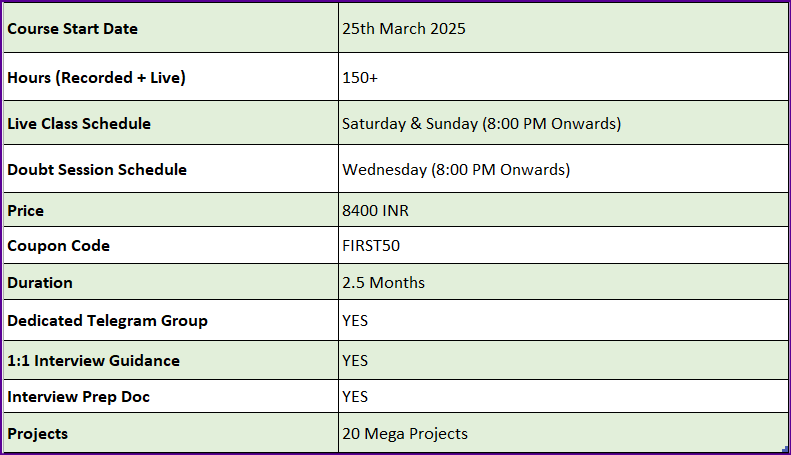
 **Project-4 | User Management In Linux**



**Enrolment Link:** <https://www.devopsshack.com/courses/Batch-9--Zero-To-Hero--DevSecOps--Cloud-DevOps-67bdb260b8143724f042a2f0>

Comprehensive shell script for user management in Ubuntu. It allows you to create, delete, list, and manage users with proper checks.

The script ensures:

* Users cannot create duplicate accounts.
* Deleting users confirms the action.
* Passwords are set securely.
* Users can be added to specific groups.
* It includes a help menu.

**Features:**

* Create users with a home directory and password.
* Delete users after confirmation.
* List all users.
* Lock/unlock users.
* Add users to groups.

**Shell Script: user\_management.sh**

#!/bin/bash

# Script Name: user\_management.sh

# Description: Manage users in Ubuntu (Create, Delete, List, Lock, Unlock)

# Usage: Run the script and choose the operation.

# Author: SHACKVERSE PRIVATE LIMITED

# Function to check if script is run as root

check\_root() {

if [[ "$EUID" -ne 0 ]]; then

echo "Error: This script must be run as root." >&2

exit 1

fi

}

# Function to create a new user

create\_user() {

read -p "Enter username to create: " username

if id "$username" &>/dev/null; then

echo "User '$username' already exists."

return

fi

read -s -p "Enter password for $username: " password

echo

useradd -m -s /bin/bash "$username"

echo "$username:$password" | chpasswd

echo "User '$username' created successfully."

read -p "Add user to a group? (y/n): " add\_group

if [[ "$add\_group" == "y" ]]; then

read -p "Enter group name: " groupname

if grep -q "^$groupname:" /etc/group; then

usermod -aG "$groupname" "$username"

echo "User '$username' added to group '$groupname'."

else

echo "Group '$groupname' does not exist."

fi

fi

}

# Function to delete a user

delete\_user() {

read -p "Enter username to delete: " username

if ! id "$username" &>/dev/null; then

echo "User '$username' does not exist."

return

fi

read -p "Are you sure you want to delete user '$username'? (y/n): " confirm

if [[ "$confirm" == "y" ]]; then

userdel -r "$username"

echo "User '$username' deleted successfully."

else

echo "User deletion aborted."

fi

}

# Function to list all users

list\_users() {

echo "Listing all system users:"

awk -F':' '{ print $1 }' /etc/passwd

}

# Function to lock a user

lock\_user() {

read -p "Enter username to lock: " username

if id "$username" &>/dev/null; then

passwd -l "$username"

echo "User '$username' has been locked."

else

echo "User '$username' does not exist."

fi

}

# Function to unlock a user

unlock\_user() {

read -p "Enter username to unlock: " username

if id "$username" &>/dev/null; then

passwd -u "$username"

echo "User '$username' has been unlocked."

else

echo "User '$username' does not exist."

fi

}

# Function to show menu

show\_menu() {

echo "--------------------------------------"

echo " Ubuntu User Management Script "

echo "--------------------------------------"

echo "1) Create a new user"

echo "2) Delete a user"

echo "3) List all users"

echo "4) Lock a user"

echo "5) Unlock a user"

echo "6) Exit"

echo "--------------------------------------"

}

# Main script execution

check\_root

while true; do

show\_menu

read -p "Choose an option: " choice

case $choice in

1) create\_user ;;

2) delete\_user ;;

3) list\_users ;;

4) lock\_user ;;

5) unlock\_user ;;

6) echo "Exiting..."; exit 0 ;;

\*) echo "Invalid option. Please select a valid choice." ;;

esac

done

**How to Use the Script**

1. **Make the script executable**:

chmod +x user\_management.sh

1. **Run the script as root**:

sudo ./user\_management.sh

1. Follow the on-screen prompts to create, delete, list, lock, or unlock users.

**Security & Best Practices**

* The script **ensures only root users can manage accounts**.
* **Passwords are set securely**.
* Users are prompted before **deletion** to avoid accidental removals.
* Users can **be added to groups** during creation.

**Real-World Scenario: User Management in an Organization Using This Script**

In an organization, managing users efficiently is crucial for security, compliance, and operational continuity. This script can be used by system administrators and DevOps engineers to **create, delete, manage, and control access to user accounts** in a structured and secure manner.

**📌 Scenario: User Lifecycle Management in a Company**

**1️. New Employee Joins the Company**

**Example:** A new DevOps Engineer named *John Doe* joins the company, and HR informs IT that he needs access.

🔹 **IT Admin uses the script to create a user**:

sudo ./user\_management.sh

* Selects **option 1 (Create a new user)**.
* Enters the username: johndoe
* Sets a password.
* Adds John to the devops group.

🔹 **Outcome:**

* John now has a home directory /home/johndoe.
* He is added to the devops group, so he has appropriate permissions.
* He can now log in and start working.

**2️. Temporary User Access for Contractors**

**Example:** A contractor (*Alice*) joins the team for **2 months** to work on a security audit.

🔹 **IT Admin creates a temporary user**:

* Runs the script and creates alice\_temp.
* Adds Alice to the security group.
* Uses chage to **set an expiration date** for the user:

sudo chage -E 2025-05-01 alice\_temp

* This ensures her access is **automatically revoked after 2 months**.

🔹 **Outcome:**

* Alice can work without the risk of forgetting to remove her account later.
* IT can audit and control temporary access effectively.

**3️. Employee Leaves the Company (User Deletion)**

**Example:** *John Doe* resigns from the company.

🔹 **IT Admin uses the script to delete John’s account**:

sudo ./user\_management.sh

* Selects **option 2 (Delete a user)**.
* Confirms deletion.

🔹 **Outcome:**

* John’s user account is **completely removed**, along with his home directory.
* Prevents ex-employees from accessing company resources.

**4️. Security Measures: Locking and Unlocking Accounts**

**Example 1:** *Mark* from Finance is on **a 2-month leave**.

* Instead of deleting his account, IT locks it:

sudo ./user\_management.sh

* + Selects **option 4 (Lock a user)**.
  + Enters mark.

🔹 **Outcome:** Mark's account is locked until he returns.

**Example 2:** When Mark returns, IT **unlocks** his account:

sudo ./user\_management.sh

* Selects **option 5 (Unlock a user)**.

🔹 **Outcome:** Mark can log in again without recreating his account.

**5️. Security & Compliance: Listing Users**

To **audit user accounts** and find unauthorized users:

sudo ./user\_management.sh

* Selects **option 3 (List all users)**.
* Admin can check for **orphaned accounts** (users who no longer work in the company).

🔹 **Outcome:** Ensures **only valid employees have access** to the system.