

Scripting Task

Task Description:

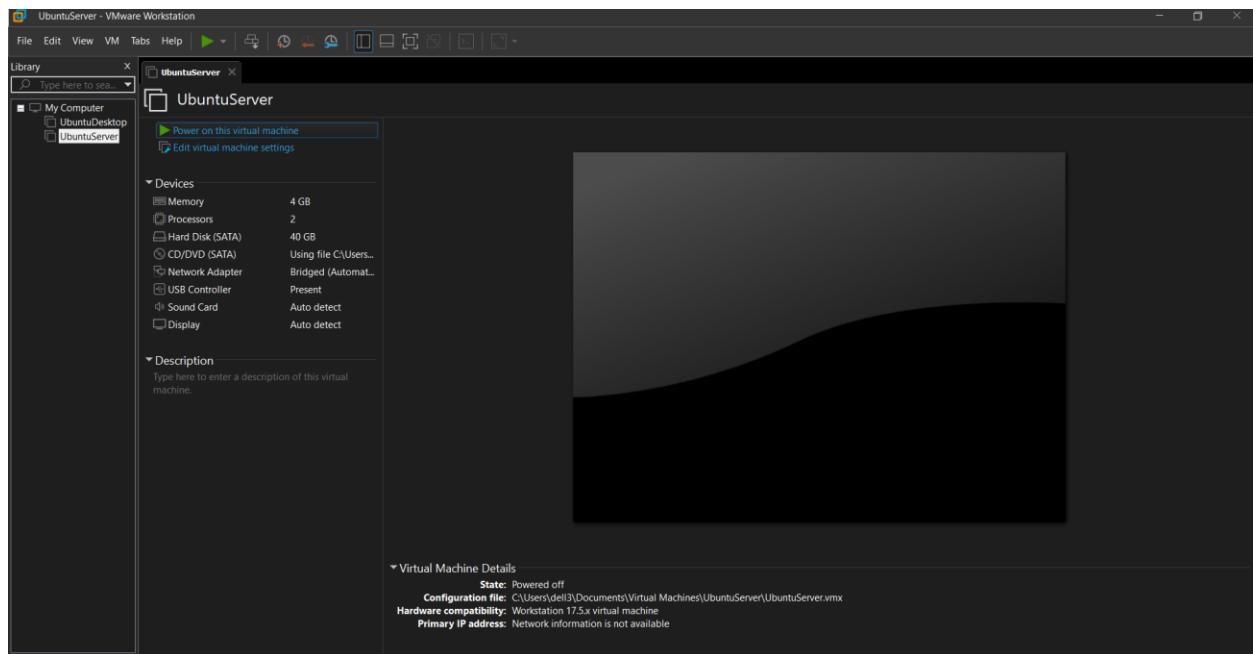
1. Create a shell script to print the HTTP error code of guvi.in & print, the success/failure message based on the error code response
2. Given a file, replace all occurrence of the word "give" with "learning" from 5th line till the end in only those lines that contain the word "welcome".

Linux and Shell Scripting Task Summary:

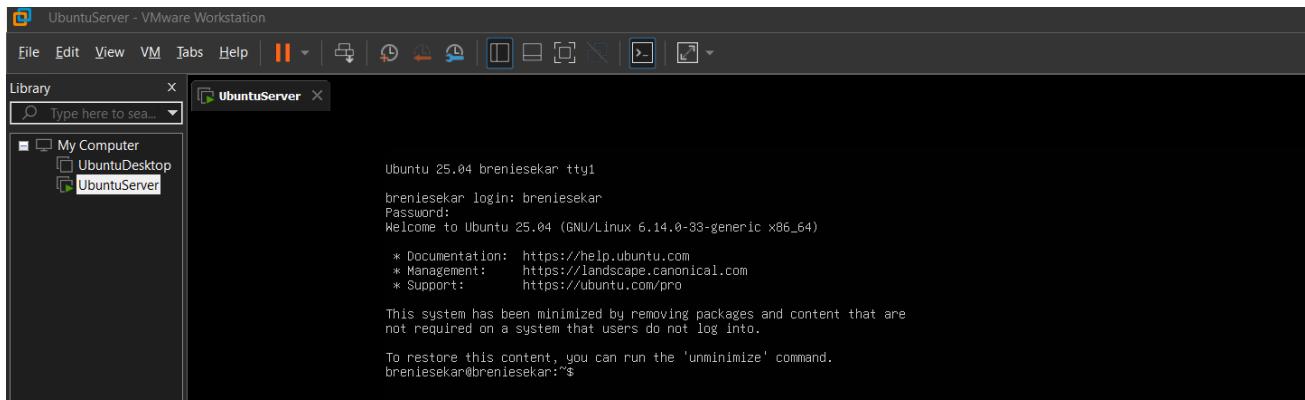
- Powering on the Linux server and connecting to it using the MobaXterm terminal via SSH.
- Shell script to print the success/failure message based on the error code response.
- Shell script to replace every occurrence of the word ‘give’ with ‘learning’ from the 5th line onward, only in lines that contain the word ‘welcome’.

Powering on the Linux server and connecting to it using the MobaXterm terminal via SSH:

Powering on the Linux Server



Login to the Linux Server



The screenshot shows a VMware Workstation interface with a single running VM named "UbuntuServer". The terminal window displays the initial boot sequence of an Ubuntu 25.04 LTS system. The user logs in with the username "breniesekar" and password "breniesekar". The terminal then presents the standard Ubuntu welcome message and documentation links. It also notes that the system has been minimized and provides instructions to restore content using the "unminimize" command.

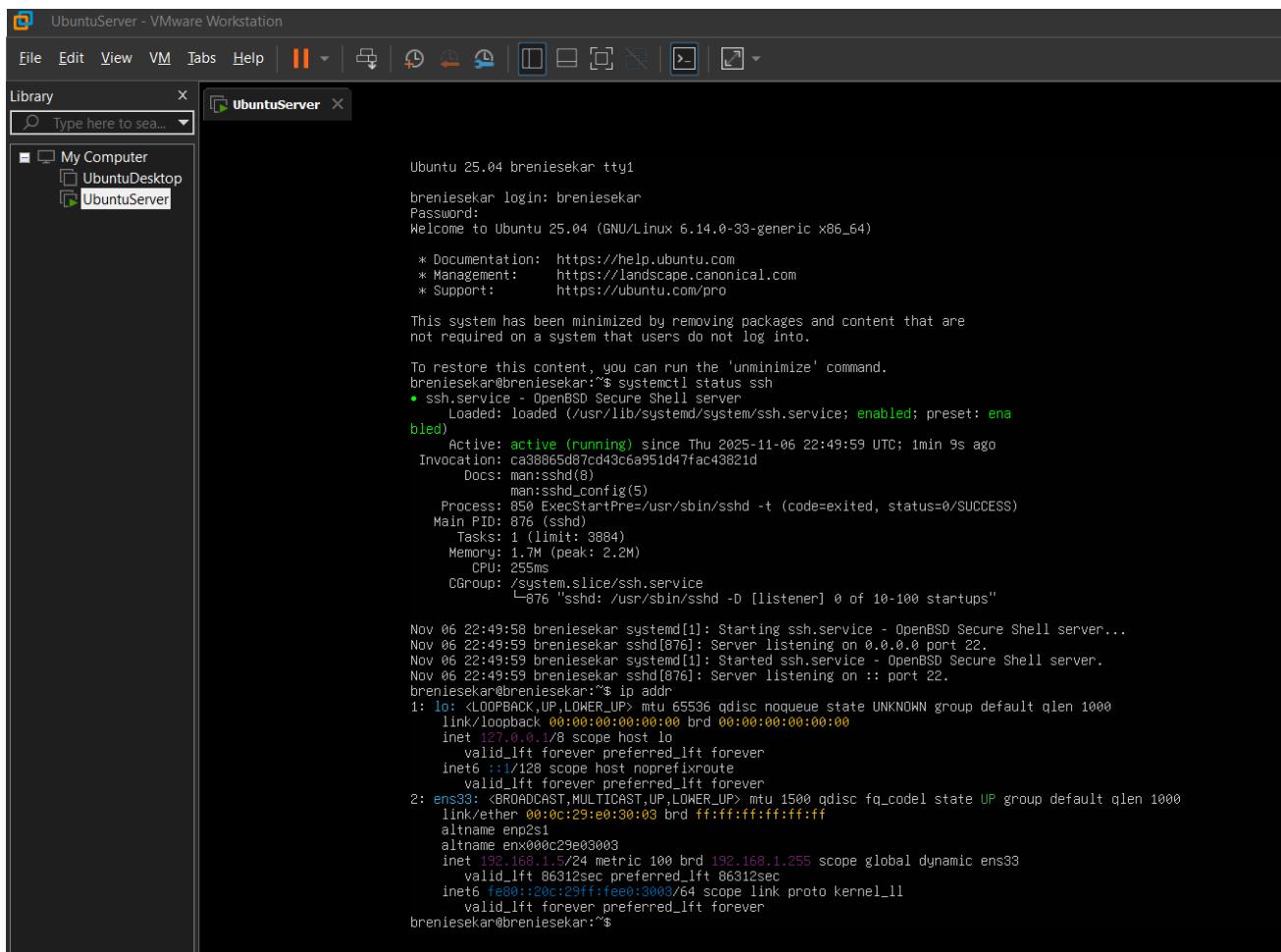
```
Ubuntu 25.04 breniesekar tty1
breniesekar login: breniesekar
Password:
Welcome to Ubuntu 25.04 (GNU/Linux 6.14.0-33-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
breniesekar@breniesekar:~$
```

Checking SSH Status and getting the IP address of the server



The screenshot shows a VMware Workstation interface with a single running VM named "UbuntuServer". The terminal window displays the output of the command "sshd -D", which shows the status of the OpenBSD Secure Shell service. The service is listed as active and loaded. The command "ip addr" is then run, showing the network configuration of the host machine. The output includes details about the loopback interface (lo) and the ens33 interface, including their MAC addresses, MTU values, and queueing disciplines.

```
Ubuntu 25.04 breniesekar tty1
breniesekar login: breniesekar
Password:
Welcome to Ubuntu 25.04 (GNU/Linux 6.14.0-33-generic x86_64)

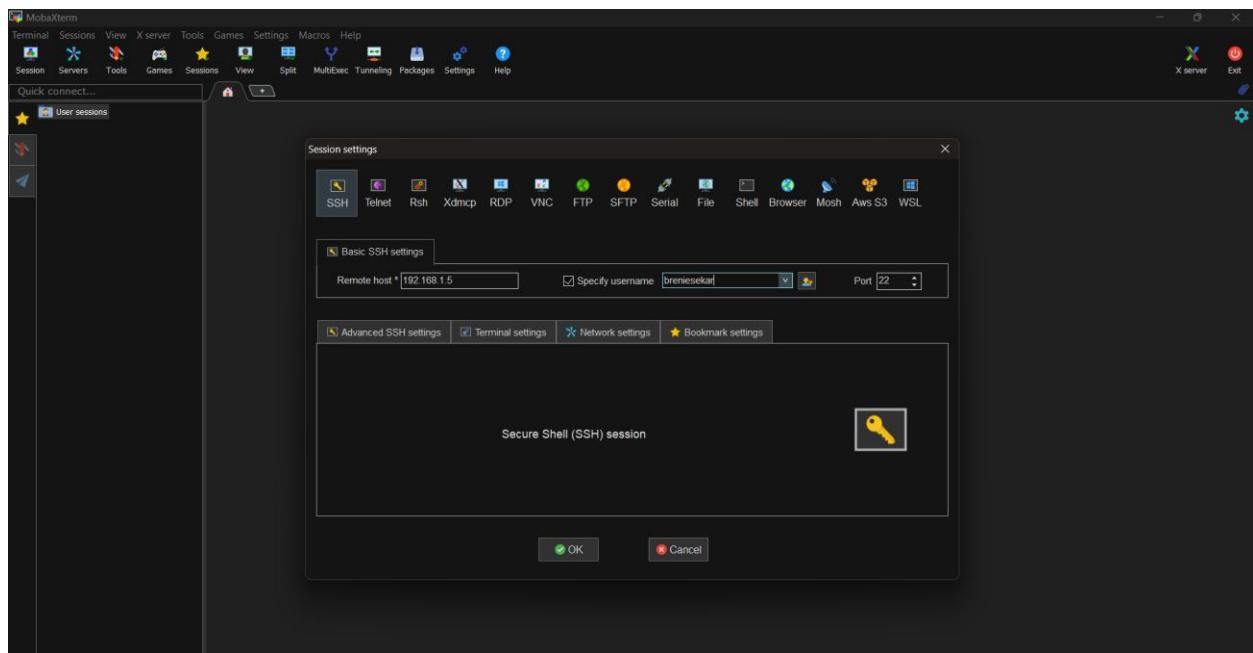
 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

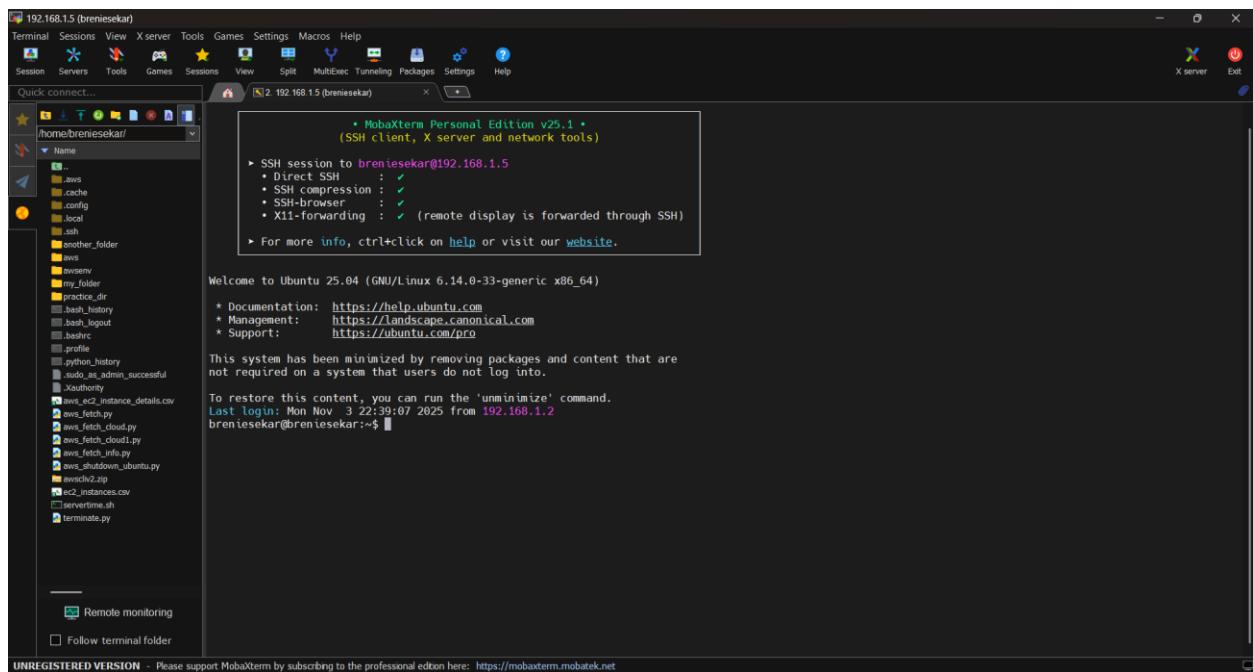
To restore this content, you can run the 'unminimize' command.
breniesekar@breniesekar:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: ena
bled)
   Active: active (running) since Thu 2025-11-06 22:49:59 UTC; 1min 9s ago
     Invocation: ca3886fd97cd43c6a951d47fac43821d
       Docs: man:sshd(8)
             man:sshd_config(5)
      Process: 856 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
      Main PID: 876 (sshd)
        Tasks: 1 (limit: 3884)
       Memory: 1.7M (peak: 2.2M)
         CPU: 255ms
        CGroup: /system.slice/ssh.service
                 └─876 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Nov 06 22:49:58 breniesekar systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Nov 06 22:49:59 breniesekar sshd[876]: Server listening on 0.0.0.0 port 22.
Nov 06 22:49:59 breniesekar systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Nov 06 22:49:59 breniesekar sshd[876]: Server listening on :: port 22.
breniesekar@breniesekar:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:e0:30:03 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    altname enx000c29e003003
    inet 192.168.1.5/24 metric 100 brd 192.168.1.255 scope global dynamic ens33
        valid_lft 86312sec preferred_lft 86312sec
    inet6 fe80::20c:29ff:fe0e:3003/64 scope link proto kernel ll
        valid_lft forever preferred_lft forever
breniesekar@breniesekar:~$
```

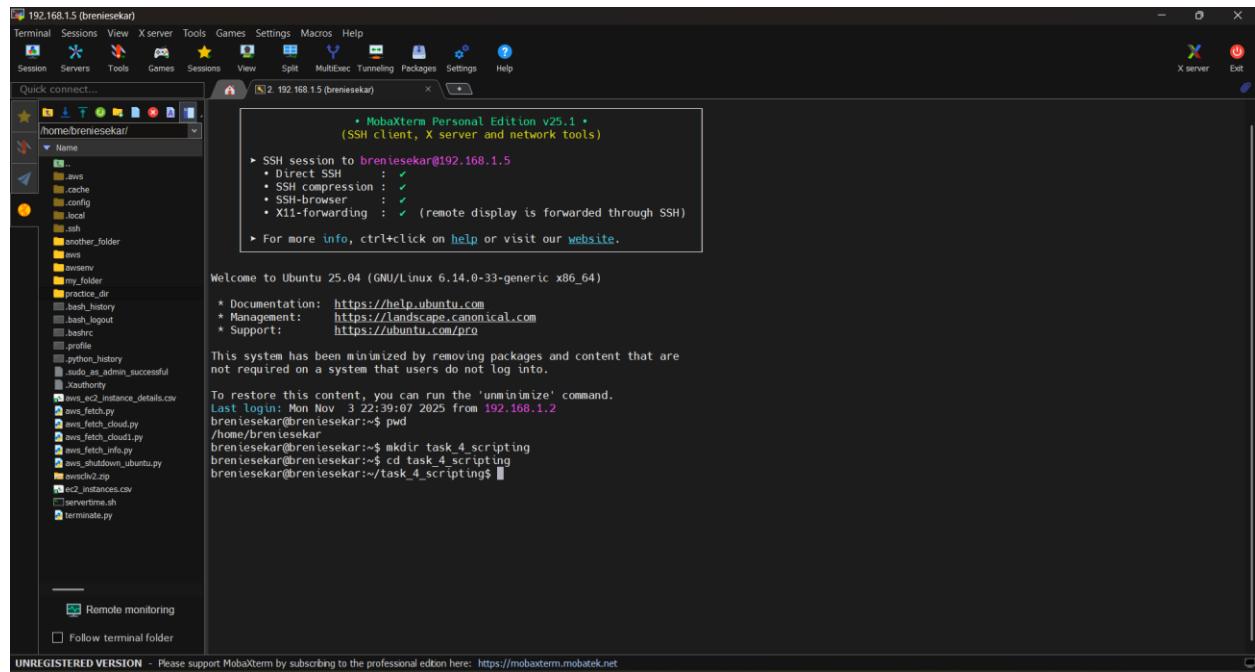
Creating a new session on MobaXterm Terminal



Login to the Sever using SSH on MobaXterm Terminal



Created a new directory for shell scripting (task_4_scripting)



Task 1: HTTP Response Handling

Create a shell script to print the HTTP error code of guvi.in & print, the success/failure message based on the error code response

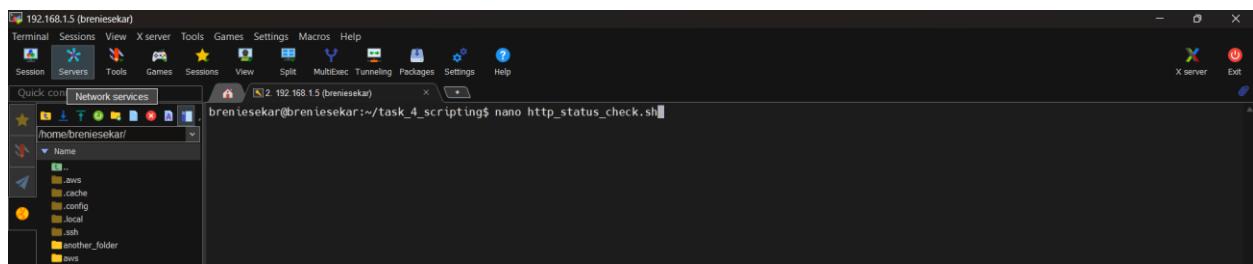
Commands Used:

Sl.no	Command	Explanation
1	nano http_status_check.sh	Creating a new .sh file
2	<pre>#!/bin/bash url="https://www.guvi.in" status_code=\$(curl -o /dev/null -s -w "%{http_code}" "\$url") echo "HTTP Status Code for \$url: \$status_code" if ["\$status_code" -eq 200]; then echo "Success: The website is reachable." else echo "Failure: The website returned an error (code \$status_code)." fi</pre>	Shell Script for http response handling
2	chmod a+x http_status_check.sh	Allowing execution permission to http_status_check.sh
3	./http_status_check.sh	Executing the code

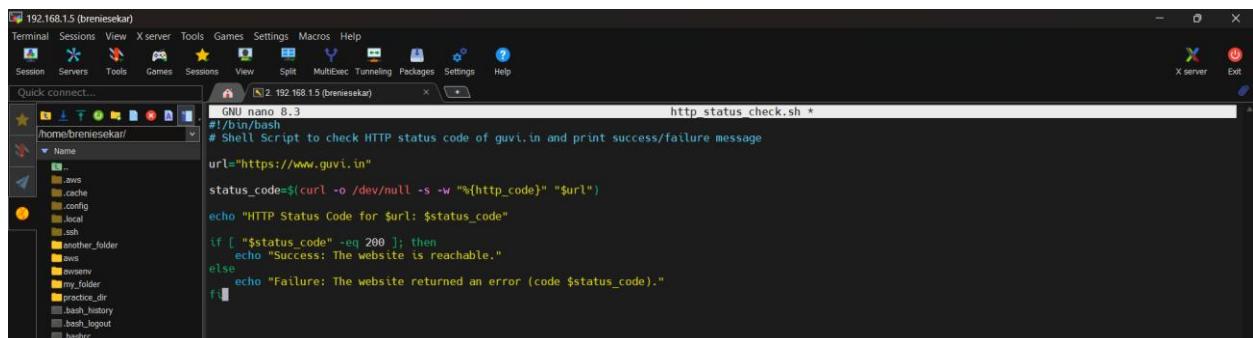
Sample Input and Expected Output :

Input	Expected output
"https://www.guvi.in"	HTTP Status Code for https://www.guvi.in : 200 Success: The website is reachable.

Created a new script file (http_status_check.sh)



Writing the script for HTTP response Handling



Shell Script with comments:

```
#!/bin/bash
#
# -----
# Script Name: http_status_check.sh
# Description: Checks the HTTP status code of guvi.in and
#               prints a success or failure message based on
#               the response.
# -----
```

Step 1: Define the website URL to check

```

url="https://www.guvi.in"

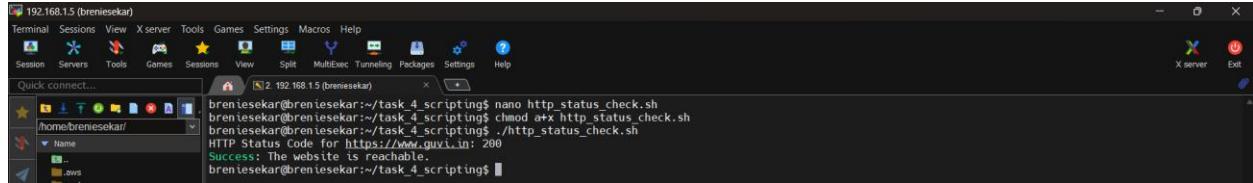
# Step 2: Use curl to retrieve only the HTTP status code
# -o /dev/null : Discards the response body
# -s           : Runs in silent mode (no progress or errors shown)
# -w "%{http_code}" : Prints only the HTTP status code
status_code=$(curl -o /dev/null -s -w "%{http_code}" "$url")

# Step 3: Display the retrieved HTTP status code
echo "HTTP Status Code for $url: $status_code"

# Step 4: Evaluate the response code and print result
if [ "$status_code" -eq 200 ]; then
    # If status code is 200, the website is reachable
    echo "Success: The website is reachable."
else
    # If not 200, it indicates a failure or error response
    echo "Failure: The website returned an error (code $status_code)."
fi

```

Providing execute permission to the shell script and executing the script



Result :

Input : <https://www.guvi.in>

Output : HTTP Status Code for <https://www.guvi.in> : 200
 "Success: The website is reachable."

Task 2: Conditional Text Replacement

Given a file, replace all occurrence of the word "give" with "learning" from 5th line till the end in only those lines that contain the word "welcome".

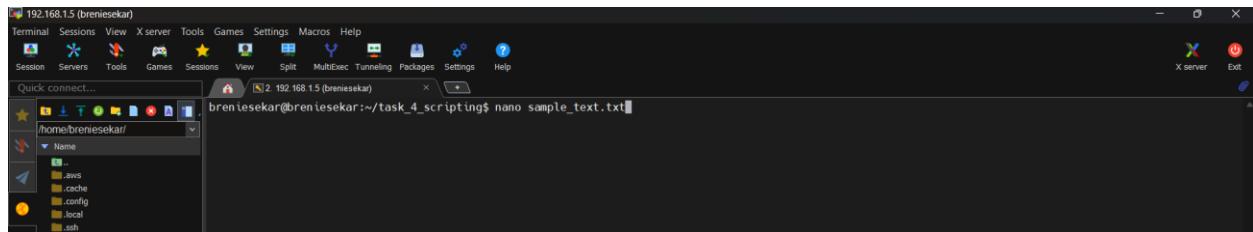
Commands Used:

Sl.no	Command	Explanation
1	<code>nano sample_text.txt</code>	Creating a sample text file
2	<code>1 welcome give your best in everything. 2 This file is used to test the sed command. 3 welcome give everyone a warm greeting. 4 Practice shell scripting daily. 5 welcome give and learn from mistakes. 6 welcome learning your knowledge freely. 7 Always try to give your 100%. 8 welcome give your ideas to others. 9 give time to understand the basics. 10 give welcome and grow every day.</code>	Writing few sample text in the .txt file for testing
3	<code>nano conditional_text_replace.sh</code> <pre>#!/bin/bash # shell script to Replace "give" with "learning" from line 5 onward in lines having "welcome" file="\$1" if [-z "\$file"]; then echo "Usage: \$0 <filename>" exit 1 elif [! -f "\$file"]; then echo "Error: File '\$file' not found!" exit 1 fi sed -i '5,\${/welcome/s/give/learning/g}' "\$file" echo "Replacement completed in '\$file'."</pre>	Creating a new .sh file Shell Script for http response handling
2	<code>chmod a+x conditional_text_replace.sh</code>	Allowing execution permission to conditional_text_replace.sh
3	<code>./conditional_text_replace.sh sample_text.txt</code>	Executing the code

Sample Input and Expected Output :

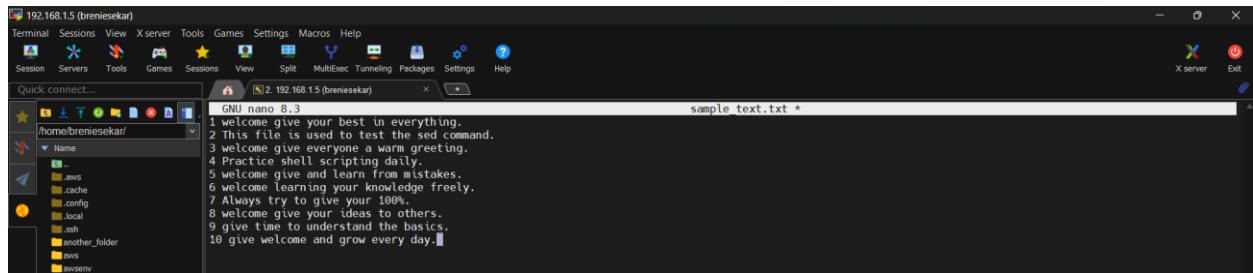
Input	Output
<p>1 welcome give your best in everything. 2 This file is used to test the sed command. 3 welcome give everyone a warm greeting. 4 Practice shell scripting daily. 5 welcome give and learn from mistakes. 6 welcome learning your knowledge freely. 7 Always try to give your 100%. 8 welcome give your ideas to others. 9 give time to understand the basics. 10 give welcome and grow every day.</p>	<p>1 welcome give your best in everything. 2 This file is used to test the sed command. 3 welcome give everyone a warm greeting. 4 Practice shell scripting daily. 5 welcome learning and learn from mistakes. 6 welcome learning your knowledge freely. 7 Always try to give your 100%. 8 welcome learning your ideas to others. 9 give time to understand the basics. 10 learning welcome and grow every day.</p>

Created a new text file (sample_text.txt)



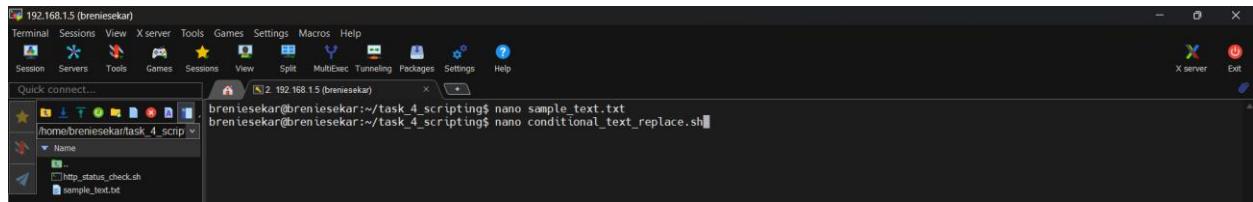
```
brentesekar@brentesekar:~/task_4_scripting$ nano sample_text.txt
```

Writing few sample texts in the sample_text.txt file for testing



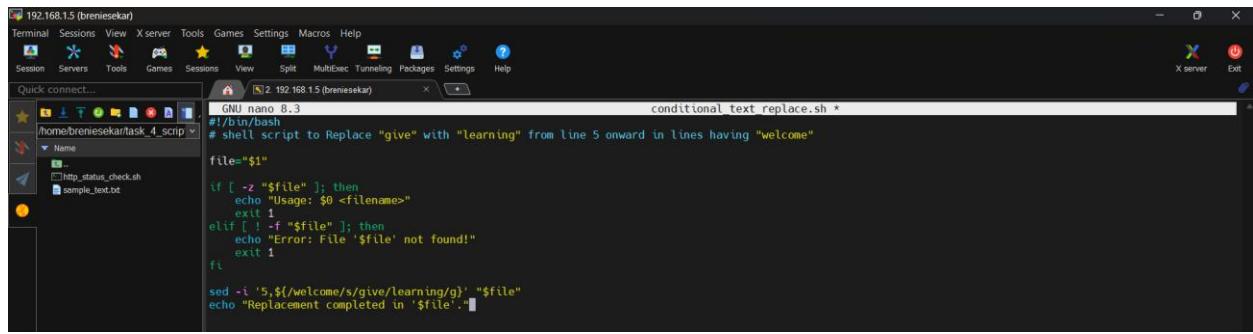
```
GNU nano 8.3
1 welcome give your best in everything.
2 This file is used to test the sed command.
3 welcome give everyone a warm greeting.
4 Practice shell scripting daily.
5 welcome give and learn from mistakes.
6 welcome learning your knowledge freely.
7 Always try to give your 100%.
8 welcome give your ideas to others.
9 give time to understand the basics.
10 give welcome and grow every day.
```

Created a new script file (conditional_text_replace.sh)



```
brentesekar@brentesekar:~/task_4_scripting$ nano sample_text.txt
brentesekar@brentesekar:~/task_4_scripting$ nano conditional_text_replace.sh
```

Writing the script for conditional text replacement



```
#!/bin/bash
# shell script to Replace "give" with "learning" from line 5 onward in lines having "welcome"
file="$1"
if [ -z "$file" ]; then
    echo "Usage: $0 <filename>"
    exit 1
elif [ ! -f "$file" ]; then
    echo "Error: File '$file' not found!"
    exit 1
fi
sed -i '5,${/welcome/s/give/learning/g}' "$file"
echo "Replacement completed in '$file'."
```

Shell Script with comments:

```
#!/bin/bash
# -----
# Script Name: conditional_text_replace.sh
# Description: Shell script to replace all occurrences of the word
#           "give" with "learning" from line 5 onward,
#           only in lines that contain the word "welcome".
# -----
# Step 1: Accept the input filename from the first command-line argument
file="$1"

# Step 2: Check if the filename argument is provided
# -z "$file" → true if the variable is empty
if [ -z "$file" ]; then
    echo "Usage: $0 <filename>"
    exit 1

# Step 3: Check if the specified file actually exists
# -f "$file" → true if it's a regular file
elif [ ! -f "$file" ]; then
    echo "Error: File '$file' not found!"
    exit 1
fi

# Step 4: Use sed to perform the replacement
# 5,$           → apply from line 5 to the end of the file
# /welcome/      → restrict action to lines containing "welcome"
```

```
# s/give/learning/g → substitute all "give" with "learning" in matching lines
# -i → edit the file in place (save changes directly)
sed -i '5,${/welcome/s/give/learning/g}' "$file"
```

```
# Step 5: Print completion message
echo "Replacement completed in '$file'."
```

Providing execution permission to the shell script and listing all the files and folders within the directory

The screenshot shows a terminal window titled '192.168.1.5 (breniesekar)'. The user has run several commands in their shell session:

```
breniesekar@breniesekar:~/task_4_scripting$ nano sample_text.txt
breniesekar@breniesekar:~/task_4_scripting$ nano conditional_text_replace.sh
breniesekar@breniesekar:~/task_4_scripting$ chmod +x conditional_text_replace.sh
breniesekar@breniesekar:~/task_4_scripting$ ls -lrt
total 12
-rwxrwxr-x 1 breniesekar breniesekar 397 Nov  6 23:55 http_status_check.sh
-rw-rw-r-- 1 breniesekar breniesekar 385 Nov  7 03:28 sample_text.txt
-rwxrwxr-x 1 breniesekar breniesekar 360 Nov  7 03:30 conditional_text_replace.sh
breniesekar@breniesekar:~/task_4_scripting$
```

Executing the script

The screenshot shows a terminal window titled '192.168.1.5 (breniesekar)'. The user has run several commands in their shell session:

```
breniesekar@breniesekar:~/task_4_scripting$ ls -lrt
total 12
-rwxrwxr-x 1 breniesekar breniesekar 397 Nov  6 23:55 http_status_check.sh
-rw-rw-r-- 1 breniesekar breniesekar 385 Nov  7 03:28 sample_text.txt
-rwxrwxr-x 1 breniesekar breniesekar 360 Nov  7 03:30 conditional_text_replace.sh
breniesekar@breniesekar:~/task_4_scripting$ ./conditional_text_replace.sh sample_text.txt
Replacement completed in 'sample_text.txt'.
breniesekar@breniesekar:~/task_4_scripting$
```

Verifying the result

The screenshot shows a terminal window titled '192.168.1.5 (breniesekar)'. The user has run several commands in their shell session:

```
GNU nano 8.3
1 Welcome give your best in everything.
2 The first step need to test the sed command.
3 welcome give everyone a warm greeting.
4 Practice shell scripting daily.
5 welcome learning and learn from mistakes.
6 welcome learning your knowledge freely.
7 Always try to give your 100%.
8 welcome learning your ideas to others.
9 give time to understand the basics.
10 learning welcome and grow every day.
```

Result :

Input	Output
<p>1 welcome give your best in everything. 2 This file is used to test the sed command. 3 welcome give everyone a warm greeting. 4 Practice shell scripting daily. 5 welcome give and learn from mistakes. 6 welcome learning your knowledge freely. 7 Always try to give your 100%. 8 welcome give your ideas to others. 9 give time to understand the basics. 10 give welcome and grow every day.</p>	<p>1 welcome give your best in everything. 2 This file is used to test the sed command. 3 welcome give everyone a warm greeting. 4 Practice shell scripting daily. 5 welcome learning and learn from mistakes. 6 welcome learning your knowledge freely. 7 Always try to give your 100%. 8 welcome learning your ideas to others. 9 give time to understand the basics. 10 learning welcome and grow every day.</p>

Task Summary :

Task No.	Task Description	Key Commands / Logic Used	Purpose / Output
1	Create a shell script to print the HTTP status code of guvi.in and display a success or failure message based on the response.	<ul style="list-style-type: none"> ➤ <code>curl -o /dev/null -s -w "%{http_code}" "\$url"</code> to get HTTP status code. ➤ <code>if ["\$status_code" -eq 200]; then ... else ... fi</code> for condition check. 	Prints the HTTP status code (e.g., 200) and displays whether the website is reachable (Success) or not (Failure).
2	Given a file, replace all occurrences of “ give ” with “ learning ” from the 5th line onward, only in lines containing “ welcome .”	<ul style="list-style-type: none"> ➤ <code>sed -i '5,\${/welcome/s/give/learning/g}' "\$file"</code> to perform the text replacement. ➤ Input validation with <code>if [-z "\$file"]</code> and <code>[! -f "\$file"]</code>. 	Modifies the given file in place, replacing “ give ” → “ learning ” only in lines containing “ welcome ” from line 5 to the end.