

TASK 4: Shell Scripting Basics

- Create a shell script file and add shebang line
Check shell " echo \$SHELL "
Create folder and go inside the folder " 'mkdir task-4-shell-scripting'
' cd task-4-shell-scripting' "
• Create First Shell Script (Shebang Included)
Create file:
"touch system_info.sh"

Open file:

"nano system_info.sh"

Content:

```
#!/bin/bash
echo "----- SYSTEM INFORMATION -----"
echo "Hostname: $(hostname)"
echo "Current User: $(whoami)"
echo "Date & Time: $(date)"
echo "Uptime:"
uptime
```

- Make Script Executable
" chmod +x system_info.sh"
Run script
"./system_info.sh"
• Use Variables in Shell Script
 - Create a new file: "nano variable.sh"

- Content in it

```
#!/bin/bash
NAME="Vishnu"
ROLE="DevOps Intern"
echo "My name is Vishnu"
echo "My role is DevOps Intern"
```

run

```
“chmod +x variables.sh
./variables.sh”
```

- Scripting using conditional statement
 - Create file “nano condition.sh”
 - Content

```
#!/bin/bash

read -p "Enter your age: " AGE

if [ $AGE -ge 18 ]
then
    echo "You are eligible to work"
else
    echo "You are not eligible to work"
fi
```

- Run “chmod +x condition.sh
./condition.sh”

STEP 5: Script Using Variables

5.1 Create file

nano variables.sh

5.2 Paste content

```
#!/bin/bash
```

```
# Script using variables
```

```
NAME="Vishnu Kumar"
```

```
COURSE="DevOps Internship"
```


```
echo "Name: $NAME"
```

```
echo "Course: $COURSE"
```

5.3 Execute

```
chmod +x variables.sh
```

```
./variables.sh
```

 Screenshot output

◆ STEP 6: Script Using Conditional Statements

6.1 Create file

```
nano condition.sh
```

6.2 Paste content

```
#!/bin/bash
```

```
# Script using if-else
```

```
read -p "Enter a number: " NUM
```

```
if [ $NUM -gt 10 ]
```

```
then
```

```
    echo "Number is greater than 10"
```

```
else
```

```
    echo "Number is less than or equal to 10"
```

```
fi
```

6.3 Execute

```
chmod +x condition.sh
```

```
./condition.sh
```

 Screenshot output

◆ STEP 7: Script Using Loop

7.1 Create file

```
nano loop.sh
```

7.2 Paste content

```
#!/bin/bash
```

```
# Script using for loop
```

```
for i in 1 2 3 4 5
```

```
do
```


```
    echo "Iteration number: $i"
```

```
done
```

7.3 Execute

```
chmod +x loop.sh
```

```
./loop.sh
```

 Screenshot output

◆ STEP 8: Automate Log Backup (DevOps Task)

8.1 Create file

```
nano log_backup.sh
```

8.2 Paste content

```
#!/bin/bash
```

```
# Log backup automation script
```

```
SOURCE_DIR="/var/log"
```

```
BACKUP_DIR="$HOME/log_backup"
```

```
mkdir -p $BACKUP_DIR
```

```
cp $SOURCE_DIR/*.log $BACKUP_DIR
```


```
echo "Log files backed up successfully"
```

8.3 Execute

```
chmod +x log_backup.sh
```

```
sudo ./log_backup.sh
```

 Screenshot execution

 Screenshot log_backup directory

◆ STEP 9: Redirect Output to File

```
./system_info.sh > system_output.txt
```

Check file:

```
cat system_output.txt
```

◆ STEP 10: Verify Comments in Scripts

Open any script:

```
nano system_info.sh
```

Confirm:

```
# Script Name
```

```
# Purpose
```

STEP 11: Create README.md (MANDATORY)

```
nano README.md
```

Paste this:

```
# Task 4 – Shell Scripting Basics
```

```
## Description
```

This task demonstrates basic shell scripting concepts used in DevOps.

```
## Scripts Created
```

- system_info.sh – Displays system information
- variables.sh – Uses shell variables
- condition.sh – Demonstrates if-else condition
- loop.sh – Demonstrates loops
- log_backup.sh – Automates log backup

Tools Used

- Bash
- Linux (Ubuntu)

Outcome

Automated repetitive system tasks using shell scripts.

Save file.

◆ STEP 12: Create Screenshots Folder

mkdir screenshots

 Put **ALL screenshots** inside this folder

◆ STEP 13: Final Folder Structure (CHECK THIS)

task-4-shell-scripting/

```
|
|— system_info.sh
|— variables.sh
|— condition.sh
|— loop.sh
|— log_backup.sh
|— system_output.txt
|— README.md
|— screenshots/
```

◆ STEP 14: Push to GitHub

git init

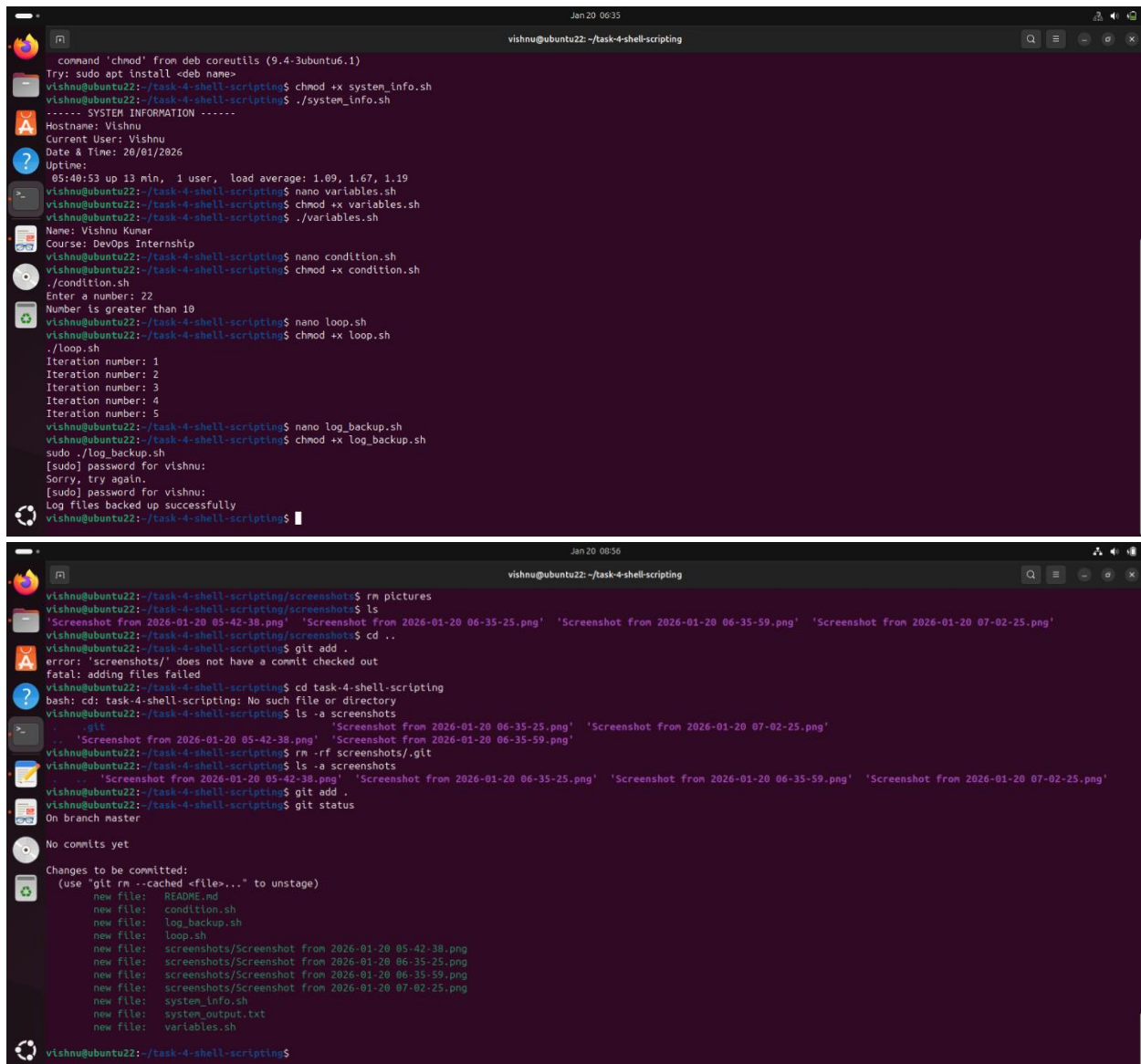
git add .

git commit -m "Completed Task 4 - Shell Scripting"

git branch -M main

git remote add origin https://github.com/<your-username>/task-4-shell-scripting.git

git push -u origin main



The image consists of two terminal screenshots. The top screenshot shows a series of commands and their outputs in a terminal window titled 'vishnu@ubuntu22: ~/task-4-shell-scripting'. The commands include 'chmod +x system_info.sh', 'nano system_info.sh', 'chmod +x variables.sh', 'nano variables.sh', 'chmod +x condition.sh', 'nano condition.sh', 'chmod +x loop.sh', 'nano loop.sh', 'chmod +x log_backup.sh', 'nano log_backup.sh', and 'sudo ./log_backup.sh'. The bottom screenshot shows the same terminal window after running 'git init', 'git add .', 'git commit -m "Completed Task 4 - Shell Scripting"', 'git branch -M main', 'git remote add origin https://github.com/<your-username>/task-4-shell-scripting.git', and 'git push -u origin main'. It also shows the output of 'git status' and 'git log'.

```
command 'chmod' from deb coreutils (9.4-3ubuntu6.1)
Try: sudo apt install <deb name>
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x system_info.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ ./system_info.sh
----- SYSTEM INFORMATION -----
Hostname: Vishnu
Current User: Vishnu
Date & Time: 26/01/2026
Uptime:
05:40:53 up 13 min, 1 user, load average: 1.09, 1.67, 1.19
vishnu@ubuntu22:~/task-4-shell-scripting$ nano variables.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x variables.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ ./variables.sh
Name: Vishnu Kumar
Course: DevOps Internship
vishnu@ubuntu22:~/task-4-shell-scripting$ nano condition.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x condition.sh
./condition.sh
Enter a number: 22
Number is greater than 10
vishnu@ubuntu22:~/task-4-shell-scripting$ nano loop.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x loop.sh
./loop.sh
Iteration number: 1
Iteration number: 2
Iteration number: 3
Iteration number: 4
Iteration number: 5
vishnu@ubuntu22:~/task-4-shell-scripting$ nano log_backup.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x log_backup.sh
sudo ./log_backup.sh
[sudo] password for vishnu:
Sorry, try again.
[sudo] password for vishnu:
Log files backed up successfully
vishnu@ubuntu22:~/task-4-shell-scripting$

vishnu@ubuntu22:~/task-4-shell-scripting$ git init
vishnu@ubuntu22:~/task-4-shell-scripting$ git add .
vishnu@ubuntu22:~/task-4-shell-scripting$ git commit -m "Completed Task 4 - Shell Scripting"
vishnu@ubuntu22:~/task-4-shell-scripting$ git branch -M main
vishnu@ubuntu22:~/task-4-shell-scripting$ git remote add origin https://github.com/<your-username>/task-4-shell-scripting.git
vishnu@ubuntu22:~/task-4-shell-scripting$ git push -u origin main
vishnu@ubuntu22:~/task-4-shell-scripting$ git status
On branch main
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   README.md
    new file:   condition.sh
    new file:   log_backup.sh
    new file:   loop.sh
    new file:   screenshots/Screenshot from 2026-01-20 05-42-38.png
    new file:   screenshots/Screenshot from 2026-01-20 06-35-25.png
    new file:   screenshots/Screenshot from 2026-01-20 06-35-59.png
    new file:   screenshots/Screenshot from 2026-01-20 07-02-25.png
    new file:   system_info.sh
    new file:   system_output.txt
    new file:   variables.sh
```

```
Jan 20 07:02
vishnu@ubuntu22: ~/task-4-shell-scripting/screenshots
vishnu@ubuntu22:~/task-4-shell-scripting/screenshots$ ls
pictures 'Screenshot from 2026-01-20 05-42-38.png' 'Screenshot from 2026-01-20 06-35-25.png' 'Screenshot from 2026-01-20 06-35-59.png'
vishnu@ubuntu22:~/task-4-shell-scripting/screenshots$
```

```
Jan 20 06:35
vishnu@ubuntu22: ~/task-4-shell-scripting
Uptime:
05:40:53 up 13 min, 1 user, load average: 1.09, 1.67, 1.19
vishnu@ubuntu22:~/task-4-shell-scripting$ nano variables.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x variables.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ ./variables.sh
Name: Vishnu Kumar
Course: DevOps Internship
vishnu@ubuntu22:~/task-4-shell-scripting$ nano condition.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x condition.sh
./condition.sh
Enter a number: 22
Number is greater than 10
vishnu@ubuntu22:~/task-4-shell-scripting$ nano loop.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x loop.sh
./loop.sh
Iteration number: 1
Iteration number: 2
Iteration number: 3
Iteration number: 4
Iteration number: 5
vishnu@ubuntu22:~/task-4-shell-scripting$ nano log_backup.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x log_backup.sh
sudo ./log_backup.sh
[sudo] password for vishnu:
Sorry, try again.
[sudo] password for vishnu:
Log files backed up successfully
vishnu@ubuntu22:~/task-4-shell-scripting$ ./system_info.sh > system_output.txt
vishnu@ubuntu22:~/task-4-shell-scripting$ cat system_output.txt
----- SYSTEM INFORMATION -----
Hostname: Vishnu
Current User: Vishnu
Date & Time: 20/01/2026
Uptime:
06:35:41 up 1:00, 1 user, load average: 0.78, 0.98, 1.12
vishnu@ubuntu22:~/task-4-shell-scripting$
```

```
Jan 20 08:56
vishnu@ubuntu22: ~/task-4-shell-scripting

.. 'Screenshot from 2026-01-20 05-42-38.png' 'Screenshot from 2026-01-20 06-35-25.png' 'Screenshot from 2026-01-20 06-35-59.png' 'Screenshot from 2026-01-20 07-02-25.png'
vishnu@ubuntu22:~/task-4-shell-scripting$ git add .
vishnu@ubuntu22:~/task-4-shell-scripting$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
        new file:   condition.sh
        new file:   log_backup.sh
        new file:   loop.sh
        new file:   screenshots/Screenshot from 2026-01-20 05-42-38.png
        new file:   screenshots/Screenshot from 2026-01-20 06-35-25.png
        new file:   screenshots/Screenshot from 2026-01-20 06-35-59.png
        new file:   screenshots/Screenshot from 2026-01-20 07-02-25.png
        new file:   system_info.sh
        new file:   system_output.txt
        new file:   variables.sh

vishnu@ubuntu22:~/task-4-shell-scripting$ git commit -m "Completed Task 4 - Shell Scripting"
[master (root-commit) a237c83] Completed Task 4 - Shell Scripting
11 files changed, 71 insertions(+)
create mode 100644 README.md
create mode 100755 condition.sh
create mode 100755 log_backup.sh
create mode 100755 loop.sh
create mode 100644 screenshots/Screenshot from 2026-01-20 05-42-38.png
create mode 100644 screenshots/Screenshot from 2026-01-20 06-35-25.png
create mode 100644 screenshots/Screenshot from 2026-01-20 06-35-59.png
create mode 100644 screenshots/Screenshot from 2026-01-20 07-02-25.png
create mode 100755 system_info.sh
create mode 100644 system_output.txt
create mode 100755 variables.sh
vishnu@ubuntu22:~/task-4-shell-scripting$
```

```
Jan 20 09:42
vishnu@ubuntu22: ~/task-4-shell-scripting

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

vishnu@ubuntu22:~$ echo $SHELL
/bin/bash
vishnu@ubuntu22:~$ mkdir task-4-shell-scripting
vishnu@ubuntu22:~$ cd task-4-shell-scripting/
vishnu@ubuntu22:~/task-4-shell-scripting$ touch system_info.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ nano system_info.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ cat system_info.sh
#!/bin/bash
# Script Name: system_info.sh
# Purpose: Display system information

echo "----- SYSTEM INFORMATION -----"
echo "Hostname: Vishnu"
echo "Current User: Vishnu"
echo "Date & Time: 20/01/2026"
echo "Uptime:"
uptime
vishnu@ubuntu22:~/task-4-shell-scripting$ chomd +x system_info.sh
Command 'chomd' not found, did you mean:
  command 'chmod' from deb coreutils (9.4-3ubuntu6.1)
Try: sudo apt install <deb name>
vishnu@ubuntu22:~/task-4-shell-scripting$ chmod +x system_info.sh
vishnu@ubuntu22:~/task-4-shell-scripting$ ./system_info.sh
----- SYSTEM INFORMATION -----
Hostname: Vishnu
Current User: Vishnu
Date & Time: 20/01/2026
Uptime:
05:40:53 up 13 min,  1 user,  load average: 1.09, 1.67, 1.19
vishnu@ubuntu22:~/task-4-shell-scripting$
```