## **Project Title**

Create a personalized CLI tool with bash

**Guide details: Student details:**

Udutha Shashidhar D. praveen

Cyber Security Trainer Course:Cyber Security

Teks Academy Teks Academy

Contact:9603343287 Contact: 6305096026

**Abstract:**

The Termi tool is a lightweight, user-friendly GUI-based system administration tool designed for Linux platforms. Built using Bash scripting and Zenati, this tool enables users to perform essential system monitoring tasks—such as checking disk usage, viewing system uptime, monitoring firewall status, listing running services, and scanning networks—without needing to use terminal commands

It provides a clean and interactive graphical menu, allowing users to select tasks easily with just a few clicks. Each time a task is performed, the result is shown in a pop-up window and also saved into a log file for future reference. This makes it easier to track the actions taken and when they were done.

Termi Tools is especially helpful for beginners, students, or anyone who finds the Linux terminal difficult. Instead of typing and memorizing commands, they can use this tool to manage their system more visually and comfortably.

The tool runs smoothly on Linux systems like Kali and Ubuntu. It also makes a great mini project for learning shell scripting, system monitoring, and how to connect backend tasks with a simple graphical interface. Overall, Termi Tools saves time, avoids command-line errors, and makes Linux system tasks much easier to handle.

**Problem Statement:**

Many Linux system administration tasks—like checking disk usage, monitoring uptime, viewing logged-in users, managing services, and scanning networks—require users to type complex terminal commands. This creates a barrier for beginners, students, and non-technical users who are not comfortable using command-line tools. They may feel confused, hesitate to explore, or even avoid important tasks due to fear of making mistakes.

As a result, learning and managing a Linux system becomes more difficult for them. Without proper tools, they miss out on understanding how the system works and how to maintain it effectively.

To solve this problem, there is a need for a simple and user-friendly tool that allows users to perform these common tasks using a graphical interface. Instead of typing commands, users can select options from a menu and view results in pop-up windows. This makes learning easier, avoids errors, and saves time.

Such a tool also helps users become more confident in using Linux, especially those who are new to the platform. It bridges the gap between advanced system functions and everyday users, turning complicated tasks into a smooth experience.

**Project Objective(s):**

The main objective of this project is to create a simple, user-friendly tool for Linux users that does not require them to use terminal commands. The tool, called Term Tools, is built using Bash scripting and Zenity to provide a graphical menu for performing basic system administration tasks.

With this tool, users can easily check disk usage, view system uptime, monitor firewall status, list running services, and perform quick network scans — all with just a few clicks. This is especially helpful for beginners or students who may not be comfortable using the command line.

The tool also automatically saves the output of each task in a log file. This allows users to review what tasks they performed and see results later if needed. In short, the goal is to make Linux system monitoring simple, visual, and accessible — even for those who are new to it.

**Project Scope:**

The scope of this project is to design and develop a lightweight, GUI-based Linux administration tool called Termi Tools. The tool focuses on simplifying common system monitoring and maintenance tasks for beginners and non-technical users. It will be built using Bash scripting and Zenity for the graphical interface. Key features within scope include:

\* Displaying logged-in users

\* Showing disk usage and system uptime \*

\*Checking firewall status using UFW

\* Listing running services

\* Performing a quick network scan using Nmap

\* Logging all outputs to a file for future reference

\* Providing a simple menu-based interface with Zenity. The tool will be designed for Ubuntu/Kali Linux systems and can be easily extended in the future with more features such as scheduled scans, system alerts, or integration with remote monitoring tools.

**Tools & Technologies:**

1. Bash (Shell Scripting)

It is used to write the core logic of the tool. Handles command execution, loops, and conditionals.

2. Zenity

provides graphical dialog boxes (menus, message boxes, info windows). Helps create a simple GUI on top of Bash scripts.

3. UFW (Uncomplicated Firewall)

Checks and displays firewall status. Simplifies firewall management via the terminal.

4.  Nmap

It is used for performing quick network scans. Detects open ports and connected devices.

5. Systemctl

Manages system services to verify if they are active or inactive. Ensures essential services are operational.

6. Logger

saves the output of each task to a log file automatically. Useful for audit and future reference.

7. GNU/Linux OS (Ubuntu/Kali)

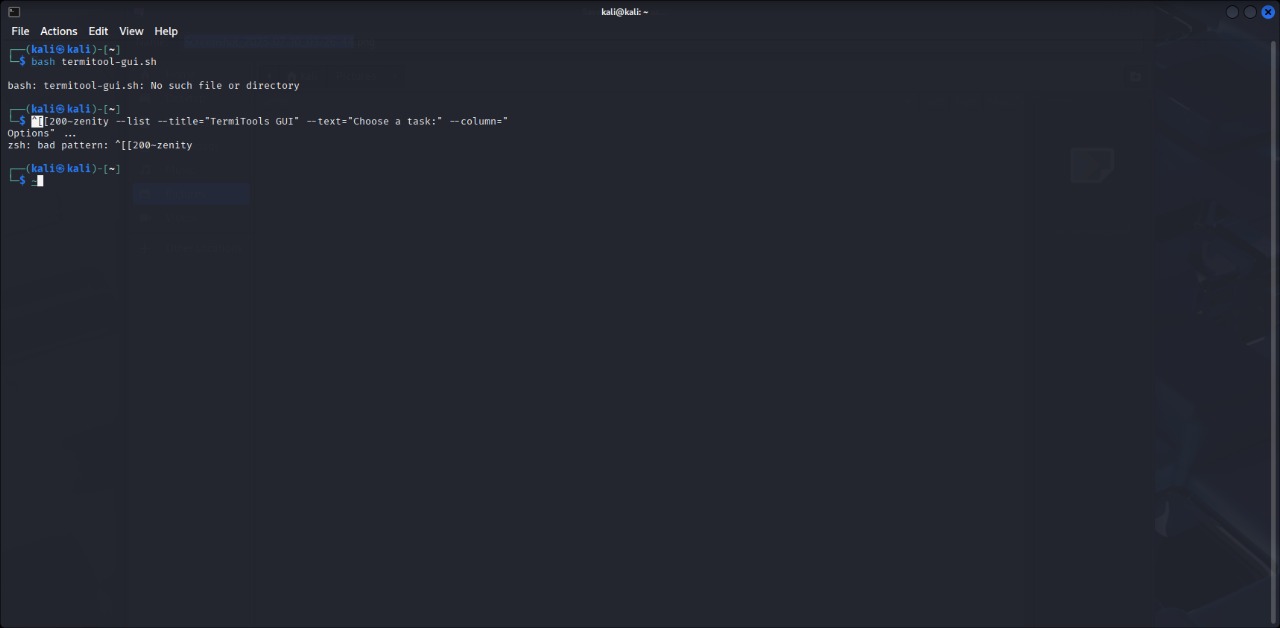
The platform where the tool runs. Compatible with most Debian-based systems.

**Timeline (Tentative):**

|  |  |
| --- | --- |
| **week** | **Day** |
| Day1 | Research & Dataset Collection |
| Day2 | Feature Engineering & Model section |
| Day3 | Training, Testing & Evaluation |
| Day4 | Interface Development & Demo |
| Day5 | Final Documentation & Presentation |

**Workflow of the project:**

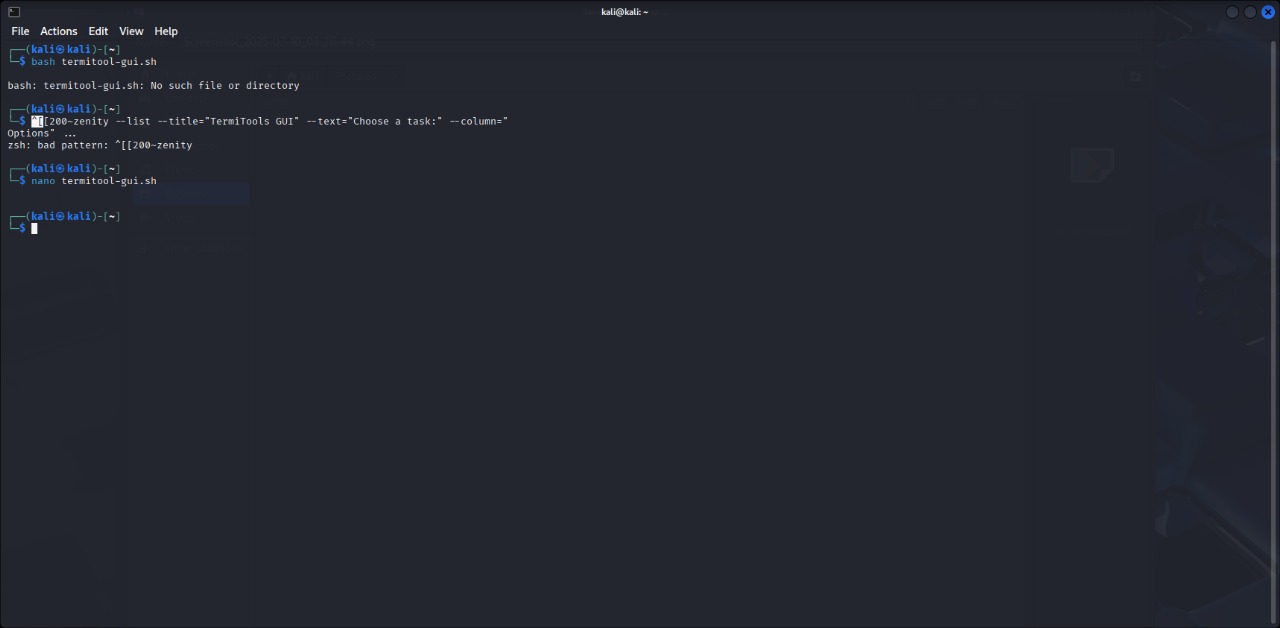
**1. Start the Script**



Command: bash termitool-gui.sh

Script starts and checks if Zenity is installed

**Step 2: Create the Script File**

****

Command: nano termitool-gui.sh

to open Nano text editor and create your script:

**Step 3: Paste the following updated script inside nano**

**#! / bin/bash**

**# Check if Zenity is installed**

**if! command -v zenity &> /dev/null; then**

**echo "Zenity not installed. Run: sudo apt install zenity"**

**exit 1**

**fi**

**# Log file location**

**LOGFILE="$HOME/termitoolplus\_gui.log"**

**# Infinite GUI loop**

**while true; do**

**CHOICE=$(zenity --list --title="Termi Tools GUI" \**

**--text="Choose a task to perform:" \**

**--column="Options" \**

**"Show Logged-in Users" \**

**"Disk Usage" \**

**"System Uptime" \**

**"Firewall Status" \**

**"Running Services" \**

**"Quick Nmap Scan" \**

**"Exit")**

**case "$CHOICE" in**

**"Show Logged-in Users")**

**who | tee -a "$LOGFILE" | zenity --text-info --title="Logged-in Users" --width=500 --height=300**

**;;**

**"Disk Usage")**

**df -h | tee -a "$LOGFILE" | zenity --text-info --title="Disk Usage" --width=500 --height=300**

**;;**

**"System Uptime")**

**uptime | tee -a "$LOGFILE" | zenity --text-info --title="System Uptime" --width=500 --height=300**

**;;**

**"Firewall Status")**

**sudo ufw status | tee -a "$LOGFILE" | zenity --text-info --title="Firewall Status" --width=500 --height=300**

**;;**

**"Running Services")**

**systemctl list-units --type=service --state=running | tee -a "$LOGFILE" | zenity --text-info --title="Running Services" --width=600 --height=400**

**;;**

**"Quick Nmap Scan")**

**N map -F localhost | tee -a "$LOGFILE" | zenity --text-info --title="Nmap Scan" --width=600 --height=400**

**;;**

**"Exit")**

**zenity-- info-- text="Thank you for using Termi Tools!" --width=300**

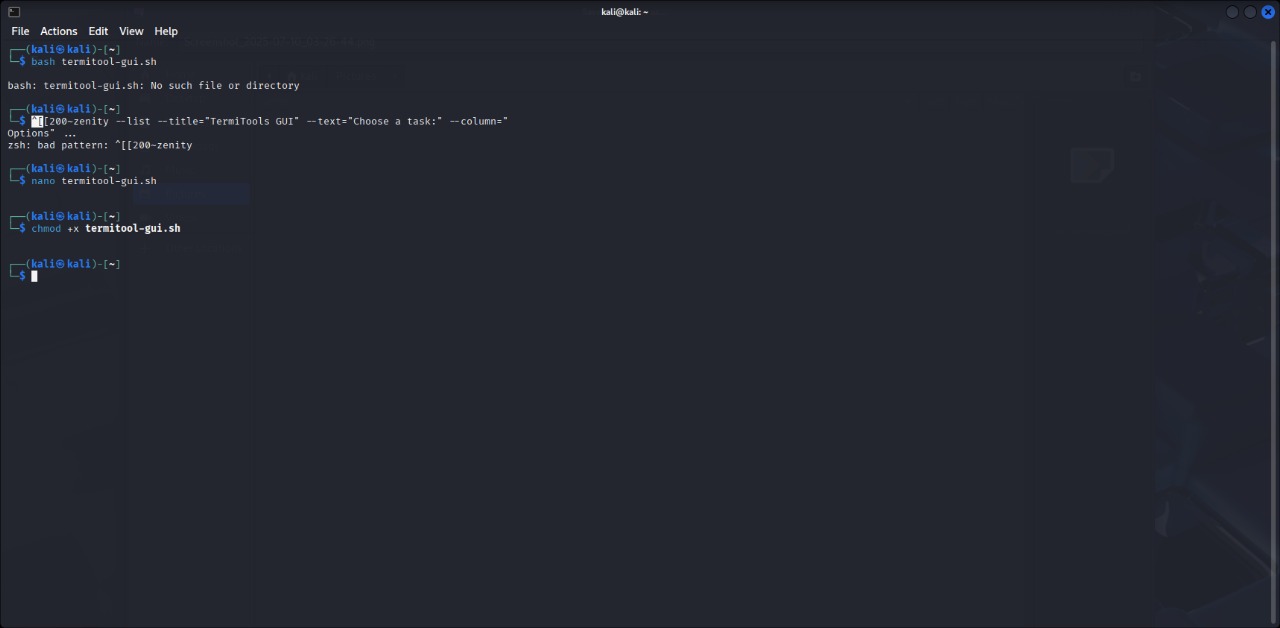
**exit**

**;;**

**Es ac**

**Done**

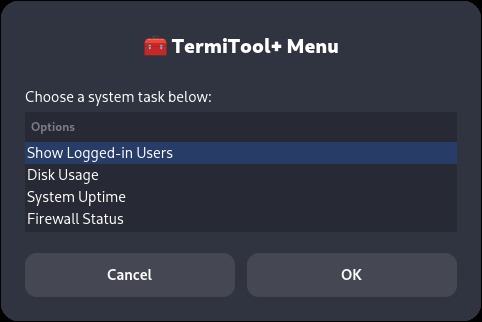
**Step 4:** **Give Execute Permission**



Command: chmod +x termitool-gui.sh

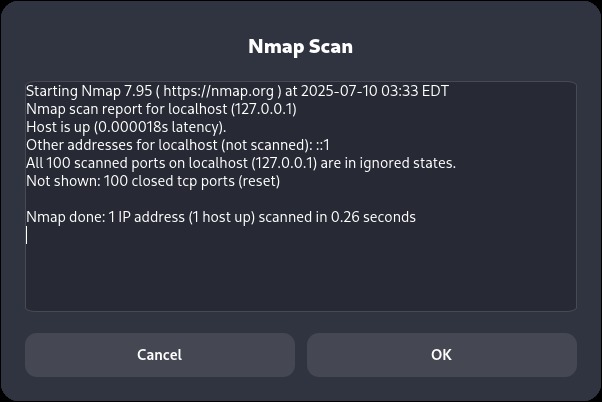
Run this to make the script executable

**Step 5: Run the Script**

****

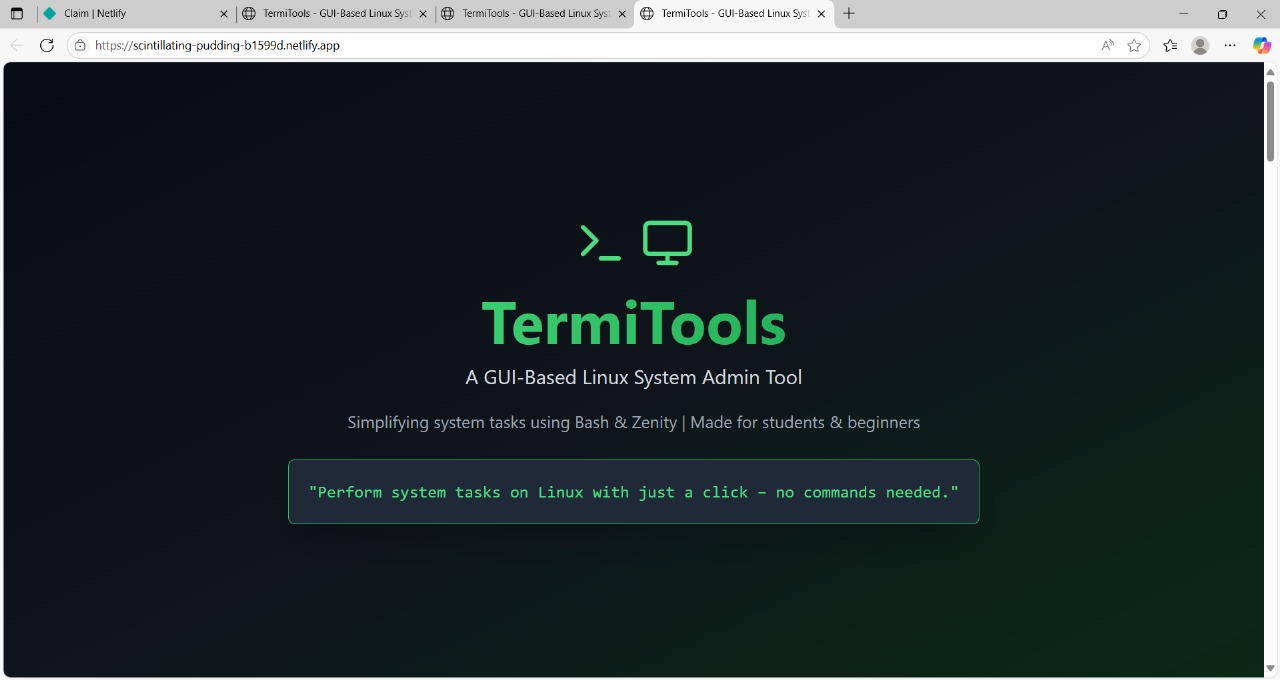
A dropdown menu will appear with all options**.**

**Final Step: Getting the required Output**

****

Gui web application

https://scintillating-pudding-b1599d.netlify.app/



Deliverables:

* Bash script with Zenity GUI (Termi Tools)
* Working menu-based Linux tool
* Log file generation feature
* Project Report (DOCX)
* PowerPoint Presentation (PPTX)