Task 2

Create a user account with the following attribute

username: islam

Fullname: Islam Askar

Password: islam

```
[root@server ~]# useradd -m -c "Islam Askar" islam
[root@server ~]# passwd islam
Changing password for user islam.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@server ~]#
```

Create a user account with the following attribute

Username: baduser

Full name: Bad User

Password: baduser

```
[root@server ~]# useradd -m -c "Bad User" baduser
useradd: warning: the home directory /home/baduser already exists.
useradd: Not copying any file from skel directory into it.
Creating mailbox file: File exists
[root@server ~]#
[root@server ~]# passwd baduser
Changing password for user baduser.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@server ~]#
```

Create a supplementary (Secondary) group called pgroup with group ID of 30000

```
[root@server ~]# groupadd -g 30000 pgroup
```

Create a supplementary group called badgroup

```
[root@server ~]# groupadd badgroup
[root@server ~]#
```

Add islam user to the pgroup group as a supplementary group

```
[root@server ~]# usermod -G pgroup islam
[root@server ~]# id islam
uid=1001(islam) gid=1001(islam) groups=1001(islam),30000(pgroup)
[root@server ~]#
```

Modify the password of islam's account to password

```
[root@server ~]# passwd islam
Changing password for user islam.
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
passwd: all authentication tokens updated successfully.
[root@server ~]#
```

Lock bad user account so he can't log in

```
[baduser@server ~]$ ssh baduser@192.168.32.128
The authenticity of host '192.168.32.128 (192.168.32.128)' can't be established.
ED25519 key fingerprint is SHA256:Rn12/K7N9seQ2KveXAKvyBw/o7M/Ve/NOKWX45Gk7jE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.32.128' (ED25519) to the list of known hosts.
baduser@192.168.32.128's password:
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Wed Nov 13 13:22:36 2024
[baduser@server ~]$
[baduser@server ~]$ su -
Password:
[root@server ~]# usermod -L baduser
[root@server ~]# ssh baduser@192.168.32.128
The authenticity of host '192.168.32.128 (192.168.32.128)' can't be established.
ED25519 key fingerprint is SHA256:Rn12/K7N9seQ2KveXAKvyBw/o7M/Ve/NOKWX45Gk7jE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.32.128' (ED25519) to the list of known hosts.
baduser@192.168.32.128's password:
Permission denied, please try again.
baduser@192.168.32.128's password:
Permission denied, please try again.
```

Delete bad user account

```
[root@server ahmed]#  userdel baduser
userdel: user baduser is currently used by process 2039
[root@server ahmed]#
```

Delete the supplementary group called badgroup

```
[root@server ~]# groupdel badgroup
[root@server ~]#
```

Using the useradd command, add accounts for the following users in your system: user1, user2, user3, user4, user5, user6 and user7.

Note: Remember to give each user a password.

```
[root@server ~]# for user in user1 user2 user3 user4 user5 user6 user7; do
    sudo useradd -m -p $(openssl passwd -1 $user) $user
done
```

```
ahmed:x:1000:1000:ahmed:/home/ahmed:/bin/bash
islam:x:1001:1001:Islam Askar:/home/islam:/bin/bash
user1:x:1002:1002::/home/user1:/bin/bash
user2:x:1003:1003::/home/user2:/bin/bash
user3:x:1004:1004::/home/user3:/bin/bash
user4:x:1005:1005::/home/user4:/bin/bash
user5:x:1006:1006::/home/user5:/bin/bash
user6:x:1007:1007::/home/user6:/bin/bash
user7:x:1008:1008::/home/user7:/bin/bash
[root@server ~]#
```

10. Using the groupadd command, add the following groups to your system.

Group

GID sales 10000 hr 10001 web 10002

```
[root@server ~]# groupadd -g 10000 sales
[root@server ~]# gro
groff       grotty      groupdel    groupmod
grops       groupadd    groupmems   groups
[root@server ~]# groupadd -g 10001 hr
[root@server ~]# groupadd -g 10001 web
groupadd: GID '10001' already exists
[root@server ~]# groupadd -g 10002 web
[root@server ~]#
```

Using the usermod command to add user1 and user2 to the sales auxiliary group, user3 and user4 to the hr auxiliary group. User5 and user6 to web auxiliary group. And add user7 to all auxiliary groups.

```
[root@server ~]# id user1
uid=1002(user1) gid=1002(user1) groups=1002(user1)
[root@server ~]# usermod -aG sales user1
[root@server ~]# id user1
uid=1002(user1) gid=1002(user1) groups=1002(user1),10000(sales)
[root@server ~]# usermod -aG sales user2
[root@server ~]# usermod -aG hr user3
[root@server ~]# usermod -aG hr user4
[root@server ~]# usermod -aG web user5
[root@server ~]# usermod -aG web user6
[root@server ~]# usermod -aG sales,hr,web user7
[root@server ~]# id user7
uid=1008(user7) gid=1008(user7) groups=1008(user7),10000(sales),10001(hr),10002(web)
[root@server ~]#
```

Login as each user and use id command to verify that they are in the appropriate groups. How else might you verify this information?

```
[root@server ~]# id user1
uid=1002(user1) gid=1002(user1) groups=1002(user1),10000(sales)
[root@server ~]# grep '^sales\|^hr\|^web' /etc/group
sales:x:10000:user1,user2,user7
hr:x:10001:user3,user4,user7
web:x:10002:user5,user6,user7
[root@server ~]#
```