feature branch
>>>>>> feature
```

After removing the conflict markers



```
feature - Notepad
File Edit Format View Help
This is a change in the development branch
This is a conflicting change in the feature branch
```

GIT WORKFLOW ASSIGNMENT

15

3. RESOLVE THE CONFLICTS

3.4 Stage and commit the Resolved Conflict:

After resolving the conflict, staged and committed the changes. Pushed the resolved changes.

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development|MERGING)
$ git add feature.txt

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development|MERGING)
$ git commit -m "Resolved the conflict"
[development f581a6d] Resolved the conflict

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git push origin development
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 385 bytes | 128.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SupriyaPejavara/devOps-assignment.git
f37e747..f581a6d  development -> development
```

GIT WORKFLOW ASSIGNMENT

16

3. RESOLVE THE CONFLICTS

3.4 Stage and commit the Resolved Conflict:

After resolving conflicts, merge pull request button is enabled.

The screenshot displays a GitHub Pull Request (PR) for the repository 'SupriyaPejavara / devOps-assignment'. The PR is titled 'Edit feature.txt in development branch #7' and is currently in the 'Open' state. The PR description indicates that 2 commits from the 'feature' branch are being merged into the 'development' branch. The PR is assigned to 'SupriyaPejavara' and has no reviews or labels. The PR is ready for merging, as indicated by the green checkmark and the text 'This branch has no conflicts with the base branch'. The 'Merge pull request' button is enabled and visible at the bottom of the PR description. The PR also includes a comment from 'SupriyaPejavara' stating 'No description provided.' and a commit history showing 'Edit feature.txt in development branch' and 'Resolved the conflict'.

GIT WORKFLOW ASSIGNMENT

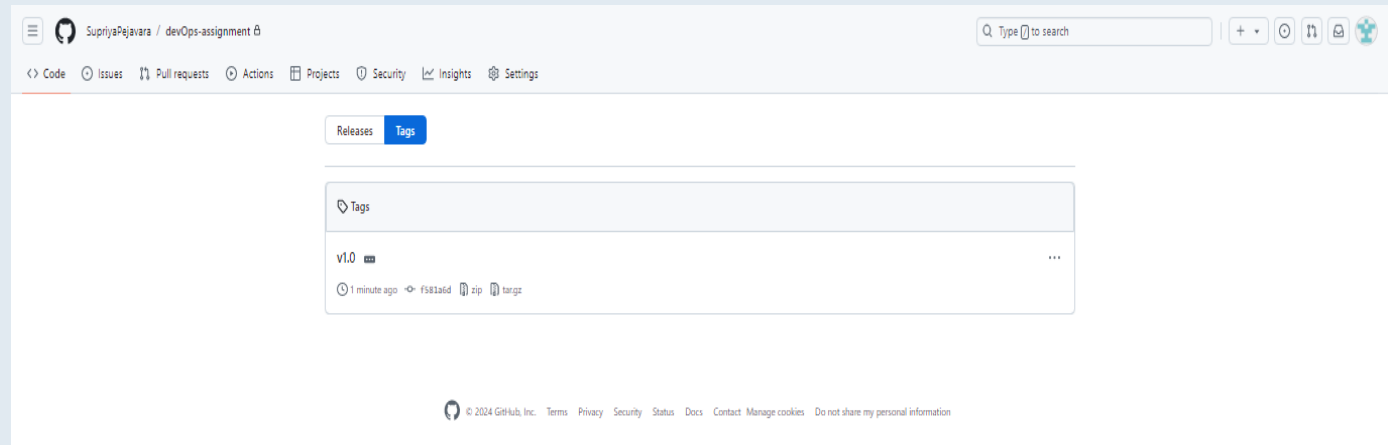
17

4. CREATE TAG SUCH AS OPEN ISSUE OR FEATURE ADDED

Created the tag locally to mark addition of the new feature and pushed the changes to repository.

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git tag -a v1.0 -m "Feature added"

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git push origin v1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 168 bytes | 84.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SupriyaPejavara/devOps-assignment.git
 * [new tag]          v1.0 -> v1.0
```



GIT WORKFLOW ASSIGNMENT

18

5. DO A FORCE PUSH/COMMIT AND THEN LATER RESET THE CHANGES

5.1 Force Push Changes

Added a temporary file and did force push

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ echo "Temporary changes" > temp.txt

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git add temp.txt
warning: in the working copy of 'temp.txt', LF will be replaced by CRLF the next
time Git touches it

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git commit -m "Temporary commit"
[development f41ac20] Temporary commit
1 file changed, 1 insertion(+)
create mode 100644 temp.txt

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git push --force origin feature
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SupriyaPejavara/devOps-assignment.git
+ 68fb357...d026fa7 feature -> feature (forced update)
```

GIT WORKFLOW ASSIGNMENT

19

5. DO A FORCE PUSH/COMMIT AND THEN LATER RESET THE CHANGES

5.2 Reset Changes

Reset to the previous commit and removed the temporary changes.

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git reset --hard Head~1
HEAD is now at f581a6d Resolved the conflict

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git push --force origin feature
Everything up-to-date
```

GIT WORKFLOW ASSIGNMENT

20

6. STAGE DEVELOPMENT BRANCH TO PRODUCTION BRANCH

Created a 'production' branch

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (development)
$ git checkout -b production
Switched to a new branch 'production'

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (production)
$ git push origin production
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'production' on GitHub by visiting:
remote:   https://github.com/SupriyaPejavara/devOps-assignment/pull/new/production
remote:
To https://github.com/SupriyaPejavara/devOps-assignment.git
 * [new branch]      production -> production
```

Merge 'development' into 'production'

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (production)
$ git checkout production
Already on 'production'

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (production)
$ git merge development
Already up to date.

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (production)
$ git push origin production
Everything up-to-date
```

GIT WORKFLOW ASSIGNMENT

21

7. SHOWCASE HOW FEATURES ARE RELEASED IN VERSIONS

7.1 Release a Version

Merged 'production' into 'master' to signify a release

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (production)
$ git checkout -b master
Switched to a new branch 'master'

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (master)
$ git merge production
Already up to date.

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/SupriyaPejavara/devOps-assignment/pull/new/master
remote:
To https://github.com/SupriyaPejavara/devOps-assignment.git
 * [new branch]      master -> master
```

GIT WORKFLOW ASSIGNMENT

22

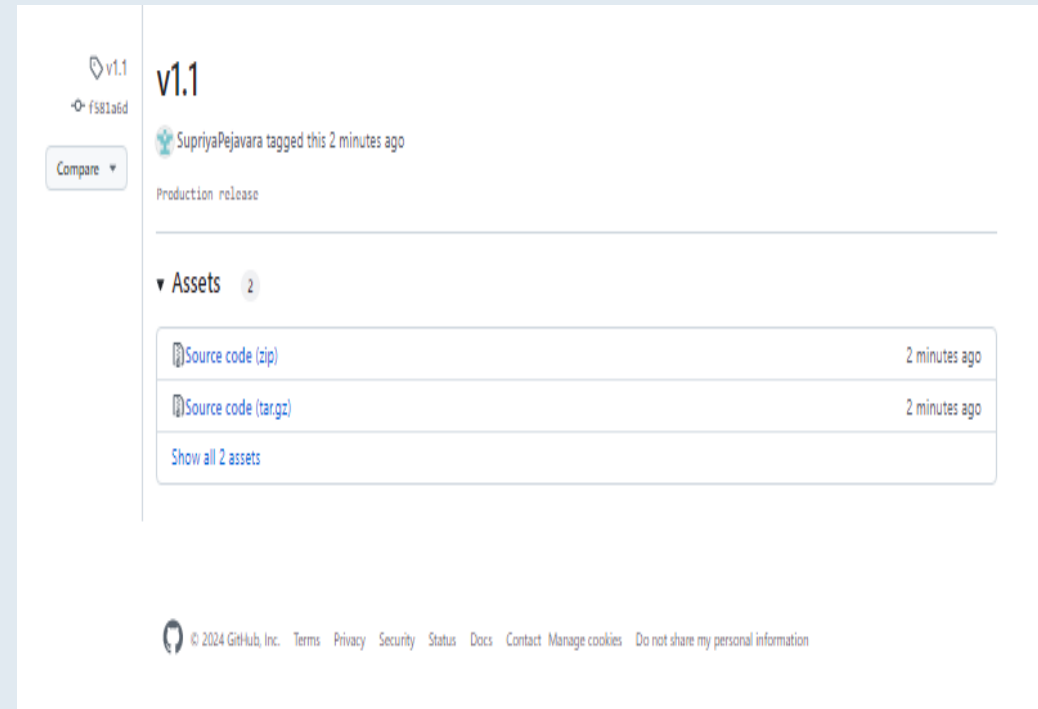
7. SHOWCASE HOW FEATURES ARE RELEASED IN VERSIONS

7.2 Tag the Release

Tagged the release version

```
sunag@Lenovo-PC MINGW64 ~/devOps-assignment (master)
$ git tag -a v1.1 -m "Production release"

sunag@Lenovo-PC MINGW64 ~/devOps-assignment (master)
$ git push origin v1.1
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 172 bytes | 86.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/SupriyaPejavara/devOps-assignment.git
 * [new tag]          v1.1 -> v1.1
```



8. IMPORTANCE OF README AND .GITIGNORE FILES

8.1 Readme.md

- The 'README.md' file provides an overview of the project.
- README.md files typically include information on:
 - What the project does
 - Why the project is useful
 - How users can get started with the project
 - Where users can get help with your project
 - Who maintains and contributes to the project
- It's vital for any distributed project as it helps collaborators and new contributors understand the purpose and structure of the project.

8. IMPORTANCE OF README AND .GITIGNORE FILES

8.2 .gitignore


- A .gitignore file specifies intentionally untracked files that Git should ignore. Files already tracked by Git are not affected.
- This is crucial in a distributed environment to avoid committing unnecessary or sensitive files like build artifacts, API keys or any local environment configurations that are not relevant to other collaborators.
- The benefit of specifying some files into .gitignore file is that one can commit all the tracked files at once without worrying about ignored files added accidentally . It is by far quicker to commit the files altogether instead of adding one by one.






GIT WORKFLOW ASSIGNMENT

25



9. PULL REQUESTS TO BE LINKED WITH E-MAIL TO THE MANAGER WHO FINALLY APPROVES THE CHANGES

Re: [SupriyaPejavara/devOps-assignment] Add feature login (PR #1)

 Supriya Pejavara <notifications@github.com>
To: ● SupriyaPejavara/devOps-assignment
Cc: ● Pejavara, Supriya; ● Assign

 Reply  Reply All  Forward  

Tue 9/3/2024 12:51 PM

 This sender notifications@github.com is from outside your organization.
 If there are problems with how this message is displayed, click here to view it in a web browser.
Click here to download pictures. To help protect your privacy, Outlook prevented automatic download of some pictures in this message.

[Unsubscribe](#)

You don't often get email from notifications@github.com. [Learn why this is important](#)

Assigned [#1](#) to [@supriyapejavar](#).

—

Reply to this email directly, [view it on GitHub](#), or [unsubscribe](#).
You are receiving this because you were assigned.



THANK YOU

SUPRIYA P

2023TM93755