Capstone Project 5

Bash Scripting Suite for System Maintenance

Automating backup, system updates, and log monitoring using Linux Bash scripts.

Project Submitted By-

Name- Ashutosh Satpathy

Regd No. – 2241019351

Batch - 13

Branch – Computer Science and Information Technology

System - Ubuntu (via WSL)

Language – Bash

Date – 9 Nov 2025

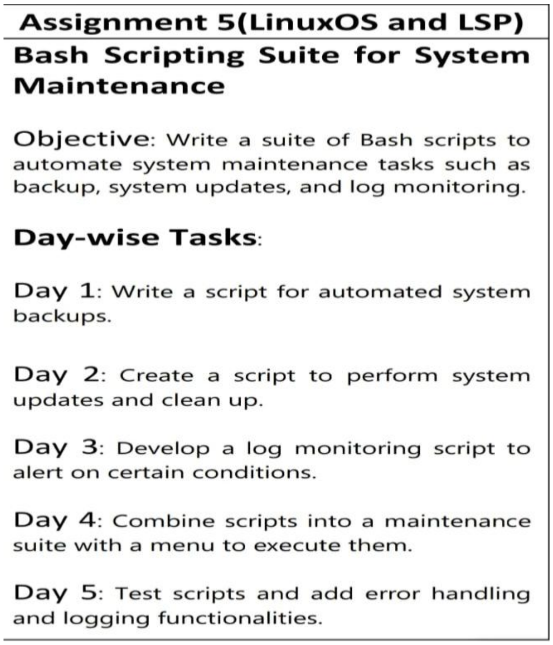
Contents

Objective

The primary objective of this project is to design and develop an intelligent and automated System Maintenance Suite using Bash scripting that serves as a smart digital assistant for Linux-based systems. The suite is intended to simplify and automate crucial system administration tasks such as performing regular data backups, executing software updates, cleaning unnecessary packages, and continuously monitoring system logs for warnings or errors. By integrating automation into these core maintenance processes, the project aims to minimize human intervention, reduce system downtime, and enhance overall operational efficiency. The suite incorporates advanced features such as timestamped logs, centralized reporting, error detection and handling mechanisms, and notification alerts, ensuring transparency and reliability during every maintenance cycle. Additionally, it offers a menu-driven, interactive interface designed to make the system user-friendly and accessible even to non-technical users, allowing them to manage complex operations through simple menu selections. Beyond automation, the project also emphasizes system health monitoring and security, ensuring that the Linux environment remains optimized, secure, and up to date at all times. Ultimately, the System Maintenance Suite represents a practical step toward intelligent system management — combining the power of shell scripting with the concept of digital assistance to create a seamless, efficient, and self-sustaining Linux ecosystem.

Tools and Environment

• Operating System: Ubuntu (WSL or Linux)
• Language: Bash shell scripting
• Text Editor: Nano or VS Code
• Scheduler: Cron
• Log files: Plain text (.log)



Assignment 5(LinuxOS and LSP)

Bash Scripting Suite for System Maintenance

Objective: Write a suite of Bash scripts to automate system maintenance tasks such as backup, system updates, and log monitoring.

Day-wise Tasks:

Day 1: Write a script for automated system backups.

Day 2: Create a script to perform system updates and clean up.

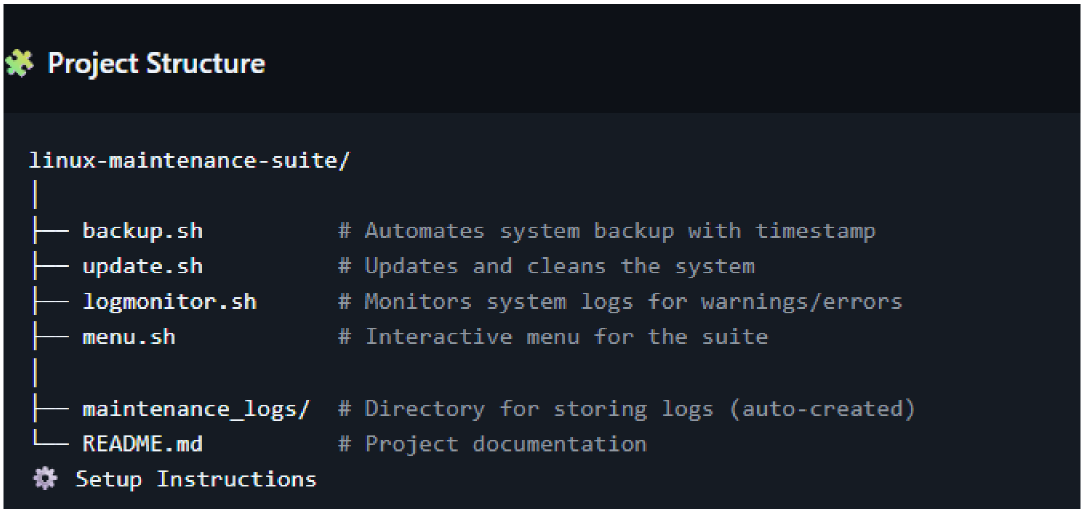
Day 3: Develop a log monitoring script to alert on certain conditions.

Day 4: Combine scripts into a maintenance suite with a menu to execute them.

Day 5: Test scripts and add error handling and logging functionalities.

Screenshots

Folder structure (maintenance_suite and maintenance_logs)

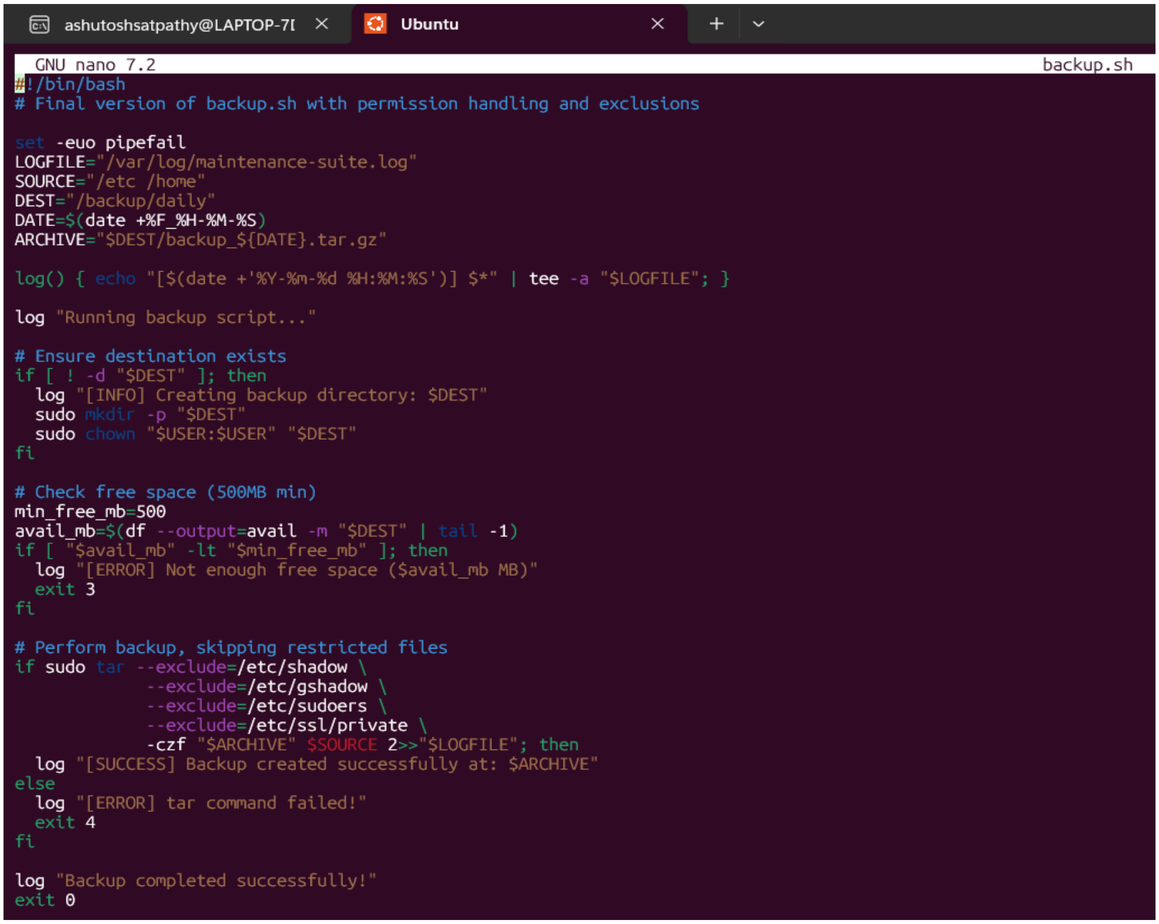


🧩 Project Structure

```
linux-maintenance-suite/
│
├── backup.sh          # Automates system backup with timestamp
├── update.sh          # Updates and cleans the system
├── logmonitor.sh      # Monitors system logs for warnings/errors
├── menu.sh            # Interactive menu for the suite
│
├── maintenance_logs/  # Directory for storing logs (auto-created)
└── README.md          # Project documentation
 ⚙️ Setup Instructions
```

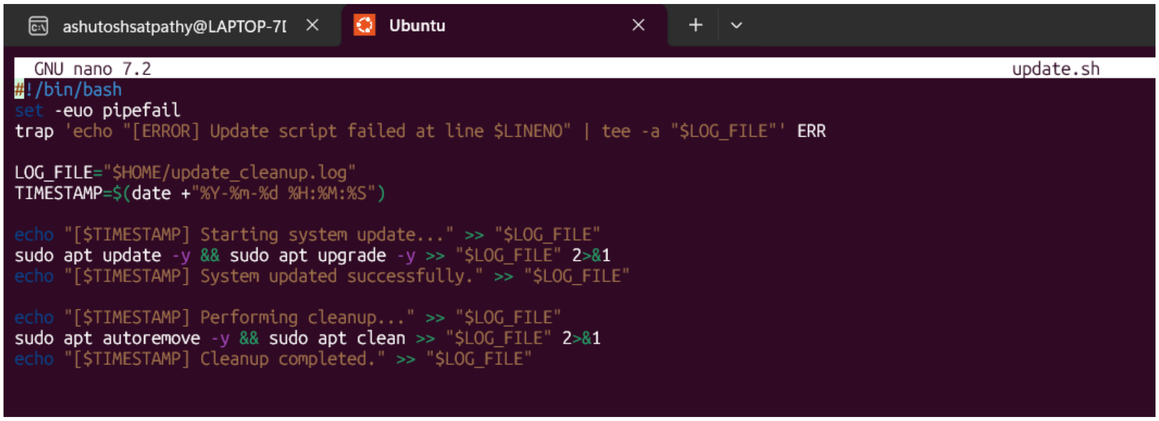
Source Code

backup.sh



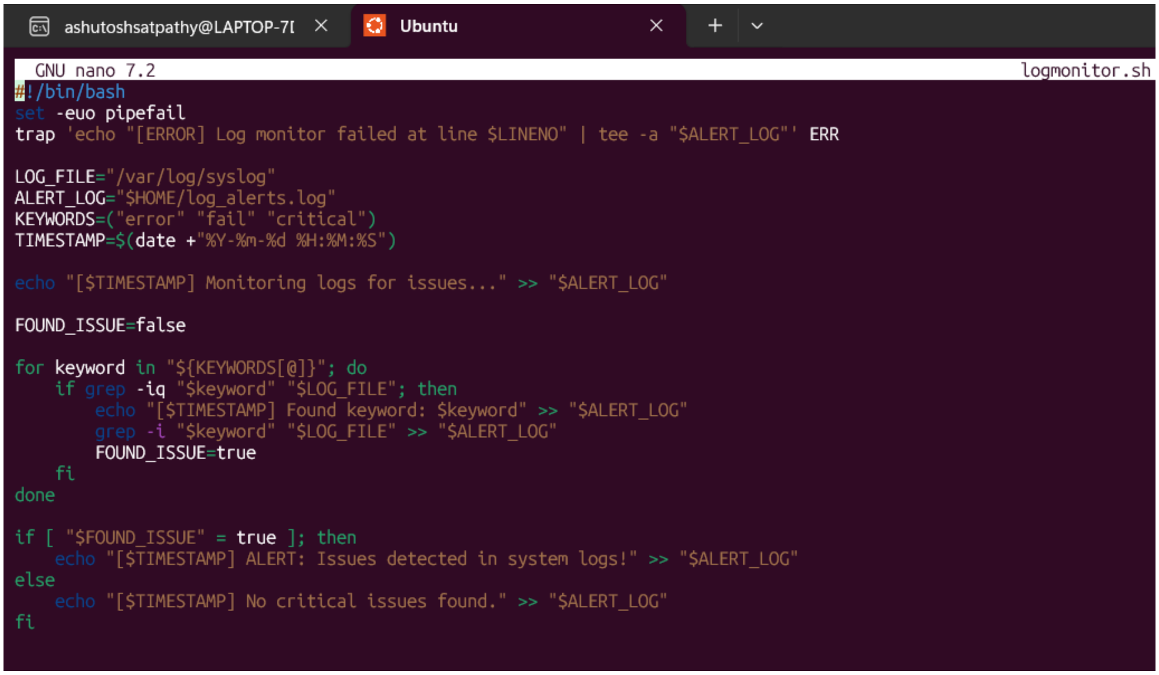
```
#!/bin/bash
# Final version of backup.sh with permission handling and exclusions

set -euo pipefail
LOGFILE="/var/log/maintenance-suite.log"
SOURCE="/etc /home"
DEST="/backup/daily"
DATE=$(date +%F_%H-%M-%S)
ARCHIVE="$DEST/backup_${DATE}.tar.gz"

log() { echo "[$(date +'%Y-%m-%d %H:%M:%S')] $*" | tee -a "$LOGFILE"; }

log "Running backup script..."

# Ensure destination exists
if [ ! -d "$DEST" ]; then
  log "[INFO] Creating backup directory: $DEST"
  sudo mkdir -p "$DEST"
  sudo chown "$USER:$USER" "$DEST"
fi

# Check free space (500MB min)
min_free_mb=500
avail_mb=$(df --output=avail -m "$DEST" | tail -1)
if [ "$avail_mb" -lt "$min_free_mb" ]; then
  log "[ERROR] Not enough free space ($avail_mb MB)"
  exit 3
fi

# Perform backup, skipping restricted files
if sudo tar --exclude=/etc/shadow \
             --exclude=/etc/gshadow \
             --exclude=/etc/sudoers \
             --exclude=/etc/ssl/private \
             -czf "$ARCHIVE" $SOURCE 2>>"$LOGFILE"; then
  log "[SUCCESS] Backup created successfully at: $ARCHIVE"
else
  log "[ERROR] tar command failed!"
  exit 4
fi

log "Backup completed successfully!"
exit 0
```

update.sh



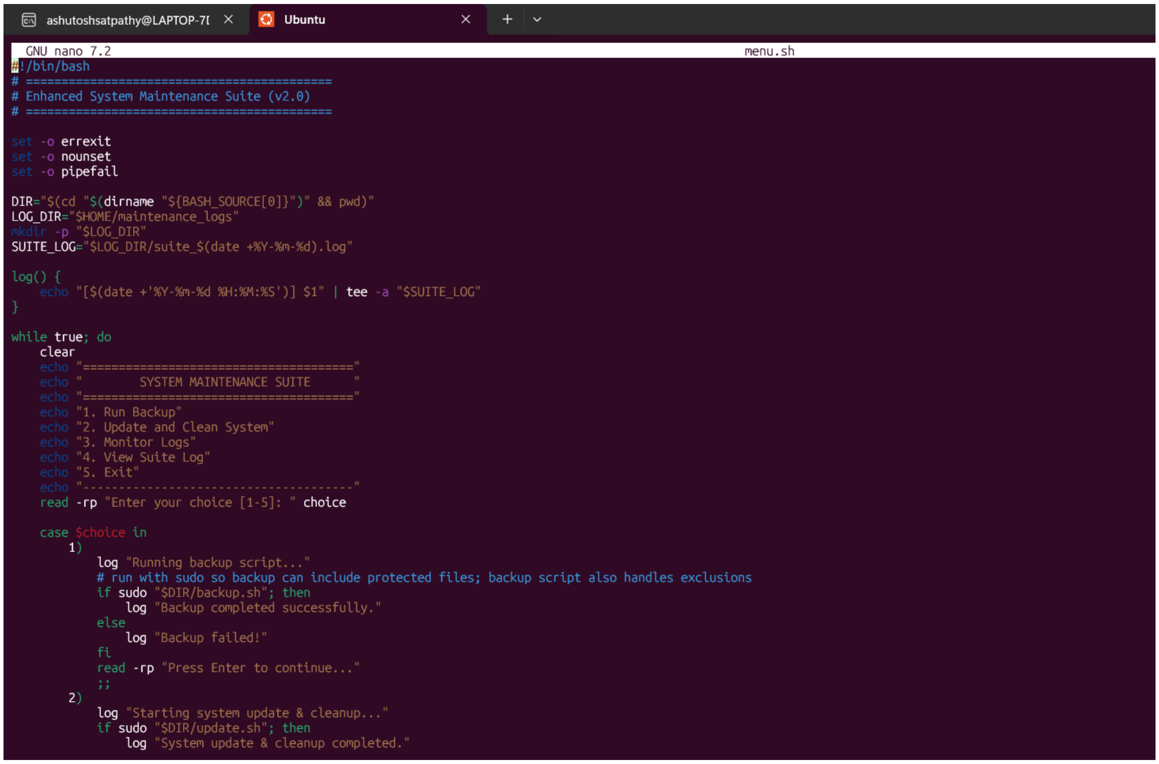
```
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Update script failed at line $LINENO" | tee -a "$LOG_FILE"' ERR

LOG_FILE="$HOME/update_cleanup.log"
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Starting system update..." >> "$LOG_FILE"
sudo apt update -y && sudo apt upgrade -y >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] System updated successfully." >> "$LOG_FILE"

echo "[$TIMESTAMP] Performing cleanup..." >> "$LOG_FILE"
sudo apt autoremove -y && sudo apt clean >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] Cleanup completed." >> "$LOG_FILE"
```

logmonitor.sh

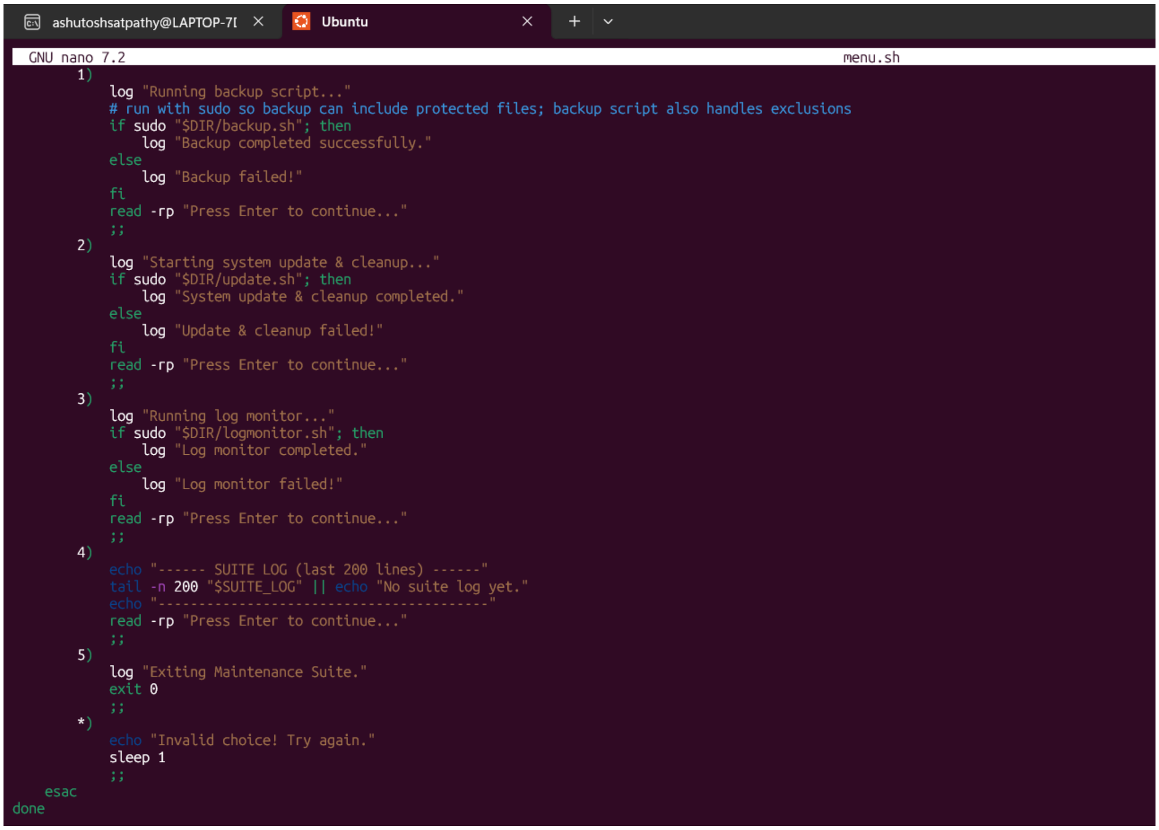


```bash
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Log monitor failed at line $LINENO" | tee -a "$ALERT_LOG"' ERR

LOG_FILE="/var/log/syslog"
ALERT_LOG="$HOME/log_alerts.log"
KEYWORDS=("error" "fail" "critical")
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Monitoring logs for issues..." >> "$ALERT_LOG"

FOUND_ISSUE=false

for keyword in "${KEYWORDS[@]}"; do
    if grep -iq "$keyword" "$LOG_FILE"; then
        echo "[$TIMESTAMP] Found keyword: $keyword" >> "$ALERT_LOG"
        grep -i "$keyword" "$LOG_FILE" >> "$ALERT_LOG"
        FOUND_ISSUE=true
    fi
done

if [ "$FOUND_ISSUE" = true ]; then
    echo "[$TIMESTAMP] ALERT: Issues detected in system logs!" >> "$ALERT_LOG"
else
    echo "[$TIMESTAMP] No critical issues found." >> "$ALERT_LOG"
fi
```

menu.sh

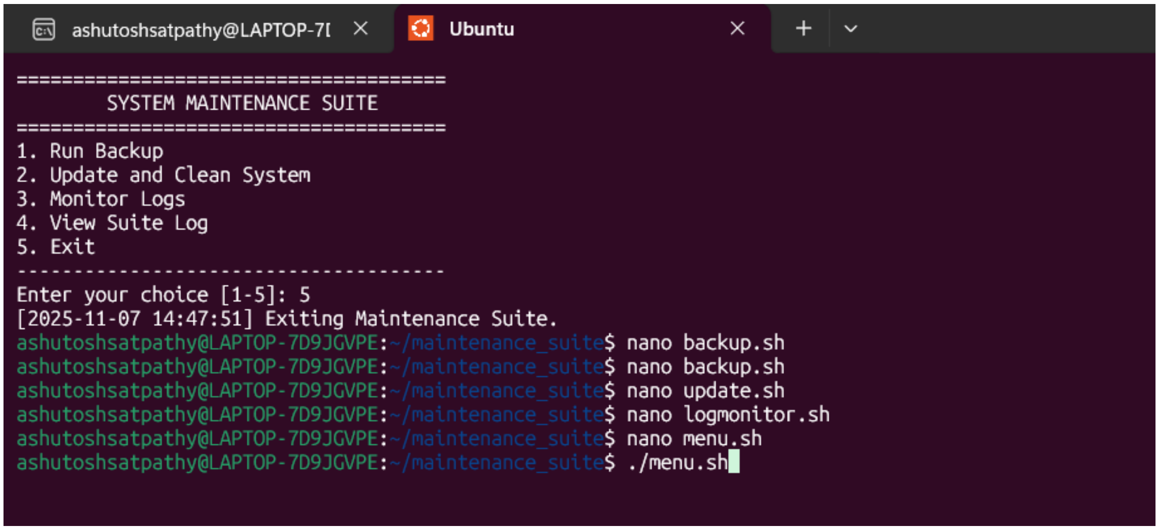


```bash
#!/bin/bash
# ===========================================
# Enhanced System Maintenance Suite (v2.0)
# ===========================================

set -o errexit
set -o nounset
set -o pipefail

DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
LOG_DIR="$HOME/maintenance_logs"
mkdir -p "$LOG_DIR"
SUITE_LOG="$LOG_DIR/suite_$(date +%Y-%m-%d).log"

log() {
    echo "[$(date +'%Y-%m-%d %H:%M:%S')] $1" | tee -a "$SUITE_LOG"
}

while true; do
    clear
    echo "======================================"
    echo "        SYSTEM MAINTENANCE SUITE      "
    echo "======================================"
    echo "1. Run Backup"
    echo "2. Update and Clean System"
    echo "3. Monitor Logs"
    echo "4. View Suite Log"
    echo "5. Exit"
    echo "--------------------------------------"
    read -rp "Enter your choice [1-5]: " choice

    case $choice in
        1)
            log "Running backup script..."
            # run with sudo so backup can include protected files; backup script also handles exclusions
            if sudo "$DIR/backup.sh"; then
                log "Backup completed successfully."
            else
                log "Backup failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        2)
            log "Starting system update & cleanup..."
            if sudo "$DIR/update.sh"; then
                log "System update & cleanup completed."
```



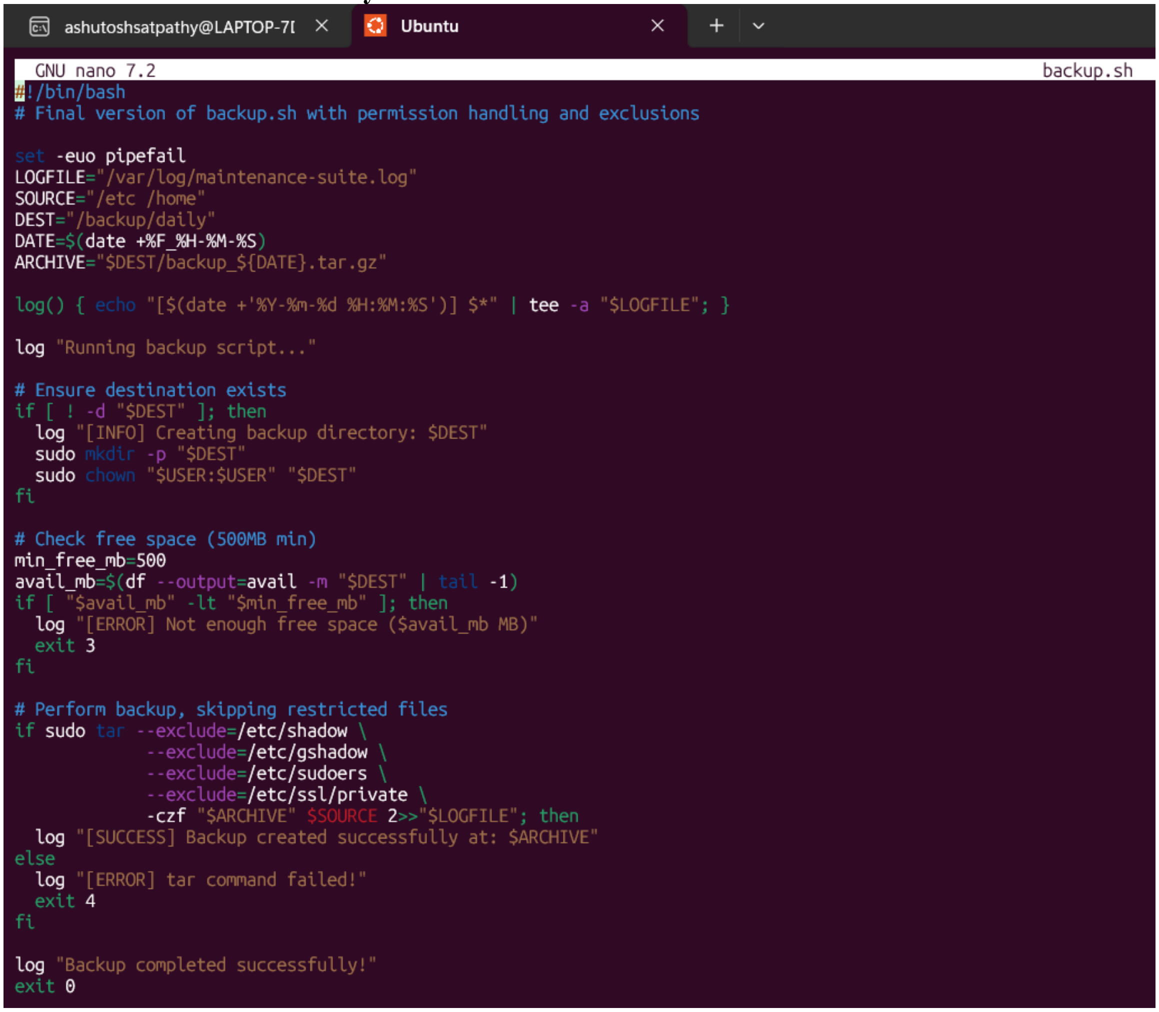
```
  GNU nano 7.2                                                                                             menu.sh
        1)
            log "Running backup script..."
            # run with sudo so backup can include protected files; backup script also handles exclusions
            if sudo "$DIR/backup.sh"; then
                log "Backup completed successfully."
            else
                log "Backup failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        2)
            log "Starting system update & cleanup..."
            if sudo "$DIR/update.sh"; then
                log "System update & cleanup completed."
            else
                log "Update & cleanup failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        3)
            log "Running log monitor..."
            if sudo "$DIR/logmonitor.sh"; then
                log "Log monitor completed."
            else
                log "Log monitor failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        4)
            echo "------ SUITE LOG (last 200 lines) ------"
            tail -n 200 "$SUITE_LOG" || echo "No suite log yet."
            echo "-----------------------------------------"
            read -rp "Press Enter to continue..."
            ;;
        5)
            log "Exiting Maintenance Suite."
            exit 0
            ;;
        *)
            echo "Invalid choice! Try again."
            sleep 1
            ;;
    esac
done
```

Editing scripts in Nano

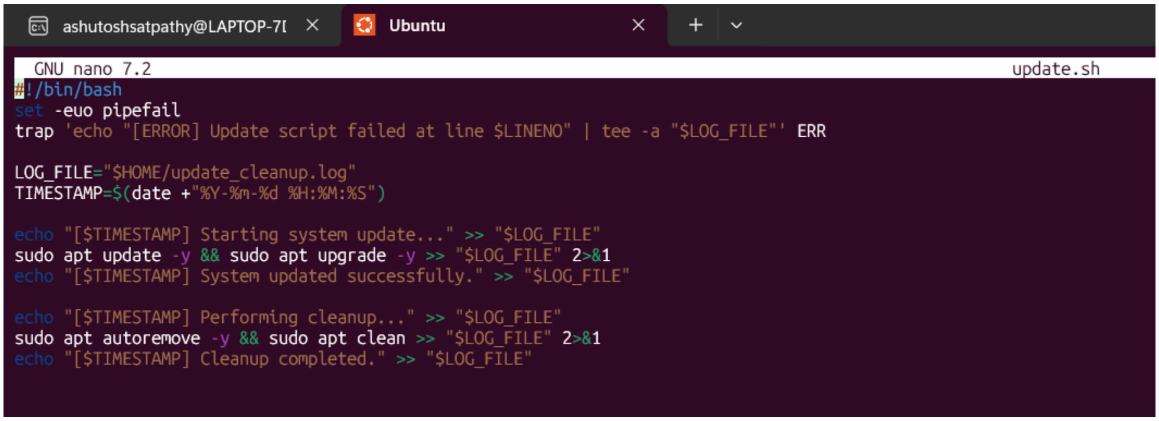


```
=======================================
        SYSTEM MAINTENANCE SUITE
=======================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
---------------------------------------
Enter your choice [1-5]: 5
[2025-11-07 14:47:51] Exiting Maintenance Suite.
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano backup.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano backup.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano update.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano logmonitor.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano menu.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ ./menu.sh
```

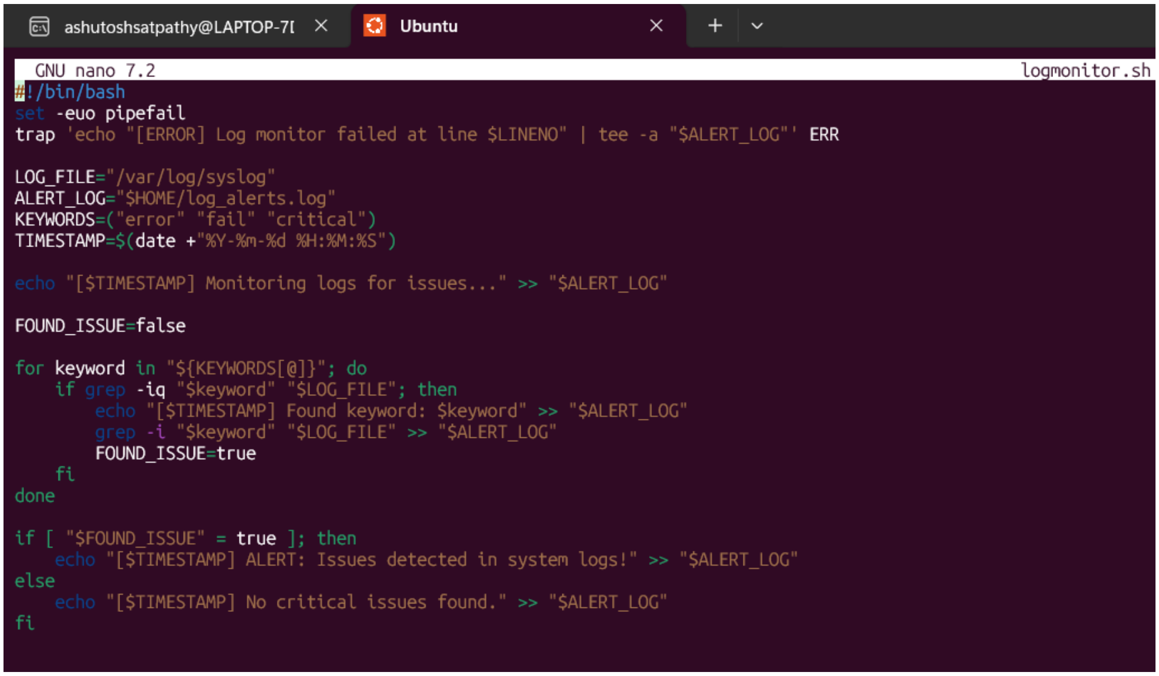
List of all .sh files in directory



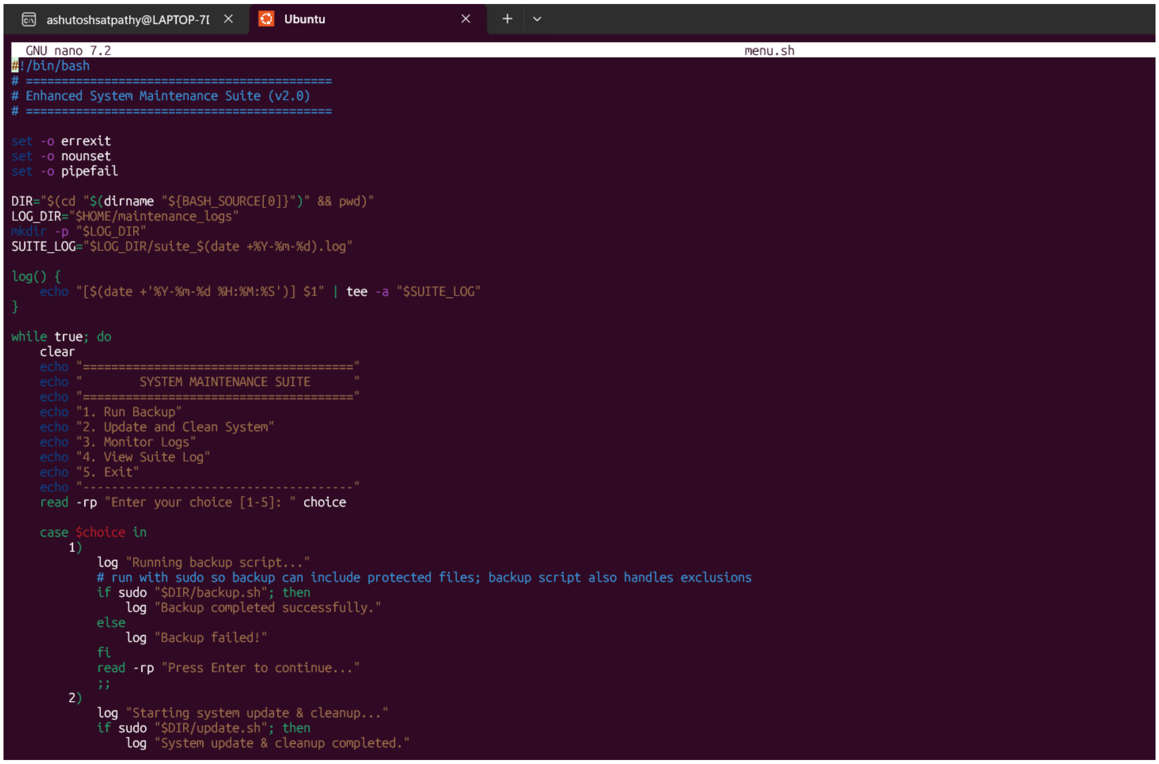
```
  GNU nano 7.2                                                                                 backup.sh
#!/bin/bash
# Final version of backup.sh with permission handling and exclusions

set -euo pipefail
LOGFILE="/var/log/maintenance-suite.log"
SOURCE="/etc /home"
DEST="/backup/daily"
DATE=$(date +%F_%H-%M-%S)
ARCHIVE="$DEST/backup_${DATE}.tar.gz"

log() { echo "[$(date +'%Y-%m-%d %H:%M:%S')] $*" | tee -a "$LOGFILE"; }

log "Running backup script..."

# Ensure destination exists
if [ ! -d "$DEST" ]; then
  log "[INFO] Creating backup directory: $DEST"
  sudo mkdir -p "$DEST"
  sudo chown "$USER:$USER" "$DEST"
fi

# Check free space (500MB min)
min_free_mb=500
avail_mb=$(df --output=avail -m "$DEST" | tail -1)
if [ "$avail_mb" -lt "$min_free_mb" ]; then
  log "[ERROR] Not enough free space ($avail_mb MB)"
  exit 3
fi

# Perform backup, skipping restricted files
if sudo tar --exclude=/etc/shadow \
             --exclude=/etc/gshadow \
             --exclude=/etc/sudoers \
             --exclude=/etc/ssl/private \
             -czf "$ARCHIVE" $SOURCE 2>>"$LOGFILE"; then
  log "[SUCCESS] Backup created successfully at: $ARCHIVE"
else
  log "[ERROR] tar command failed!"
  exit 4
fi

log "Backup completed successfully!"
exit 0
```



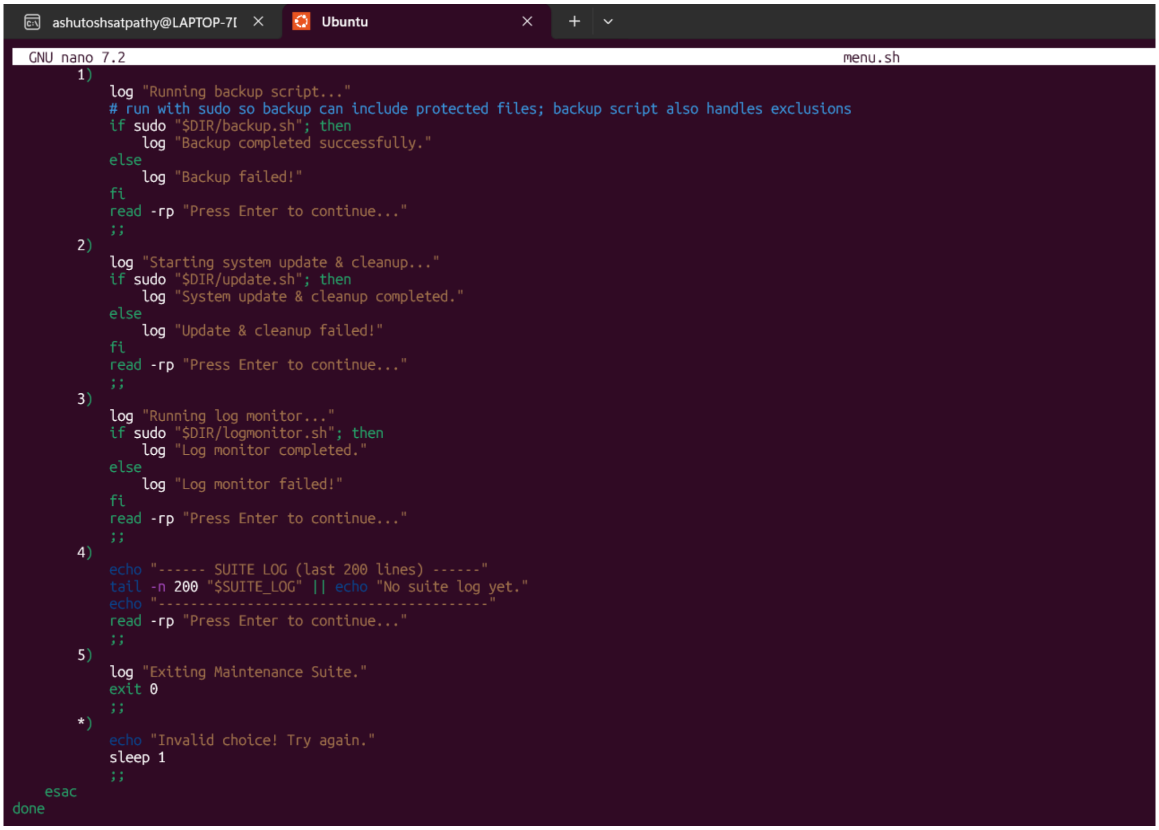
```
  GNU nano 7.2                                                                              update.sh
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Update script failed at line $LINENO" | tee -a "$LOG_FILE"' ERR

LOG_FILE="$HOME/update_cleanup.log"
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Starting system update..." >> "$LOG_FILE"
sudo apt update -y && sudo apt upgrade -y >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] System updated successfully." >> "$LOG_FILE"

echo "[$TIMESTAMP] Performing cleanup..." >> "$LOG_FILE"
sudo apt autoremove -y && sudo apt clean >> "$LOG_FILE" 2>&1
echo "[$TIMESTAMP] Cleanup completed." >> "$LOG_FILE"
```



```bash
#!/bin/bash
set -euo pipefail
trap 'echo "[ERROR] Log monitor failed at line $LINENO" | tee -a "$ALERT_LOG"' ERR

LOG_FILE="/var/log/syslog"
ALERT_LOG="$HOME/log_alerts.log"
KEYWORDS=("error" "fail" "critical")
TIMESTAMP=$(date +"%Y-%m-%d %H:%M:%S")

echo "[$TIMESTAMP] Monitoring logs for issues..." >> "$ALERT_LOG"

FOUND_ISSUE=false

for keyword in "${KEYWORDS[@]}"; do
    if grep -iq "$keyword" "$LOG_FILE"; then
        echo "[$TIMESTAMP] Found keyword: $keyword" >> "$ALERT_LOG"
        grep -i "$keyword" "$LOG_FILE" >> "$ALERT_LOG"
        FOUND_ISSUE=true
    fi
done

if [ "$FOUND_ISSUE" = true ]; then
    echo "[$TIMESTAMP] ALERT: Issues detected in system logs!" >> "$ALERT_LOG"
else
    echo "[$TIMESTAMP] No critical issues found." >> "$ALERT_LOG"
fi
```



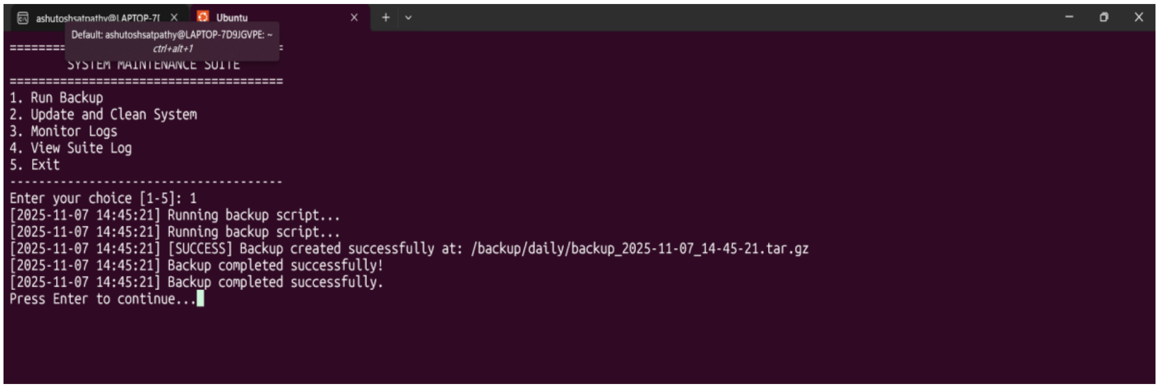
```bash
#!/bin/bash
# ===========================================
# Enhanced System Maintenance Suite (v2.0)
# ===========================================

set -o errexit
set -o nounset
set -o pipefail

DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
LOG_DIR="$HOME/maintenance_logs"
mkdir -p "$LOG_DIR"
SUITE_LOG="$LOG_DIR/suite_$(date +%Y-%m-%d).log"

log() {
    echo "[$(date +'%Y-%m-%d %H:%M:%S')] $1" | tee -a "$SUITE_LOG"
}

while true; do
    clear
    echo "======================================"
    echo "        SYSTEM MAINTENANCE SUITE      "
    echo "======================================"
    echo "1. Run Backup"
    echo "2. Update and Clean System"
    echo "3. Monitor Logs"
    echo "4. View Suite Log"
    echo "5. Exit"
    echo "--------------------------------------"
    read -rp "Enter your choice [1-5]: " choice

    case $choice in
        1)
            log "Running backup script..."
            # run with sudo so backup can include protected files; backup script also handles exclusions
            if sudo "$DIR/backup.sh"; then
                log "Backup completed successfully."
            else
                log "Backup failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        2)
            log "Starting system update & cleanup..."
            if sudo "$DIR/update.sh"; then
                log "System update & cleanup completed."
```



```
        1)
            log "Running backup script..."
            # run with sudo so backup can include protected files; backup script also handles exclusions
            if sudo "$DIR/backup.sh"; then
                log "Backup completed successfully."
            else
                log "Backup failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        2)
            log "Starting system update & cleanup..."
            if sudo "$DIR/update.sh"; then
                log "System update & cleanup completed."
            else
                log "Update & cleanup failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        3)
            log "Running log monitor..."
            if sudo "$DIR/logmonitor.sh"; then
                log "Log monitor completed."
            else
                log "Log monitor failed!"
            fi
            read -rp "Press Enter to continue..."
            ;;
        4)
            echo "------ SUITE LOG (last 200 lines) ------"
            tail -n 200 "$SUITE_LOG" || echo "No suite log yet."
            echo "----------------------------------------"
            read -rp "Press Enter to continue..."
            ;;
        5)
            log "Exiting Maintenance Suite."
            exit 0
            ;;
        *)
            echo "Invalid choice! Try again."
            sleep 1
            ;;
    esac
done
```

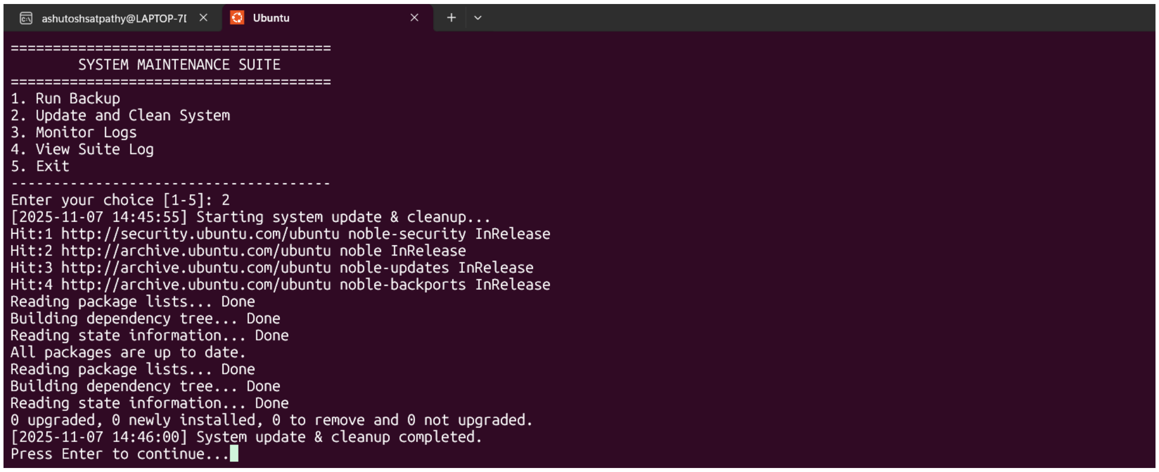
Ouput Screenshot

Running backup.sh successfully


ashutoshsatpathy@LAPTOP-7I
Ubuntu
Default: ashutoshsatpathy@LAPTOP-7D9JGVPE: ~
ctrl+alt+1

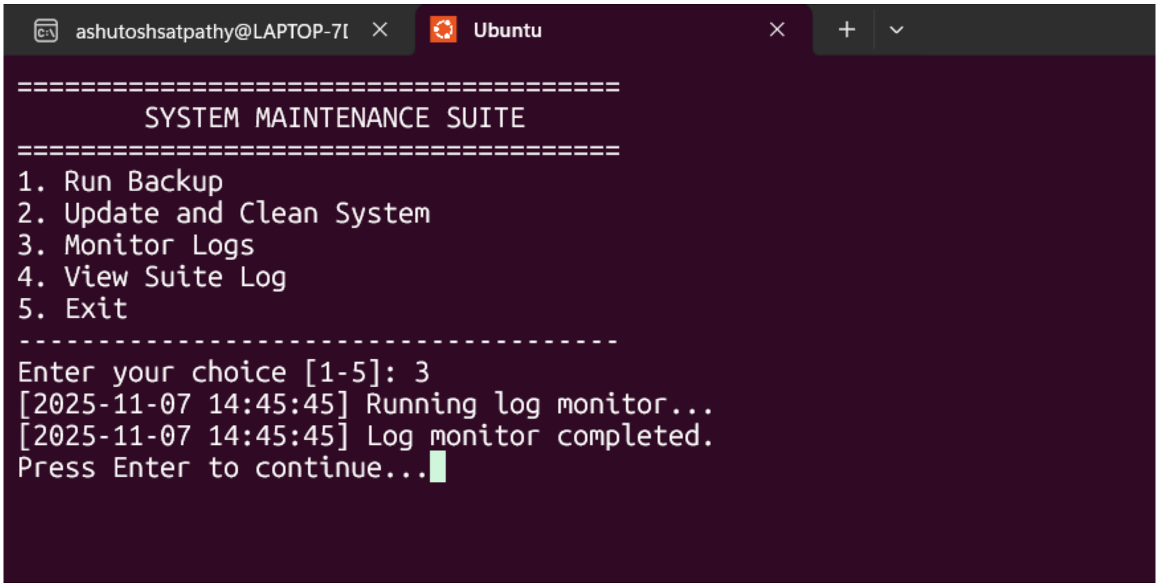
```
========                              =
        SYSTEM MAINTENANCE SUITE
=======================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
---------------------------------------
Enter your choice [1-5]: 1
[2025-11-07 14:45:21] Running backup script...
[2025-11-07 14:45:21] Running backup script...
[2025-11-07 14:45:21] [SUCCESS] Backup created successfully at: /backup/daily/backup_2025-11-07_14-45-21.tar.gz
[2025-11-07 14:45:21] Backup completed successfully!
[2025-11-07 14:45:21] Backup completed successfully.
Press Enter to continue...
```

Running update.sh and cleanup.sh successfully



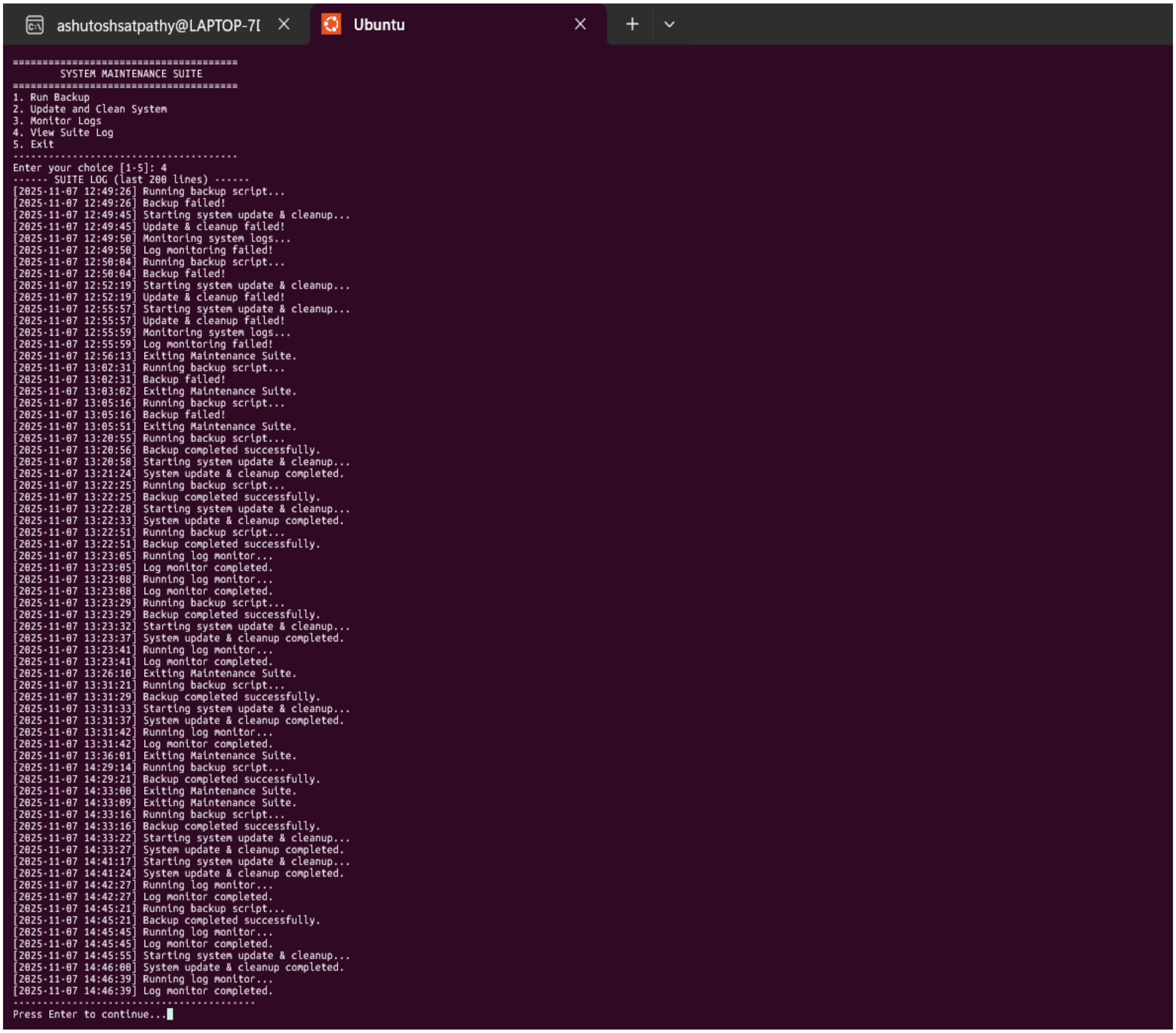
```
=======================================
        SYSTEM MAINTENANCE SUITE
=======================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
---------------------------------------
Enter your choice [1-5]: 2
[2025-11-07 14:45:55] Starting system update & cleanup...
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
[2025-11-07 14:46:00] System update & cleanup completed.
Press Enter to continue...
```

Running Monitor Logs successfully



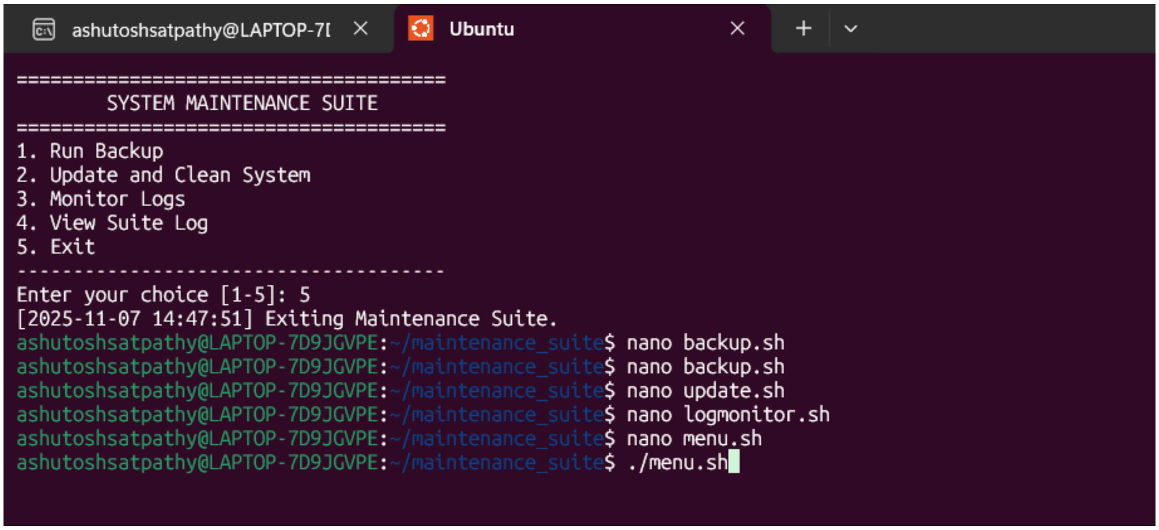
```
======================================
        SYSTEM MAINTENANCE SUITE
======================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
--------------------------------------
Enter your choice [1-5]: 3
[2025-11-07 14:45:45] Running log monitor...
[2025-11-07 14:45:45] Log monitor completed.
Press Enter to continue...
```

Viewing Suite Log



```
=======================================
        SYSTEM MAINTENANCE SUITE
=======================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
---------------------------------------
Enter your choice [1-5]: 4
------ SUITE LOG (last 200 lines) ------
[2025-11-07 12:49:26] Running backup script...
[2025-11-07 12:49:26] Backup failed!
[2025-11-07 12:49:45] Starting system update & cleanup...
[2025-11-07 12:49:45] Update & cleanup failed!
[2025-11-07 12:49:50] Monitoring system logs...
[2025-11-07 12:49:50] Log monitoring failed!
[2025-11-07 12:50:04] Running backup script...
[2025-11-07 12:50:04] Backup failed!
[2025-11-07 12:52:19] Starting system update & cleanup...
[2025-11-07 12:52:19] Update & cleanup failed!
[2025-11-07 12:55:57] Starting system update & cleanup...
[2025-11-07 12:55:57] Update & cleanup failed!
[2025-11-07 12:55:59] Monitoring system logs...
[2025-11-07 12:55:59] Log monitoring failed!
[2025-11-07 12:56:13] Exiting Maintenance Suite.
[2025-11-07 13:02:31] Running backup script...
[2025-11-07 13:02:31] Backup failed!
[2025-11-07 13:03:02] Exiting Maintenance Suite.
[2025-11-07 13:05:16] Running backup script...
[2025-11-07 13:05:16] Backup failed!
[2025-11-07 13:05:51] Exiting Maintenance Suite.
[2025-11-07 13:20:55] Running backup script...
[2025-11-07 13:20:56] Backup completed successfully.
[2025-11-07 13:20:58] Starting system update & cleanup...
[2025-11-07 13:21:24] System update & cleanup completed.
[2025-11-07 13:22:25] Running backup script...
[2025-11-07 13:22:25] Backup completed successfully.
[2025-11-07 13:22:28] Starting system update & cleanup...
[2025-11-07 13:22:33] System update & cleanup completed.
[2025-11-07 13:22:51] Running backup script...
[2025-11-07 13:22:51] Backup completed successfully.
[2025-11-07 13:23:05] Running log monitor...
[2025-11-07 13:23:05] Log monitor completed.
[2025-11-07 13:23:08] Running log monitor...
[2025-11-07 13:23:08] Log monitor completed.
[2025-11-07 13:23:29] Running backup script...
[2025-11-07 13:23:29] Backup completed successfully.
[2025-11-07 13:23:32] Starting system update & cleanup...
[2025-11-07 13:23:37] System update & cleanup completed.
[2025-11-07 13:23:41] Running log monitor...
[2025-11-07 13:23:41] Log monitor completed.
[2025-11-07 13:26:10] Exiting Maintenance Suite.
[2025-11-07 13:31:21] Running backup script...
[2025-11-07 13:31:29] Backup completed successfully.
[2025-11-07 13:31:33] Starting system update & cleanup...
[2025-11-07 13:31:37] System update & cleanup completed.
[2025-11-07 13:31:42] Running log monitor...
[2025-11-07 13:31:42] Log monitor completed.
[2025-11-07 13:36:01] Exiting Maintenance Suite.
[2025-11-07 14:29:14] Running backup script...
[2025-11-07 14:29:21] Backup completed successfully.
[2025-11-07 14:33:00] Exiting Maintenance Suite.
[2025-11-07 14:33:09] Exiting Maintenance Suite.
[2025-11-07 14:33:16] Running backup script...
[2025-11-07 14:33:16] Backup completed successfully.
[2025-11-07 14:33:22] Starting system update & cleanup...
[2025-11-07 14:33:27] System update & cleanup completed.
[2025-11-07 14:41:17] Starting system update & cleanup...
[2025-11-07 14:41:24] System update & cleanup completed.
[2025-11-07 14:42:27] Running log monitor...
[2025-11-07 14:42:27] Log monitor completed.
[2025-11-07 14:45:21] Running backup script...
[2025-11-07 14:45:21] Backup completed successfully.
[2025-11-07 14:45:45] Running log monitor...
[2025-11-07 14:45:45] Log monitor completed.
[2025-11-07 14:45:55] Starting system update & cleanup...
[2025-11-07 14:46:00] System update & cleanup completed.
[2025-11-07 14:46:39] Running log monitor...
[2025-11-07 14:46:39] Log monitor completed.
-----------------------------------------
Press Enter to continue...
```

Final Exit



```
======================================
        SYSTEM MAINTENANCE SUITE
======================================
1. Run Backup
2. Update and Clean System
3. Monitor Logs
4. View Suite Log
5. Exit
--------------------------------------
Enter your choice [1-5]: 5
[2025-11-07 14:47:51] Exiting Maintenance Suite.
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano backup.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano backup.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano update.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano logmonitor.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ nano menu.sh
ashutoshsatpathy@LAPTOP-7D9JGVPE:~/maintenance_suite$ ./menu.sh
```

Conclusion

The System Maintenance Suite acts like a smart humanoid assistant for your computer — it performs routine maintenance tasks automatically, just like a system caretaker. It takes care of backups, software updates, cleaning unnecessary files, and keeping logs — all without constant human supervision. With built-in error handling and detailed logs, it ensures everything runs smoothly and transparently. This project shows how automation through Bash scripting can make a system more self-sufficient, efficient, and reliable — much like giving your computer a human-like ability to maintain itself.