The code for web Scrapping is in the link https://github.com/Thapa-G/Web_scrapping.git

- Here as the task suggests I have written the code in two phases.

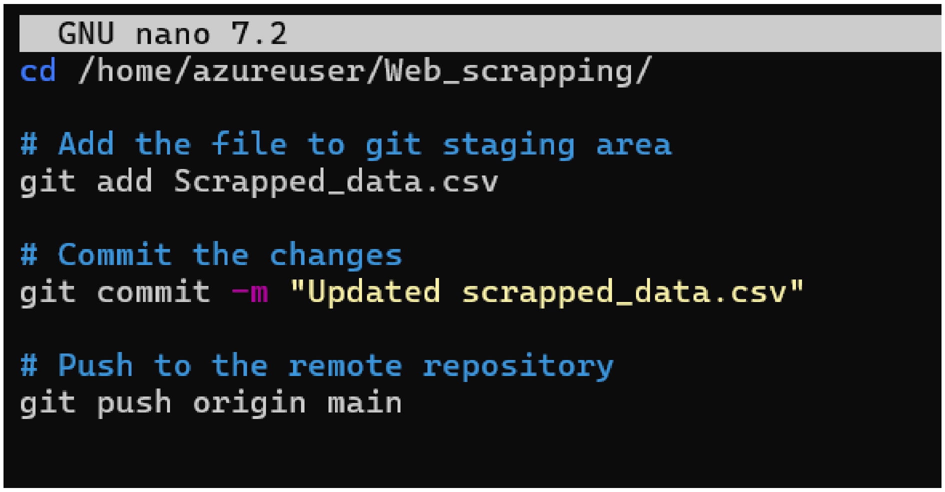
   1) Extracting the links of all the vehicles to be sold and downloading the OuterHTML of location and Details.
      - Used Libraries:
         - Selenium
         - Pandas
      - Stored:
         - OuterHTML:
            a) Stored as html files
            b) Links stored in Scrapped_data_copy.csv

   2) Extracting Details from the stored OuterHTML:
      - Used Libraries:
         - BeautifulSoup
         - Pandas
      - Stored:

        In Csv file names as Scrapped_data.csv

   3) This past is app.py that runs both code one after another where the codes were executed as subprocesses.

- I used Microsoft Azure VM machine to deploy. The code and images are displayed bellow. Commands:
   - Ssh Connection: `ssh -i Task_key.pem azureuser@20.2.80.82`
   - `Sudo apt update`
   - `Sudo apt install git`
   - `pip3 install virtualenv`
   - `git clone https://github.com/Thapa-G/Web_scrapping.git`
   - `cd Web_scrapping`
   - `python3 –m venv venv`
   - `sourse venv/bin/activate`
   - `pip install –r requirement.text`
   - `python3 app.py`
   - `touch logfiles.log`
   - `touch gitpush_files.log`
   - `nano  /home/azureuser/Web_scraapping/push_git.sh`



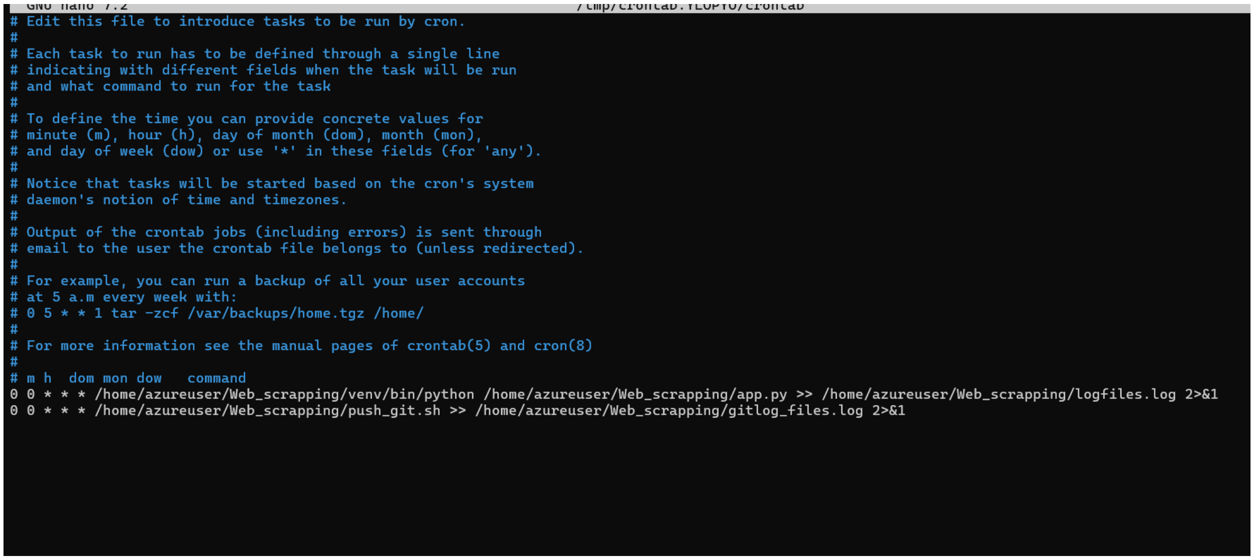
```
  GNU nano 7.2
cd /home/azureuser/Web_scrapping/

# Add the file to git staging area
git add Scrapped_data.csv

# Commit the changes
git commit -m "Updated scrapped_data.csv"

# Push to the remote repository
git push origin main
```

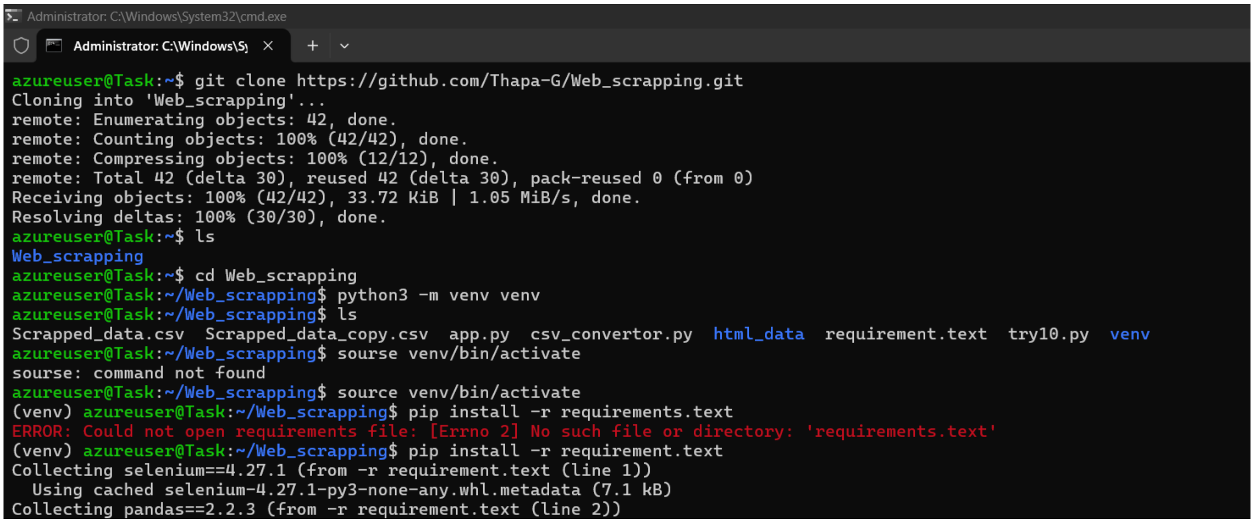
- Chmod +x /home/azureuser/Web_scrapping/push_git.sh
- Crontab –e



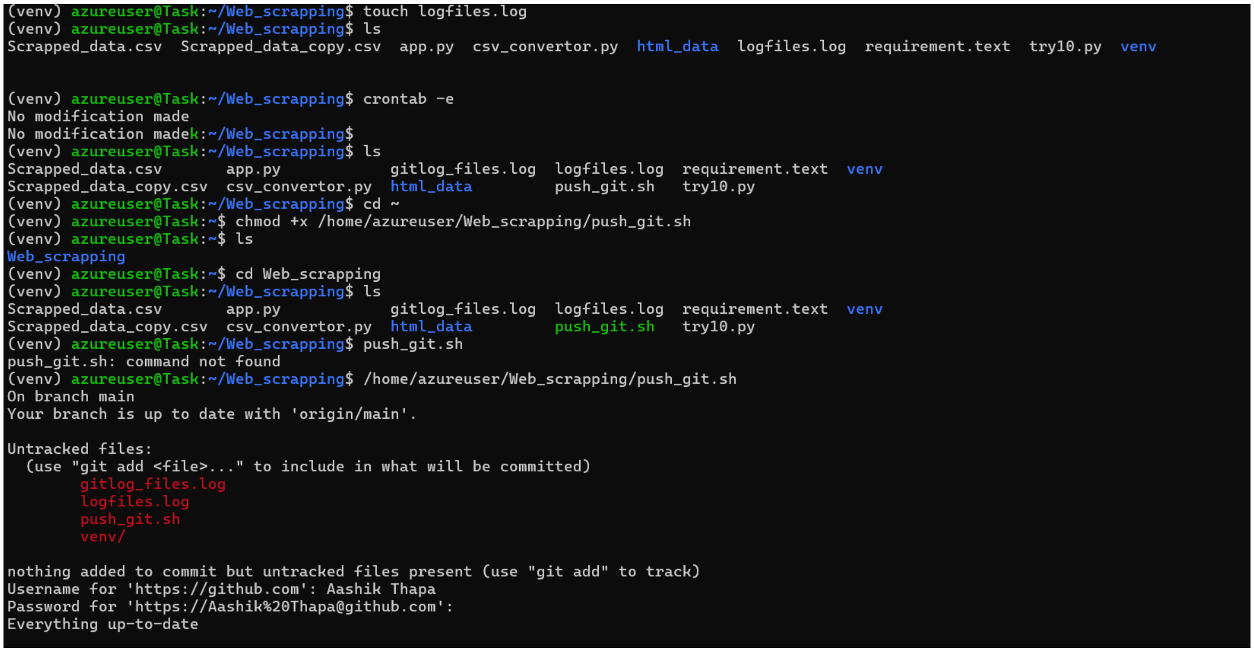
```
  GNU nano 7.2                                                          /tmp/crontab.YLOPYO/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to 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
0 0 * * * /home/azureuser/Web_scrapping/venv/bin/python /home/azureuser/Web_scrapping/app.py >> /home/azureuser/Web_scrapping/logfiles.log 2>&1
0 0 * * * /home/azureuser/Web_scrapping/push_git.sh >> /home/azureuser/Web_scrapping/gitlog_files.log 2>&1
```

I have schedule the **“app.py”** to run every day on mid night.
What it does is, the code runs every day in the midnight which update the Scrapped_data.csv and push the change in the github repository such that it can be viewed by all the members.

Other pic:


Administrator: C:\Windows\System32\cmd.exe
Administrator: C:\Windows\Sy

```
azureuser@Task:~$ git clone https://github.com/Thapa-G/Web_scrapping.git
Cloning into 'Web_scrapping'...
remote: Enumerating objects: 42, done.
remote: Counting objects: 100% (42/42), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 42 (delta 30), reused 42 (delta 30), pack-reused 0 (from 0)
Receiving objects: 100% (42/42), 33.72 KiB | 1.05 MiB/s, done.
Resolving deltas: 100% (30/30), done.
azureuser@Task:~$ ls
Web_scrapping
azureuser@Task:~$ cd Web_scrapping
azureuser@Task:~/Web_scrapping$ python3 -m venv venv
azureuser@Task:~/Web_scrapping$ ls
Scrapped_data.csv  Scrapped_data_copy.csv  app.py  csv_convertor.py  html_data  requirement.text  try10.py  venv
azureuser@Task:~/Web_scrapping$ sourse venv/bin/activate
sourse: command not found
azureuser@Task:~/Web_scrapping$ source venv/bin/activate
(venv) azureuser@Task:~/Web_scrapping$ pip install -r requirements.text
ERROR: Could not open requirements file: [Errno 2] No such file or directory: 'requirements.text'
(venv) azureuser@Task:~/Web_scrapping$ pip install -r requirement.text
Collecting selenium==4.27.1 (from -r requirement.text (line 1))
  Using cached selenium-4.27.1-py3-none-any.whl.metadata (7.1 kB)
Collecting pandas==2.2.3 (from -r requirement.text (line 2))
```



```
(venv) azureuser@Task:~/Web_scrapping$ touch logfiles.log
(venv) azureuser@Task:~/Web_scrapping$ ls
Scrapped_data.csv  Scrapped_data_copy.csv  app.py  csv_convertor.py  html_data  logfiles.log  requirement.text  try10.py  venv


(venv) azureuser@Task:~/Web_scrapping$ crontab -e
No modification made
No modification madek:~/Web_scrapping$
(venv) azureuser@Task:~/Web_scrapping$ ls
Scrapped_data.csv       app.py            gitlog_files.log  logfiles.log  requirement.text  venv
Scrapped_data_copy.csv  csv_convertor.py  html_data         push_git.sh   try10.py
(venv) azureuser@Task:~/Web_scrapping$ cd ~
(venv) azureuser@Task:~$ chmod +x /home/azureuser/Web_scrapping/push_git.sh
(venv) azureuser@Task:~$ ls
Web_scrapping
(venv) azureuser@Task:~$ cd Web_scrapping
(venv) azureuser@Task:~/Web_scrapping$ ls
Scrapped_data.csv       app.py            gitlog_files.log  logfiles.log  requirement.text  venv
Scrapped_data_copy.csv  csv_convertor.py  html_data         push_git.sh   try10.py
(venv) azureuser@Task:~/Web_scrapping$ push_git.sh
push_git.sh: command not found
(venv) azureuser@Task:~/Web_scrapping$ /home/azureuser/Web_scrapping/push_git.sh
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        gitlog_files.log
        logfiles.log
        push_git.sh
        venv/

nothing added to commit but untracked files present (use "git add" to track)
Username for 'https://github.com': Aashik Thapa
Password for 'https://Aashik%20Thapa@github.com':
Everything up-to-date
```