UnderSupervision: Eng. Romany
By: Amira Mhammad

Interactive Linux User & Group Management Tool

(An easy-to-use Bash script with a whiptail-based menu)

🛠 Project Idea:

This script is a **Bash-based interactive tool** using the whiptail dialog interface. It helps in managing users and groups on a Linux system with a user-friendly GUI-like experience, instead of typing complex terminal commands manually.

🔥 What the Script Can Do:

1. Add a new user with a password.
2. Modify a user (change name, UID, home directory,).
3. List all system users.
4. Create a new group.
5. Modify a group (rename, change GID, show members, set group password).
6. List all groups.
7. Enable (unlock) a user account.
8. Disable (lock) a user account.
9. Change a user's password.
10. Exit the program.

✳ Part 1: Add New User (Full Name – Username – Email – Password – Group)

Part 2: Modify User Data (Change Name – UID – Home Directory – Lock/Unlock Account)

amiramohamed@localhost:~/whiptailFunctions — sudo nano modify_user_module.sh

GNU nano 5.6.1 modify_user_module.sh

```
#!/bin/bash

user_to_edit=$(whiptail --title "User Editor" --inputbox "Enter the username to update:" 8 50 3>&1 1>&2 2>&3)

if getent passwd "$user_to_edit" > /dev/null; then
    selection=$(whiptail --title "User Modification" --menu "Select an option for '$user_to_edit':" 15 60 4 \
        "1" "Rename user" \
        "2" "Update UID" \
        "3" "Add to group" \
        "4" "Change login shell" 3>&1 1>&2 2>&3)

    case $selection in
        1)
            new_user=$(whiptail --title "Rename User" --inputbox "Enter new username:" 8 50 3>&1 1>&2 2>&3)
            if getent passwd "$new_user" > /dev/null; then
                whiptail --title "Name Taken" --msgbox "Username '$new_user' is already in use." 8 50
            else
                usermod -l "$new_user" "$user_to_edit"
                whiptail --title "Updated" --msgbox "Username changed to '$new_user'." 8 50
            fi
            ;;
        2)
            new_uid=$(whiptail --title "Change UID" --inputbox "Enter new UID for '$user_to_edit':" 8 50 3>&1 1>&2 2>&3)
            if getent passwd | cut -d: -f3 | grep -qx "$new_uid"; then
                whiptail --title "UID Exists" --msgbox "UID '$new_uid' is already assigned." 8 50
            else
                usermod -u "$new_uid" "$user_to_edit"
                whiptail --title "Success" --msgbox "UID updated to '$new_uid'." 8 50
            fi
            ;;
        3)
            grp_name=$(whiptail --title "Add Group" --inputbox "Enter group name to add user to:" 8 50 3>&1 1>&2 2>&3)
            if getent group "$grp_name" > /dev/null; then
                usermod -aG "$grp_name" "$user_to_edit"
                whiptail --title "Success" --msgbox "User '$user_to_edit' added to group '$grp_name'." 8 50
            else
                whiptail --title "Group Not Found" --msgbox "Group '$grp_name' does not exist." 8 50
            fi
            ;;
        4)
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo M-6 Copy

Part 3: List all users

amiramohamed@localhost:~/whiptailFunctions — sudo nano list_User_module.sh

GNU nano 5.6.1 list_User_module.sh

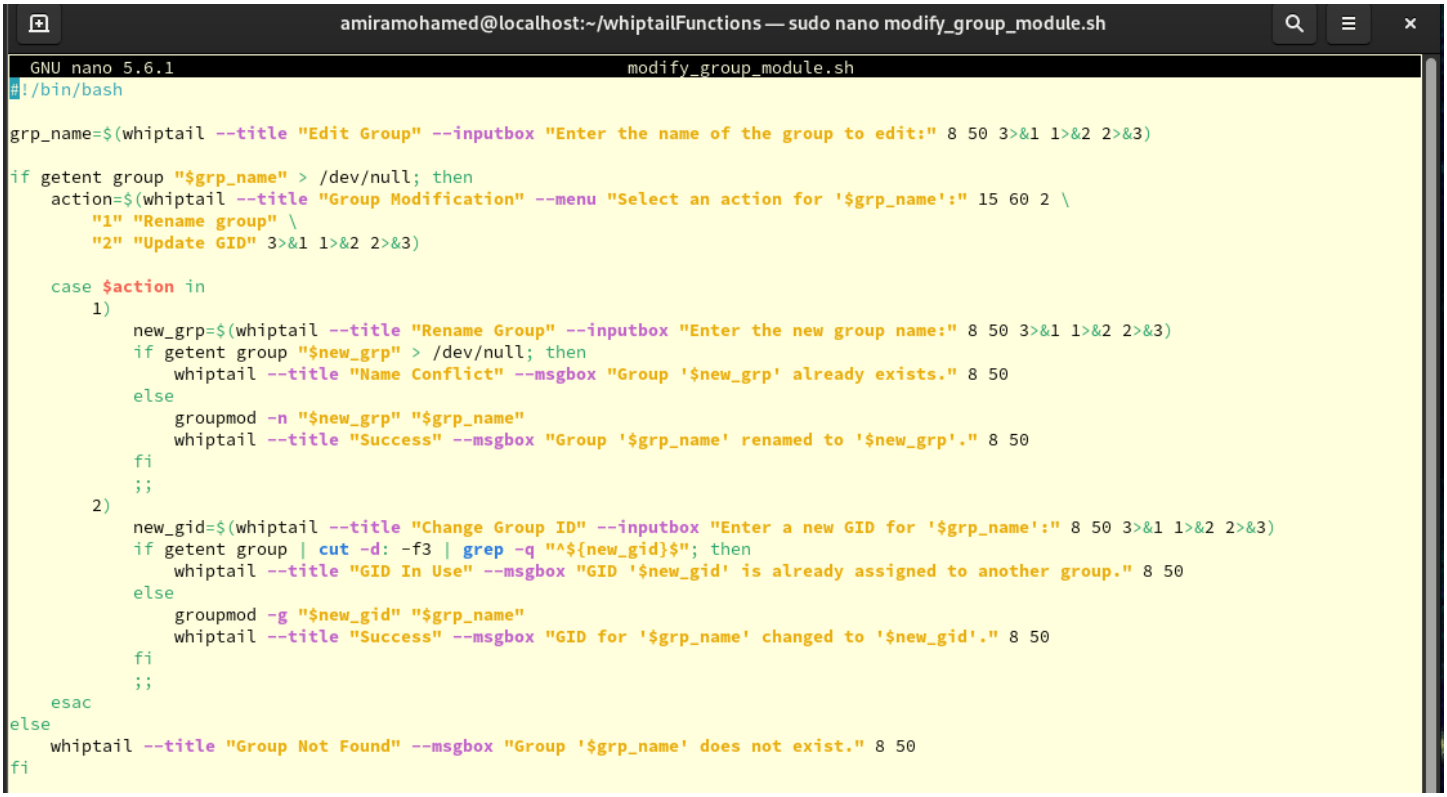
```
#!/bin/bash

choice=$(whiptail --title "User Listing Options" --menu "Select the type of listing:" 12 60 2 \
        "1" "Usernames Only" \
        "2" "Full User Info" 3>&1 1>&2 2>&3)

case "$choice" in
    1)
        user_list=$(cut -d: -f1 /etc/passwd)
        whiptail --title "Usernames" --scrolltext --msgbox "$user_list" 20 60
        ;;
    2)
        full_info=$(< /etc/passwd)
        whiptail --title "User Details" --scrolltext --msgbox "$full_info" 30 80
        ;;
esac
```

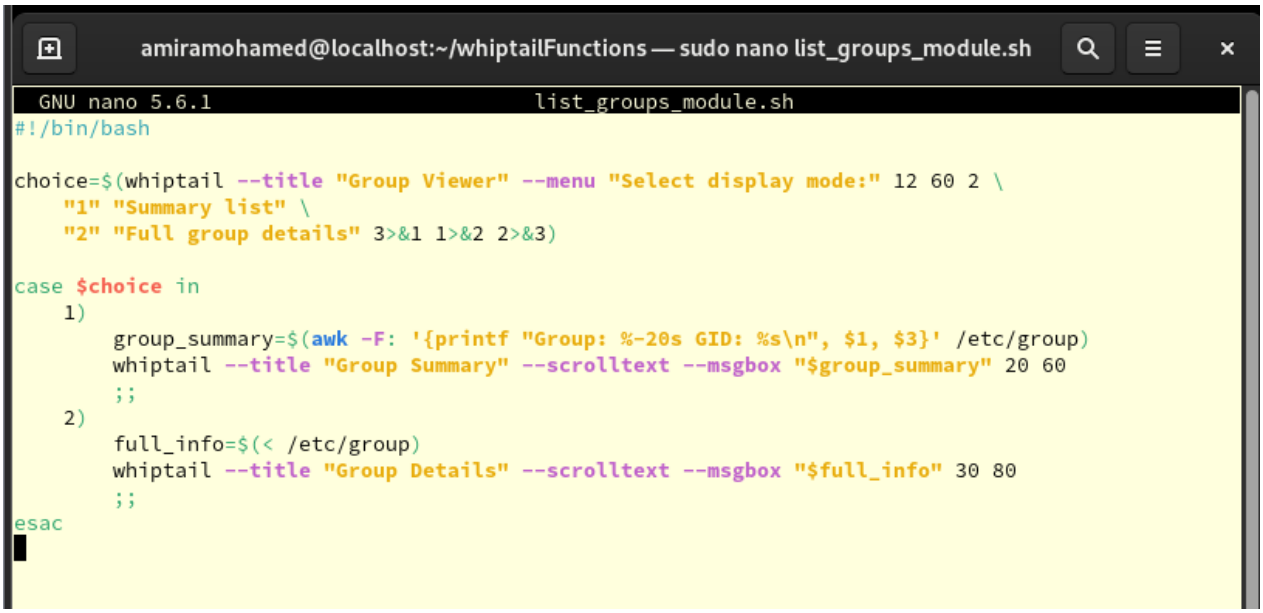
Part 4: Create New Group (Group Name)

Part 5: Modify Group (Rename – GID – Show Members – Set Password)

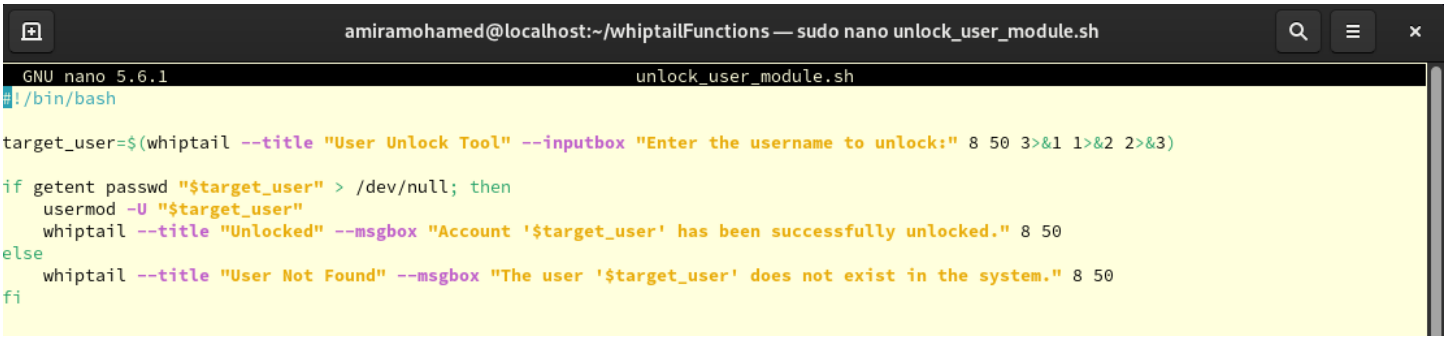


Part 6: List All Groups

Displays all the collections in the system in a formatted way using cut and sort, and displays them in a scrollable screen using less.



Part 7: Unlock a User Account



Part 8: Lock a User Account

Part 9: Change a User's Password

amiramohamed@localhost:~/whiptailFunctions — sudo nano change_user_password_module.sh

GNU nano 5.6.1 change_user_password_module.sh

```
#!/bin/bash

target_user=$(whiptail --title "Update User Password" --inputbox "Enter the username:" 8 50 3>&1 1>&2 2>&3)

if id "$target_user" &>/dev/null; then
    new_pass=$(whiptail --title "Set New Password" --passwordbox "Enter the new password for $target_user:" 8 50 3>&1 1>&2 2>&3)
    echo "$new_pass" | passwd "$target_user" --stdin
    whiptail --title "Password Changed" --msgbox "Password successfully updated for '$target_user'." 8 50
else
    whiptail --title "User Not Found" --msgbox "The user '$target_user' does not exist in the system." 8 50
fi
```

Part 10: Exit the Program

10)

whiptail --msgbox "Exiting program. Bye Amira! 😊" 10 50
exit
;;

Main Menu – User & Group Management (Call Modules – View Users – Manage Groups Lock/Unlock Accounts – Change Password – Exit)

GNU nano 5.6.1 main_user_menu.sh Modified

```
#!/bin/bash

# Check for root privileges
if [[ $EUID -ne 0 ]]; then
    whiptail --title "Permission Denied" --msgbox "This script must be run as root!" 8 50
    exit 1
fi

# Define modules directory
MODULES_DIR="./"

# Main menu loop
while true; do
    selection=$(whiptail --title "Admin Panel" --menu "Choose an action:" 20 60 10 --nocancel \
        "1" "Add a new user" \
        "2" "Modify a user" \
        "3" "List all users" \
        "4" "Add a new group" \
        "5" "Modify a group" \
        "6" "List all groups" \
        "7" "Lock a user account" \
        "8" "Unlock a user account" \
        "9" "Change user password" \
        "10" "Exit" 3>&1 1>&2 2>&3)

    case $selection in
        1) bash "${MODULES_DIR}add_user_module.sh" ;;
        2) bash "${MODULES_DIR}modify_user_module.sh" ;;
        3) bash "${MODULES_DIR}list_User_module.sh" ;;
        4) bash "${MODULES_DIR}create_group_module.sh" ;;
        5) bash "${MODULES_DIR}modify_group_module.sh" ;;
        6) bash "${MODULES_DIR}list_groups_module.sh" ;;
        7) bash "${MODULES_DIR}lock_user_module.sh" ;;
        8) bash "${MODULES_DIR}unlock_user_module.sh" ;;
        9) bash "${MODULES_DIR}change_user_password_module.sh" ;;
        10) exit 0 ;;
    esac
```

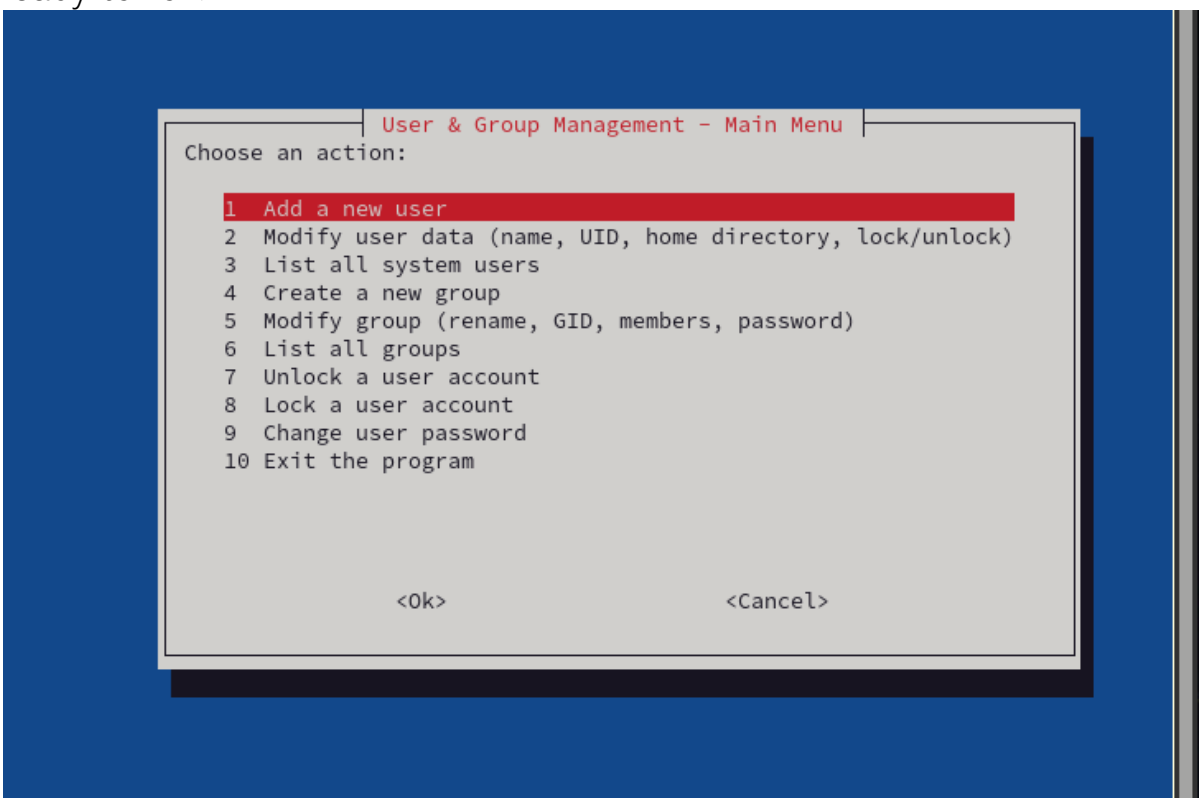
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo M-6 Copy

```
amiramohamed@localhost:~
[amiramohamed@localhost ~]$ sudo nano add_user_module.sh
[sudo] password for amiramohamed:
[amiramohamed@localhost ~]$ sudo nano modify_user_module.sh
[amiramohamed@localhost ~]$ sudo chmod +x add_user_module.sh
[amiramohamed@localhost ~]$ sudo chmod +x modify_user_module.sh
[amiramohamed@localhost ~]$ sudo nano main_user_menu.sh
[sudo] password for amiramohamed:
[amiramohamed@localhost ~]$ sudo chmod +x main_user_menu.sh
[amiramohamed@localhost ~]$ sudo nano create_group_module.sh
[sudo] password for amiramohamed:
[amiramohamed@localhost ~]$ sudo nano modify_group_module.sh
[amiramohamed@localhost ~]$ sudo chmod +x create_group_module.sh
[amiramohamed@localhost ~]$ sudo chmod +x modify_group_module.sh
[amiramohamed@localhost ~]$ sudo nano list_groups_module.sh
[sudo] password for amiramohamed:
[amiramohamed@localhost ~]$ sudo nano unlock_user_module.sh
[amiramohamed@localhost ~]$ sudo chmod +x unlock_user_module.sh
[amiramohamed@localhost ~]$ sudo nano lock_user_module.sh
[amiramohamed@localhost ~]$ sudo chmod +x lock_user_module.sh
[amiramohamed@localhost ~]$ sudo nano change_user_password_module.sh
[amiramohamed@localhost ~]$ chmod +x change_user_password_module.sh
chmod: changing permissions of 'change_user_password_module.sh': Operation not permitted
[amiramohamed@localhost ~]$ sudo chmod +x change_user_password_module.sh
[amiramohamed@localhost ~]$ ./main_user_menu.sh
[amiramohamed@localhost ~]$
```

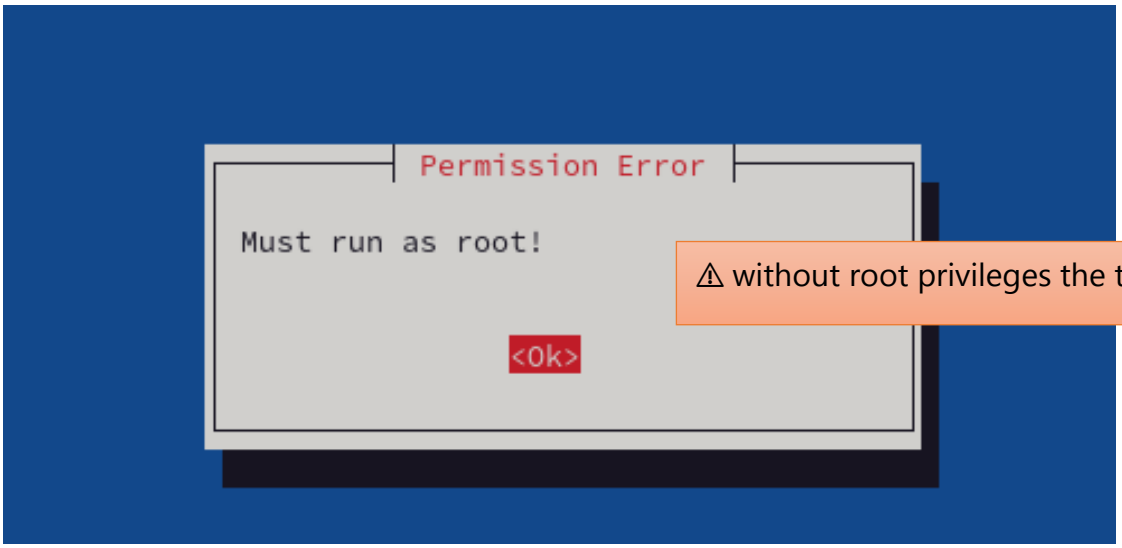
2️⃣ Make the 'main' file executable chmod +x main

3️⃣ Start the tool with superuser access sudo ./main

🚀 Ready to roll!



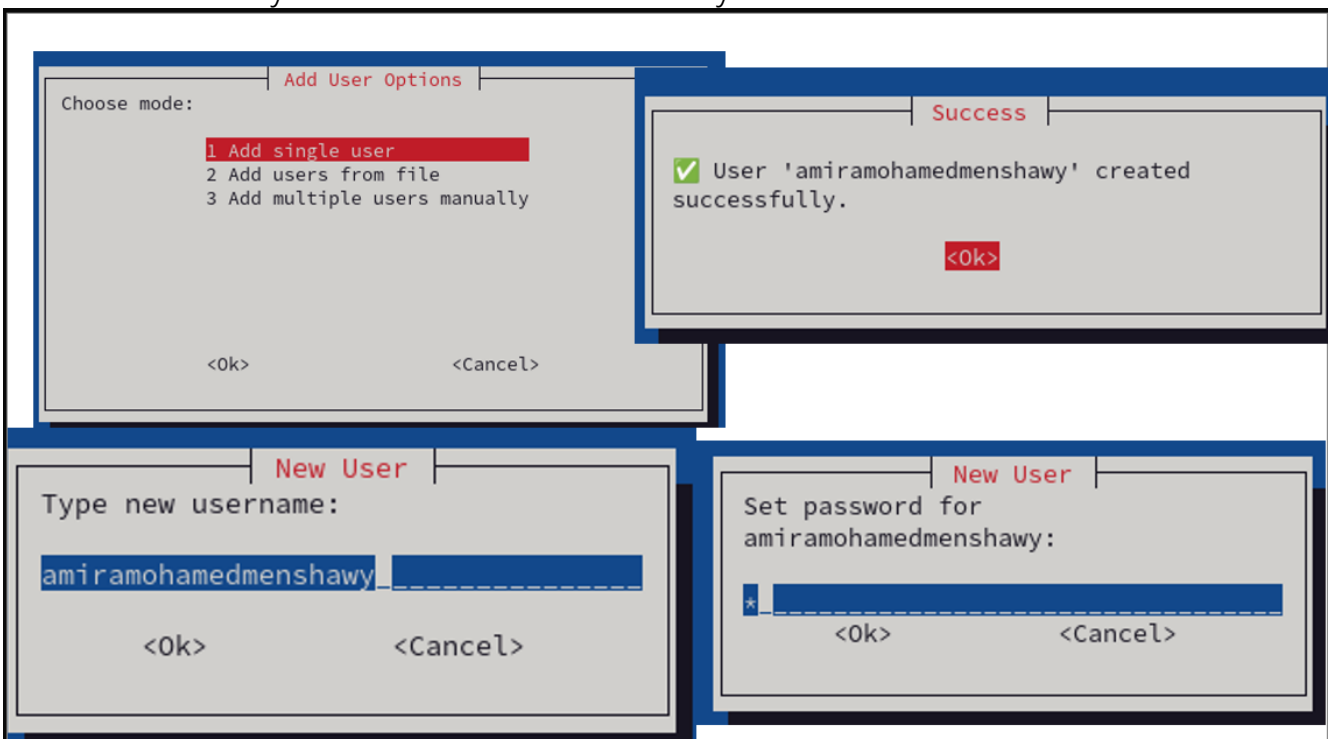
✳ Option 1: Add a new user



⚠ without root privileges the tool won't run

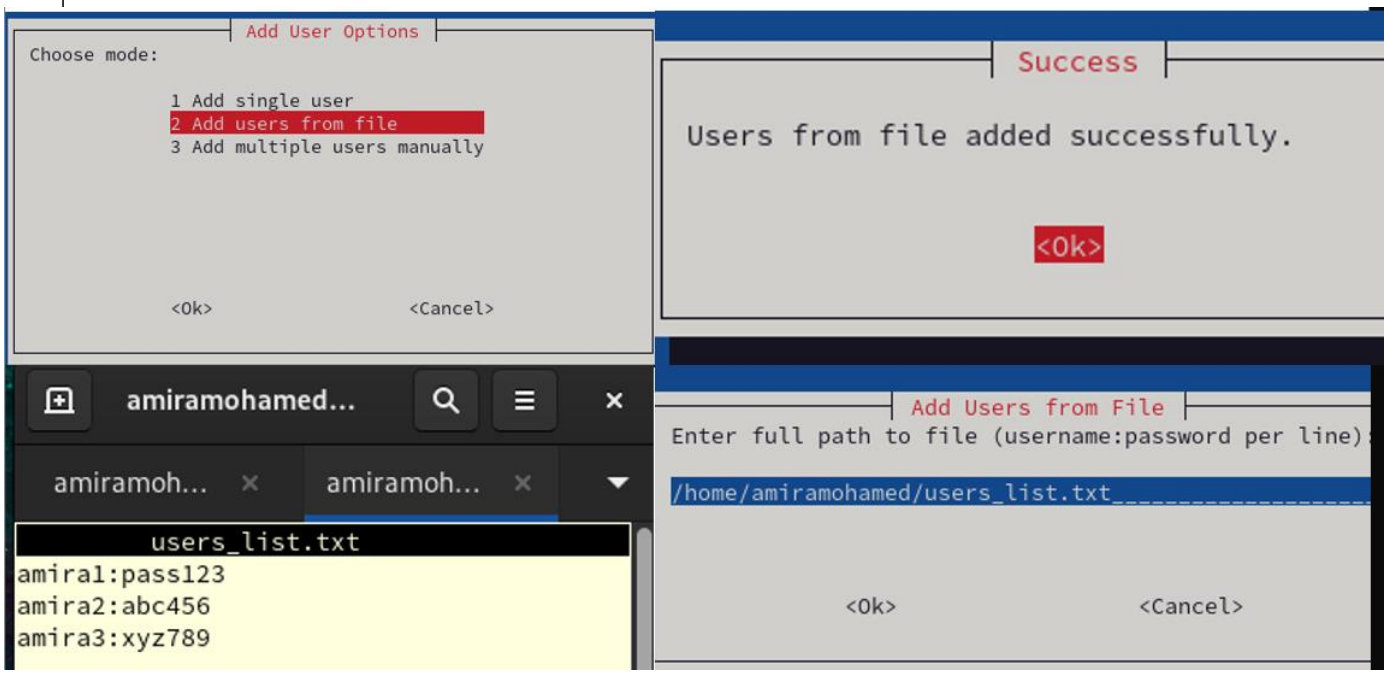
👤 Adding a New User
From the Main Menu, select "Add User" — you'll then pick from 3 available modes.

🧩 Mode 1: Add a User
This is a simple function that asks you to enter a username and password,
then automatically creates the user on the system.



📂 Mode 2: Add Users from a File
This mode simplifies the process by automatically creating users from a file (/home/amiramohamed/users_list.txt).

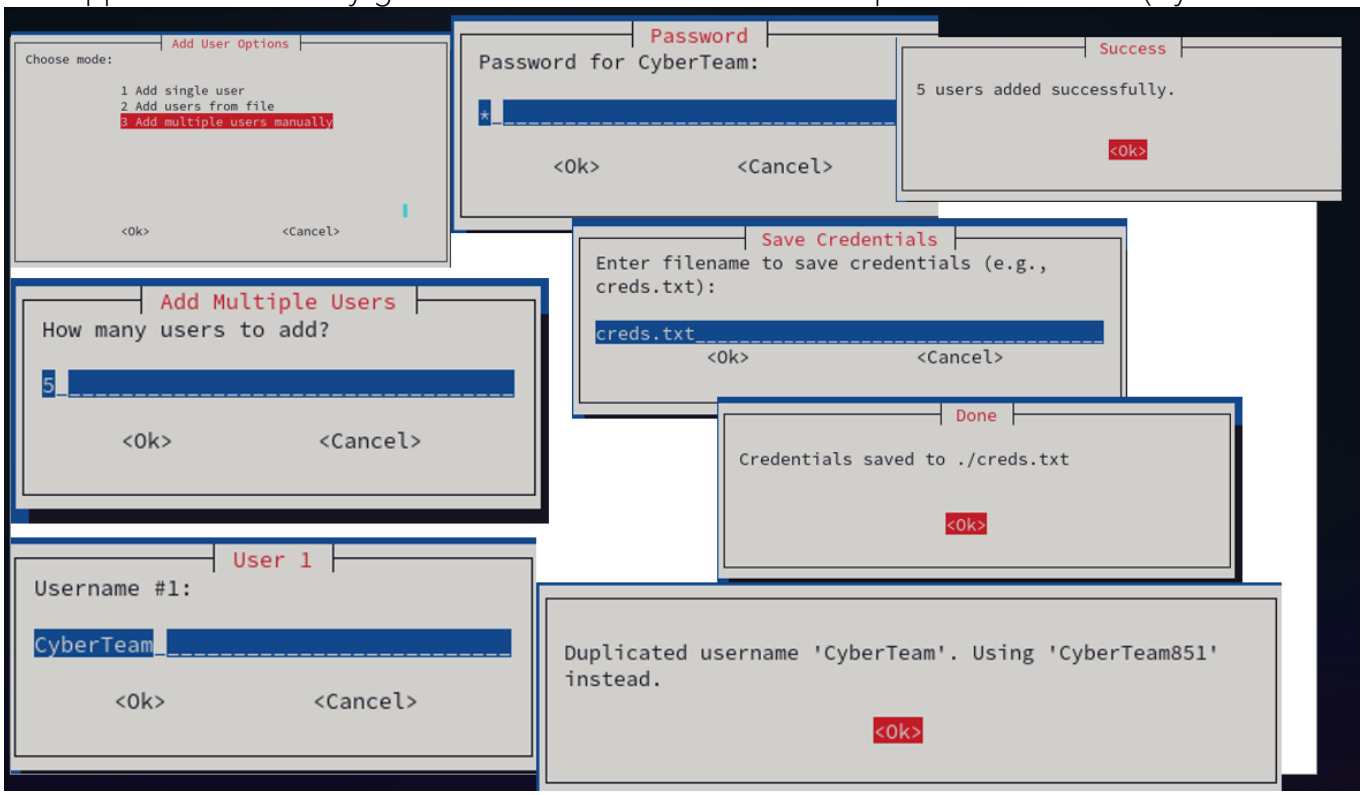
The tool reads the file, extracts the credentials, and creates the accounts accordingly.
🔄 If it detects a duplicate username, it appends a random number to the end to ensure uniqueness.



✨Mode 3 – Add Multiple Users Manually

🚀 This option is ideal for quickly setting up multiple user accounts for a team or department,
handling both account creation and credential storage automatically.

For example, the tool first creates a user named (CyberTeam).
When it attempts to create another user with the same name and detects a duplicate,
it appends a randomly generated number to create a unique username like (CyberTeam827).



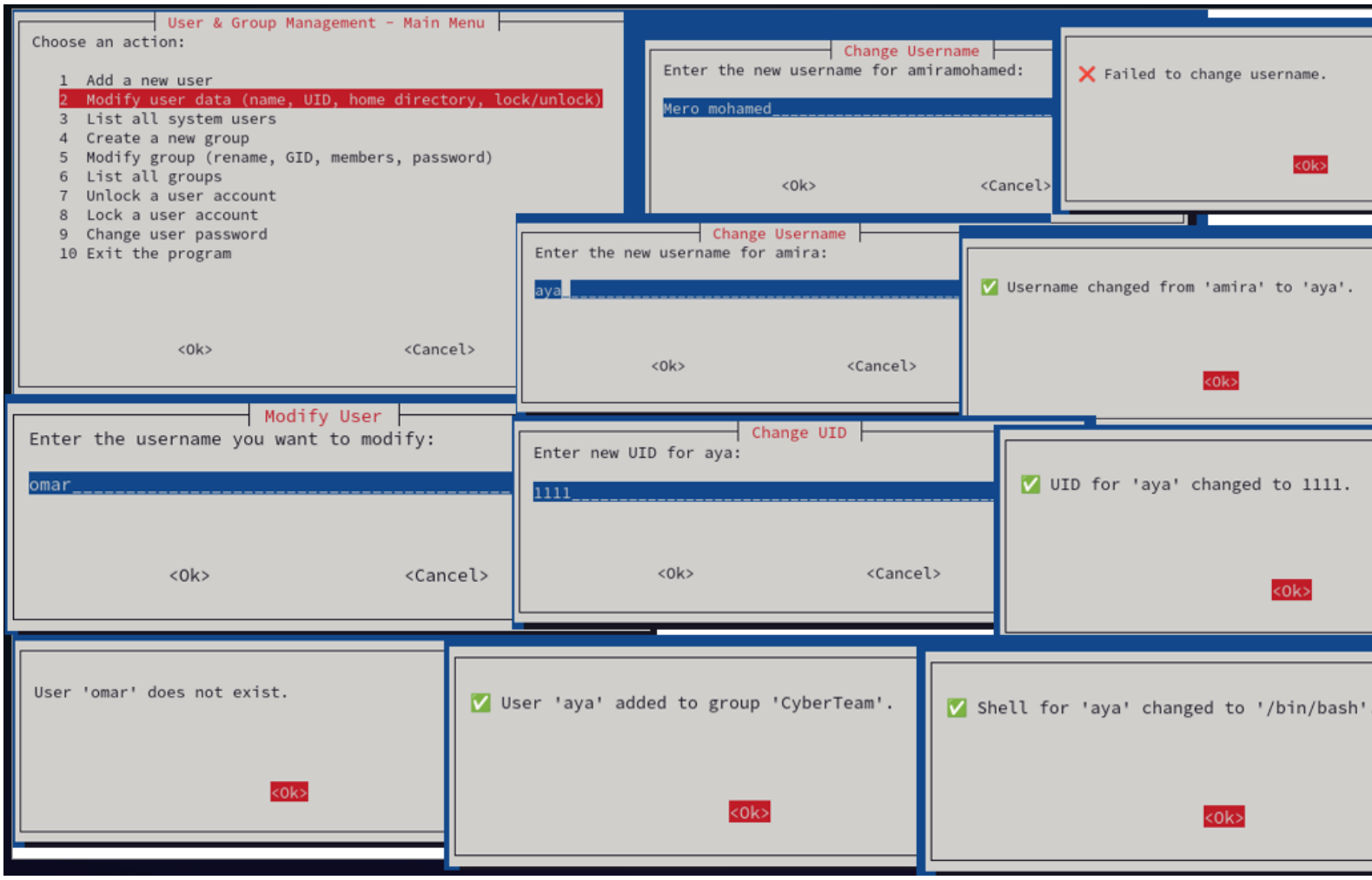
✳️ Option 2: Modify user data

🔍 If you enter a group name that doesn't exist, the tool will display an error message indicating the issue.

✅ When a valid group name is entered, the user will be successfully added to that group.

🐚 Option : Change Shell

This feature allows you to modify the user's default shell to a different one of your choice.

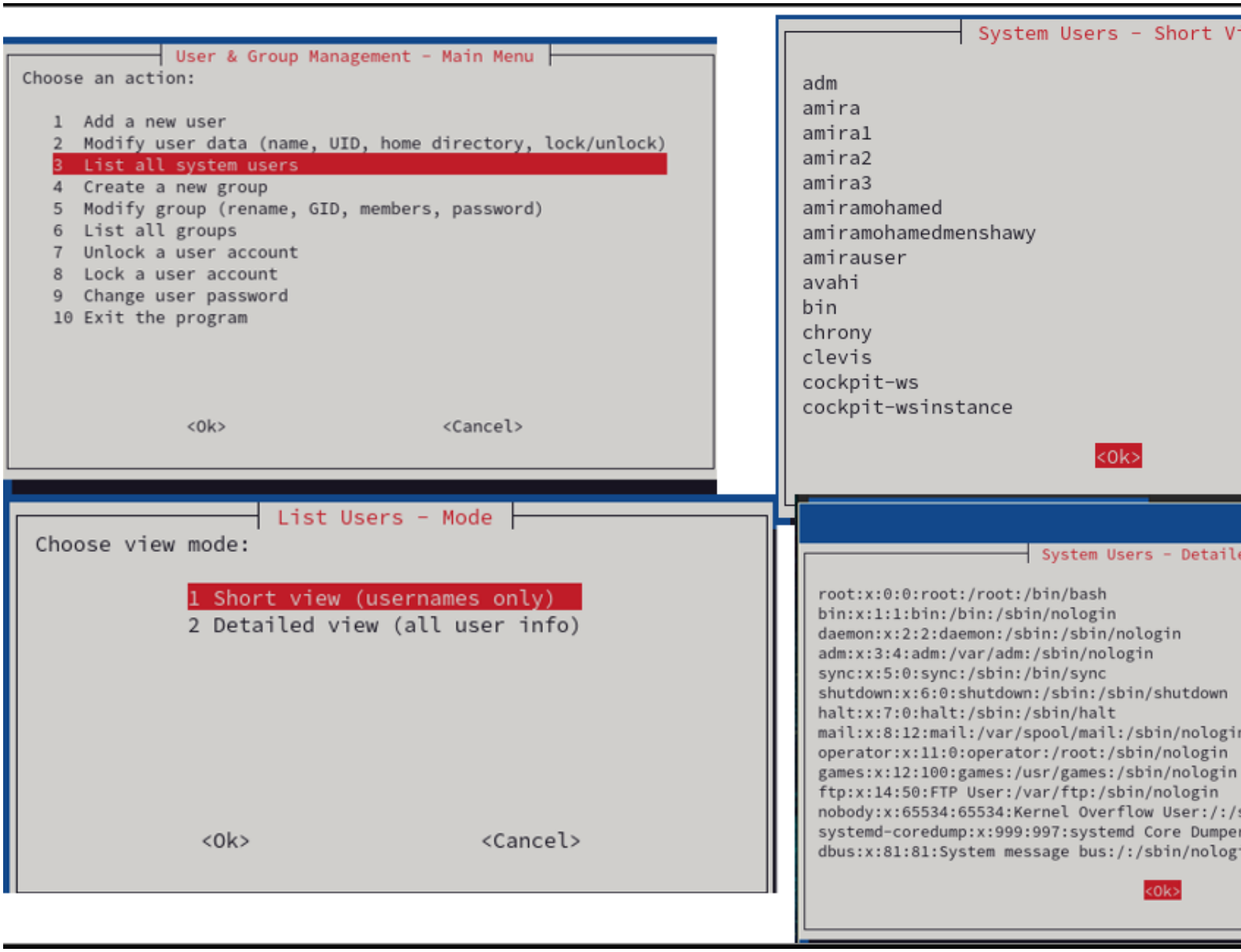


✳ Option 3: List all users

Selecting "List All Users" from the Main Menu will prompt you to choose a display mode:

🔹 Mode 1: Short View
Shows a simple list of all usernames on the system.

🔸 Mode 2: Detailed View
Displays comprehensive information about each user, including all available account details.

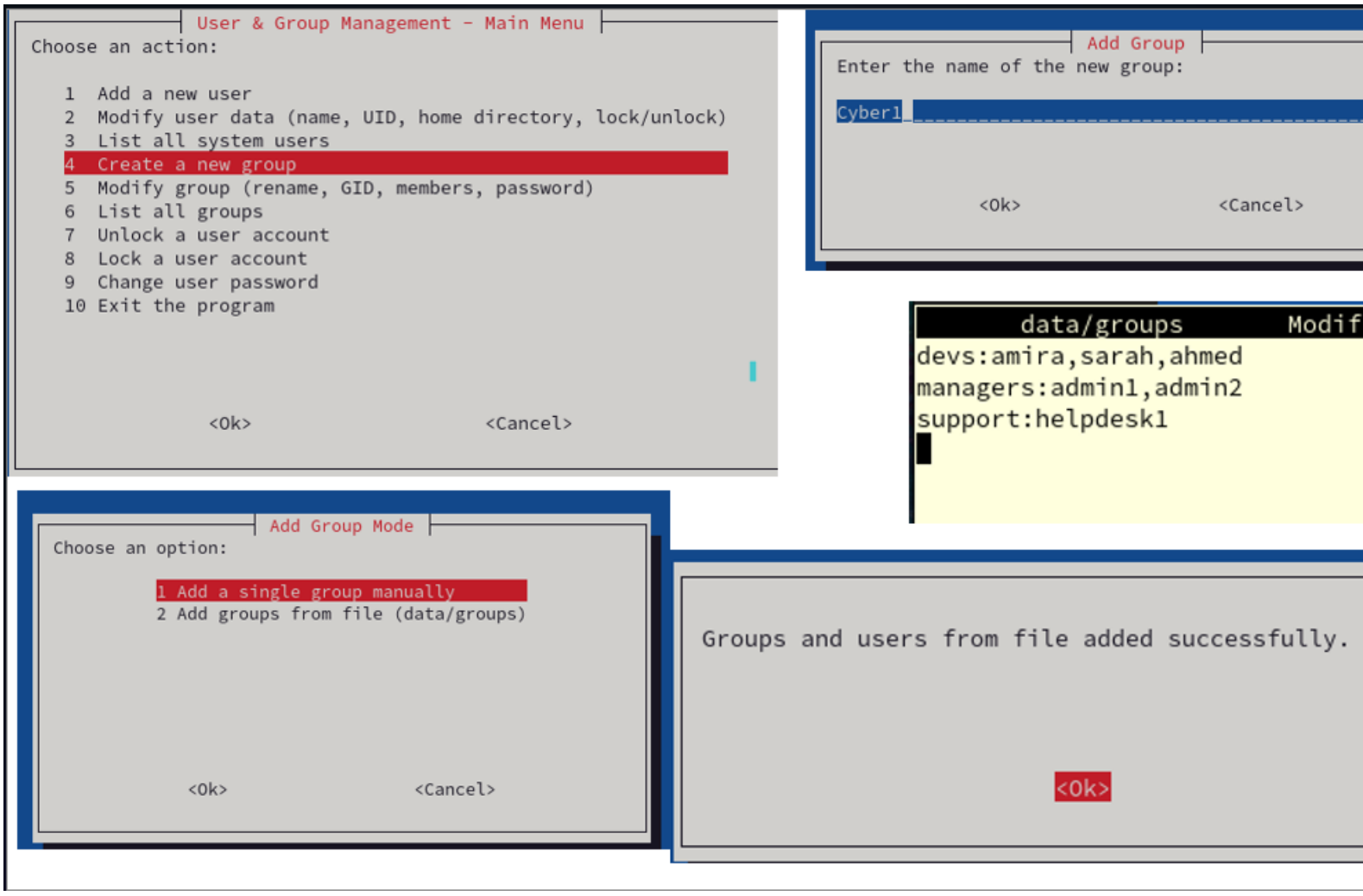


✳ Option 4: Add a Group

When selecting "Add a Group" from the Main Menu, you'll be asked to choose a mode:

🔹 Mode 1: Add a Group
A simple option that asks for a group name and creates it on the system.

🔸 Mode 2: Add Groups from a File
Automates the process by reading group info from the file `data/groups`.
The tool reads the group name, creates it, then adds its associated users.
👤 If a user mentioned in the file doesn't exist, the tool creates a new user account, appending a random number to ensure uniqueness.
📌 If the group already exists, it simply adds the users to that group.



✳️ Option 5: Modify a Group

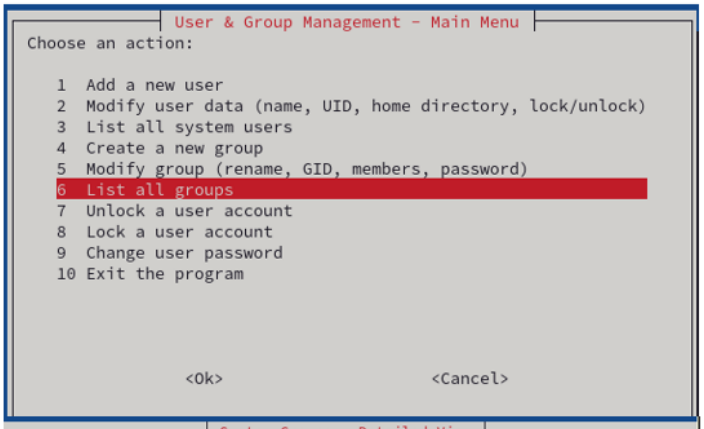
From the Main Menu, selecting "Modify a Group" will prompt you to enter the group name you wish to modify.

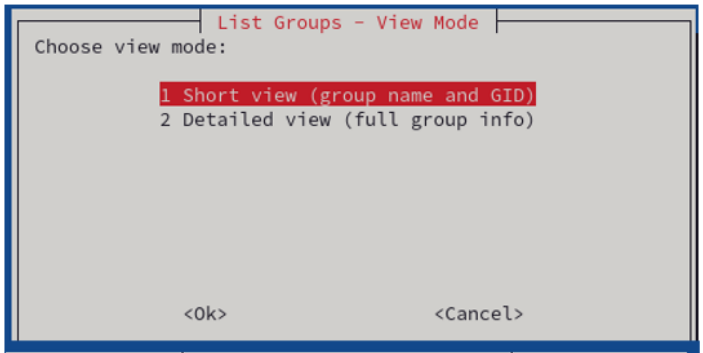
❌ If the group name doesn't exist, the tool will display an error message.
✅ If a valid group name is entered, a modification menu will appear with the following options:

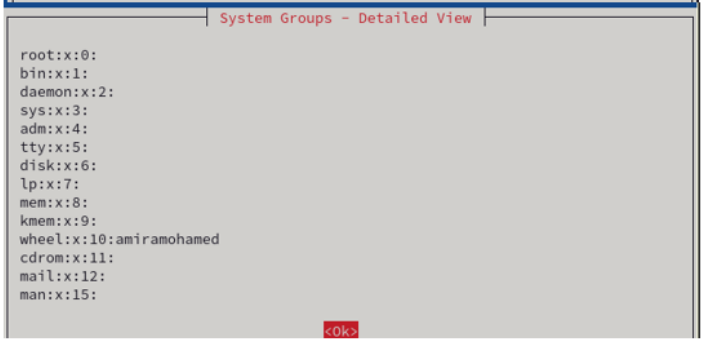
🔹 Option 1: Change Group Name
- If the new group name already exists, an error will be shown.
- Otherwise, the group name will be successfully updated.

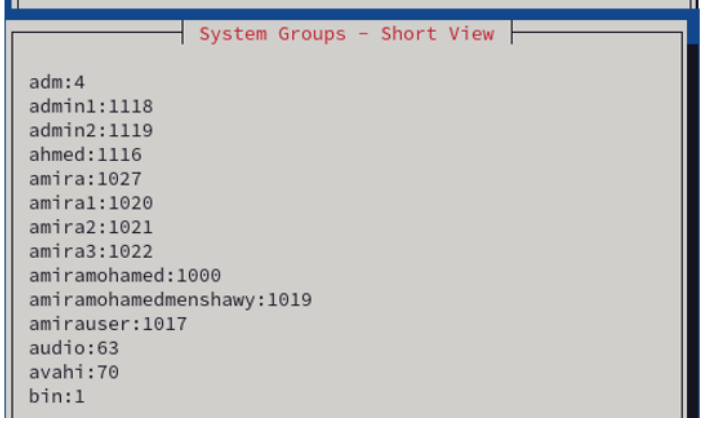
🔸 Option 2: Change Group ID (GID)
- If the entered GID is already in use by another group, the tool will display an error.
- If the GID is available, it will be assigned to the selected group without issues.

✳ Option 6: Lists all Groups





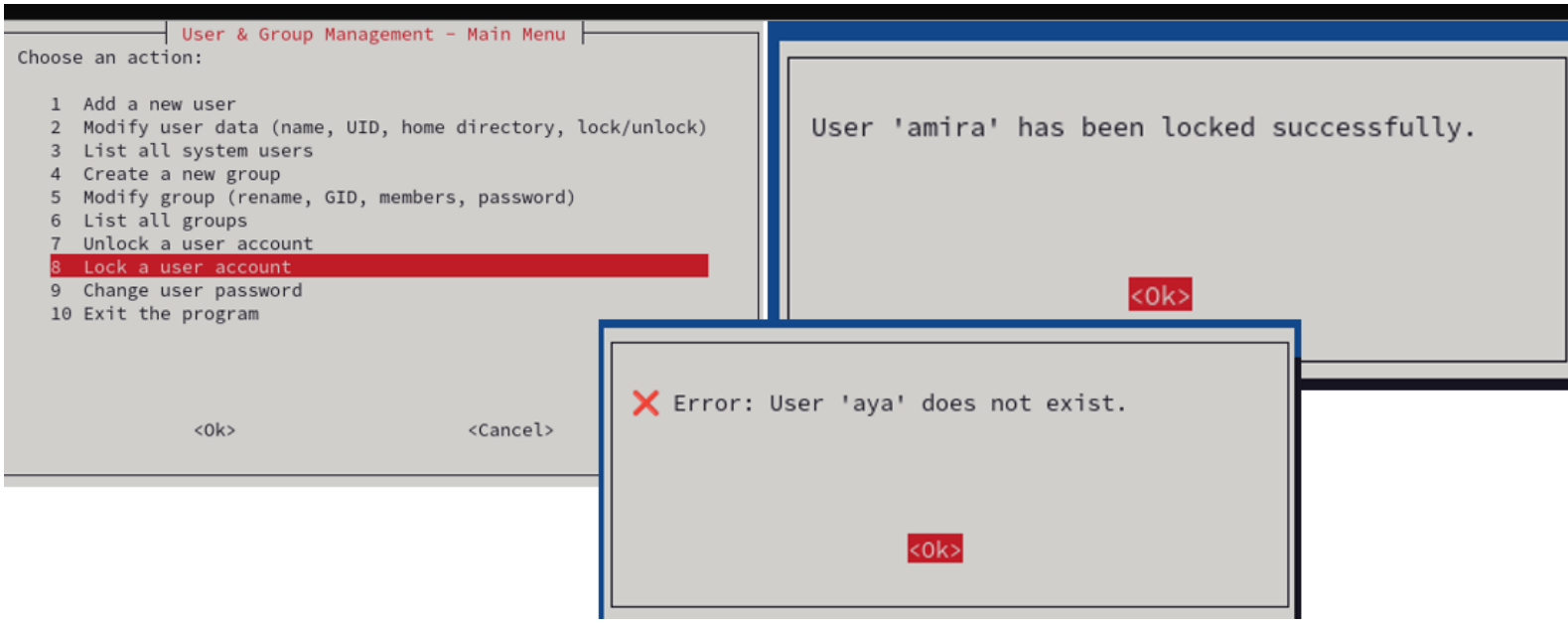




✳ Option 8 : Lock account

Selecting "Lock a User Account" from the Main Menu will prompt you to enter the target username.

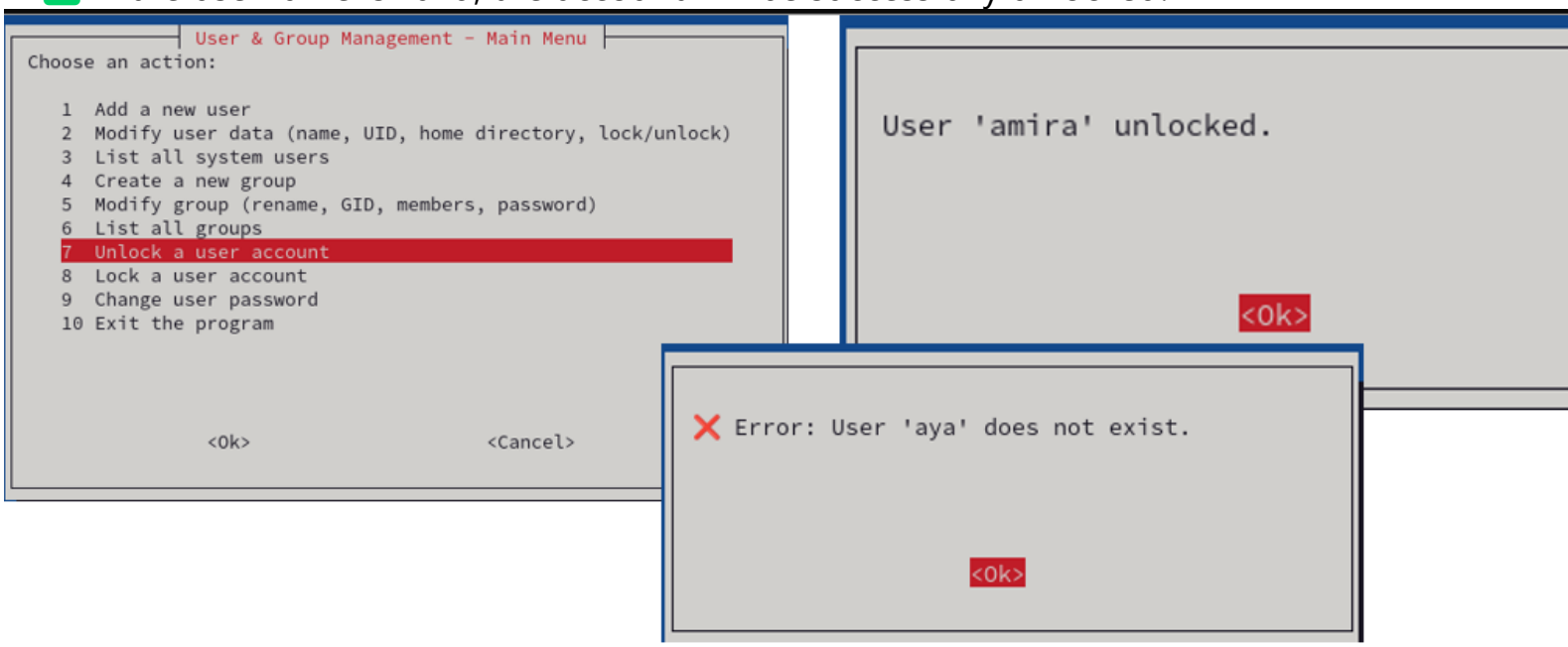
❌ If the entered username doesn't exist, the tool will display an error message.
✅ If the username is valid and exists on the system, the account will be successfully locked.



✳ Option 7 : UnLock account

Choosing "Unlock a User Account" from the Main Menu will prompt you to enter the username you want to unlock.

❌ If the username doesn't exist on the system, an error message will be shown.
✅ If the username is valid, the account will be successfully unlocked.



✳ Option 9 : Change a user's password