**Assignment 2: Introduction to Git & GitHub**

**Course:** BIA 4650 — Data Management & Security in the Cloud **Points:** 20 — **Due Date: September 22nd, 2025**

You must submit a Word document or PDF which shows screenshots of all the tasks being successfully completed with screenshots attached and with the proper captions.

# Learning Goals

* Create a clean local Git repository with a simple folder structure.
* Add, stage, and commit files with clear commit messages.
* Create a GitHub repository, and push the local Git repository to the remote GitHub repository via HTTPS URL.
* Document your steps with numbered screenshots so another student can follow them.

# Scenario

As a final-year undergraduate entering the job market, you’re hosting your work on GitHub to track it and share it publicly with recruiters.

# Conventions

* Replace placeholders like Firstname Lastname with your name.
* Run commands in **Terminal (macOS)** or **Git Bash / Command Prompt (Windows)**.
* **Required Screenshots** are labeled #1, #2, with proper captions.
* All the screenshots should be of the full screen.

# Part 1: Create a Local Repository (2 points)

1. Go to your folder where you want to create the local Git repository using the following command:

cd ~/Downloads/course\_material/Fall\_2025/

1. Make and initialize your repo:

mkdir BIA\_4650\_Fall25\_Firstname\_Lastname

cd BIA\_4650\_Fall25\_Firstname\_Lastname git init

1. Check status:

git status

1. **Screenshot #1:** Terminal showing git init and git status in the new repo.

**#1 git init and git status outputs**

**A screenshot of a computer

AI-generated content may be incorrect.**

# Part 2: Add Assignment Files (2 points)

1. Create a folder for your first assignment:

mkdir Assignment\_1

1. Put your Assignment 1 Excel (e.g., Analysis.xlsx) and Word file (e.g., Report.docx) in Assignment 1/. You can drag-and-drop or use:

mv ~/Downloads/course\_material/Fall\_2025/Analysis.xlsx Assignment\_1/ mv ~/Downloads/course\_material/Fall\_2025/Report.docx Assignment\_1/

1. Show the content of the folder using the following command:

ls

1. **Screenshot #2:** Terminal showing ls with both files in Assignment 1/.

**#2 move assignment 1 file to newly created repo**

**A screen shot of a computer

AI-generated content may be incorrect.**

# Part 3: Stage and Commit (3 points)

1. Stage and check:

git add . git status

1. Commit with a clear message:

git commit -m "Initial commit: add Assignment\_1 Excel and Word files"

1. Show a short history:

git log --oneline

1. **Screenshot #3:** Terminal showing git add, git commit, and the git log.

**#3 git add, it commit, and git log outputs**

A screenshot of a computer

AI-generated content may be incorrect.

# Part 4: Create a GitHub Repository and Connect Remote (3 points)

1. In a browser, go to [github.com.](https://github.com/) Create a **new repository** named BIA 4650 Fall25 Firstname Lastname **Do not** initialize it with a README, license, or .gitignore.
2. Copy the repository **HTTPS** URL from GitHub.
3. Add the URL you copied as remote and verify:

git remote add origin <PASTE\_HTTPS\_URL\_HERE> git remote -v

1. **Screenshot #4:** Terminal showing git remote -v with correct fetch/push URLs.

**#4 git remote output**

A screenshot of a computer screen

AI-generated content may be incorrect.

# Part 5: Clear Cache Memory (1 point)

Before you push your Git repo to GitHub repo, you need to clear any cached passwords. For that, you need to run the following code:

* **macOS (Keychain helper):**

printf "protocol=https\nhost=github.com\n" | git credential-osxkeychain erase

* **Windows (Git Bash):**

printf "protocol=https\nhost=github.com\n" | git credential-manager-core erase

# Part 6: Generate Personal Access Token (3 points)

1. Go to your profile settings.
2. Go to developer settings.
3. Select Personal Access Tokens.
4. Generate new token.
5. Fill in the details for token name and token description according to you.
6. Under Repository access, choose Selected repositories.
7. Under Permissions, select Contents and give Read & Write Permission.
8. Generate PAT.

# Part 7: Authenticate and Push (4 points)

*Note:* For HTTPS pushes, GitHub uses a Personal Access Token (PAT). Therefore, you should have cleared the cache memory and should have PAT up to this step. If you are done with these steps, then you can use the following command to push the local repo to online repo.

1. First push using:

git push origin main

1. It should ask for username. When asked for username, use the username you see on GitHub.
2. Then it will ask for password, so you will copy the PAT generated and paste it here.
3. **Screenshot #5:** Terminal showing a successful git push (line ending with "main -> main").

**#5 git push output**

A screen shot of a computer

AI-generated content may be incorrect.

# Part 8: Verify on GitHub (2 points)

1. Refresh the repository page and confirm your folders/files.
2. **Screenshot #6:** GitHub page showing Assignment 1/ and your files.

**#6 GitHub showing file from Assignment 1**

A screenshot of a computer

AI-generated content may be incorrect.