**OST CA3**

**Report**

**Group Information**

|  |  |  |
| --- | --- | --- |
| **Sno** | **PRN** | **Name** |
| **1** | **23070123008** | **Aditya Sthawarmath** |
| **2** | **23070123011** | **Adrish Purkayastha** |
| **3** | **23070123073** | **Kautik Verma** |
| **4** | **23070123169** | **Gayatri Ratnaparkhi** |
| **5** | **24070123505** | **Sakshi Sonawane** |

**Submitted Under the Guidance of :**

Dr. Amol Kamble

1. **Working of Project**

**Introduction:**

The Network Analytics Suite is a Bash based modular framework to monitor, analyse and log a range of network performance on Linux systems.

The main purpose of this suite will be to offer real-time information about the network behaviour, speed, connection and reliability using automated shell scripts.

The performance of network is one of the essential aspects of contemporary computing where the stability of the networks and the efficient internet usage are of utmost importance in the functioning of the applications. This suite makes the task of collecting performance data easier because it brings a number of network diagnostic tools together, under one unified framework

**System Overview:**

**The suite is provided with five autonomous modules and one master controller:**

1. Network speed test- Checks download and upload speeds and tests ping latency.
2. Network Interface Info - List and recorded IP address, MAC address and the operational status of all network interfaces.
3. DNS Resolver Check- Tests the availability and responsiveness of a number of DNS servers.
4. Signal Strength Monitor Signal Quality and Strength (in dBm).
5. HTTP Availability Checker -Checks the availability and responsiveness of websites.

**Requirements:**

1. Linux OS
2. Bash shell
3. Utilities: ping, curl, ip, iwconfig, nslookup
4. Optional (for detailed DNS checks): dig (dnsutils package)
5. GitHub

**Screenshots of Execution:**

A screenshot of a computer

AI-generated content may be incorrect.

*Signal Strength*

A computer screen shot of a black screen

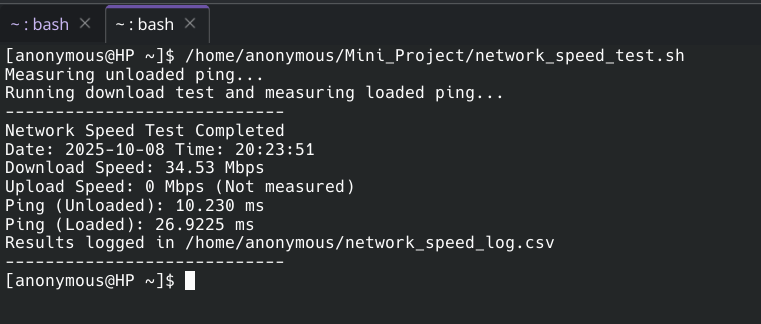
AI-generated content may be incorrect.

*Network Interface*

A computer screen with white text

AI-generated content may be incorrect.

*DNS Resolver*



*Speed Test*

A computer screen with white text

AI-generated content may be incorrect.

*Website Availability Checker*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**How to Run the Project:**

**Steps to Execute:**

1. Clone the repository**:** git clone [https://github.com/Aditya-2805/Mini\_Project.git](https://github.com/Aditya-2805/Mini\_Project.git)
2. Navigate to the project directory: cd Mini\_Project
3. Make the scripts executable: chmod +x \*.sh
4. Run any of the desired scripts**:**
   * To check network interfaces: ./network\_interface.sh
   * To test network speed: ./network\_speed\_test.sh
   * To check Wi-Fi signal: ./signal\_strength.sh
   * To check DNS resolvers: ./dns\_resolver\_check.sh
5. **Innovation / Novelty / Contribution**

**Unique Feature or Enhancement**

The primary innovation of this project lies in its consolidation and automation. While the underlying commands (dig, speedtest-cli, etc.) are standard, our toolkit integrates them into a user-friendly suite. The key enhancements are:

* Automated Logging: Every script automatically appends its output to a corresponding .csv file, creating a time-stamped log of network performance without any manual intervention.
* Simplicity and Portability: The toolkit is built entirely with shell scripts, requiring minimal dependencies that are common on most Linux systems. This makes it extremely lightweight and portable.
* Centralized Toolkit: It provides a single, consistent interface for multiple network diagnostic tasks, saving the user from having to remember and type out various complex commands.

**Student’s Individual Contribution**

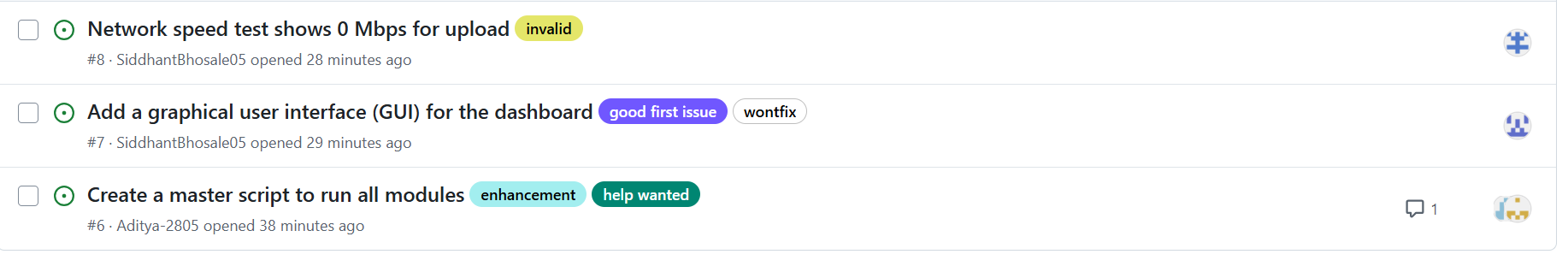
| **Student Name** | **Role/Contribution** | **GitHub Profile** | **Key Commits / Pull Requests** |
| --- | --- | --- | --- |
| **1.Aditya Sthawarmath** | Developed speed\_test.sh | [Aditya-2805](https://github.com/Aditya-2805) | Master Script |
| **2.Adrish Purkayastha** | Developed nerwork\_interface.sh | [adrish20 (Adrish Purkayastha)](https://github.com/adrish20) | Separate tests for loaded and unloaded ping measurements |
| **3.Kautik Verma** | Developed dns\_resolver\_check.sh | [kautik4 (kautik)](https://github.com/kautik4) | Inaccurate network speed measurements |
| **4.Gayatri Ratnaparkhi** | Developed http\_availability\_checker.sh | [Gayatri-Ratnaparkhi (Gayatri)](https://github.com/Gayatri-Ratnaparkhi) | Availability Checker |
| **5.Sakshi Sonwane** | Developed signal\_strength.sh | [sonawane-sakshi](https://github.com/sonawane-sakshi) | Handled missing iwconfig command |

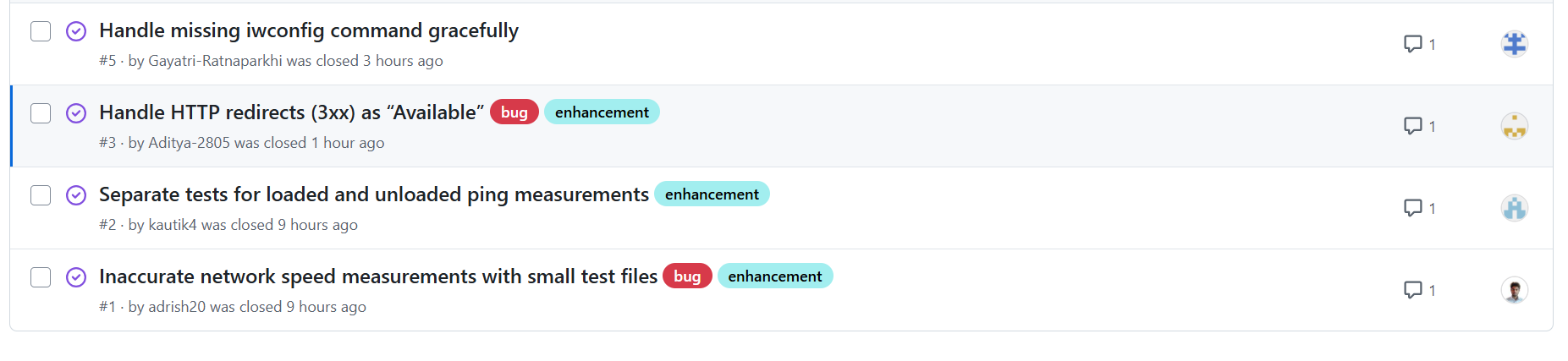
**Challenges Faced and Solved:**

* **Inconsistent Network Speed Measurement:** Small test file sizes, such as 1 MB, resulted in inaccurate speed measurements on high-speed connections. Using larger files provided more accuracy but led to increased execution time and bandwidth consumption.
* **Upload Speed Testing Limitations:** It was difficult to reliably test upload speed without external tools like speedtest-cli, because basic utilities such as curl do not easily measure upload performance.
* **Dependency Management:** Scripts depended on commands like dig or iwconfig that were not pre-installed on all systems, which caused them to fail on minimal Linux setups.
* **Permission and Access Issues:** Depending on the system's configuration, reading from /sys/class/net/ or running network commands sometimes required elevated permissions with sudo.
* **Dynamic Interface Detection:** Automatically identifying the correct network and Wi-Fi interfaces was challenging due to inconsistent naming conventions across different Linux distributions (e.g., wlan0, wlp3s0).
* **Data Formatting for Logging:** To avoid corrupt log entries, multi-line outputs from commands required careful cleanup before being logged in CSV format.
* **Error Handling:** Initially, the scripts would crash or produce empty logs when no internet connection was available or a DNS query failed. This required adding checks and fallback messages.
* **Performance Overhead:** Running all modules in sequence increased the total execution time, especially for scripts that needed to query external servers for DNS or HTTP checks.
* **Cross-Distribution Compatibility:** Ensuring the scripts worked on different systems was a challenge, as commands like iwconfig are deprecated on newer Linux versions, which favor tools like ip and iw.

1. **Timely Submission**

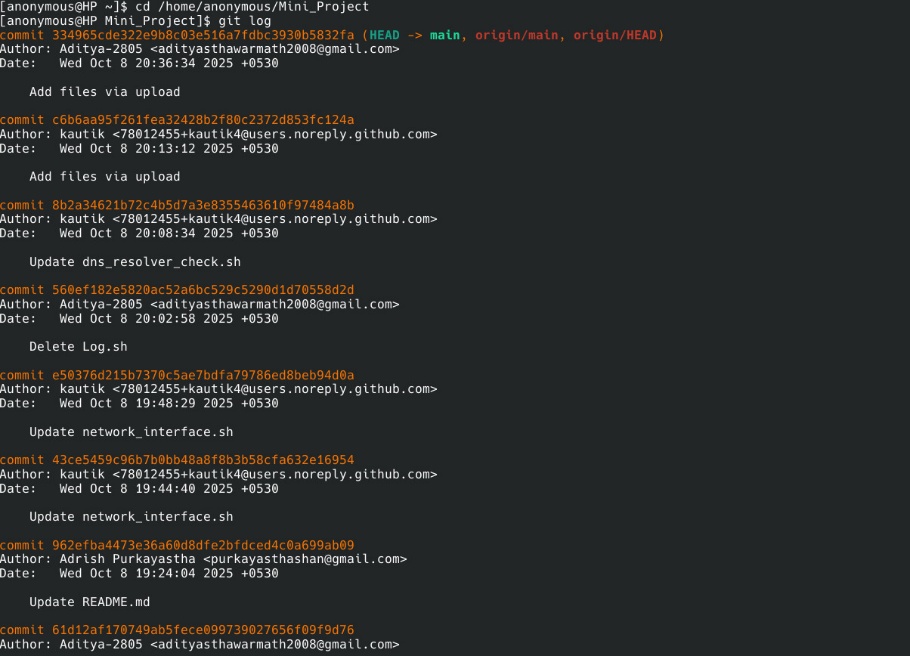
* **Submission Timeline:**

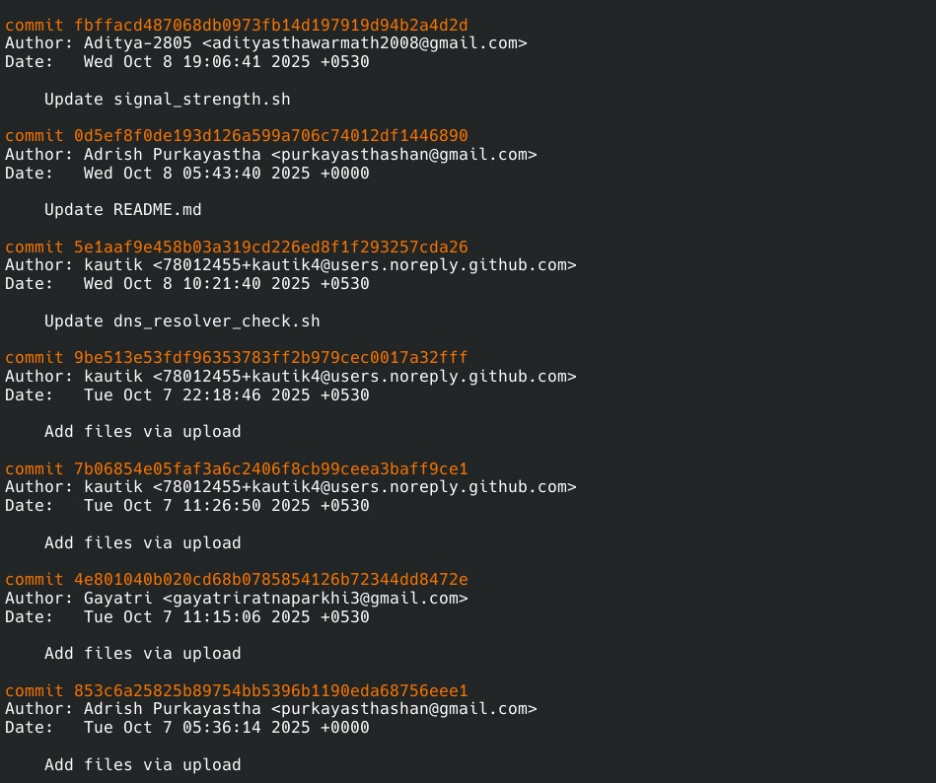
****

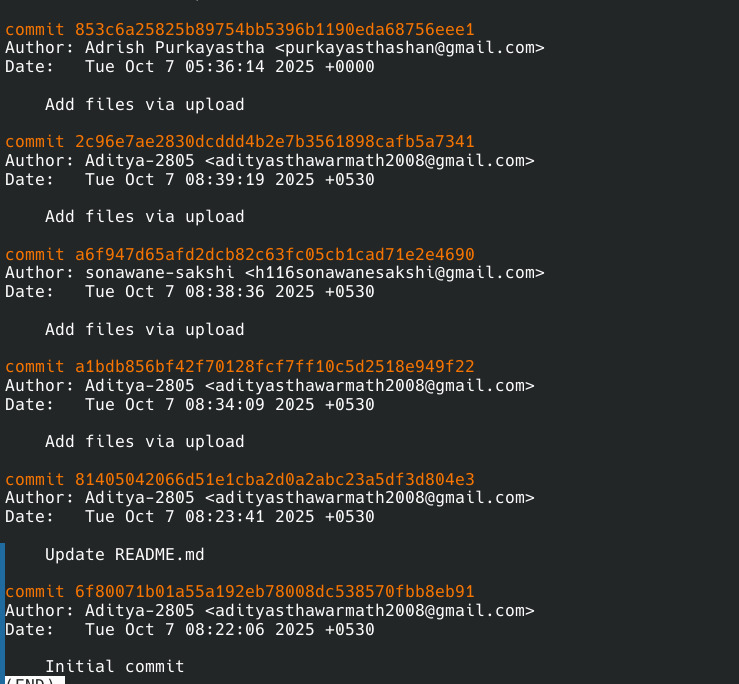
****

* **Evidence of Progress**

****

****

****

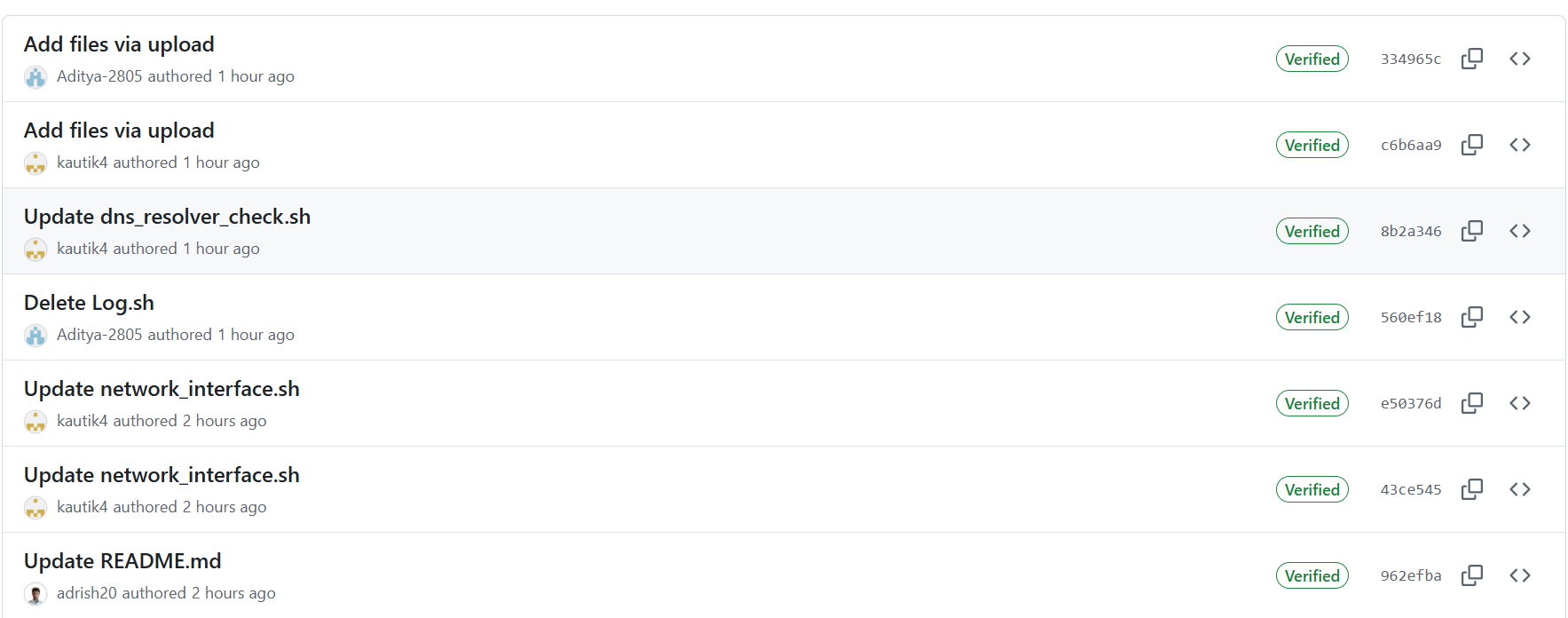
****

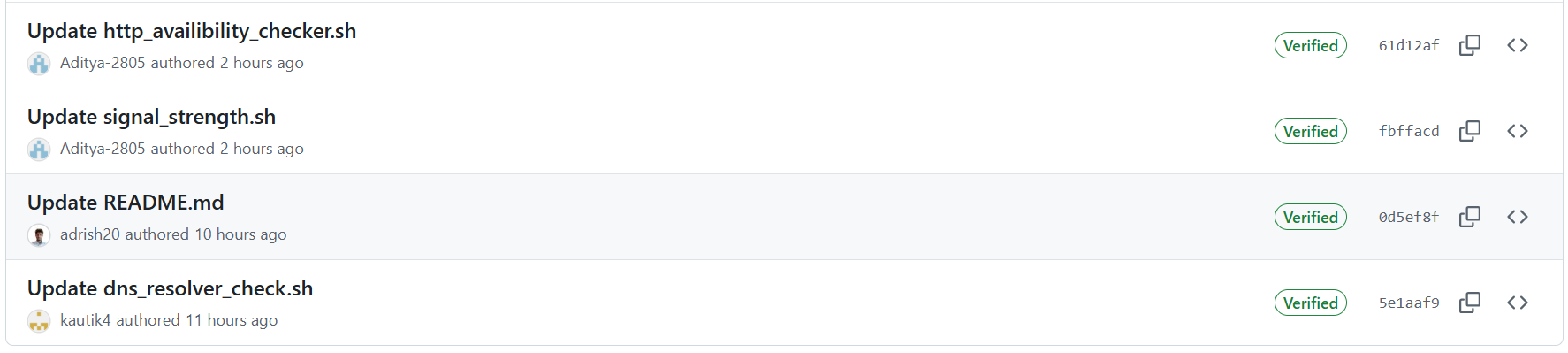
1. **Pushing the Project to GitHub**

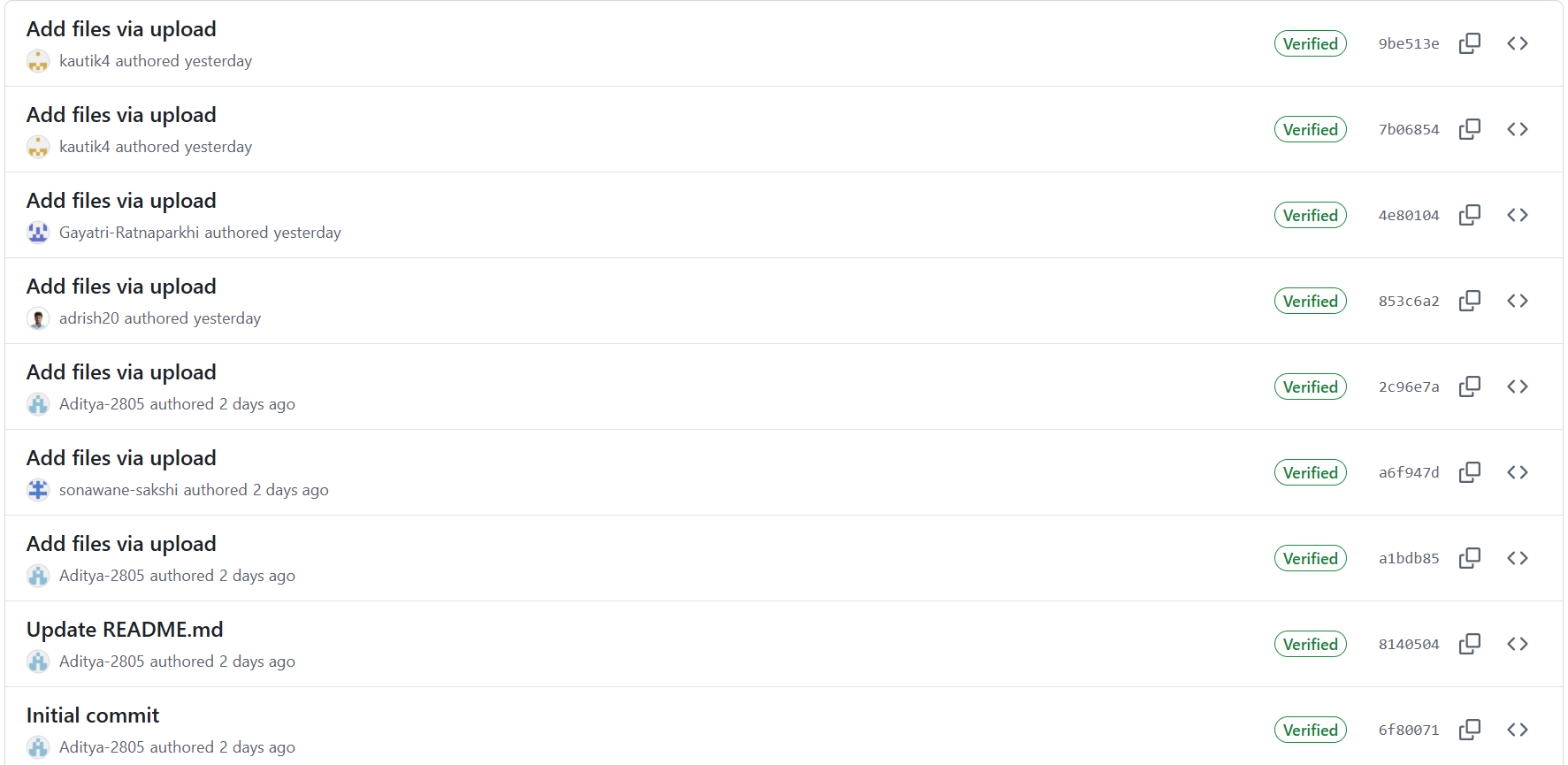
**GitHub Repository Link:**

<https://github.com/Aditya-2805/Mini_Project>

**Commit History:**



****

****

**README File:**

**Network Analytics Suite**

A modular Bash-based toolkit for comprehensive network performance monitoring on Linux systems.

Description:

This suite provides a collection of simple, powerful Bash scripts designed to monitor and log various aspects of your network. It is useful for system administrators, developers, or power users to keep a close eye on network health and performance, logging everything to clean, easy-to-analyze CSV files.

Core Features:

- Modular Design: Each function is a self-contained script, allowing selective usage.

- CSV Logging: Automatically logs results in CSV format for easy data analysis.

- Linux Native: Built with standard Linux utilities like ping, curl, ip, and iwconfig for maximum compatibility.

- Cron-Ready: The master script can be scheduled with cron for automated, periodic monitoring.

Modules Included:

1. Network Speed Test: Measures download speed, upload speed, and ping latency.

2. Network Interface Info: Logs IP/MAC addresses and link status for all network interfaces.

3. DNS Resolver Check: Tests the response time and availability of DNS servers.

4. Signal Strength Monitor: Monitors Wi-Fi signal strength and link quality over time.

5. HTTP Availability Checker: Checks the status code and response time of critical websites or endpoints.

6. Master Script: Runs all modules sequentially and aggregates the results.

Requirements:

- OS: Any Linux distribution

- Shell: bash

- Core Utilities: ping, curl, ip, iwconfig, nslookup

Installation:

1. Clone the repository:

git clone https://github.com/kautik4/Network-Analytics

cd Network-Analytics

2. Grant execute permissions to all scripts:

chmod +x \*.sh

Usage:

- Run each module individually by executing the respective script:

./network\_speed\_test.sh

./network\_interface\_info.sh

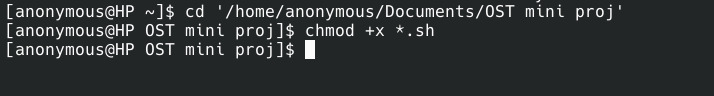
./dns\_resolver\_check.sh

./signal\_strength.sh

./http\_availibility\_checker.sh

- Or run the master script to execute all modules and aggregate results.

**Shell scripting and CLI commands used during development (Bash)**



* ping: Checks network connectivity and latency, and is used in the Network Speed Tester to measure response time to a known server like 8.8.8.8.
* curl: Performs data transfer tests and checks HTTP availability by downloading small files to estimate speed and to verify website status.
* wget: Serves as an alternative to curl for file downloads and is used for speed measurement when curl is unavailable.
* dig: Queries DNS servers to measure their response times, used specifically in the DNS Resolver Check script.
* ip: Displays network interface and IP information, used to extract the IPv4 address and operational state of each interface.
* iwconfig: Shows wireless interface details such as SSID and signal strength. It is used to log the connected SSID, signal level, and link quality.
* cat: Reads attributes from system files in /sys/class/net/, used to fetch MAC addresses, interface status, and speed information.
* awk / grep / cut: A suite of tools for text parsing and filtering. They are used throughout the project to extract specific values from command outputs.
* date: Gets the current date and time, used in all scripts for timestamping log entries.
* tr: Transforms or cleans text, used in the master script to replace newlines with spaces before logging to CSV files.
* echo: Displays data on the terminal and logs it to files. It is used to print results and append formatted lines to the logs.
* ls: Lists network interfaces, used to iterate through all available network adapters in the system.