#!/bin/bash

create\_user()

{

echo -n "Enter username: "

read user\_name

# Check if the user already exists

if sudo id "$user\_name" &>/dev/null

#/dev/null Redirecting output to it is a way to effectively discard or ignore unwanted information.

then

echo "Error: User '$user\_name' already exists."

else

echo -n "Enter password for '$user\_name': "

read -s password

echo

# Create the new user

sudo useradd -m -p $(openssl passwd -1 "$password") "$user\_name"

if [ $? -eq 0 ]; then

echo "User '$user\_name' created successfully."

else

echo "Error creating user '$user\_name'."

fi

fi

}

reset\_password()

{

echo -n "Enter username to reset password: "

read user\_name

# Check if the user exists

if sudo id "$user\_name" &>/dev/null

then

echo -n "Enter new password for '$user\_name': "

#The -n option suppresses this newline, allowing the next output to appear on the same line.

read -s new\_password

#the user's input for the password will not be visible on the screen as they type it due to the use of read -s.

echo

# Reset the password

echo "$new\_password" | sudo passwd --stdin "$user\_name"

# The --stdin option indicates that the new password will be provided through the standard input (stdin) rather than prompting the user.

if [ $? -eq 0 ]

then

echo "Password for user '$user\_name' reset successfully."

else

echo "Error resetting password for user '$user\_name'."

fi

else

echo "Error: User '$user\_name' not exists."

fi

}

list\_of\_users()

{

echo "List of all users:"

cut -d: -f1 /etc/passwd

}

delete\_user()

{

echo -n "Enter username to delete: "

read user\_name

if sudo id "$user\_name" &>/dev/null

then

sudo userdel -r "$user\_name"

# -r option removes home directory and mail spool

# Alternatively, you can use 'sudo deluser --remove-home "$user\_name"' with 'deluser' command

echo "User '$user\_name' deleted successfully."

else

echo "Error: User '$user\_name' not exists."

fi

}

#help

user\_help()

{

echo "User Management Console"

echo " -c, --create Create a new user account"

echo " -d, --delete Delete an existing user account"

echo " -r, --reset Reset a password for an existing user account"

echo " -l, --list List all users accounts on the system"

echo " -h, --help Display this Help and exit"

echo "Exiting User Management Console."

exit 0

}

#main program

while true

do

echo "User Management Console"

echo “Option:”

echo " -c, --create Create a new user account"

echo " -d, --delete Delete an existing user account"

echo " -r, --reset Reset a password for an existing user account"

echo " -l, --list List all users accounts on the system"

echo " -h, --help Display this Help and exit"

read option

case $option in

c) create\_user ;;

d) delete\_user ;;

r) reset\_password ;;

l) list\_of\_users ;;

h) user\_help ;;

\*)

echo "Invalid option. Use -h for help."

;;

esac

done