Name : Sasanka Sekhar kundu

Roll Number : 150096725118

Subject : Operating System (OS)

Semister : 1 Sprint : 1

Collage : ITM Skills University

Cohort : Demis Hassabis

Batch : 2025 - 2026

Problem Statement :

A company frequently updates its database host, username, and password stored in configuration files. Manual updates are error-prone, inconsistent, and may lead to downtime.

To solve this, automation through shell scripting is proposed. Using the Linux `sed` command, the script will:

- Replace credentials,
- Insert backup information,
- Append connection status,
- Delete outdated entries, and
- Preview changes before applying.

This approach ensures updates are consistent, error-free, and efficient.

Objectives :

The objectives of this case study are to design and implement a shell script that automates database configuration updates. The specific goals are:

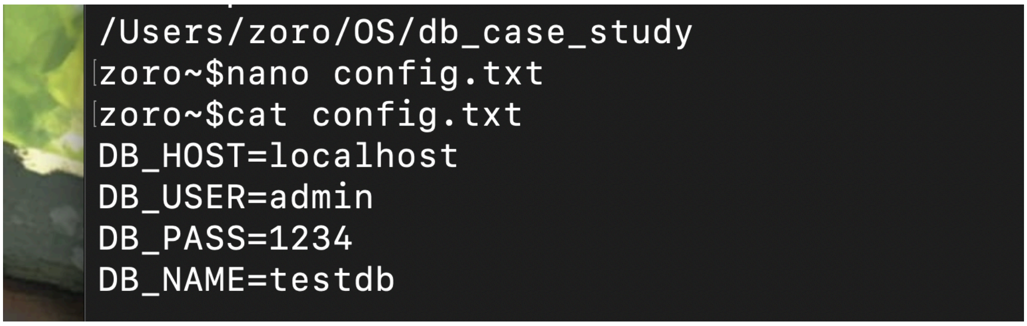
1. Replace existing database credentials
   - Update database host, username, and password automatically using the `s` (substitute) operation.
2. Insert backup information
   - Ensure reliability by adding backup details into the configuration file using the `i` (insert) operation.
3. Append connection status
   - Record connection details (e.g., `CONNECTION=active`) at the end of the file using the `a` (append) operation.
4. Delete outdated database entries
   - Remove unnecessary or old entries (such as obsolete database names) using the `d` (delete) operation.
5. Preview updated configuration
   - Display modified lines before deployment using the `p` (print) operation to verify correctness.

Methodology / Workflow :

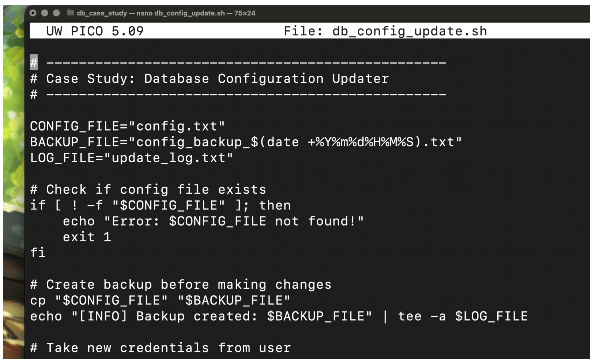
To automate updates of database configuration files, a shell script was implemented using `sed`. The workflow reduces manual errors and ensures consistency.

Step	Description
Setup	Created project folder `db_case_study` and sample config file `config.txt`.
Script Development	Wrote `db_config_update.sh` with user input, backup creation, error checking, and logging.
Operations	`s`: substitute credentials `i`: insert backup info `a`: append connection status `d`: delete outdated entries `p`: preview changes
Execution	Run script → enter new values → backup created → config updated → preview displayed.
Verification	Checked updated file with `cat` and reviewed backup + log.

Config File (Before Update)



```
/Users/zoro/OS/db_case_study
zoro~$nano config.txt
zoro~$cat config.txt
DB_HOST=localhost
DB_USER=admin
DB_PASS=1234
DB_NAME=testdb
```

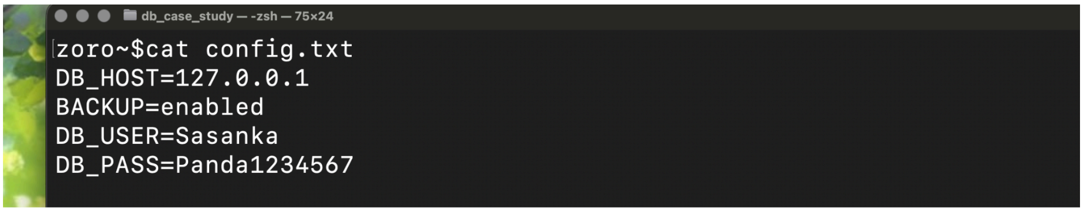


```
  UW PICO 5.09                     File: db_config_update.sh

# -------------------------------------------------
# Case Study: Database Configuration Updater
# -------------------------------------------------

CONFIG_FILE="config.txt"
BACKUP_FILE="config_backup_$(date +%Y%m%d%H%M%S).txt"
LOG_FILE="update_log.txt"

# Check if config file exists
if [ ! -f "$CONFIG_FILE" ]; then
    echo "Error: $CONFIG_FILE not found!"
    exit 1
fi

# Create backup before making changes
cp "$CONFIG_FILE" "$BACKUP_FILE"
echo "[INFO] Backup created: $BACKUP_FILE" | tee -a $LOG_FILE

# Take new credentials from user
```

Results :

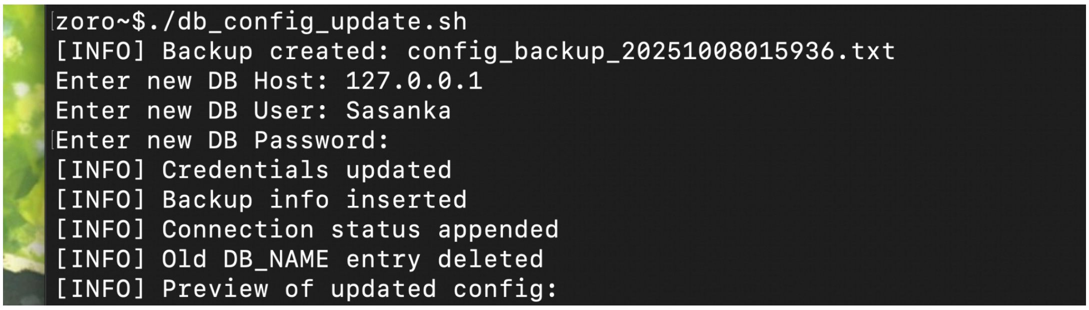
The script `db_config_update.sh` was executed successfully. The following outputs were observed:

(a) Config File (After Update)



```
zoro~$cat config.txt
DB_HOST=127.0.0.1
BACKUP=enabled
DB_USER=Sasanka
DB_PASS=Panda1234567
```

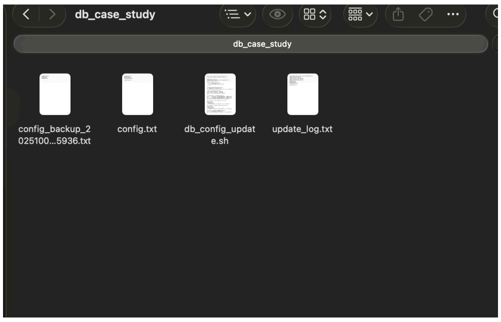
(b) Execution Log (update_log.txt)



```
zoro~$./db_config_update.sh
[INFO] Backup created: config_backup_20251008015936.txt
Enter new DB Host: 127.0.0.1
Enter new DB User: Sasanka
Enter new DB Password:
[INFO] Credentials updated
[INFO] Backup info inserted
[INFO] Connection status appended
[INFO] Old DB_NAME entry deleted
[INFO] Preview of updated config:
```

Verification :

- Updated config file reflected all changes correctly.
- Backup file was generated (`config_backup_...`).
- Log file confirmed step-by-step execution.



Conclusion :

This case study showed how Linux `sed` commands can automate database configuration updates. The script replaced credentials, added backup info, appended connection status, deleted outdated entries, and provided a preview. Automation reduces errors, ensures consistency, and improves reliability.

References :

[Linux sed Manual]()
[Mastering sed Command in Linux: A Comprehensive Guide]()
Notebook
Github
THANK YOU…….