1. If you were able to push the code in your branch to the given repo then you answered well.

2. Script

```bash
#!/bin/bash

echo "Ente the Type "Directory" or "File" ,as per your search"
read DirectoryORFile
echo "Type "Directory" or "File" ,as per your search"
read name
echo "Enter the location of File or Directory"
read location
echo "Enter the number days created before  "
read d
case "$DirectoryORFile" in
      "Directory")
if [ -d $location/$name ];
 then
        echo "Directory exists at  $location"/"$name"
        echo "it will be deleted if entered time matches"
        cd $location

   if [[ $(find . -name "$name" -mtime $d) ]]
   then
       sudo find . -name "$name" -mtime $d -exec rm -rf {} \;
    else
       echo "Directory found at  $location"/"$name but time does not match"
    fi
else
       echo "Directory not found"
fi
;;
"File")
if [ -f $location/$name ];
then
        echo "File exists at $location"/"$name"
        echo "it will be deleted if entered time matches"
        cd $location

   if [[ $(find . -name "$name" -mtime $d) ]]
```

```bash
    then
        find . -name "$name" -mtime $d -exec rm -rf {} \;
    else
        echo "File found at  $location"/"$name but time does not match"
    fi
else
        echo "File not found  "
fi
;;
Esac

###Another Way

echo "Please enter Filename or Directory"
read fname
echo "Please enter date like (Apr  1)"
read fdate
echo "Enter the location"
read location
cd $location

if [[ -d "$location/$fname" ]]; then
   echo "$fname is a directory"
   tfile=`find . -name $fname 2>/dev/null`
   dfile=`ls -ld $tfile|grep " $fdate "|awk -F " " '{print $NF}'`
   echo $tfile
   echo $dfile
   rm -rf $dfile
elif [[ -f $fname ]]; then
   echo "$fname is a file"
   tfile=`find . -name $fname 2>/dev/null`
   dfile=`ls -ld $tfile|grep " $fdate "|awk -F " " '{print $NF}'`
   echo $tfile
   echo $dfile
   rm -r $dfile
else
   echo "$fname is not valid"
   exit 1
fi
```
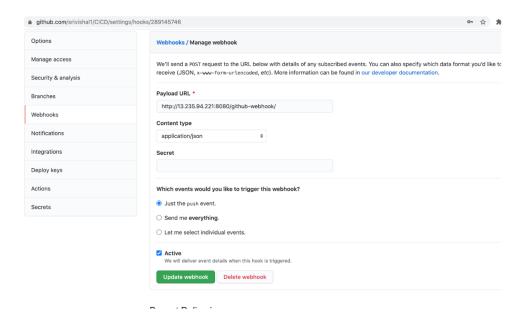
3. Script

```bash
#!/bin/bash
FILE=$(grep -hr "/" /etc/passwd  | head -4 > /tmp/file) # This will first 4 users and
redirect to the /tmp/file
FILENAME="/tmp/file"
while IFS=: read -r username password userid groupid home shell
do
echo "username = " "$username"
echo "password = " "$password"
echo "userid = " "$userid"
echo "groupid = " "$groupid"
echo "home = " "$home"
echo "shell = " "$shell"
done < $FILENAME


## OR Another Method
cat /etc/passwd | grep -v '^#' | head -n 4 |awk -F ':' '{ print " \n username= " $1
"\n password= " $2  " \n userid= " $3 " \n groupid= " $4 "\n userinfo= " $5 "\n home=
" $6  " \n shell= " $7}'
```

4. Steps for Jenkins Job.
a) Configure github webhook with Jenkins

Validating whether webhook configured properly -

Setup a New Jenkins Job named as "AssignmentJob"

1. Configure Github repository path
2. Select "Restrict where this project can be run" and Select the created slave

b) Create the slave configuration

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
🖥	master	Linux (amd64)	In sync	5.35 GB	⛔ 0 B	5.35 GB	0ms	⚙
🖥	slave1	Linux (amd64)	In sync	5.35 GB	⛔ 0 B	5.35 GB	166ms	⚙
	Data obtained	17 min	17 min	17 min	17 min	17 min	17 min

c) Create a free style Job with any name.

d) Configure the job to be bound with a managed selected slave.

General   Source Code Management   Build Triggers   Build Environment   Build   Post-build Actions

Description

[Safe HTML] Preview

☐ Discard old builds   ❓
☐ GitHub project
☐ This build requires lockable resources
☐ This project is parameterized   ❓
☐ Throttle builds   ❓
☐ Disable this project   ❓
☐ Execute concurrent builds if necessary   ❓
☑ Restrict where this project can be run   ❓
Label Expression   ❓

slave

Label slave matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced...

Save   Apply

e) Configure git repo.

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Source Code Management

○ None

● Git   ❓

  Repositories   ❓

    Repository URL   ❓

    https://github.com/guptacompanys/demo-tsest.git

    Credentials   ❓

    ⌄   🔑 Add ▾

    Name   ❓

    Refspec   ❓

                                                                          Add Repository

  Branches to build   ❓

                                                                                    X   ❓

  Save   Apply

f) Configure the Poll SCM to check every 5 min job.

Add ▾

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)   ❓
☐ Build after other projects are built   ❓
☐ Build periodically   ❓
☐ GitHub hook trigger for GITScm polling   ❓
☑ Poll SCM   ❓
  Schedule   ❓

  H/5 * * * *

  Would last have run at Tuesday, April 6, 2021 9:30:07 AM UTC; would next run at Tuesday, April 6, 2021 9:35:07 AM UTC.

☐ Ignore post-commit hooks   ❓

5. commands
    # touch credentials   # create the empty file
    # ls -lhrta credentials # check the permissions
    # chmod o-r credentials # This removes read permission from others.
    # setfacl -m u:<username>:r credentials # to set the read permission to the required user
    # getfacl credentials # To check the new permissions.

6.

```
NDI-LAP-685:Devops_Upskill pradeep.rawat$ cat emp
echo "Name    : Pradep Rawat"
echo "EmplCode: 1234PR"
NDI-LAP-685:Devops_Upskill pradeep.rawat$ chmod a+x emp
NDI-LAP-685:Devops_Upskill pradeep.rawat$ mv emp /usr/local/bin/
NDI-LAP-685:Devops_Upskill pradeep.rawat$ emp
Name    : Pradep Rawat
EmplCode: 1234PR
NDI-LAP-685:Devops_Upskill pradeep.rawat$ ▯
```

Note: There are multiple ways to get these tasks done and few of them I have mentioned. Maybe you have a different approach to achieve it but before choosing any approach you should see which is appropriate for the current scenarios.