Assignment 10 - Git Basics

TW Lab 2025

In today's lab, you will explore basic Git commands used to create and update a repository.

Installation

Use *apt* to install *git* on Ubuntu: Check its version.

```
sudo apt install git
```

Instructions

Create a project directory (folder) on your local machine to experiment with Git com mands. Within that directory, create a file using Python (vecadd.py) to input two vectors and print their dot product.

Perform the following tasks step by step, carefully following the description provided. After running each command take one or more screenshots of the terminal, as required, and add it to a Google Doc. At the end of this assignment, you will submit a PDF file containing all these screenshots along with brief description of what happened at each step.

1. Navigate to a new folder (cd folder-path) where your sample project will be located and initialize it as a Git repository.
   git init

2. Configure the Git repository with your name and email.
   git config user.name "Your Name"
   git config user.email "username@provider.com"

3. Create a simple text file "readme.txt" in your project folder. Add a description of your project to this file. Use the following commands to stage the new file. git add readme.txt
   After creating and adding the file, check the status of your repository using: git status

4. After adding the file, commit your changes to the repository
   using: git commit -m "Adding first file."
   Check status again to verify that the working directory is clean after the

5. Open ”readme.txt” and write another line in the file. Then use the following com mand to view the changes you made: git diff

6. Explore the git restore command and observe its impact at both stages (modified & staged) of commit creation. Capture the output of ”git restore” at various stages. Commit your changes again using the commands learnt above.

7. Observe the history of the repository, including the commits made so far. git log

8. Revert the previous commit using the following command:
git revert <commit-hash>
Observe, how this creates another commit to undo the changes, rather than remov ing the old commit from the log.

9. Using the commands learnt above, commit your Python code (vecadd.py). Cap ture the output of git log as a screenshot.

10. Now, remove ”readme.txt” from your repository using:
git rm readme.txt
Commit the removal, so that the file is removed from the repository as well.

11. Navigate to the previous commit using git checkout <commit-hash>. Identify the hash (SHA-1) of an earlier commit from the log and check out that commit. After checking out an earlier version of the project, view the contents of the working directory. Also observe the log. Note down your observations regarding how this command compares to git revert.

12. To return to the latest commit, use:
git checkout <branch-name>.

Submission

Upload the following to Google Classroom:

1. The source file

2. Generated PDF file

Please ensure that the document is well-organized, with clear headings for each

task. 2