



**BICOL UNIVERSITY**  
**POLANGUI**  
Polangui, Albay



# **IT 123 – System Administration and Maintenance**

1<sup>st</sup> Semester 2025-2026

## **Lab Report 2 – User and Group Management in Windows & Linux**

John Omar C. Clutario  
Simone Andreas M. Mannhuel B. Nate  
Joshua A. Obstaculo  
**BSIT – 4B**

Guillermo V. Red, DIT  
**Instructor**

**09-02-2025**

## I. Objectives

- To create and manage user accounts and groups in both Windows Server and Ubuntu Server.
- To assign appropriate permissions for faculty and student users.
- To verify access rights based on group membership.
- To practice documentation and version control through GitHub submission.

## II. Scenario

You are appointed as the new system administrator of a university lab. You are tasked to configure Windows and Linux servers so that faculty and students have proper access permissions.

## III. Procedures

### 1. Windows Server – User & Group Management (No GUI)

Part 1 – Create a New User

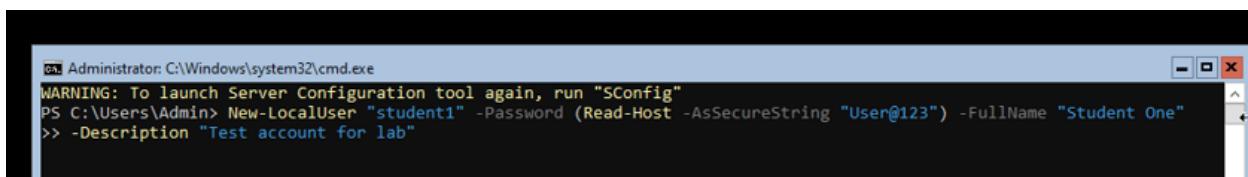
Run in PowerShell (as Administrator):

```
net user student1 User@123 /add
```

- Creates user student1 with password User@123.

```
net user student2 User@123 /add
```

- Creates user student2 with password User@123.



The screenshot shows a Windows Server command prompt window titled "Administrator: C:\Windows\system32\cmd.exe". It displays the following PowerShell command and its output:

```
WARNING: To launch Server Configuration tool again, run "SConfig"
PS C:\Users\Admin> New-LocalUser "student1" -Password (Read-Host -AsSecureString "User@123") -FullName "Student One"
>> -Description "Test account for lab"
```

Below the command prompt window, a separate black terminal window shows the results of the second command:

```
WARNING: To launch Server Configuration tool again, run "SConfig"
PS C:\Users\Admin> net user student2 User@123 /add
The command completed successfully.
```

## Part 2 – Modify User Properties

Add a description for student1:

```
wmic useraccount where name="student1" set description="Test account for Week 3 Lab"
```

Force password change on first login:

```
net user student1 /logonpasswordchg:yes
```

```
PS C:\Users\Admin> Set-LocalUser -Name "student1" -Description "Test account for lab"
PS C:\Users\Admin> Get-LocalUser

Name          Enabled Description
----          ----- -----
Admin         True
Administrator True   Built-in account for administering the computer/domain
DefaultAccount False  A user account managed by the system.
Guest          False  Built-in account for guest access to the computer/domain
student1      True   Test account for lab
WDAGUtilityAccount False A user account managed and used by the system for Windows Defender Application Guard sce...
```

## Part 3 – Create a Group

Create new group **LabUsers**:

```
net localgroup LabUsers /add
```

Add both students to the group:

```
net localgroup LabUsers student1 /add
```

```
net localgroup LabUsers student2 /add
```

```
PS C:\Select Administrator: C:\Windows\system32\cmd.exe
PS C:\Users\Admin> Set-LocalUser -Name "student1" -Description "Test account for lab"
PS C:\Users\Admin> Get-LocalUser

Name          Enabled Description
----          ----- -----
Admin         True
Administrator True   Built-in account for administering the computer/domain
DefaultAccount False  A user account managed by the system.
Guest          False  Built-in account for guest access to the computer/domain
student1      True   Test account for lab
WDAGUtilityAccount False A user account managed and used by the system for Windows Defender Application Guard sce...
```

```
PS C:\Users\Admin> Set-LocalUser -Name "student1" -UserMayChangePassword $true -PasswordNeverExpires $false
PS C:\Users\Admin> New-LocalGroup -Name "LabUsers" -Description "Lab Test Group"

Name          Description
----          -----
LabUsers     Lab Test Group

PS C:\Users\Admin> Add-LocalGroupMember -Group "LabUsers" -Member "student1"
PS C:\Users\Admin> Get-LocalGroupMember -Group "LabUsers"

ObjectClass Name          PrincipalSource
----- ----          -----
User        WINDOWS$SERVER\student1 Local

PS C:\Users\Admin>
```

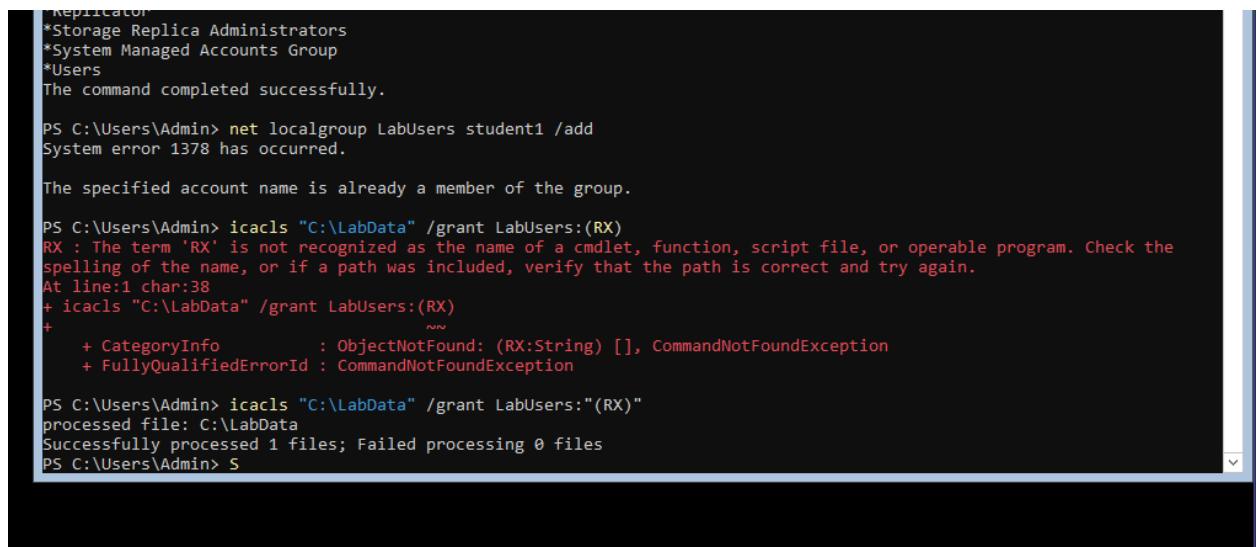
## Part 4 – Apply Folder Permissions

Create a folder:

```
mkdir C:\LabData
```

Set NTFS permissions so only LabUsers have Read & Execute:

```
icacls C:\LabData /grant LabUsers:(RX)
```



```
Replicator
*Storage Replica Administrators
*System Managed Accounts Group
*Users
The command completed successfully.

PS C:\Users\Admin> net localgroup LabUsers student1 /add
System error 1378 has occurred.

The specified account name is already a member of the group.

PS C:\Users\Admin> icacls "C:\LabData" /grant LabUsers:(RX)
RX : The term 'RX' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:38
+ icacls "C:\LabData" /grant LabUsers:(RX)
+ ~~~
+ CategoryInfo          : ObjectNotFound: (RX:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Admin> icacls "C:\LabData" /grant LabUsers:"(RX)"
processed file: C:\LabData
Successfully processed 1 files; Failed processing 0 files
PS C:\Users\Admin> S
```

## Part 5 – Verification

Test with student1:

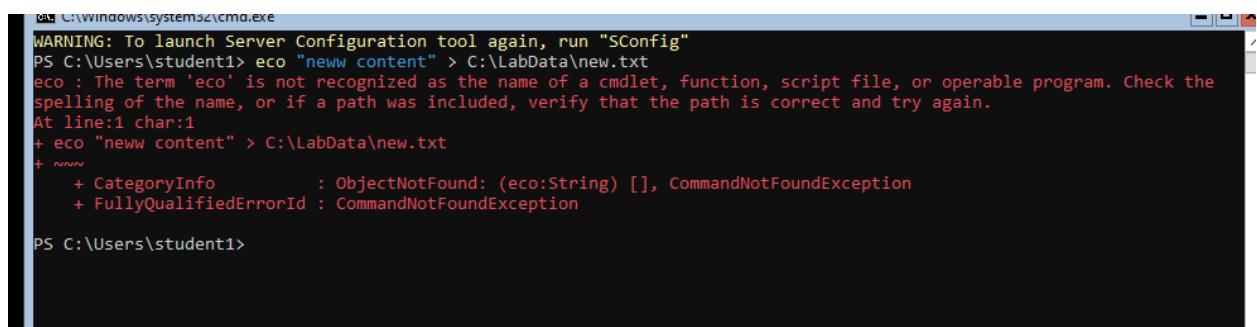
Login as .\student1

Check file contents (should work):

```
type C:\LabData\file.txt
```

Try to create a file (should fail):

```
echo "test" > C:\LabData\new.txt
```



```
C:\> C:\Windows\system32\cmd.exe
WARNING: To launch Server Configuration tool again, run "SConfig"
PS C:\Users\student1> echo "neww content" > C:\LabData\new.txt
echo : The term 'echo' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ echo "neww content" > C:\LabData\new.txt
+ ~~~
+ CategoryInfo          : ObjectNotFound: (echo:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\student1>
```

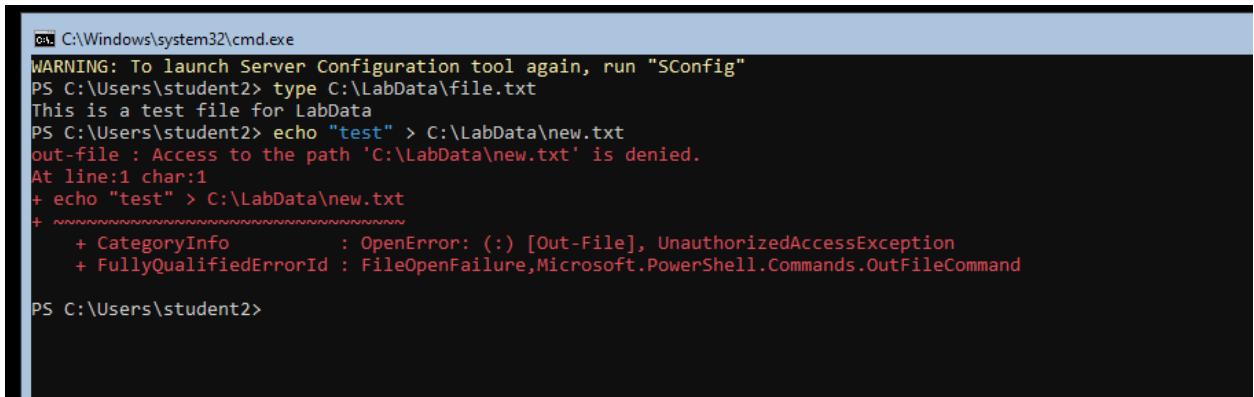
**Test with student2:**

Login as .\student2

Repeat the same:

type C:\LabData\file.txt

echo "test2" > C:\LabData\another.txt



```
C:\Windows\system32\cmd.exe
WARNING: To launch Server Configuration tool again, run "SConfig"
PS C:\Users\student2> type C:\LabData\file.txt
This is a test file for LabData
PS C:\Users\student2> echo "test" > C:\LabData\new.txt
out-file : Access to the path 'C:\LabData\new.txt' is denied.
At line:1 char:1
+ echo "test" > C:\LabData\new.txt
+ ~~~~~
+ CategoryInfo          : OpenError: (:) [Out-File], UnauthorizedAccessException
+ FullyQualifiedErrorId : FileOpenFailure,Microsoft.PowerShell.Commands.OutFileCommand

PS C:\Users\student2>
```

- This matches your lab manual, just adapted for Server Core (no GUI).

## Windows Server – Exercise for Students (No GUI)

### Part 1 – Create Users

Create two new users with passwords:

```
net user faculty1 User@123 /add
```

```
net user student3 User@123 /add
```

```
More help is available by typing NET HELPMSG 3506.

PS C:\Users\Admin> net user faculty1 Faculty@123 /add
The command completed successfully.

PS C:\Users\Admin> net user student3 User@123 /add
The command completed successfully.
```

### Part 2 – Create Groups

Create groups FacultyGroup and StudentGroup:

```
net localgroup FacultyGroup /add
```

```
net localgroup StudentGroup /add
```

```
PS C:\Users\Admin> net localgroup FacultyGroup /add
The command completed successfully.

PS C:\Users\Admin> net localgroup StudentGroup /add
The command completed successfully.
```

### Part 3 – Add Users to Groups

Assign each user to their respective group:

```
net localgroup FacultyGroup faculty1 /add
```

```
net localgroup StudentGroup student3 /add
```

```
PS C:\Users\Admin> net localgroup FacultyGroup faculty1 /add
The command completed successfully.

PS C:\Users\Admin> net localgroup StudentGroup student3 /add
The command completed successfully.
```

## Part 4 – Create Folders

Make two directories:

```
mkdir C:\FacultyData
```

```
mkdir C:\StudentData
```

```
PS C:\Users\Admin> mkdir C:\FacultyData

Directory: C:\

Mode          LastWriteTime         Length Name
----          -----          ----- 
d---          9/2/2025   3:09 PM           FacultyData
```

```
PS C:\Users\Admin> mkdir C:\StudentData

Directory: C:\

Mode          LastWriteTime         Length Name
----          -----          ----- 
d---          9/2/2025   3:10 PM           StudentData
```

## Part 5 – Apply Permissions

Grant FacultyGroup Modify rights on C:\FacultyData:

```
icacls C:\FacultyData /grant FacultyGroup:(M)
```

Grant StudentGroup Read-only rights on C:\StudentData:

```
icacls C:\StudentData /grant StudentGroup:(RX)
```

```
PS C:\Users\Admin> icacls C:\FacultyData /grant FacultyGroup:"(M)"
processed file: C:\FacultyData
Successfully processed 1 files; Failed processing 0 files
PS C:\Users\Admin> icacls C:\StudentData /grant StudentGroup:"(R)"
processed file: C:\StudentData
Successfully processed 1 files; Failed processing 0 files
PS C:\Users\Admin>
```

## Part 6 – Verification

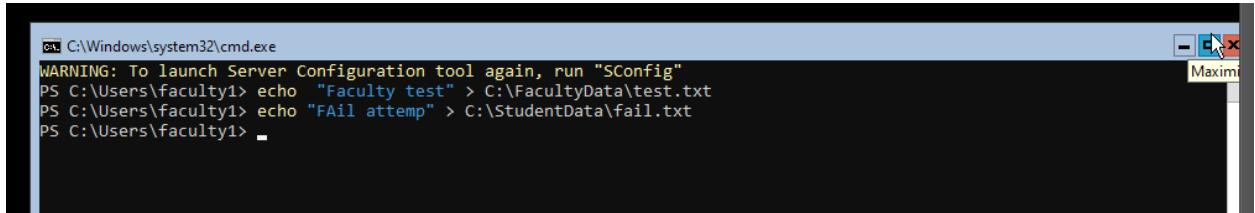
Test as faculty1:

- Should be able to create/modify files in C:\FacultyData.
- Should be denied access to C:\StudentData.

```
echo "Faculty test" > C:\FacultyData\test.txt
```

```
type C:\FacultyData\test.txt
```

```
echo "Fail attempt" > C:\StudentData\fail.txt
```



The screenshot shows a Windows Command Prompt window titled 'cmd' with the path 'C:\Windows\system32\cmd.exe'. It displays the following command-line session:

```
C:\Windows\system32\cmd.exe
WARNING: To launch Server Configuration tool again, run "SConfig"
PS C:\Users\faculty1> echo "Faculty test" > C:\FacultyData\test.txt
PS C:\Users\faculty1> echo "Fail attempt" > C:\StudentData\fail.txt
PS C:\Users\faculty1> -
```

**Test as student3:**

- Should only be able to read from C:\StudentData.
- Should not create files in C:\StudentData.
- Should not access C:\FacultyData.

```
type C:\StudentData\test.txt
```

```
echo "Student attempt" > C:\StudentData\new.txt
```

```
echo "Fail faculty access" > C:\FacultyData\fail2.txt
```

```
C:\Windows\system32\cmd.exe
WARNING: To launch Server Configuration tool again, run "SConfig"
PS C:\Users\student3> type C:\StudentData\test.txt
type : Cannot find path 'C:\Users\student3\StudentData\test.txt' because it does not exist.
At line:1 char:1
+ type C:\StudentData\test.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\student3\StudentData\test.txt:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetContentCommand

PS C:\Users\student3> echo "Student attempt" > C:\StudentData\new.txt
PS C:\Users\student3> type C:\StudentData\test.txt
type : Cannot find path 'C:\StudentData\test.txt' because it does not exist.
At line:1 char:1
+ type C:\StudentData\test.txt
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\StudentData\test.txt:String) [Get-Content], ItemNotFoundException
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.GetContentCommand

PS C:\Users\student3> echo "Fail faculty access" > C:\FacultyData\fail2.txt
PS C:\Users\student3>
```

## 2. Ubuntu Server (User and Group Management)

### Part 1 – Create Users

Run the following commands to create two users:

```
sudo adduser faculty2
```

```
sudo adduser student4
```

This creates two accounts: faculty2 and student4.



```
Ubuntu_Server [Running] - Oracle VirtualBox
File Machine View Input Devices Help

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

admin@UbuntuServer:$ sudo adduser faculty2
[sudo] password for admin:
info: Adding user `faculty2' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `faculty2' (1001) ...
info: Adding new user `faculty2' (1001) with group `faculty2 (1001)' ...
info: Creating home directory `/home/faculty2' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for faculty2
Enter the new value, or press ENTER for the default
  Full Name []: faculty2
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
info: Adding new user `faculty2' to supplemental / extra groups `users' ...
info: Adding user `faculty2' to group `users' ...
admin@UbuntuServer:$ sudo adduser student4
info: Adding user `student4' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `student4' (1002) ...
info: Adding new user `student4' (1002) with group `student4 (1002)' ...
info: Creating home directory `/home/student4' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for student4
Enter the new value, or press ENTER for the default
  Full Name []: student4
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
info: Adding new user `student4' to supplemental / extra groups `users' ...
info: Adding user `student4' to group `users' ...
admin@UbuntuServer:~$
```

## Part 2 – Create Groups and Assign Users

Run the following commands to create two groups:

```
sudo groupadd facultygrp
```

```
sudo groupadd studentgrp
```

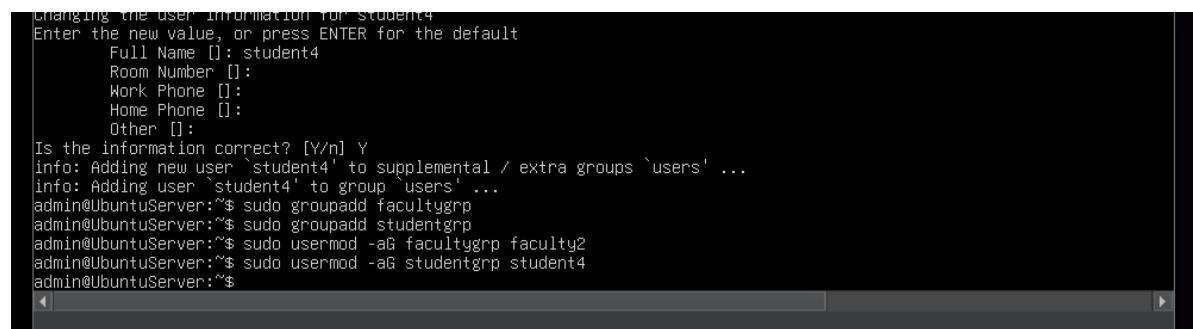
This creates the groups facultygrp and studentgrp.

Assign each user to their group:

```
sudo usermod -aG facultygrp faculty2
```

```
sudo usermod -aG studentgrp student4
```

This places faculty2 in the faculty group and student4 in the student group.



```
Changing the user information for student4
Enter the new value, or press ENTER for the default
  Full Name []: student4
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] Y
info: Adding new user 'student4' to supplemental / extra groups `users' ...
info: Adding user `student4' to group `users' ...
admin@UbuntuServer:~$ sudo groupadd facultygrp
admin@UbuntuServer:~$ sudo groupadd studentgrp
admin@UbuntuServer:~$ sudo usermod -aG facultygrp faculty2
admin@UbuntuServer:~$ sudo usermod -aG studentgrp student4
admin@UbuntuServer:~$
```

## Part 3 – Create Directories and Apply Permissions

Run the following commands to create directories:

```
sudo mkdir /facultydata
```

```
sudo mkdir /studentdata
```

Set ownership and permissions:

```
# Faculty group – read/write
```

```
sudo chown :facultygrp /facultydata
```

```
sudo chmod 770 /facultydata
```

```
# Student group – read-only
```

```
sudo chown :studentgrp /studentdata
```

```
sudo chmod 750 /studentdata
```

This gives facultygrp full access to /facultydata and studentgrp read-only access to /studentdata.

```
info: Adding new user 'student4' to supplemental / extra groups: 'users' ...
info: Adding user 'student4' to group 'users' ...
admin@UbuntuServer:~$ sudo groupadd facultygrp
admin@UbuntuServer:~$ sudo groupadd studentgrp
admin@UbuntuServer:~$ sudo usermod -aG facultygrp faculty2
admin@UbuntuServer:~$ sudo usermod -aG studentgrp student4
admin@UbuntuServer:~$ sudo mkdir /facultydata
admin@UbuntuServer:~$ sudo mkdir /studentdata
admin@UbuntuServer:~$ sudo chown :facultygrp /facultydata
admin@UbuntuServer:~$ sudo chmod 770 /facultydata
admin@UbuntuServer:~$ sudo chown :studentgrp /studentdata
admin@UbuntuServer:~$ sudo chmod 550 /studentdata
admin@UbuntuServer:~$ groups faculty2
faculty2 : faculty2 users facultygrp
admin@UbuntuServer:~$ groups student4
student4 : student4 users studentgrp
admin@UbuntuServer:~$
```

## Part 4 – Testing and Verification

Testing was performed by logging in with the created accounts:

- faculty2 was able to read and write files in /facultydata.
- student4 was able to read files in /studentdata but could not modify them.

This confirmed that the permissions were applied correctly.

```
admin@UbuntuServer: $ groups student4
student4 : student4 users studentgrp
admin@UbuntuServer:~$ su - student4
Password:
student4@UbuntuServer:~$ whoami
student4
student4@UbuntuServer:~$ groups
student4 users studentgrp
student4@UbuntuServer:~$ cd /studentdata
student4@UbuntuServer:/studentdata$ ls
student4@UbuntuServer:/studentdata$ cat file
cat: file: No such file or directory
student4@UbuntuServer:/studentdata$ touch newfile.txt
touch: cannot touch 'newfile.txt': Permission denied
student4@UbuntuServer:/studentdata$ su - faculty2
Password:
su: Authentication failure
student4@UbuntuServer:/studentdata$ su - faculty2
Password:
faculty2@UbuntuServer:~$ cd /facultydata
faculty2@UbuntuServer:/facultydata$ ls
faculty2@UbuntuServer:/facultydata$ touch test.txt
faculty2@UbuntuServer:/facultydata$
```

## **IV. Results**

After performing the procedures:

1. Two users were successfully created:
  - faculty2
  - student4
2. The users were verified in the system by checking the /etc/passwd file.
3. Both accounts were able to log in using their respective credentials.
4. Group permissions and access restrictions worked as expected, allowing proper separation between users.

## **V. Conclusion**

In this activity, I learned how to manage user accounts in Ubuntu by creating, verifying, and testing multiple users. Through this process, I understood the importance of:

- Properly creating and configuring accounts for different roles.
- Using Linux commands like adduser, id, and checking /etc/passwd for verification.
- Setting permissions to ensure security and organized system management.

This hands-on practice emphasized how user and group management is essential in real-world system administration.