System Maintenance Suite – Automated Backup, Update, and Log Monitoring Tool

Author: Sanata Dash

1. Objective

The project aims to develop an automated Bash-based maintenance suite capable of performing system backups, updates and cleanup operations, and log scanning for critical issues. It provides a unified command-line menu for simplified system management.

2. Codes

maint.sh

```
#!/usr/bin/env bash
set -Eeuo pipefail
banner() { echo -e "\n========== $* ==========
"; }
while true; do
  clear
  echo "System Maintenance Suite"
  echo "1) Run Backup"
  echo "2) Update & Cleanup"
  echo "3) Scan Logs for Issues"
  echo "4) Configure Cron Schedules"
  echo "5) View Logs Directory"
  echo "6) Exit"
  read -rp "Choose an option: " opt
  case "$opt" in
    1) banner "Backup"; ./scripts/backup.sh; read -rp "Press Enter...";;
    2) banner "Update & Cleanup"; ./scripts/update_cleanup.sh; read -rp "Press Enter...";;
    3) banner "Log Monitor"; ./scripts/log_monitor.sh; read -rp "Press Enter...";;
    4) banner "Cron Setup"; echo "Example: 0 2 * * * $(pwd)/scripts/backup.sh"; echo "Open
crontab editor now? (y/n)"; read yn; [ "$yn" = "y" ] && crontab -e;;
    5) banner "Logs"; ls -lah logs; read -rp "Press Enter...";;
    6) exit 0;;
    *) echo "Invalid"; sleep 1;;
  esac
done
```

backup.sh

```
#!/usr/bin/env bash
set -euo pipefail
```

```
SRC="${1:-$HOME}"
DEST_DIR="${2:-$(pwd)/backups}"
mkdir -p "$DEST_DIR"
STAMP="$(date +'%Y-%m-%d_%H-%M-%S')"
ARCHIVE="$DEST_DIR/backup_$STAMP.tar.gz"
tar --exclude="$HOME/.cache" --exclude="*/node_modules" -czf "$ARCHIVE" "$SRC"
echo "Backup created: $ARCHIVE"
```

update_cleanup.sh

```
#!/usr/bin/env bash
set -euo pipefail
LOG_DIR="$(pwd)/logs"
mkdir -p "$LOG_DIR"
LOG_FILE="$LOG_DIR/update_$(date +'%Y-%m-%d_%H-%M-%S').log"
run() { echo "+ $*" | tee -a "$LOG_FILE"; "$@" |& tee -a "$LOG_FILE"; }
if command -v apt >/dev/null 2>&1; then
 PKG="apt"
 run sudo apt update
 run sudo apt -y upgrade
 run sudo apt -y autoremove
 run sudo apt -y autoclean
elif command -v dnf >/dev/null 2>&1; then
 PKG="dnf"
 run sudo dnf -y upgrade --refresh
 run sudo dnf -y autoremove
else
 echo "Unsupported package manager" | tee -a "$LOG_FILE"
 exit 1
fi
echo "Updates complete via $PKG. Log: $LOG_FILE"
```

log_monitor.sh

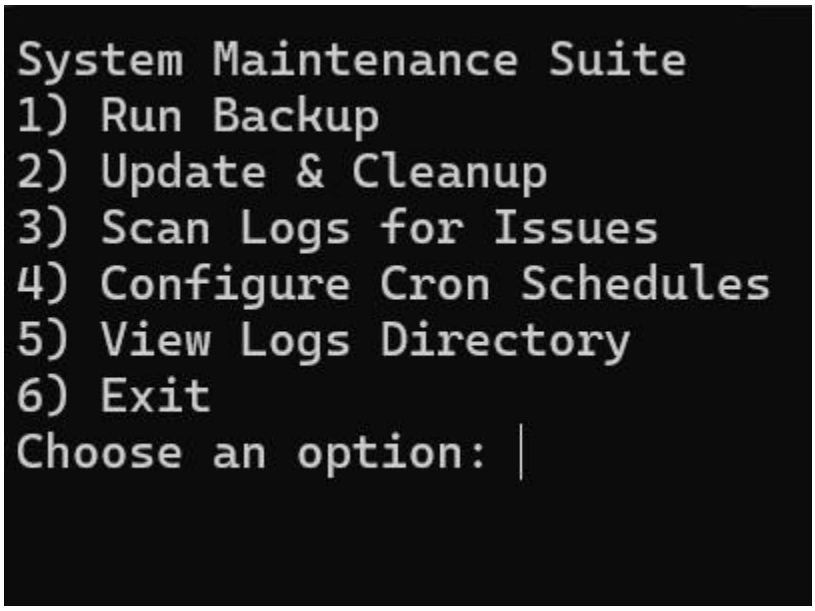
```
#!/usr/bin/env bash
set -euo pipefail
LOG_SRC=""
if [ -f /var/log/syslog ]; then LOG_SRC="/var/log/syslog";
elif [ -f /var/log/messages ]; then LOG_SRC="/var/log/messages";
else echo "No system log found"; exit 1; fi
OUT_DIR="$(pwd)/logs"
mkdir -p "$OUT_DIR"
OUT_FILE="$OUT_DIR/alerts_$(date +'%Y-%m-%d').log"
```

```
PATTERN="${1:-'(error|failed|critical|panic)'}"
TAIL_LINES="${2:-500}"
echo "Scanning $LOG_SRC (last $TAIL_LINES lines) for: $PATTERN"
grep -Ei "$PATTERN" <(tail -n "$TAIL_LINES" "$LOG_SRC") | tee -a "$OUT_FILE" || true
echo "Results saved to: $OUT_FILE"
```

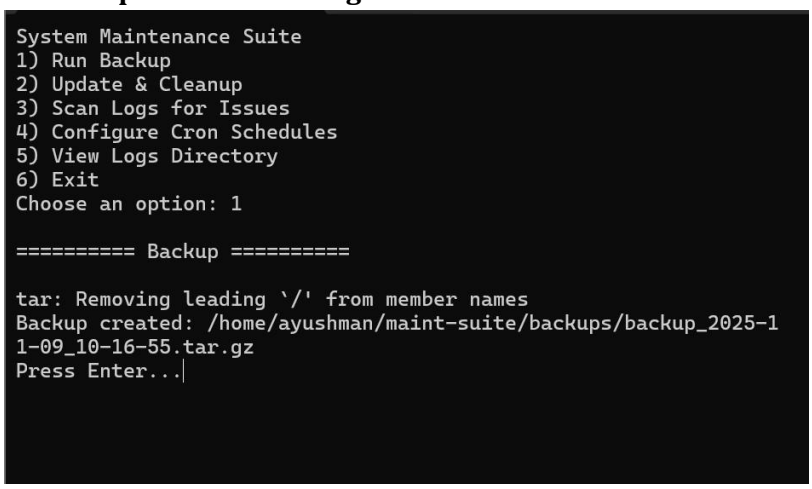
3. Screenshots

Screenshots based on execution:

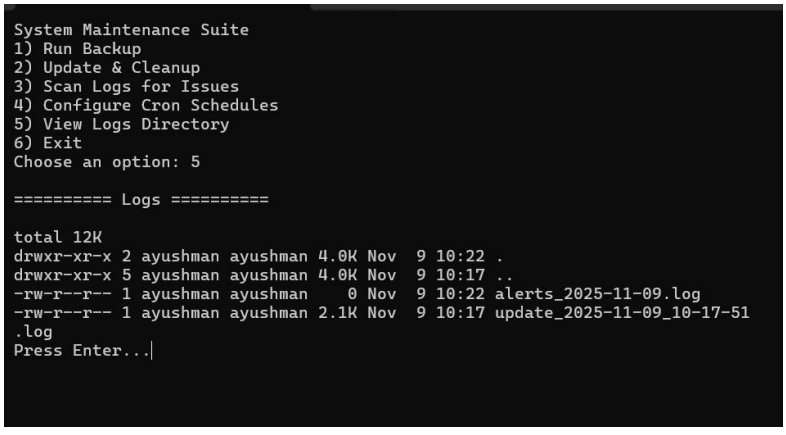
1. Main menu interface.



2. Backup creation message.

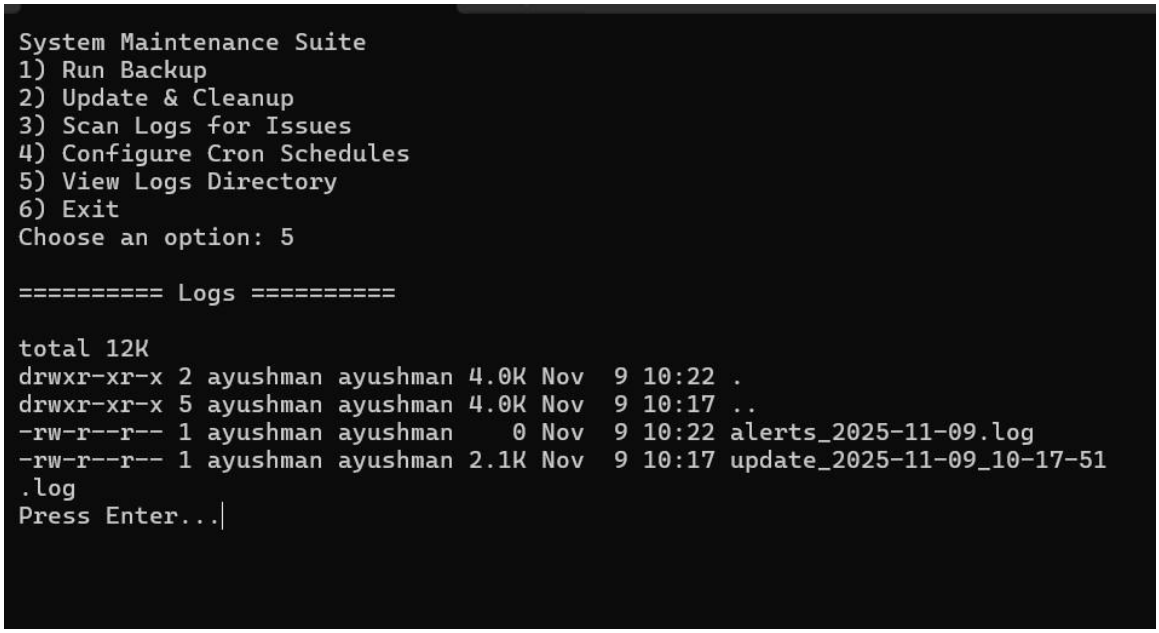


3. Update and cleanup log creation.

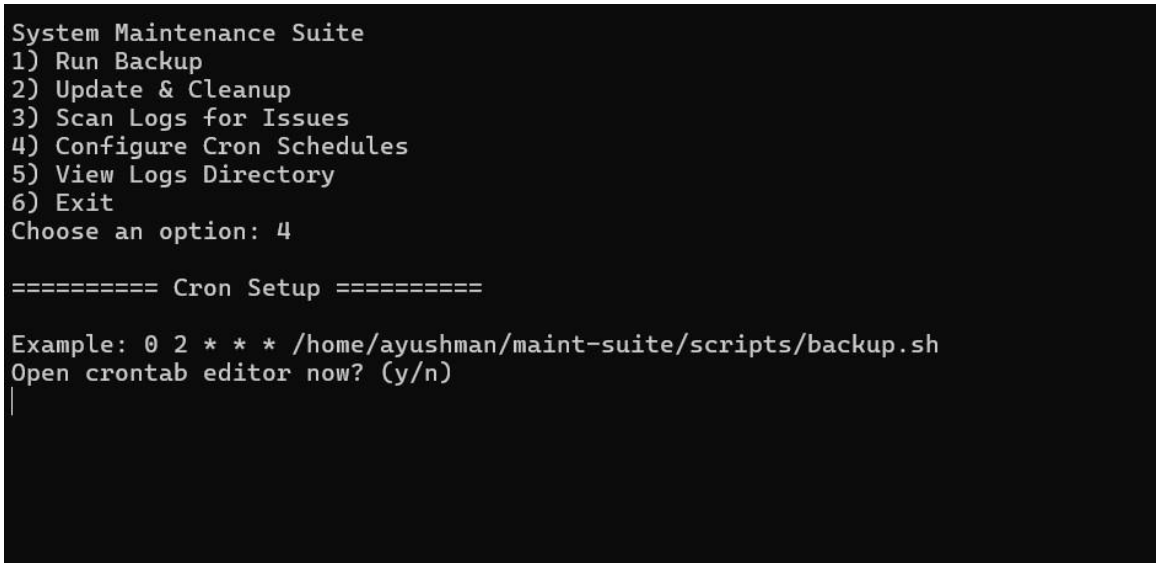


4. Log monitor output.

5. Cron setup prompt.



6. Logs directory listing.



4. Conclusion

The **System Maintenance Suite** project successfully demonstrates how automation can simplify routine system administration tasks using Bash scripting.
It integrates essential maintenance features such as **backup creation, system updates, and log monitoring** into a single unified command-line interface, reducing manual effort and minimizing the risk of system failure due to overlooked maintenance.

Through this project, I gained practical experience in **shell scripting, file management, process automation, and Linux system operations**. The modular script design ensures scalability — new maintenance modules can easily be added in the future.

Overall, the project enhances system reliability, saves time for users, and provides a strong foundation for further development into a **smart automated maintenance assistant**.