Dungo, Jinnel I.

IV-ACSAD

General Instructions: This activity is expected to take approximately 1.5 hours. Please ensure careful adherence to the provided instructions.

Objective:

• This exercise focuses on using Git in a local environment without remote repositories. You'll practice setting up a local Git repository, working with branches, managing commits, and resolving conflicts locally.

Problem Statement: You are working on a project locally, and need to use Git for version control. Your task is to initialize a Git repository, implement features using branches, manage commits, and handle conflicts that may arise when merging different branches

1. Open Microsoft Word or any word processing software that supports document creation.

2. Start a new document.

3. Copy and paste the tasks for Machine Problem #1 into the document. Ensure proper formatting and readability.

4. Insert your source code into the document. You can either copy and paste it directly or take a screenshot of the code and insert the image

into the document.

5. Capture screenshots of the output of your code execution. Make sure the screenshots are clear and readable.

6. Insert the screenshots into the document. You can either directly insert the images or embed them as links, depending on your preference. and the document's requirements

7. Arrange the content in a logical and organized manner. You may want to use headings, subheadings, and bullet points to make the document easy to follow.

8. Review the document for any errors or formatting issues. Make necessary adjustments to ensure clarity and correctness. 9. Save the document as a PDF file. Most ward processing software offers an option to save or export documents as PDFs. Choose this option and follow the prompts to save your document in PDF format.

10. Once saved, review the PDF file to ensure that all content, including instructions, source code, and screenshots, is accurately captured and presented

11. If everything looks good, your Word document containing instructions, source code, and screenshots, saved as a PDF file, is ready for submission or sharing.

Please follow the filename format (minus 10 for the wrong filename) MP1<Lastname>.pdf

Example: MP1Mansueto.pdf

Scenario:

You're working on a personal project named "<YourLastname>_local_project and want to experiment with new features without affecting the main codebase

Example Mansueto local_project

Note:

Ensure Git is installed on your machine.

Set up your Git usemame and email (if not already configuredi

Create a directory called Your Lastname_local project, navigate into it, and initialize it as a Git repository

Multiple commits reflecting changes in different branches

Evidence of conflict resolution during branch merging (optional)

A Git tag-marking version v1.0 (optional).

Tasks:

1. Create a new branch:

Create a new branch named "feature_x" to isolate your experimental changes

2. Stage changes:

Make modifications to your project files.

Stage the changes using the gat add command.

3. Commit changes:

Commit the staged changes to your local repository using the git commit command. Provide a clear and concise commit message.

4. Switch back to the main branch:

Use the git checkout command to switch back to the main branch.

5. Merge changes from the feature branch:

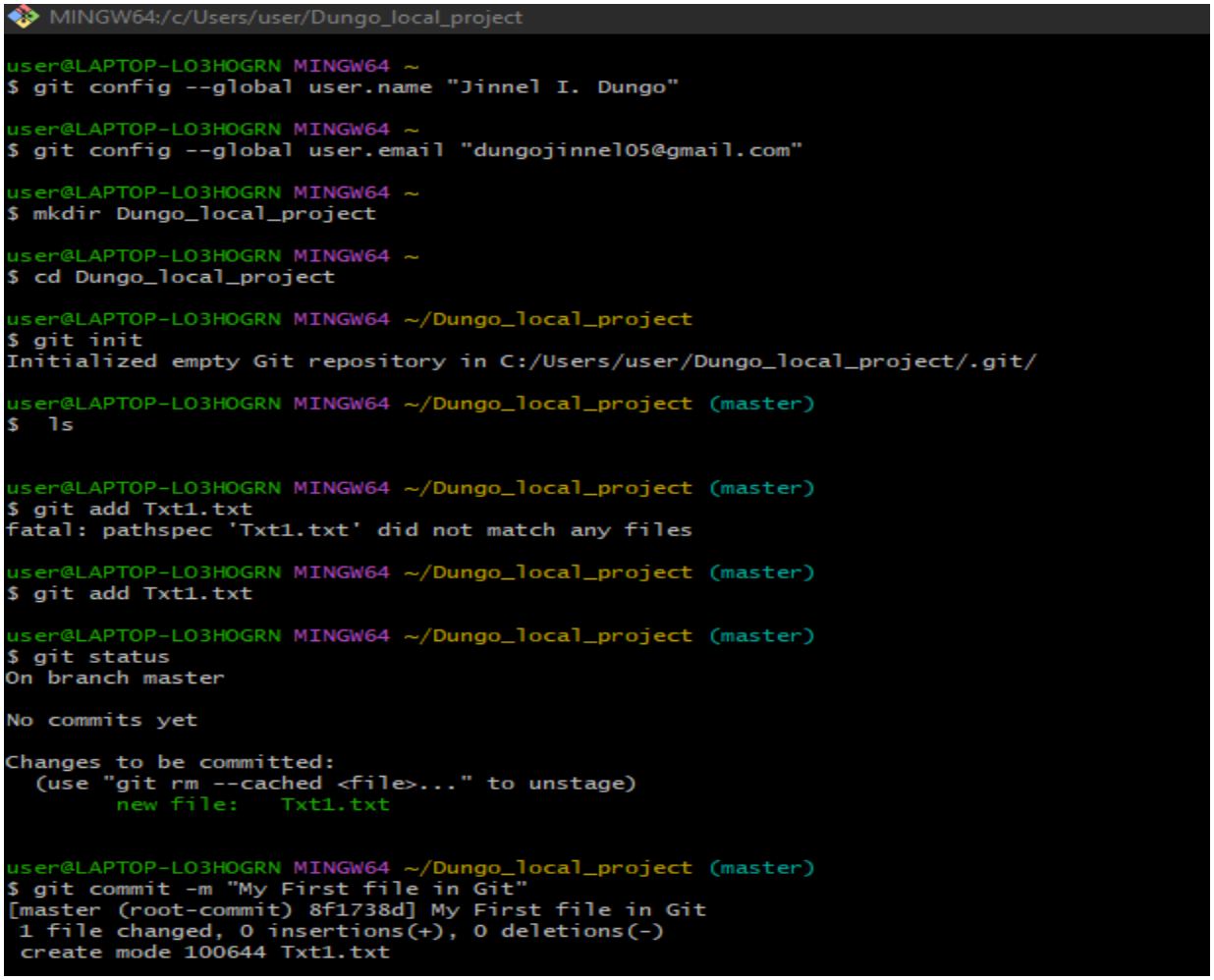
Use the git merge command to merge the changes from the feature x branch into the main branch. Resolve any merge conflicts that may arise

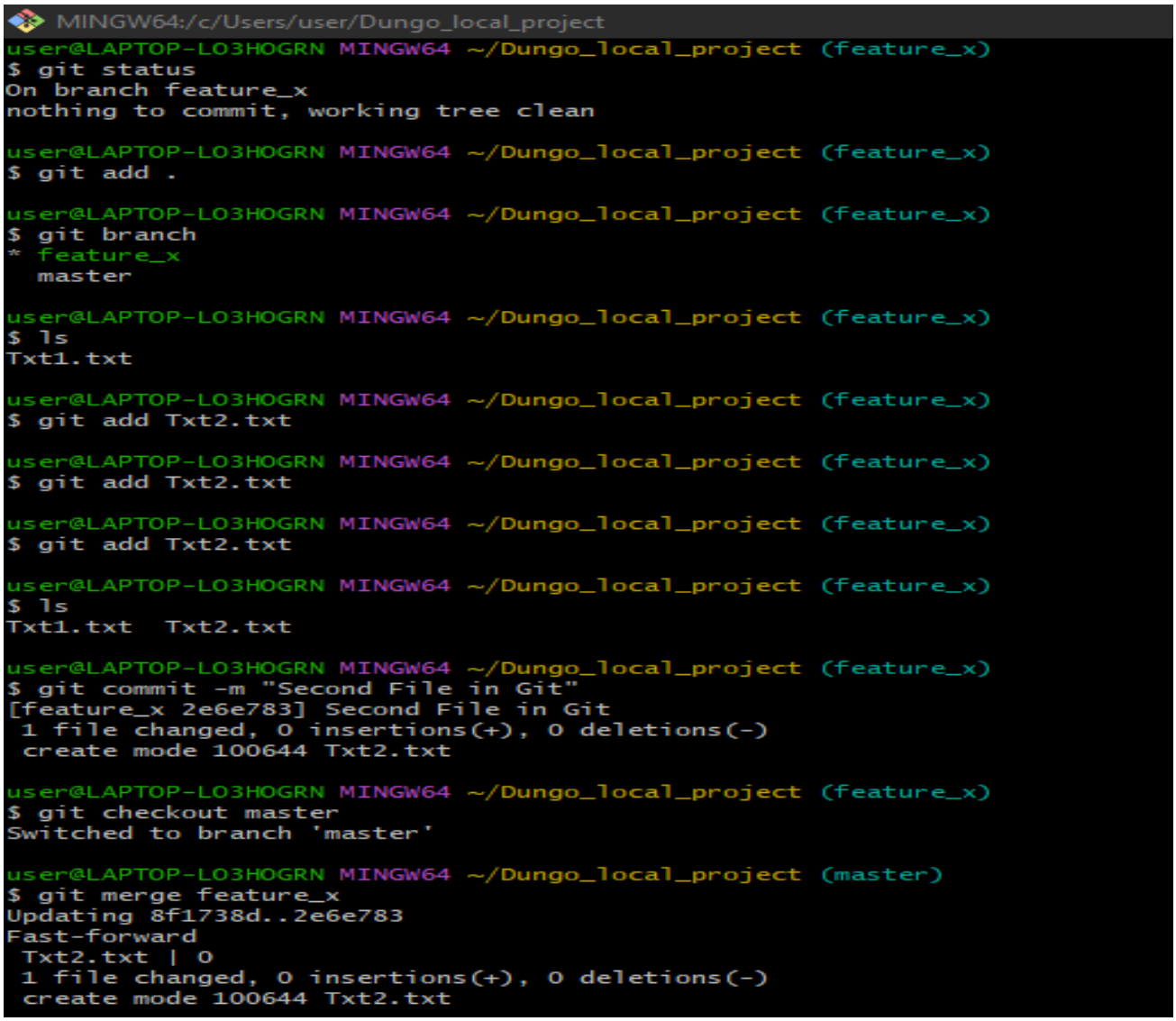
6. Viewing the Commit History:

Use Git commands to view the commit history and ensure that all commits are in place.

7. Tagging Versions (Optional):

After completing the merge, use the it tags command to create a tag named v1.0 to mark the first version of your project.




MINGW64:/c/Users/user/Dungo_local_project
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git status
On branch feature_x
nothing to commit, working tree clean
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git add .
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git branch
* feature_x
master
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ ls
Txt1.txt
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git add Txt2.txt
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git add Txt2.txt
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git add Txt2.txt
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ ls
Txt1.txt Txt2.txt
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git commit -m "Second File in Git"
[feature_x 2e6e783] Second File in Git
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Txt2.txt
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (feature_x)
$ git checkout master
Switched to branch 'master'
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git merge feature_x
Updating 8f1738d..2e6e783
Fast-forward
Txt2.txt | 0
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Txt2.txt

```
MINGW64:/c/Users/user/Dungo_local_project
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git merge feature_x
Updating 8f1738d..2e6e783
Fast-forward
 Txt2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Txt2.txt

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git tag v1.0

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git log --online
fatal: unrecognized argument: --online

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git checkout -b lastbranch
Switched to a new branch 'lastbranch'

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git add .

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ ls
Txt1.txt  Txt2.txt  Txt3.txt

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'?
hint: Disable this message with "git config advice.addEmptyPathspec false"

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git add .

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git status
On branch lastbranch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Txt3.txt


user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git commit -b "Third File in Git"
error: unknown switch `b'
usage: git commit [-a | --interactive | --patch] [-s] [-v] [-u<mode>] [--amend]
                  [--dry-run] [(-c | -C | --squash) <commit> | --fixup [(amend|r
eword):]<commit>]
```

```
user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git commit -m "Third File in Git"
[lastbranch 21cf9ea] Third File in Git
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Txt3.txt

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git status
On branch lastbranch
nothing to commit, working tree clean

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (lastbranch)
$ git checkout master
Switched to branch 'master'

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ get merge lastbranch
bash: get: command not found

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git merge lastbranch
Updating 2e6e783..21cf9ea
Fast-forward
 Txt3.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Txt3.txt

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git tag v1.0
fatal: tag 'v1.0' already exists

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$ git log --online
fatal: unrecognized argument: --online

user@LAPTOP-LO3HOGRN MINGW64 ~/Dungo_local_project (master)
$
```