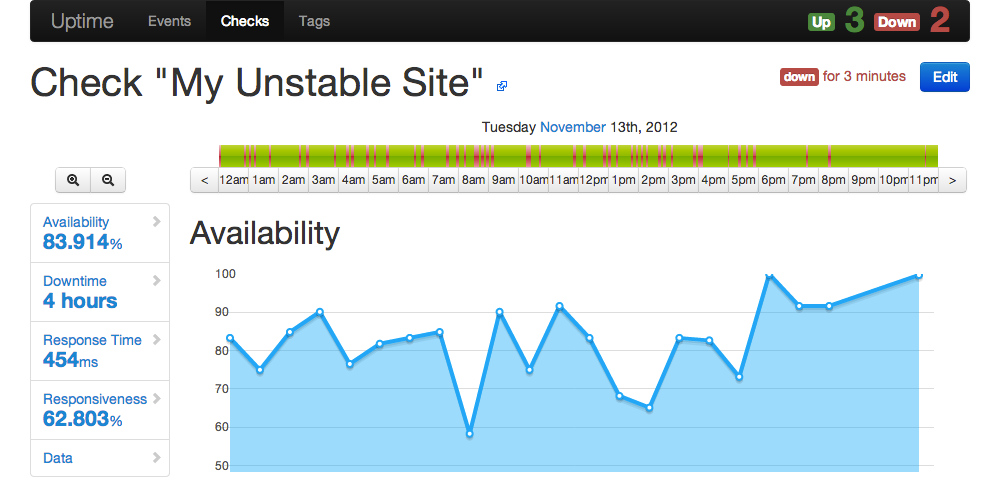
# Operating System Principles Lab: Group (6)-Members:

* **M.Fayyaz-077**
* **Aun Haider-065**
* **Ali Haider-058**
* **Asim Tahir-064**

**Project Report**

**Website Availability Checker**

****

**1. Introduction** The objective of this project is to create a simple yet effective Website Availability Checker using a shell script in a Linux environment. This tool checks the availability of a website by sending HTTP requests and logs the response status and time. This report details the steps to build the project, the software requirements, the usage of the curl command, and provides space for screenshots to document the implementation.

**2. Software and Tools**

* **VMware Workstation 7**: A virtualization software that allows users to run multiple operating systems on a single machine. For this project, VMware is used to run Fedora Linux.
* **Fedora Linux 42**: A popular Linux distribution chosen for its stability and developer-friendly features. It serves as the operating system where the shell script is executed.
* **Curl**: A command-line tool for transferring data using various network protocols. In this project, curl is used to send HTTP requests and capture response codes and times.

**3. Curl and Its Role in the Project**

curl is a versatile tool for interacting with web servers. It supports various protocols like HTTP, HTTPS, FTP, and more. In this project, curl is specifically used to:

1. Send HTTP GET requests to the provided website URL.
2. Retrieve the HTTP response status code.
3. Measure the total response time of the website.

Command format:

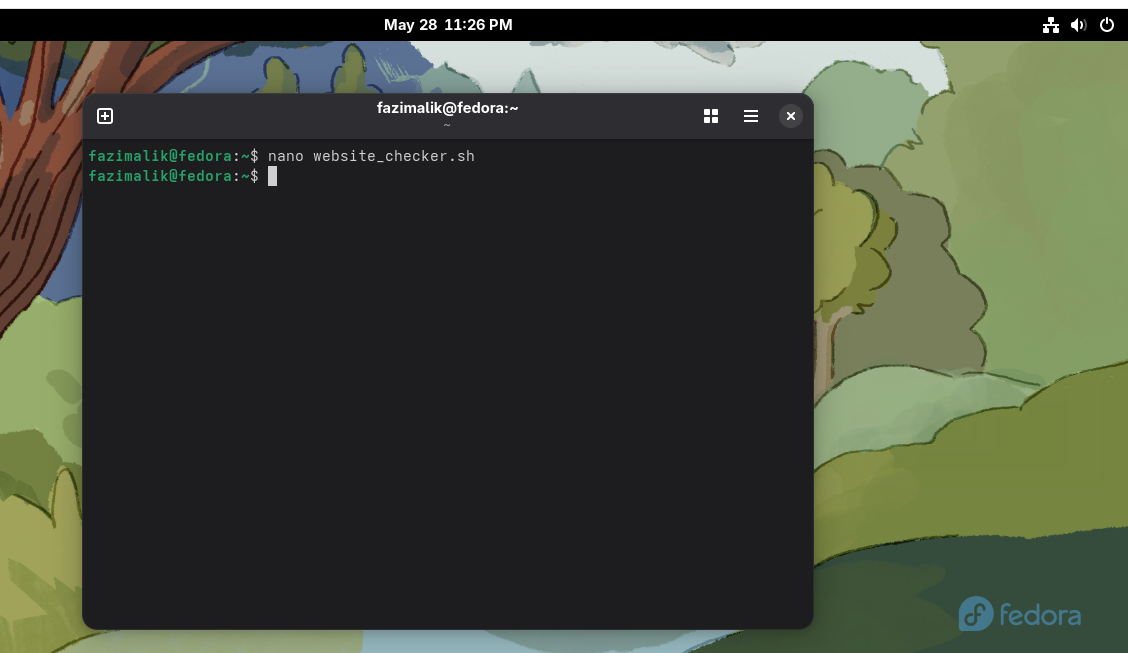
curl -o /dev/null -s -w "%{http\_code} %{time\_total}\n" "<URL>"

* **-o /dev/null**: Discards the response body.
* **-s**: Silent mode to suppress unnecessary output.
* **-w**: Outputs the desired metrics (e.g., status code and response time).

**4. Step-by-Step Implementation**

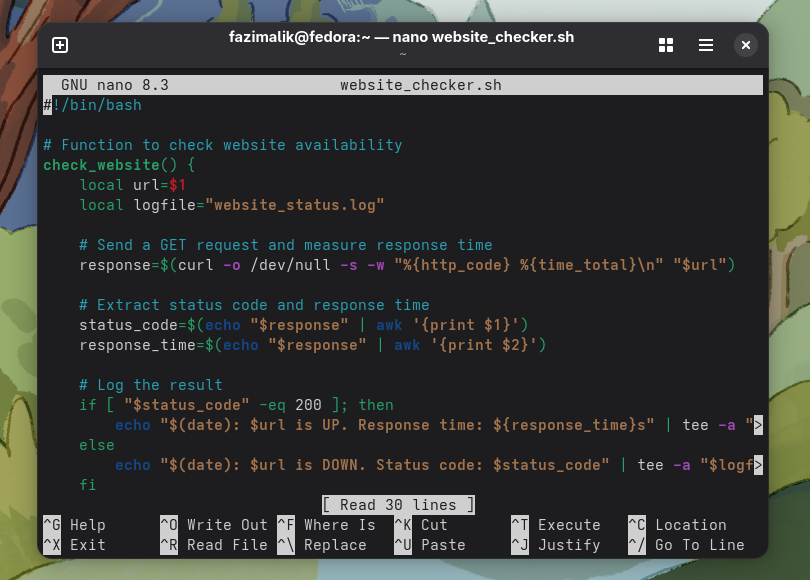
1. **Open the Terminal and Create a New Script File** Command:

nano website\_checker.sh



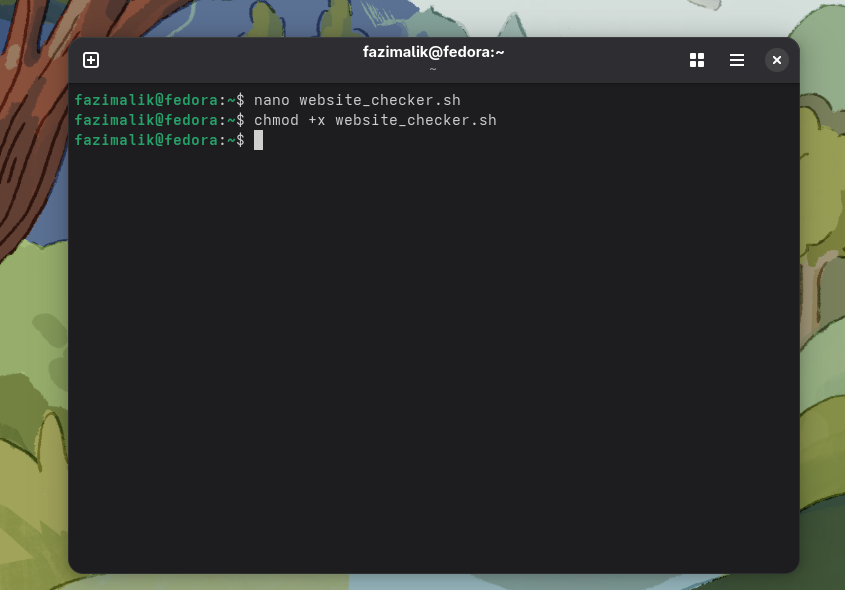
1. **Write the Bash Script** Paste the following code into the script file:
2. #!/bin/bash
3. # Function to check website availability
4. check\_website() {
5. local url=$1
6. local logfile="website\_status.log"
7. # Send a GET request and measure response time
8. response=$(curl -o /dev/null -s -w "%{http\_code} %{time\_total}\n" "$url")
9. # Extract status code and response time
10. status\_code=$(echo "$response" | awk '{print $1}')
11. response\_time=$(echo "$response" | awk '{print $2}')
12. # Log the result
13. if [ "$status\_code" -eq 200 ]; then
14. echo "$(date): $url is UP. Response time: ${response\_time}s" | tee -a "$logfile"
15. else
16. echo "$(date): $url is DOWN. Status code: $status\_code" | tee -a "$logfile"
17. fi
18. }
19. # Ensure a URL is provided
20. if [ -z "$1" ]; then
21. echo "Usage: $0 <website\_url>"
22. exit 1
23. fi
24. # Call the function with the provided URL

check\_website "$1"



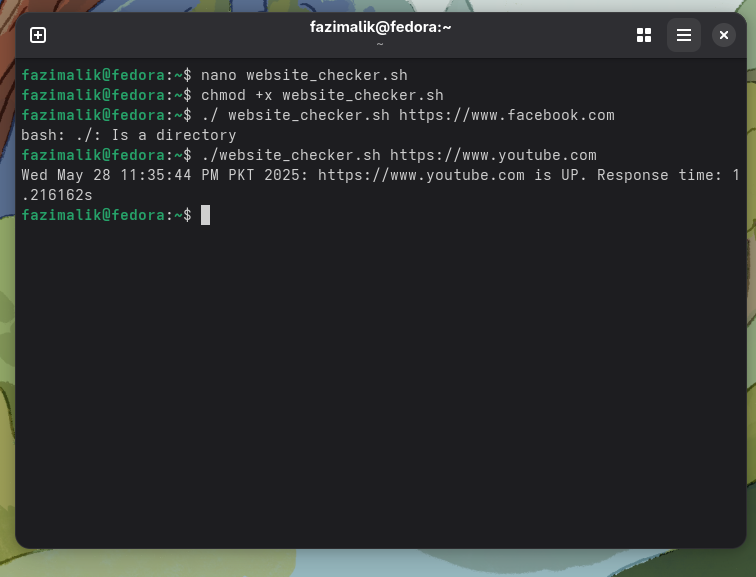
1. **Save and Exit the File**
   * Press Ctrl + O to save the file.
   * Press Ctrl + X to exit.
2. **Make the Script Executable** Command:

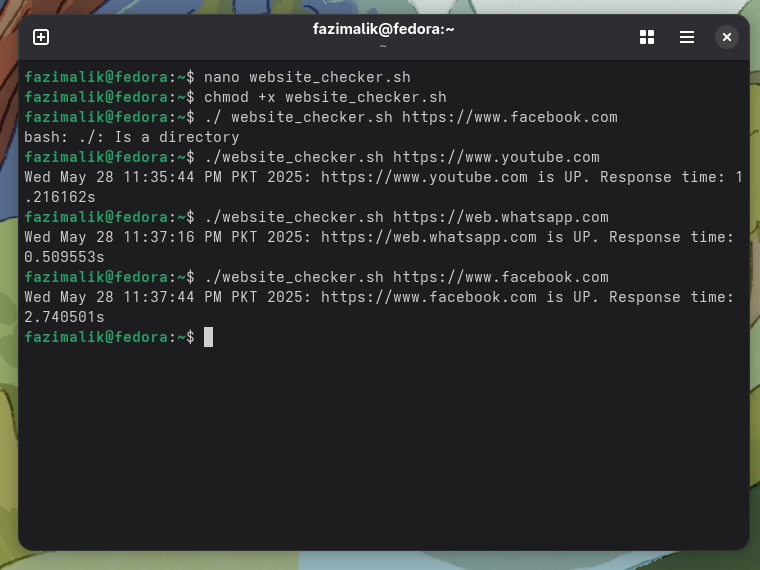
chmod +x website\_checker.sh



1. **Run the Script** Command:

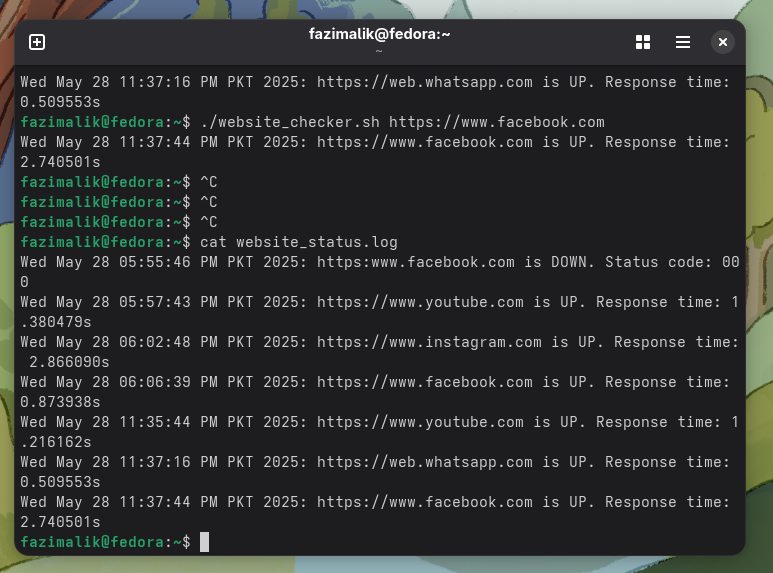
./website\_checker.sh https://example.com





1. **Verify the Log File** Command:

cat website\_status.log



**5. Enhancements**

1. **Check Multiple Websites**: Modify the script to include a list of URLs to check in one execution.
2. **Automate with Cron**: Schedule the script to run periodically using cron. Command:

crontab -e

Add a job like this to check every hour:

0 \* \* \* \* /path/to/website\_checker.sh

**6. Conclusion**

The Website Availability Checker is a simple yet powerful tool for monitoring the uptime of websites. Using Linux, curl, and shell scripting, students can build an automated solution for detecting downtime and logging response times. This project also introduces foundational scripting and system administration skills essential for IT professionals.