Task 8: Website Availability Checker

1. Create your list of websites:

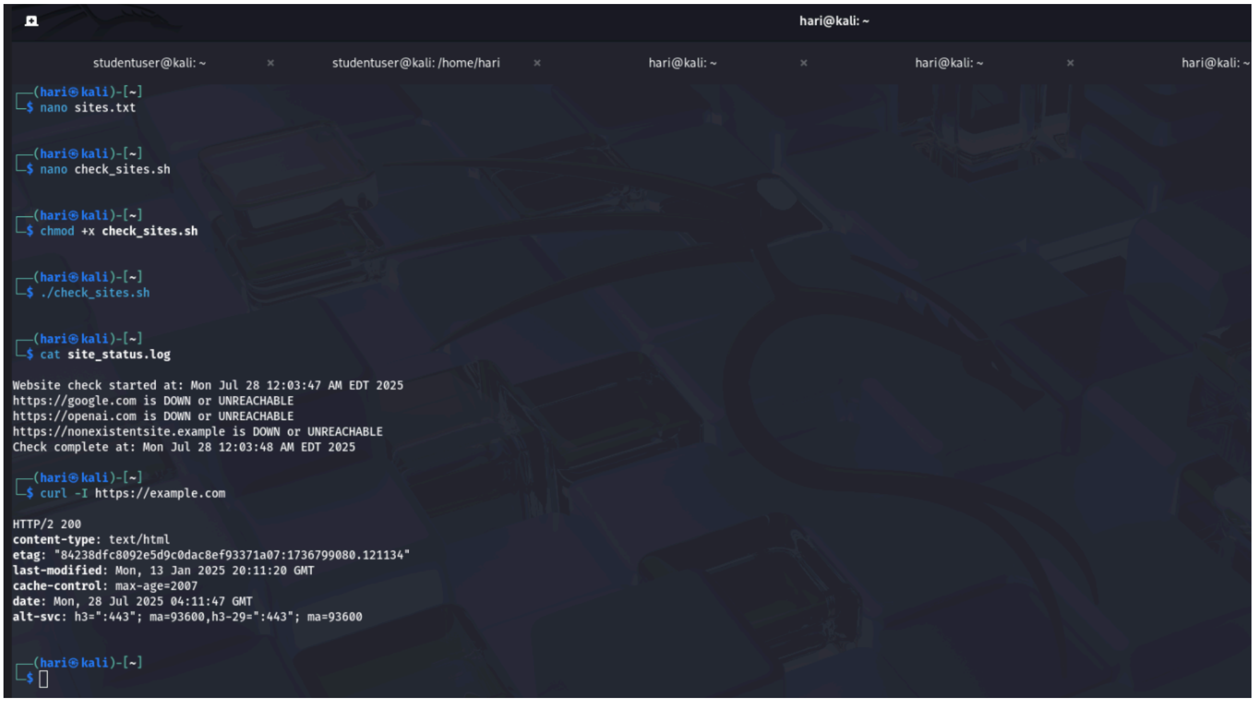
```
nano sites.txt
https://google.com
https://openai.com
https://nonexistentsite.example
```

2. Create the script:

```
nano check_sites.sh
#!/bin/bash
LOG_FILE="site_status.log"
DATE=$(date)
echo "Website check started at: $DATE" > $LOG_FILE
while read -r SITE; do
if curl -Is --max-time 5 "$SITE" | grep "200 OK" > /dev/null; then
echo "$SITE is UP" >> $LOG_FILE
else
echo "$SITE is DOWN or UNREACHABLE" >> $LOG_FILE
fi
done < sites.txt
echo "Check complete at: $(date)" >> $LOG_FILE
```

3. Run it:

```
chmod +x check_sites.sh
./check_sites.sh
cat site_status.log
```



```
┌──(hari㉿kali)-[~]
└─$ nano sites.txt

┌──(hari㉿kali)-[~]
└─$ nano check_sites.sh

┌──(hari㉿kali)-[~]
└─$ chmod +x check_sites.sh

┌──(hari㉿kali)-[~]
└─$ ./check_sites.sh

┌──(hari㉿kali)-[~]
└─$ cat site_status.log
Website check started at: Mon Jul 28 12:03:47 AM EDT 2025
https://google.com is DOWN or UNREACHABLE
https://openai.com is DOWN or UNREACHABLE
https://nonexistentsite.example is DOWN or UNREACHABLE
Check complete at: Mon Jul 28 12:03:48 AM EDT 2025

┌──(hari㉿kali)-[~]
└─$ curl -I https://example.com
HTTP/2 200
content-type: text/html
etag: "84238dfc8092e5d9c0dac8ef93371a07:1736799080.121134"
last-modified: Mon, 13 Jan 2025 20:11:20 GMT
cache-control: max-age=2007
date: Mon, 28 Jul 2025 04:11:47 GMT
alt-svc: h3=":443"; ma=93600,h3-29=":443"; ma=93600


┌──(hari㉿kali)-[~]
└─$ ▯
```

Conclusion: Task 8 focused on checking website availability by reading URLs from `sites.txt` and using `curl` or `ping` to test connectivity. The results were logged to `site_status.log`, helping automate uptime checks for multiple websites.