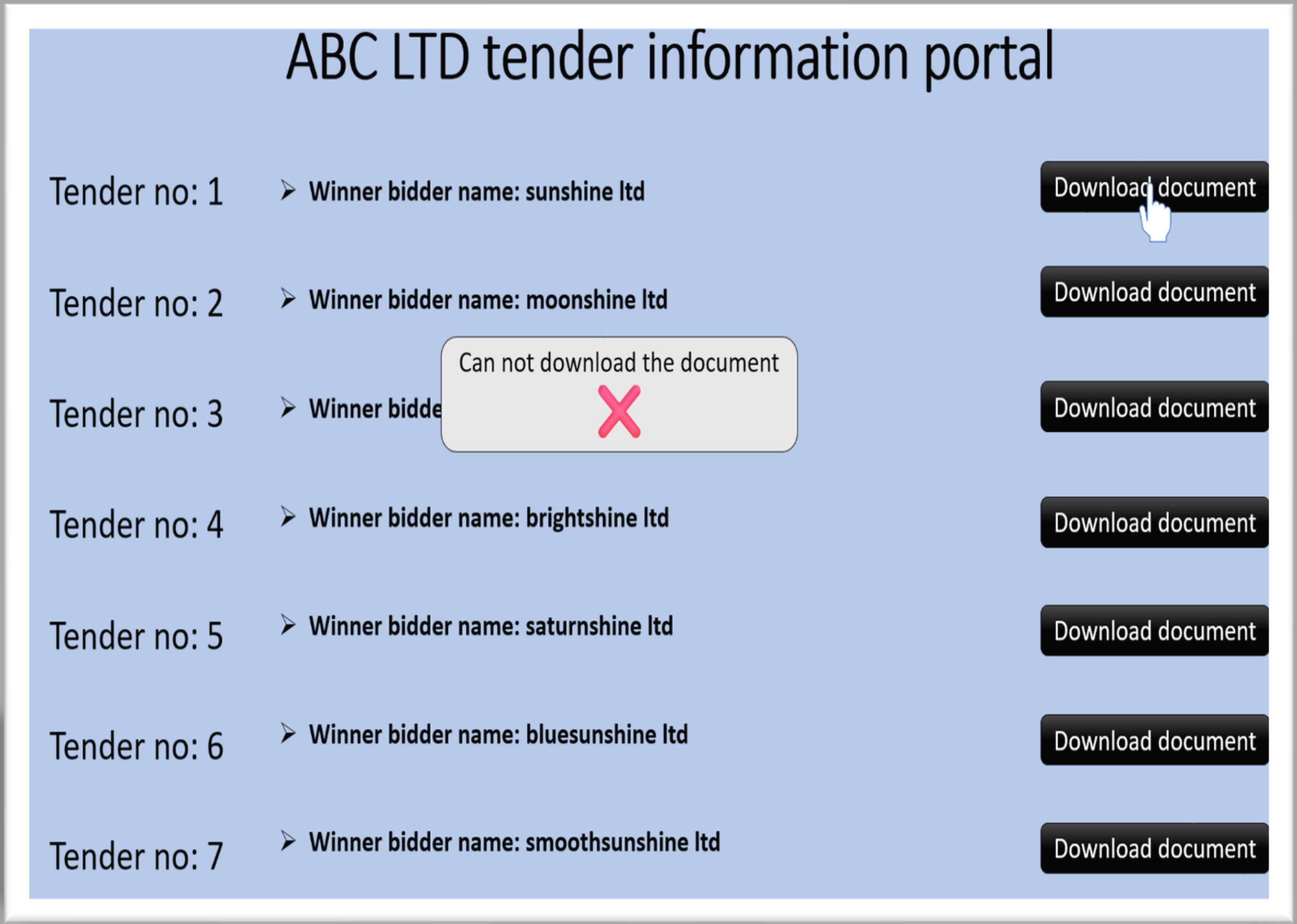
# Problem Statement

ABC Ltd provides tender information services to clients across India.

Clients often report that they can see tender details(winner bidder), can not download the documents and report customer-care.

Employee send documents but process becomes time-consuming when handling multiple tenders or clients, causing delays.



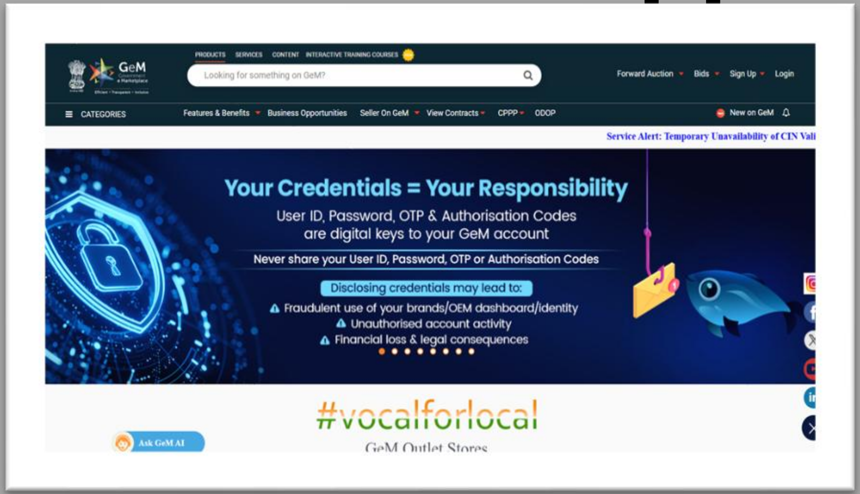
## ABC LTD tender information portal

| | |
|---|---|
| Tender no: 1 | ➢ Winner bidder name: sunshine ltd |
| Tender no: 2 | ➢ Winner bidder name: moonshine ltd |
| Tender no: 3 | ➢ Winner bidde |
| Tender no: 4 | ➢ Winner bidder name: brightshine ltd |
| Tender no: 5 | ➢ Winner bidder name: saturnshine ltd |
| Tender no: 6 | ➢ Winner bidder name: bluesunshine ltd |
| Tender no: 7 | ➢ Winner bidder name: smoothsunshine ltd |

Can not download the document ✗

# Process of customer support employee

**Step 1** open the gem site



Service Alert: Temporary Unavailability of CIN Vali

Your Credentials = Your Responsibility

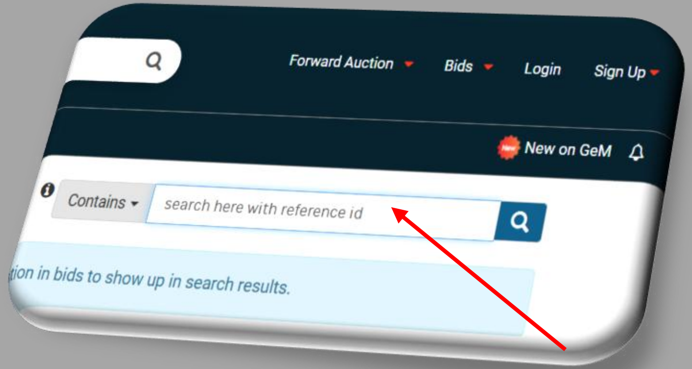
User ID, Password, OTP & Authorisation Codes are digital keys to your GeM account

Never share your User ID, Password, OTP or Authorisation Codes

Disclosing credentials may lead to:

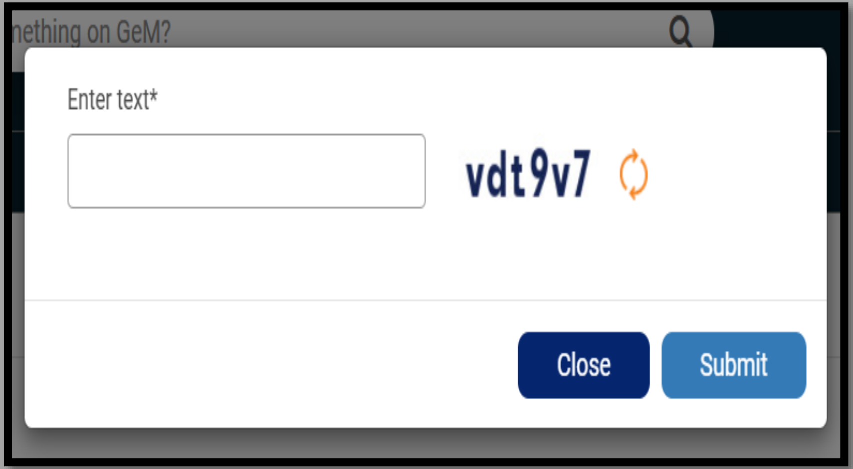
- Fraudulent use of your brands/OEM dashboard/identity
- Unauthorised account activity
- Financial loss & legal consequences

**Step 2** fill the reference id

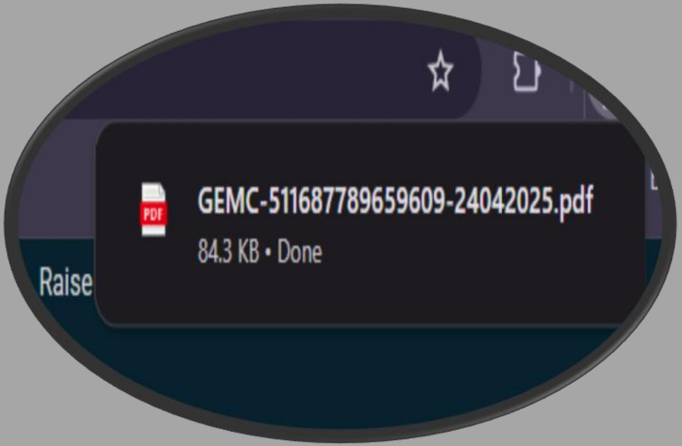


ion in bids to show up in search results.

**Step 3** solve the captcha



**Step 4** download document



# What is the solution?

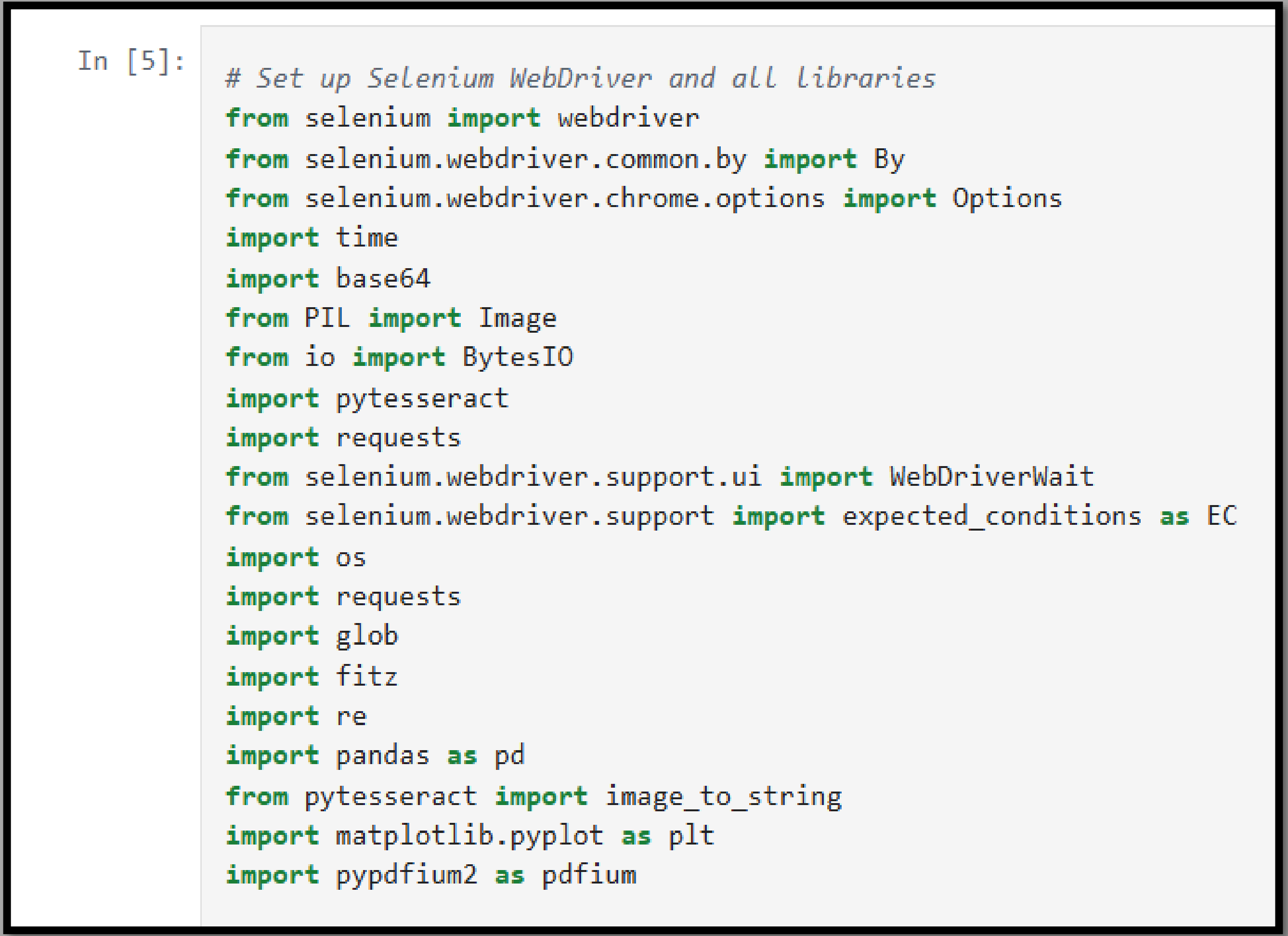
To avoid this lengthy process, what if you paste each reference id and run the code and you will get each tender documents downloaded automatically and save in your selected folder

**How it is possible?**

**Let’s understand this step by step with my python code**

# Important libraries

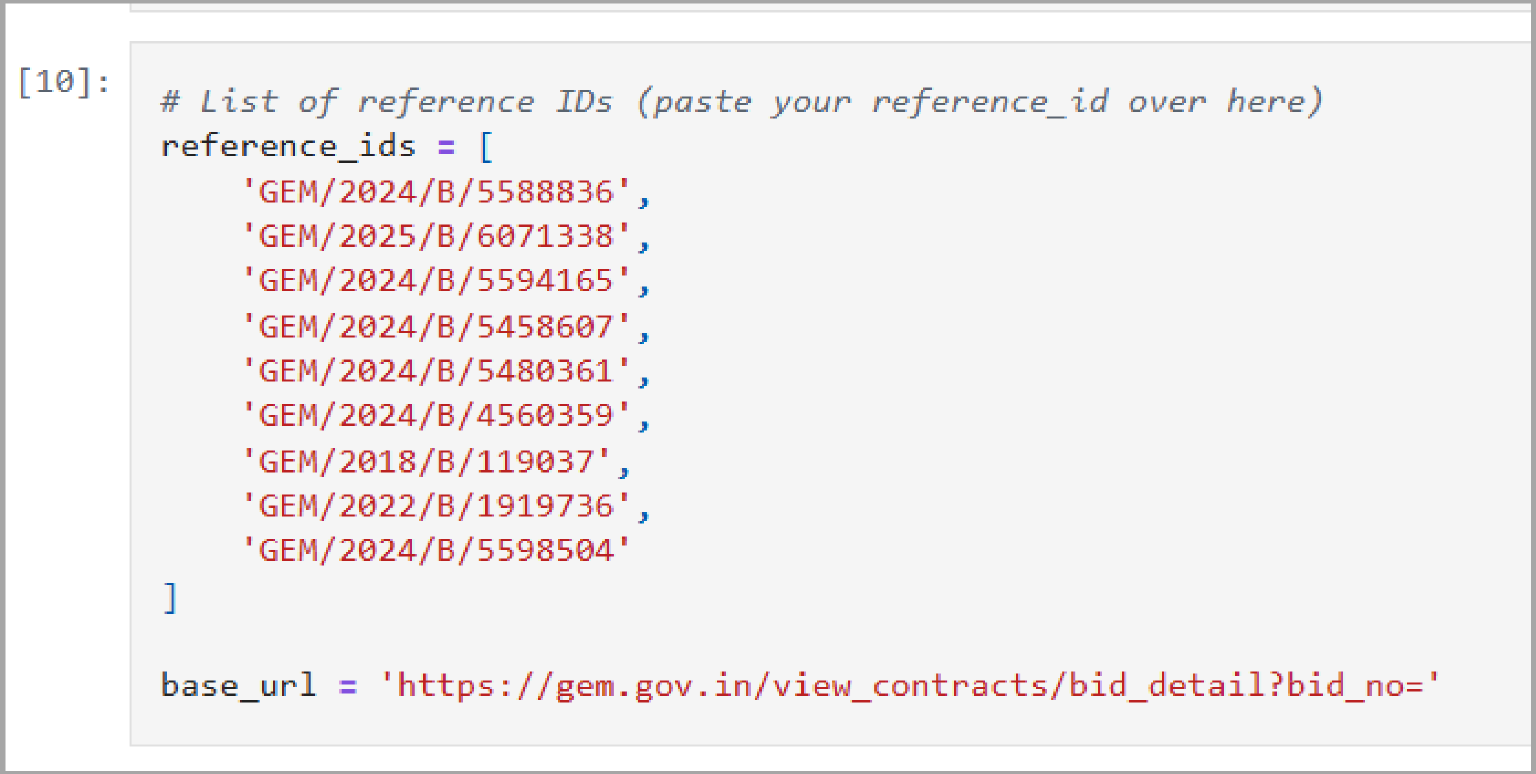
in web scrapping we will selenium and chrome web-driver, pytesseract is  a free captcha solving option which is call ocr method.



```
# Set up Selenium WebDriver and all libraries
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
import time
import base64
from PIL import Image
from io import BytesIO
import pytesseract
import requests
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import os
import requests
import glob
import fitz
import re
import pandas as pd
from pytesseract import image_to_string
import matplotlib.pyplot as plt
import pypdfium2 as pdfium
```

# This is where we paste tender no/reference id

We will paste comma separated reference id for each tender

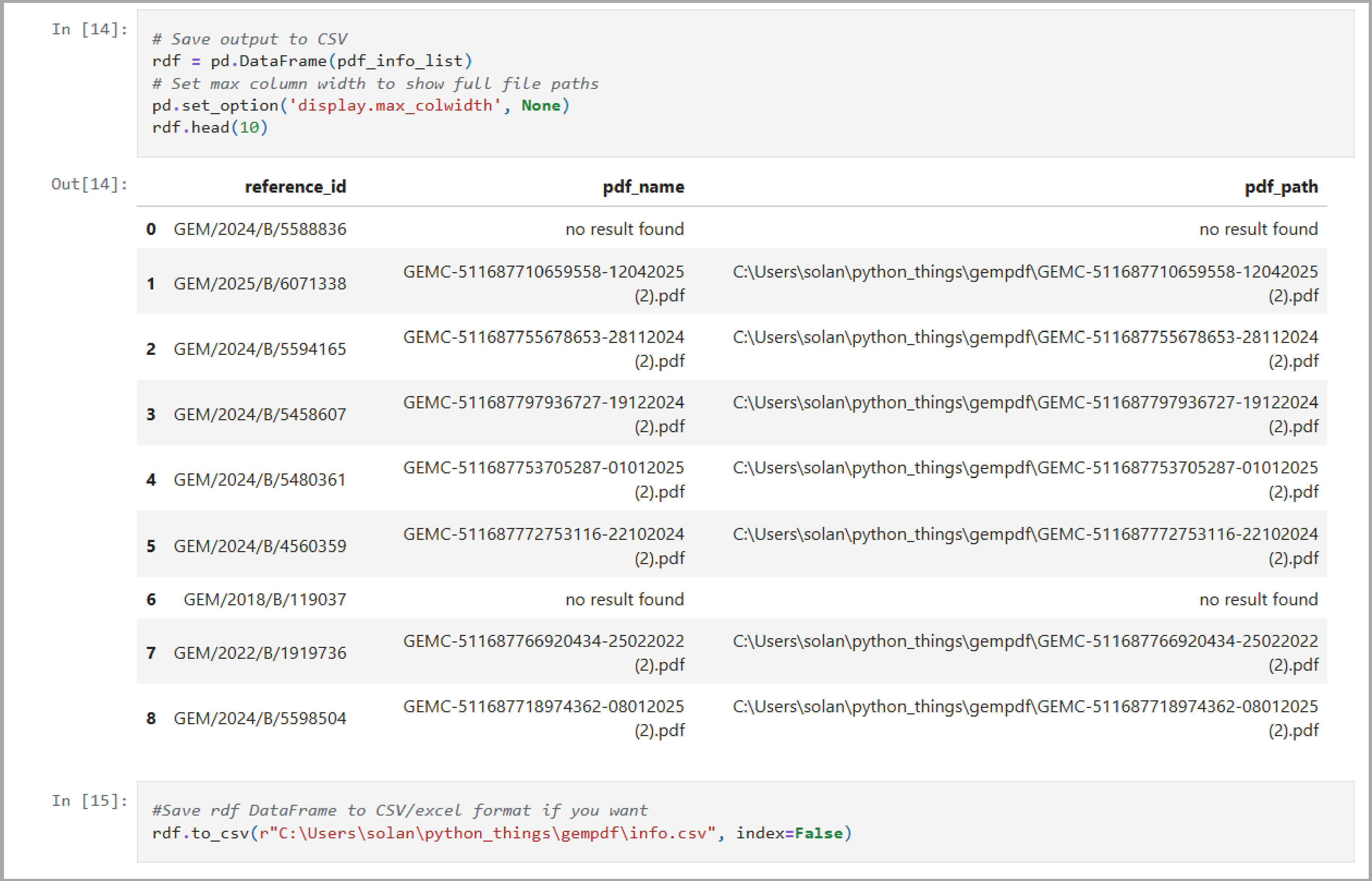


```python
# List of reference IDs (paste your reference_id over here)
reference_ids = [
    'GEM/2024/B/5588836',
    'GEM/2025/B/6071338',
    'GEM/2024/B/5594165',
    'GEM/2024/B/5458607',
    'GEM/2024/B/5480361',
    'GEM/2024/B/4560359',
    'GEM/2018/B/119037',
    'GEM/2022/B/1919736',
    'GEM/2024/B/5598504'
]

base_url = 'https://gem.gov.in/view_contracts/bid_detail?bid_no='
```

# Output

This is how the output looks like and we will see the location of each document that we downloaded in local device.



In [14]:

```
# Save output to CSV
rdf = pd.DataFrame(pdf_info_list)
# Set max column width to show full file paths
pd.set_option('display.max_colwidth', None)
rdf.head(10)
```

Out[14]:

| | reference_id | pdf_name | pdf_path |
|---|---|---|---|
| **0** | GEM/2024/B/5588836 | no result found | no result found |
| **1** | GEM/2025/B/6071338 | GEMC-511687710659558-12042025 (2).pdf | C:\Users\solan\python_things\gempdf\GEMC-511687710659558-12042025 (2).pdf |
| **2** | GEM/2024/B/5594165 | GEMC-511687755678653-28112024 (2).pdf | C:\Users\solan\python_things\gempdf\GEMC-511687755678653-28112024 (2).pdf |
| **3** | GEM/2024/B/5458607 | GEMC-511687797936727-19122024 (2).pdf | C:\Users\solan\python_things\gempdf\GEMC-511687797936727-19122024 (2).pdf |
| **4** | GEM/2024/B/5480361 | GEMC-511687753705287-01012025 (2).pdf | C:\Users\solan\python_things\gempdf\GEMC-511687753705287-01012025 (2).pdf |
| **5** | GEM/2024/B/4560359 | GEMC-511687772753116-22102024 (2).pdf | C:\Users\solan\python_things\gempdf\GEMC-511687772753116-22102024 (2).pdf |
| **6** | GEM/2018/B/119037 | no result found | no result found |
| **7** | GEM/2022/B/1919736 | GEMC-511687766920434-25022022 (2).pdf | C:\Users\solan\python_things\gempdf\GEMC-511687766920434-25022022 (2).pdf |
| **8** | GEM/2024/B/5598504 | GEMC-511687718974362-08012025 (2).pdf | C:\Users\solan\python_things\gempdf\GEMC-511687718974362-08012025 (2).pdf |

In [15]:

```
#Save rdf DataFrame to CSV/excel format if you want
rdf.to_csv(r"C:\Users\solan\python_things\gempdf\info.csv", index=False)
```

# In what situation we won’t get the documents

- on gem web-site, document officially not available
- Tender is in process, the department won’t publish the winner bidder yet
- By the department, tender is officially get cancel
- Re-publish the tender means RA stage