**📄 Project Title: Automated Backup Script (backup.sh)**

**🧑‍💻 Author: [Souvik Mandal]**

**🏢 Organization: ABC International Inc.**

**📅 Project Goal:**

To automate the process of backing up encrypted password files modified in the past 24 hours, reducing human error and improving efficiency.

**📜 Script Breakdown and Explanation**

bash

CopyEdit

#!/bin/bash

**Shebang** – Specifies the interpreter to execute the script, in this case /bin/bash.

bash

CopyEdit

if [[ $# != 2 ]]

then

echo "backup.sh target\_directory\_name destination\_directory\_name"

exit

fi

**Task**: Validate that exactly two arguments are passed to the script.

* $# holds the number of passed arguments.
* If not 2, show usage instruction and exit.

bash

CopyEdit

if [[ ! -d $1 ]] || [[ ! -d $2 ]]

then

echo "Invalid directory path provided"

exit

fi

**Task**: Validate that both passed arguments are valid directories.

* -d checks if the path is a directory.

bash

CopyEdit

targetDirectory=$1

destinationDirectory=$2

**Task 1**: Assign the first and second command-line arguments to respective variables.

bash

CopyEdit

echo "Target Directory: $targetDirectory"

echo "Destination Directory: $destinationDirectory"

**Task 2**: Print the provided directories for confirmation.

bash

CopyEdit

currentTS=$(date +%s)

**Task 3**: Capture the current timestamp (in seconds since Epoch) into currentTS.

bash

CopyEdit

backupFileName="backup-$currentTS.tar.gz"

**Task 4**: Set the name of the backup file using the timestamp.

bash

CopyEdit

origAbsPath=$(pwd)

**Task 5**: Save the absolute path of the original directory (before any cd).

bash

CopyEdit

cd "$destinationDirectory" || exit

destDirAbsPath=$(pwd)

**Task 6**: Change to destination directory and get its absolute path.  
|| exit ensures script exits if cd fails.

bash

CopyEdit

cd "$origAbsPath" || exit

cd "$targetDirectory" || exit

**Task 7**: Navigate to the target directory where files will be scanned.

bash

CopyEdit

yesterdayTS=$((currentTS - 24 \* 60 \* 60))

**Task 8**: Calculate the timestamp 24 hours ago for comparison.

bash

CopyEdit

declare -a toBackup

Initialize an array toBackup to store names of files to archive.

bash

CopyEdit

for file in \*

**Task 9**: Loop through all files and directories in the current folder.

bash

CopyEdit

if [[ $(date -r "$file" +%s) -gt $yesterdayTS ]]

**Task 10**: Check if file's last modified time is within the last 24 hours.

bash

CopyEdit

toBackup+=("$file")

**Task 11**: Add qualified file to the toBackup array.

bash

CopyEdit

tar -czvf $backupFileName ${toBackup[@]}

**Task 12**: Archive and compress all files in the toBackup array into one .tar.gz file.

bash

CopyEdit

mv $backupFileName $destDirAbsPath

**Task 13**: Move the backup file to the destination directory.

**✅ Output Example**

When run correctly:

bash

CopyEdit

Target Directory: /home/user/docs

Destination Directory: /home/user/backups

backup-1718892304.tar.gz

**🕒 Crontab Example**

To run this script daily:

bash

CopyEdit

0 0 \* \* \* /usr/local/bin/backup.sh /path/to/source /path/to/destination

To run every 1 minute for testing:

bash

CopyEdit

\*/1 \* \* \* \* /usr/local/bin/backup.sh /path/to/source /path/to/destination