**Backup Script Code And Explanation**#!/bin/bash

WEB\_DIR="/var/www/html/wordpress"

BACKUP\_DIR="/home/ubuntu/website\_backups"

DB\_NAME="m\_ali"

DB\_USER="m\_ali"

DB\_PASS="pak@123"

DATE=$(date +"%Y-%m-%d\_%H-%M")

LOG\_FILE="$BACKUP\_DIR/backup\_log.txt"

DAYS\_TO\_KEEP=7

if [ ! -d "$BACKUP\_DIR" ]; then   
mkdir "$BACKUP\_DIR"  
fi

BACKUP\_PATH="$BACKUP\_DIR/backup-$DATE"

mkdir "$BACKUP\_PATH"

echo "Backup started on $(date)" >> "$LOG\_FILE"

echo "Copying website files..." >> "$LOG\_FILE"

tar -czf "$BACKUP\_PATH/wordpress\_files.tar.gz" "$WEB\_DIR"

echo "Saving database..." >> "$LOG\_FILE"

mysqldump -u "$DB\_USER" -p"$DB\_PASS" "$DB\_NAME" > "$BACKUP\_PATH/db\_backup.sql"

echo "Backup done! Saved in $BACKUP\_PATH" >> "$LOG\_FILE"

echo "Cleaning up old backups..." >> "$LOG\_FILE"

find "$BACKUP\_DIR" -maxdepth 1 -type d -name "backup-\*" -mtime +$DAYS\_TO\_KEEP -exec rm -rf {} \; >> "$LOG\_FILE"

echo "Finished on $(date)" >> "$LOG\_FILE"

echo "------------------------------" >> "$LOG\_FILE"

**Explanation**

The following is a backup script for backing up website files and MySQL database for a website using WordPress. It starts with a line declaring it uses Bash shell to run it. It sets a number of variables for use in the process. These include where website files are (/var/www/html/wordpress), where backups are stored (/home/ubuntu/website\_backups), database name (m\_ali), database username (m\_ali), and password (pak@123). It also sets a variable as current date and time, which it uses as a prefix while naming backups so no backup would overwrite an older one. It uses a different variable as a location of a log file documenting script activity and yet a different one as how many days' backups should be retained (7 days in this script).  
  
The script then searches for a backup directory, and if it does not already have one, it creates it. Creates a new folder for this specific backup with a date as well as timestamp. Logs the date and timestamp as a start time for record keeping in a log file. Backs up the website files. Logs, compresses the entire folder containing WordPress into a single archive file (in .tar.gz form), then moves it into the new backup folder. Once the backup file is complete, the script diligently marks the beginning database backup before using the mysqldump command to export the contents of the WordPress database into an .sql file, also in a backup folder. Having backed up the database as well as site files, the script marks backup completion alongside where it has been stored.  
  
To prevent disk space from filling up as well as declutter, the script also searches for older backups within the backup directory. It searches for any directories with a name in the format and more than 7 days old and removes them automatically. It furthermore marks as finished the time a process has completed as well as contains a divider line within its log file, separating it from other backup entries, making it more readable. The script as a whole is a great way of backing up a WordPress site on a regular basis as well as keeping backups organized.