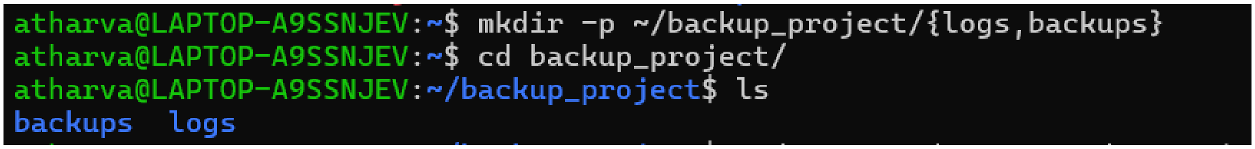
Project Title: "Automated Backup and Rotation System with Google Drive Integration using Python and rclone"

Objective:

The goal of this project is to automate the process of backing up important project folders.
It creates `.zip` files from a source folder, stores them locally, uploads them to Google Drive, and manages (deletes) old backups.
It also sends a notification to confirm the backup and writes a log for tracking.
This system runs daily using a cron job, and helps ensure that data is never lost.

Steps:

1. Create Project Folder Structure:



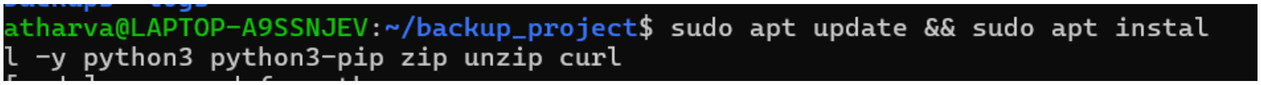
```
atharva@LAPTOP-A9SSNJEV:~$ mkdir -p ~/backup_project/{logs,backups}
atharva@LAPTOP-A9SSNJEV:~$ cd backup_project/
atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls
backups   logs
```

Purpose:

`logs/`: Stores a log file showing all backup operations

backups/: Temporarily stores zipped files before uploading to Google Drive

2. Install Required Tools:



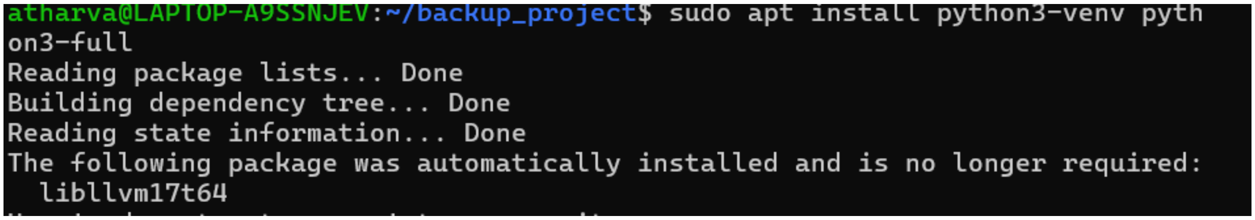
```
atharva@LAPTOP-A9SSNJEV:~/backup_project$ sudo apt update && sudo apt instal
l -y python3 python3-pip zip unzip curl
```

why:

1) `python3`, `pip3`: For running Python scripts
2) `zip`/`unzip`: For creating `.zip` files
3) `curl`: For uploading to webhook

4) `python-dotenv` is required for your script to read `.env` variables:

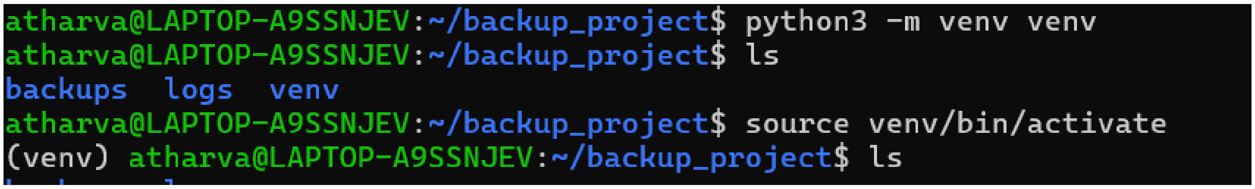
i) To enable `python3 -m venv` command



```
atharva@LAPTOP-A9SSNJEV:~/backup_project$ sudo apt install python3-venv pyth
on3-full
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libllvm17t64
```

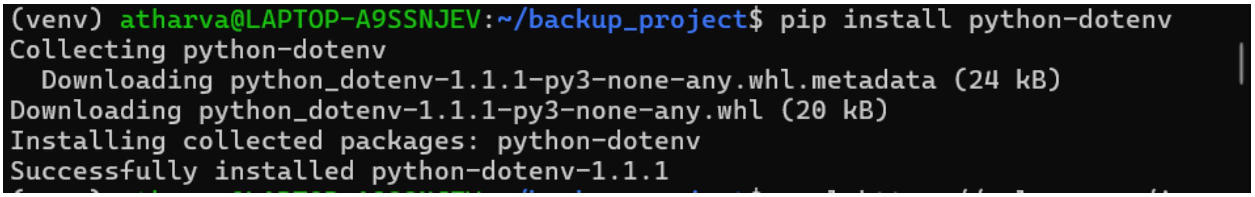
ii) python3 -m venv venv-To create isolated python environment

iii) source venv/bin/activate - To enter the environment



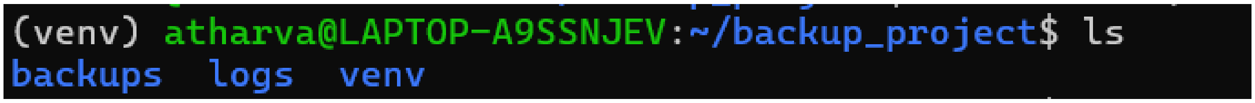
```
atharva@LAPTOP-A9SSNJEV:~/backup_project$ python3 -m venv venv
atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls
backups  logs  venv
atharva@LAPTOP-A9SSNJEV:~/backup_project$ source venv/bin/activate
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls
```

iii) pip install python-dotenv - To read .env file in your script



```
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ pip install python-dotenv
Collecting python-dotenv
  Downloading python_dotenv-1.1.1-py3-none-any.whl.metadata (24 kB)
Downloading python_dotenv-1.1.1-py3-none-any.whl (20 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.1.1
```

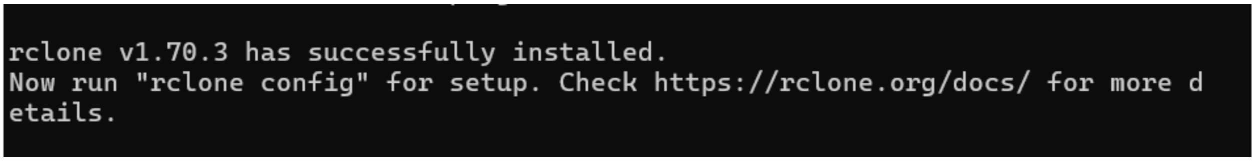
iv)venv folder – Keeps project clean and dependency-safe



```
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls
backups  logs  venv
```

3. Install and Configure rclone:

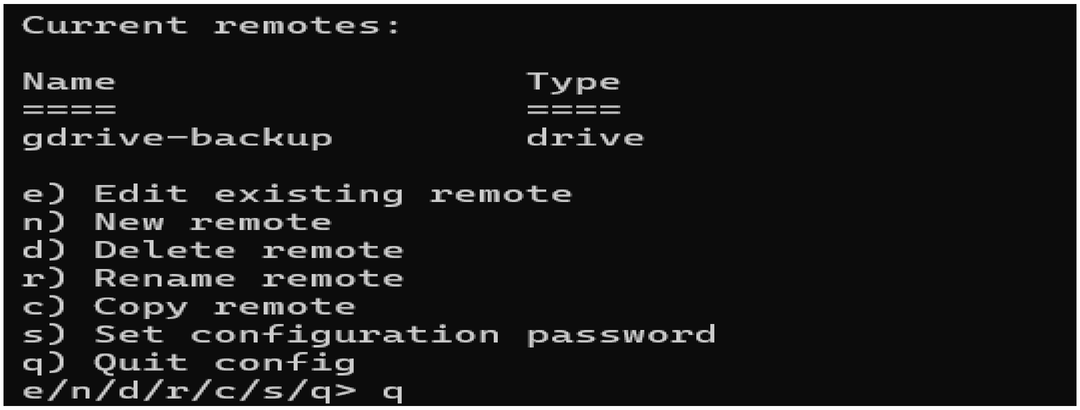
Installed rclone (used to upload to Google Drive):



```
rclone v1.70.3 has successfully installed.
Now run "rclone config" for setup. Check https://rclone.org/docs/ for more d
etails.
```

I)Using - “curl https://rclone.org/install.sh | sudo bash

ii)rclone config.



```
Current remotes:

Name                 Type
====                 ====
gdrive-backup        drive

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q
```

During `rclone config`:

- We created a remote called `gdrive-backup`
- Selected `drive` as the storage type (Google Drive)
- Logged in to Google to give rclone permission
- We skipped setting client ID/secret (left it empty)
- Chose full access scope
- Did not use service account or shared drive

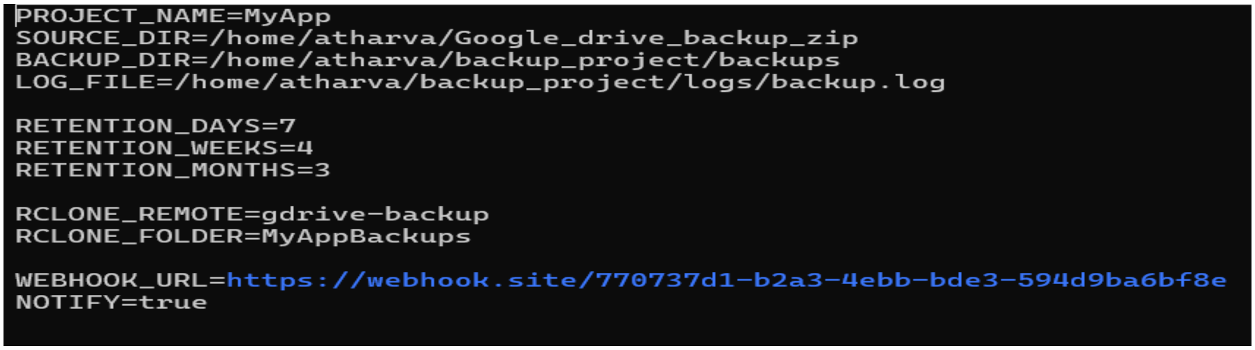
Why:

rclone acts as the middleman to upload zipped backups to Google Drive

4.Create Configuration File:

We created the config file:

`nano ~/.backup_config.env`



```
PROJECT_NAME=MyApp
SOURCE_DIR=/home/atharva/Google_drive_backup_zip
BACKUP_DIR=/home/atharva/backup_project/backups
LOG_FILE=/home/atharva/backup_project/logs/backup.log

RETENTION_DAYS=7
RETENTION_WEEKS=4
RETENTION_MONTHS=3

RCLONE_REMOTE=gdrive-backup
RCLONE_FOLDER=MyAppBackups

WEBHOOK_URL=https://webhook.site/770737d1-b2a3-4ebb-bde3-594d9ba6bf8e
NOTIFY=true
```

Ii)we get the webhook url from webhook website:

Why:

Stores values like source folder, backup location, retention policy, etc.

Helps make the Python script reusable and easy to change later

5.Write the python Backup Script:

We created the script:

```
nano ~/backup_project/backup_script.py
```

This script does the following:

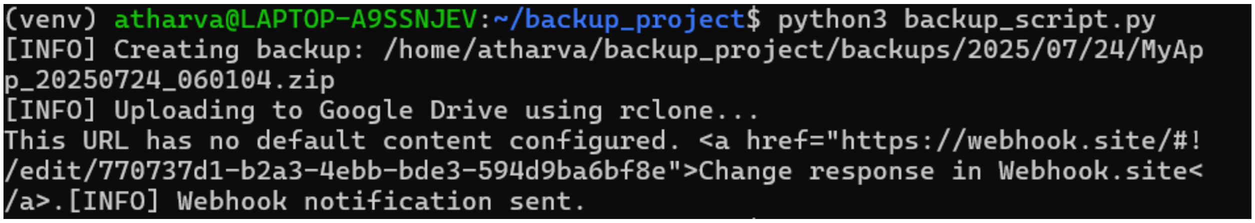
- Loads variables from the .env file
- Creates a .zip file of the source directory
- Stores it in the backup folder with a timestamp
- Uploads the zip to Google Drive using rclone
- Sends a webhook notification (optional)
- Logs each operation
- Deletes old backups based on retention policy

We made it executable:

```
chmod +x ~/backup_project/backup_script.py
```

6.Test the Script:

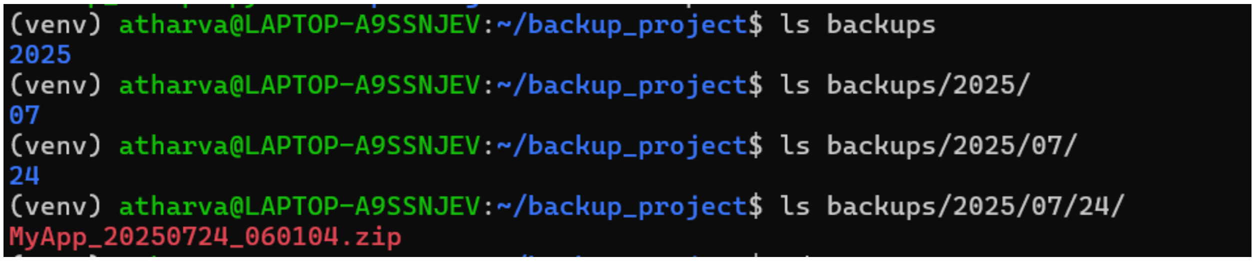
We tested the script manually:



```
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ python3 backup_script.py
[INFO] Creating backup: /home/atharva/backup_project/backups/2025/07/24/MyAp
p_20250724_060104.zip
[INFO] Uploading to Google Drive using rclone...
This URL has no default content configured. <a href="https://webhook.site/#!
/edit/770737d1-b2a3-4ebb-bde3-594d9ba6bf8e">Change response in Webhook.site<
/a>.[INFO] Webhook notification sent.
```

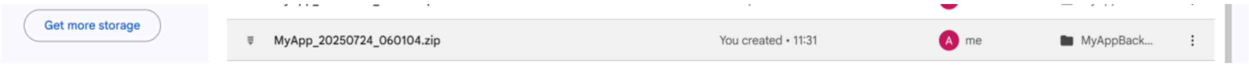
What we checked:

i) New zip file created inside backups/YYYY/MM/DD/



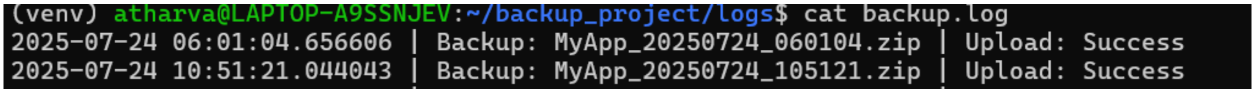
```
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls backups
2025
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls backups/2025/
07
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls backups/2025/07/
24
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project$ ls backups/2025/07/24/
MyApp_20250724_060104.zip
```

```
ii) File was uploaded to Google Drive
```



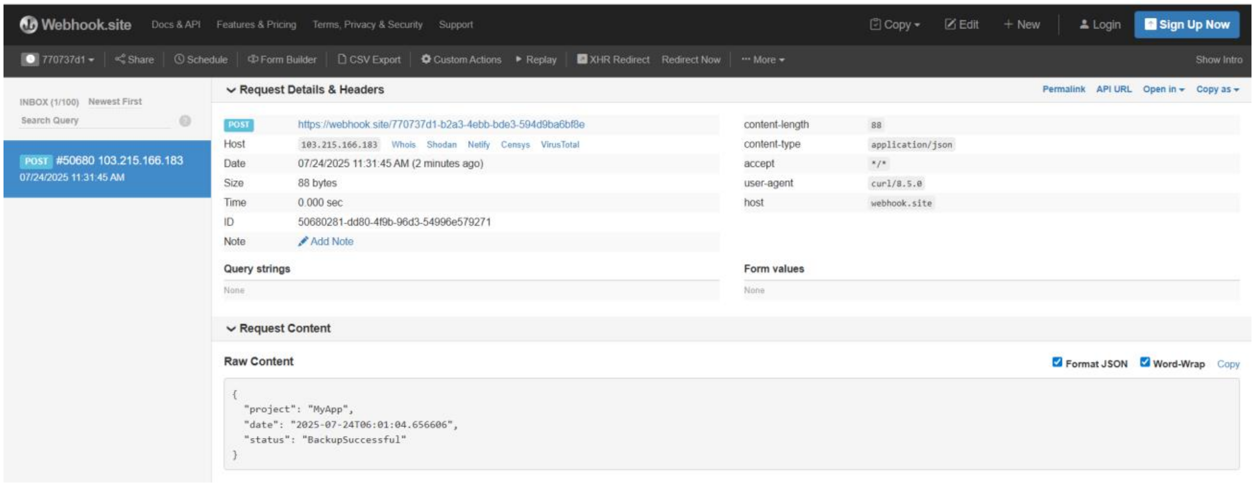
MyApp_20250724_060104.zip You created • 11:31 me MyAppBack...

```
iii) Log was added in logs/backup.log
```



```
(venv) atharva@LAPTOP-A9SSNJEV:~/backup_project/logs$ cat backup.log
2025-07-24 06:01:04.656606 | Backup: MyApp_20250724_060104.zip | Upload: Success
2025-07-24 10:51:21.044043 | Backup: MyApp_20250724_105121.zip | Upload: Success
```

```
iv) Notification seen on webhook.site
```



POST #50680 103.215.166.183
07/24/2025 11:31:45 AM

Request Details & Headers

POST https://webhook.site/770737d1-b2a3-4ebb-bde3-594d9ba6bf8e
Host 103.215.166.183 Whois Shodan Netify Censys VirusTotal
Date 07/24/2025 11:31:45 AM (2 minutes ago)
Size 88 bytes
Time 0.000 sec
ID 50680281-dd80-4f9b-96d3-54996e579271
Note Add Note

content-length 88
content-type application/json
accept */*
user-agent curl/8.5.0
host webhook.site

Query strings
None

Form values
None

Request Content

Raw Content ☑ Format JSON ☑ Word-Wrap

```
{
  "project": "MyApp",
  "date": "2025-07-24T06:01:04.656606",
  "status": "BackupSuccessful"
}
```

7. Automating the Script via Cron with Virtual Environment:

Problem occured when running cron job without shell script:

The cron job was not running the Python backup script properly, even though it worked when we ran it manually in the terminal.

Why This Happened:

Cron runs in a very minimal shell environment:

- It doesn't know your virtual environment exists.
- It doesn’t automatically load your environment variables or installed Python packages like `python-dotenv`.

So, when cron tried to run the Python script, it failed silently because `dotenv` and other dependencies were not available.

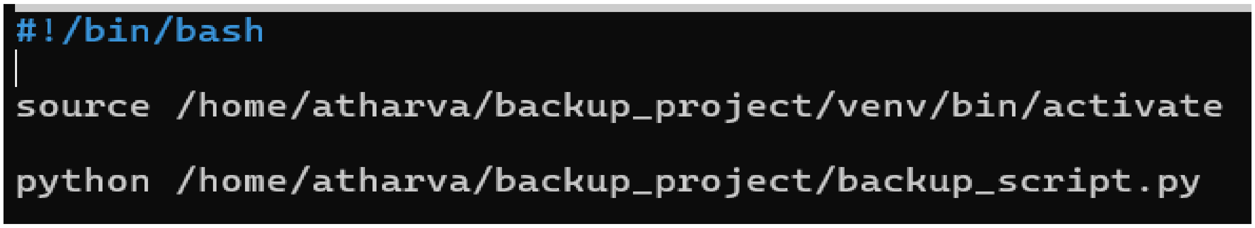
Solution:

We fixed this by creating a shell script that activates the virtual environment before running the Python script.

Steps:

1) Create `run_backup.sh` Shell Script:

“nano ~/backup_project/run_backup.sh”



```
#!/bin/bash

source /home/atharva/backup_project/venv/bin/activate

python /home/atharva/backup_project/backup_script.py
```

Why:

- This script makes sure the virtual environment is loaded.
- So when the Python script runs, it has access to `python-dotenv` and other packages.

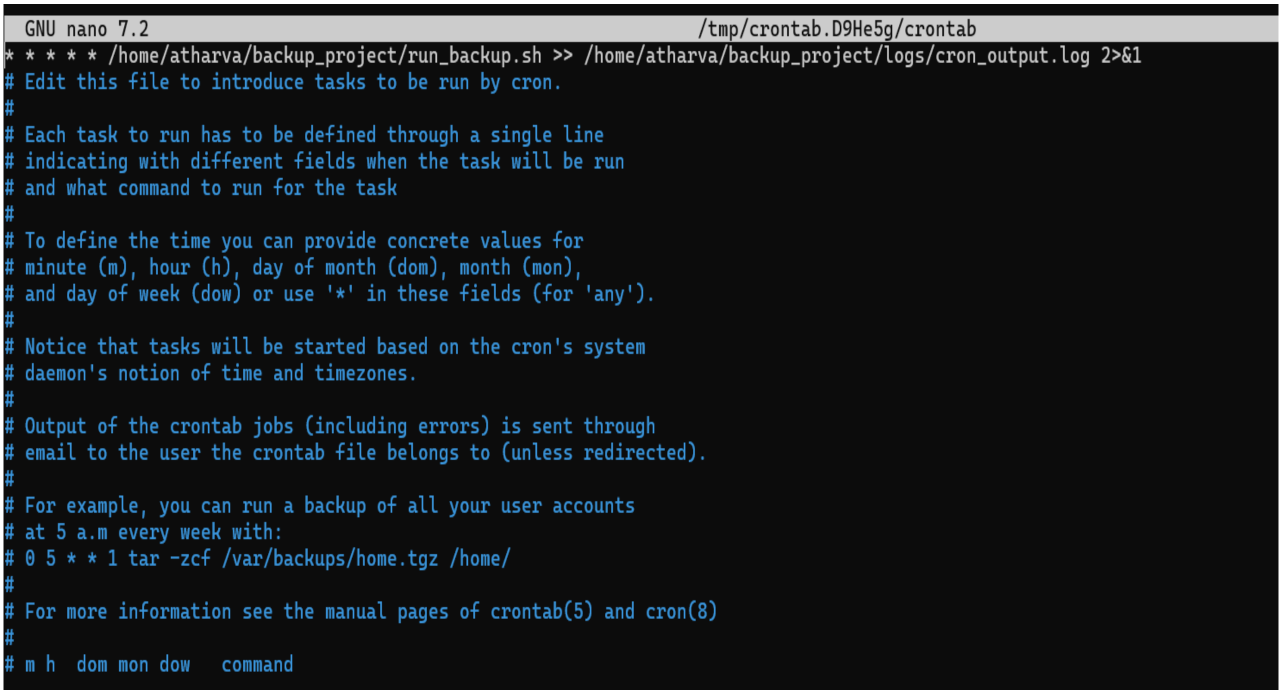
2) Make It Executable:

“chmod +x ~/backup_project/run_backup.sh”

3) Schedule It with Cron:

“crontab -e”

4)Add this line to run it every minute or you can modify it according to you:

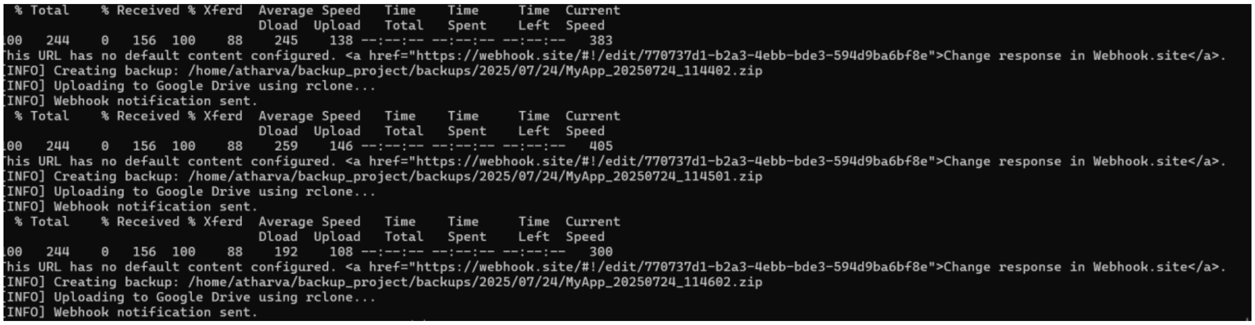


```
  GNU nano 7.2                                                        /tmp/crontab.D9He5g/crontab
* * * * * /home/atharva/backup_project/run_backup.sh >> /home/atharva/backup_project/logs/cron_output.log 2>&1
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
```

What this does:

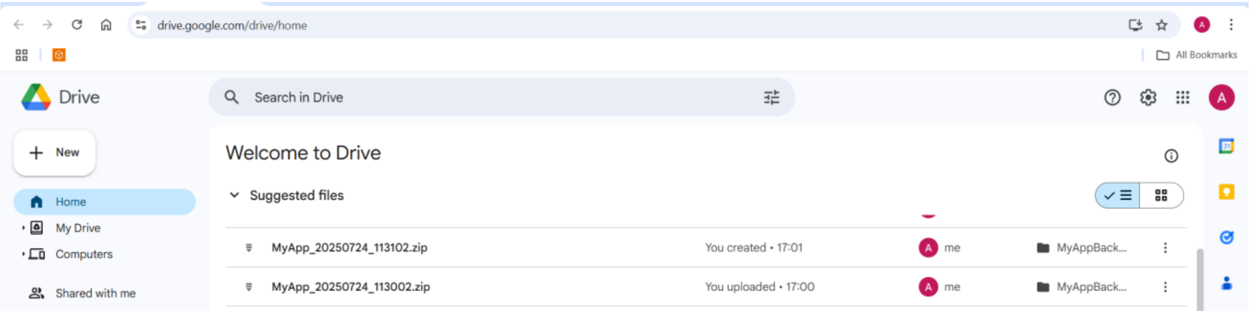
- Runs the backup automatically every 1 minute
- Saves any output (success/failure logs) into `cron_output.log`.

Output:



```
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
00   244    0   156  100    88    245    138 --:--:-- --:--:-- --:--:--   383
his URL has no default content configured. <a href="https://webhook.site/#!/edit/770737d1-b2a3-4ebb-bde3-594d9ba6bf8e">Change response in Webhook.site</a>.
INFO] Creating backup: /home/atharva/backup_project/backups/2025/07/24/MyApp_20250724_114402.zip
INFO] Uploading to Google Drive using rclone...
INFO] Webhook notification sent.
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
00   244    0   156  100    88    259    146 --:--:-- --:--:-- --:--:--   405
his URL has no default content configured. <a href="https://webhook.site/#!/edit/770737d1-b2a3-4ebb-bde3-594d9ba6bf8e">Change response in Webhook.site</a>.
INFO] Creating backup: /home/atharva/backup_project/backups/2025/07/24/MyApp_20250724_114501.zip
INFO] Uploading to Google Drive using rclone...
INFO] Webhook notification sent.
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
00   244    0   156  100    88    192    108 --:--:-- --:--:-- --:--:--   300
his URL has no default content configured. <a href="https://webhook.site/#!/edit/770737d1-b2a3-4ebb-bde3-594d9ba6bf8e">Change response in Webhook.site</a>.
INFO] Creating backup: /home/atharva/backup_project/backups/2025/07/24/MyApp_20250724_114602.zip
INFO] Uploading to Google Drive using rclone...
INFO] Webhook notification sent.
```

On google drive:



MyApp_20250724_113102.zip	You created • 17:01	me	MyAppBack...
MyApp_20250724_113002.zip	You uploaded • 17:00	me	MyAppBack...

Project Summary:

This project automates the process of backing up a local project directory by compressing it into a timestamped `.zip` file, uploading it to a specified Google Drive folder using `rclone`, and managing backup rotation (daily/weekly/monthly retention). It also sends a webhook notification upon successful backup and logs all activities. The entire process is configurable via an `.env` file and scheduled with a `cron` job.

Key Features:

Environment-based Configuration (`.`env file)
⍰ **Timestamped Backup Zip Creation**
⍰ **Google Drive Upload via rclone**
Retention Policy: Automatically deletes older backups (daily, weekly, monthly)
Webhook Notification after successful backup
Logging: Logs each action (success/failure/deletions)
Automated Execution via Cron