**Aniket Kumar**(12211057)

**aniketkumar.nikeman@gmail.com**

Task 1: User Account Creation Documentation

# Introduction

This document provides detailed documentation for the shell script created to automate the process of creating 100 user accounts for new engineers at MyComp. The script ensures that the user accounts are created with the specified requirements and includes functionality, error handling, and a mechanism to notify users of password expiration.

# Script Functionality

The shell script performs the following tasks:  
1. Creates 100 user accounts with usernames in the format mycompusr1, mycompusr2, ..., up to mycompusr100.  
2. Creates a home directory for each user in the format /home/mycompusr1, /home/mycompusr2, ..., /home/mycompusr100.  
3. Sets the password for each user to be the same as the username.  
4. Sets the password to expire every 30 days.  
5. Sets the permissions for each home directory to 700.  
6. Adds each user to the 'wheel' group to grant administrative privileges.  
7. Implements a mechanism to notify users of their password expiration.

# Script Clarity

The script is clearly written with appropriate variable names, comments, and structured to follow best practices for shell scripting. The code is organized into logical sections for user creation, password setting, permission setting, and password expiration notification.

# Documentation

The script is thoroughly documented with comments explaining the purpose of each major step. Below is the complete script:

#!/bin/bash  
  
# Script to create 100 user accounts for new engineers at MyComp  
  
# Variables  
USER\_PREFIX="mycompusr"  
HOME\_DIR\_PREFIX="/home/"  
GROUP="wheel"  
PASSWORD\_EXPIRY\_DAYS=30  
  
# Function to check if the user already exists  
user\_exists() {  
 id "$1" &>/dev/null  
}  
  
# Loop to create 100 user accounts  
for i in {1..100}; do  
 USERNAME="${USER\_PREFIX}${i}"  
 HOME\_DIR="${HOME\_DIR\_PREFIX}${USERNAME}"  
 PASSWORD="${USERNAME}"  
  
 # Check if user already exists  
 if user\_exists "$USERNAME"; then  
 echo "User $USERNAME already exists. Skipping..."  
 continue  
 fi  
  
 # Create user with home directory  
 useradd -m -d "$HOME\_DIR" -s /bin/bash -G "$GROUP" "$USERNAME"  
 if [ $? -eq 0 ]; then  
 echo "User $USERNAME created successfully."  
 else  
 echo "Error creating user $USERNAME."  
 continue  
 fi  
  
 # Set the user's password  
 echo "$USERNAME:$PASSWORD" | chpasswd  
 if [ $? -eq 0 ]; then  
 echo "Password for $USERNAME set successfully."  
 else  
 echo "Error setting password for $USERNAME."  
 fi  
  
 # Set home directory permissions  
 chmod 700 "$HOME\_DIR"  
 if [ $? -eq 0 ]; then  
 echo "Permissions for $HOME\_DIR set to 700."  
 else  
 echo "Error setting permissions for $HOME\_DIR."  
 fi  
  
 # Set password expiration  
 chage -M "$PASSWORD\_EXPIRY\_DAYS" "$USERNAME"  
 if [ $? -eq 0 ]; then  
 echo "Password for $USERNAME set to expire in $PASSWORD\_EXPIRY\_DAYS days."  
 else  
 echo "Error setting password expiration for $USERNAME."  
 fi  
done  
  
# Password expiration notification mechanism  
NOTIFY\_SCRIPT="/etc/cron.daily/password\_expiry\_notify.sh"  
  
cat << 'EOF' > "$NOTIFY\_SCRIPT"  
#!/bin/bash  
  
# Script to notify users of password expiration  
  
# Loop through all users  
for user in $(getent passwd | awk -F: '{ print $1 }'); do  
 # Get the number of days until password expiration  
 days\_left=$(chage -l $user | grep "Password expires" | awk -F: '{ print $2 }' | awk '{ print $1 }')  
   
 # Notify user if password expires in less than 7 days  
 if [ "$days\_left" -lt 7 ]; then  
 echo "Hello $user, your password will expire in $days\_left days. Please change it soon." | mail -s "Password Expiry Notification" $user  
 fi  
done  
EOF  
  
# Make the notification script executable  
chmod +x "$NOTIFY\_SCRIPT"  
  
echo "Password expiration notification mechanism set up successfully."  
echo "User account creation script completed."

# Error Handling

The script includes error handling mechanisms to ensure robustness:  
1. Checks if a user already exists before attempting to create the account.  
2. Checks the success or failure of each command (useradd, chpasswd, chmod, chage) and provides appropriate messages.  
3. Skips to the next user if an error occurs during the creation process.

# Creativity

The script incorporates a creative solution for password expiration notification by creating a daily cron job that checks for upcoming password expirations and sends email notifications to the users.

# Verification

To verify the script's functionality, perform the following steps:  
1. Save the script to a file named task1\_user\_account\_creation.sh.  
2. Make the script executable: chmod +x task1\_user\_account\_creation.sh  
3. Run the script with superuser privileges: sudo ./task1\_user\_account\_creation.sh  
4. Verify the creation of user accounts, home directories, group membership, and password expiration using the provided verification commands.

# Script’s Screenshots :

