Lab6

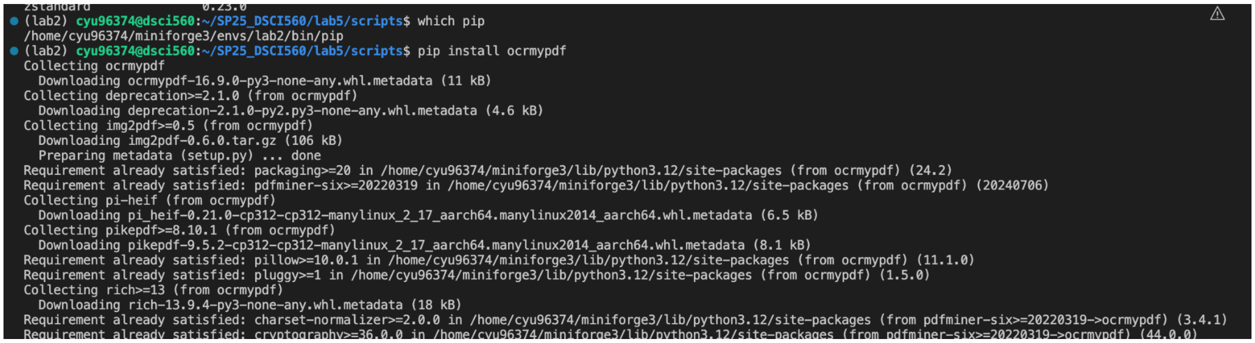
1. Team Details

Name	USC ID
Chenxiao Yu	6024079123
Yiqing Hong	4395913002

2. Github link:
https://github.com/AiChiMoCha/SP25_DSCI560/tree/main/lab6

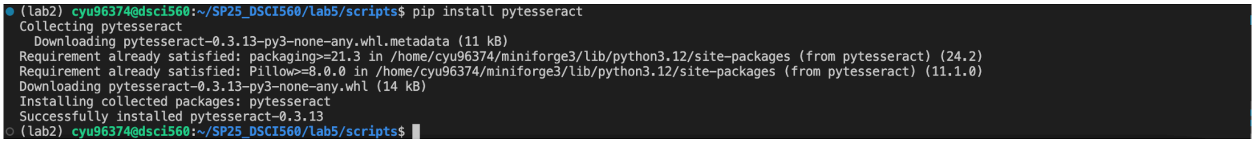
Env settings

- **ocrmypdf**



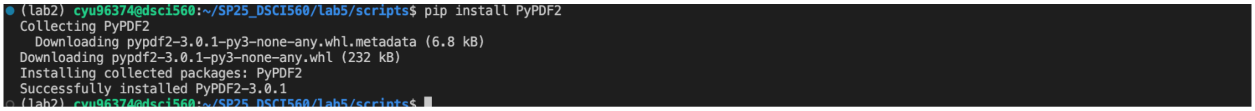
```
zstandard               0.23.0
● (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ which pip
/home/cyu96374/miniforge3/envs/lab2/bin/pip
● (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ pip install ocrmypdf
Collecting ocrmypdf
  Downloading ocrmypdf-16.9.0-py3-none-any.whl.metadata (11 kB)
Collecting deprecation>=2.1.0 (from ocrmypdf)
  Downloading deprecation-2.1.0-py2.py3-none-any.whl.metadata (4.6 kB)
Collecting img2pdf>=0.5 (from ocrmypdf)
  Downloading img2pdf-0.6.0.tar.gz (106 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: packaging>=20 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from ocrmypdf) (24.2)
Requirement already satisfied: pdfminer-six>=20220319 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from ocrmypdf) (20240706)
Collecting pi-heif (from ocrmypdf)
  Downloading pi_heif-0.21.0-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl.metadata (6.5 kB)
Collecting pikepdf>=8.10.1 (from ocrmypdf)
  Downloading pikepdf-9.5.2-cp312-cp312-manylinux_2_17_aarch64.manylinux2014_aarch64.whl.metadata (8.1 kB)
Requirement already satisfied: pillow>=10.0.1 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from ocrmypdf) (11.1.0)
Requirement already satisfied: pluggy>=1 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from ocrmypdf) (1.5.0)
Collecting rich>=13 (from ocrmypdf)
  Downloading rich-13.9.4-py3-none-any.whl.metadata (18 kB)
Requirement already satisfied: charset-normalizer>=2.0.0 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from pdfminer-six>=20220319->ocrmypdf) (3.4.1)
Requirement already satisfied: cryptography>=36.0.0 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from pdfminer-six>=20220319->ocrmypdf) (44.0.0)
```

- PyPDF2



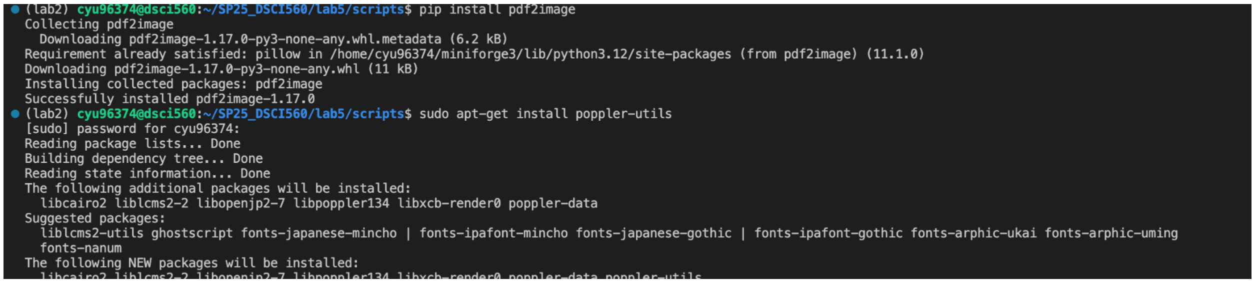
```
● (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ pip install pytesseract
Collecting pytesseract
  Downloading pytesseract-0.3.13-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: packaging>=21.3 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from pytesseract) (24.2)
Requirement already satisfied: Pillow>=8.0.0 in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from pytesseract) (11.1.0)
Downloading pytesseract-0.3.13-py3-none-any.whl (14 kB)
Installing collected packages: pytesseract
Successfully installed pytesseract-0.3.13
○ (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ █
```

- pytesseract



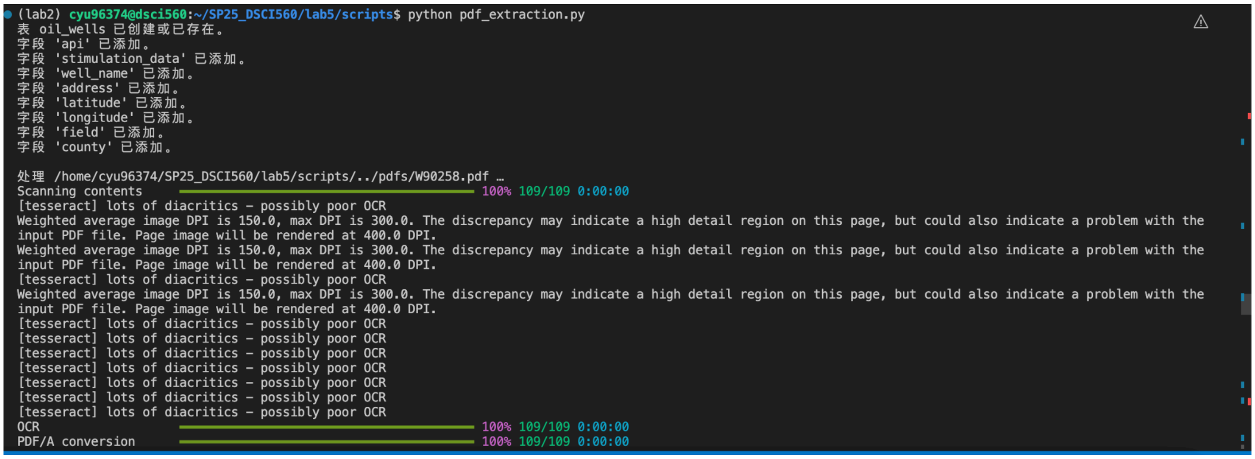
```
● (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ pip install PyPDF2
Collecting PyPDF2
  Downloading pypdf2-3.0.1-py3-none-any.whl.metadata (6.8 kB)
Downloading pypdf2-3.0.1-py3-none-any.whl (232 kB)
Installing collected packages: PyPDF2
Successfully installed PyPDF2-3.0.1
○ (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ █
```

- poppler & pdf2image



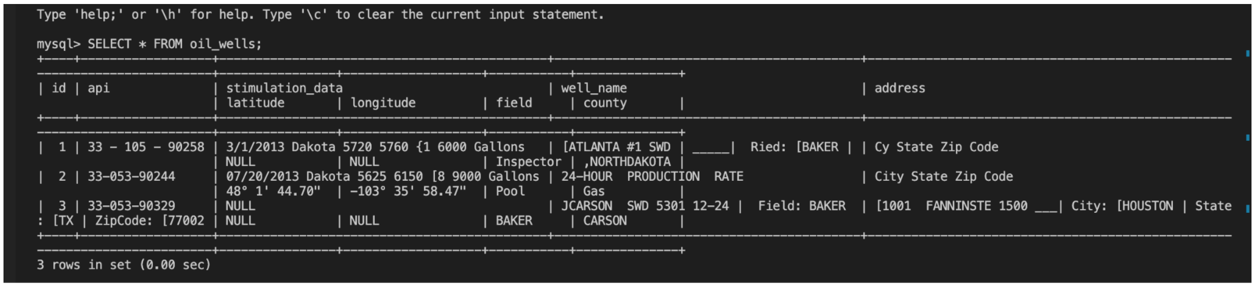
```
● (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ pip install pdf2image
Collecting pdf2image
  Downloading pdf2image-1.17.0-py3-none-any.whl.metadata (6.2 kB)
Requirement already satisfied: pillow in /home/cyu96374/miniforge3/lib/python3.12/site-packages (from pdf2image) (11.1.0)
Downloading pdf2image-1.17.0-py3-none-any.whl (11 kB)
Installing collected packages: pdf2image
Successfully installed pdf2image-1.17.0
● (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ sudo apt-get install poppler-utils
[sudo] password for cyu96374:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcairo2 liblcms2-2 libopenjp2-7 libpoppler134 libxcb-render0 poppler-data
Suggested packages:
  liblcms2-utils ghostscript fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming
  fonts-nanum
The following NEW packages will be installed:
  libcairo2 liblcms2-2 libopenjp2-7 libpoppler134 libxcb-render0 poppler-data poppler-utils
```

PDF Extraction



```
● (lab2) cyu96374@dsci560:~/SP25_DSCI560/lab5/scripts$ python pdf_extraction.py
  表 oil_wells 已创建或已存在。
  字段 'api' 已添加。
  字段 'stimulation_data' 已添加。
  字段 'well_name' 已添加。
  字段 'address' 已添加。
  字段 'latitude' 已添加。
  字段 'longitude' 已添加。
  字段 'field' 已添加。
  字段 'county' 已添加。

  处理 /home/cyu96374/SP25_DSCI560/lab5/scripts/../pdfs/W90258.pdf …
  Scanning contents     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 109/109 0:00:00
  [tesseract] lots of diacritics - possibly poor OCR
  Weighted average image DPI is 150.0, max DPI is 300.0. The discrepancy may indicate a high detail region on this page, but could also indicate a problem with the
  input PDF file. Page image will be rendered at 400.0 DPI.
  Weighted average image DPI is 150.0, max DPI is 300.0. The discrepancy may indicate a high detail region on this page, but could also indicate a problem with the
  input PDF file. Page image will be rendered at 400.0 DPI.
  [tesseract] lots of diacritics - possibly poor OCR
  Weighted average image DPI is 150.0, max DPI is 300.0. The discrepancy may indicate a high detail region on this page, but could also indicate a problem with the
  input PDF file. Page image will be rendered at 400.0 DPI.
  [tesseract] lots of diacritics - possibly poor OCR
  [tesseract] lots of diacritics - possibly poor OCR
  [tesseract] lots of diacritics - possibly poor OCR
  [tesseract] lots of diacritics - possibly poor OCR
  [tesseract] lots of diacritics - possibly poor OCR
  [tesseract] lots of diacritics - possibly poor OCR
  [tesseract] lots of diacritics - possibly poor OCR
  OCR                   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 109/109 0:00:00
  PDF/A conversion      ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 109/109 0:00:00
```

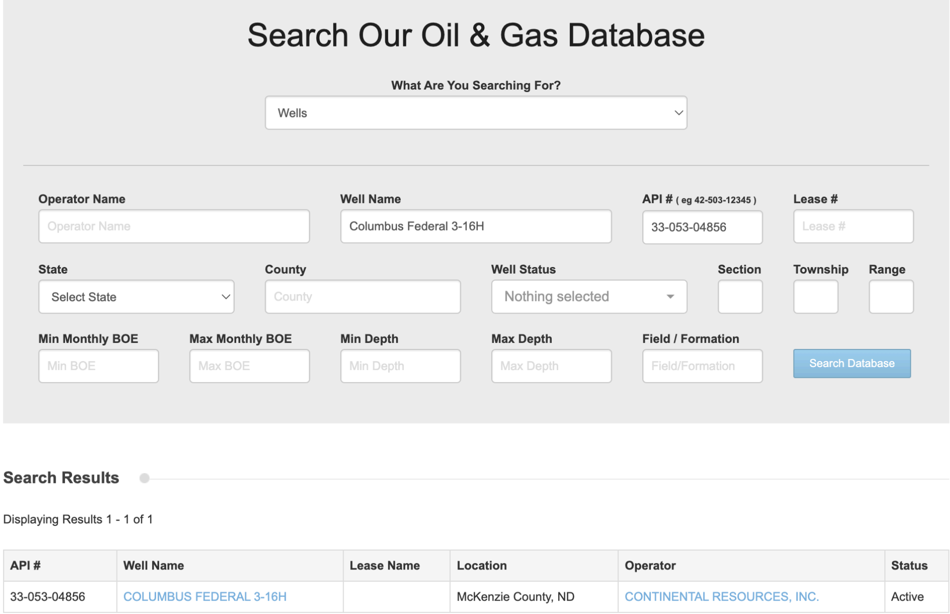
stored info into databases



```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SELECT * FROM oil_wells;
+----+--------------------+------------------------------------------+-----------------------------------------------+------------------------------------------------------------
---------------------+----------------+--------------------+-----------+-------------+
| id | api                | stimulation_data                         | well_name                                     | address
                     | latitude       | longitude          | field     | county      |
+----+--------------------+------------------------------------------+-----------------------------------------------+------------------------------------------------------------
---------------------+----------------+--------------------+-----------+-------------+
|  1 | 33 - 105 - 90258 | 3/1/2013 Dakota 5720 5760 {1 6000 Gallons   | [ATLANTA #1 SWD | _____|  Ried: [BAKER | | Cy State Zip Code
                     | NULL           | NULL               | Inspector | ,NORTHDAKOTA |
|  2 | 33-053-90244       | 07/20/2013 Dakota 5625 6150 [8 9000 Gallons | 24-HOUR  PRODUCTION  RATE                  | City State Zip Code
                     | 48° 1' 44.70"  | -103° 35' 58.47"   | Pool      | Gas         |
|  3 | 33-053-90329       | NULL                                     | JCARSON  SWD 5301 12-24 |  Field: BAKER   | [1001  FANNINSTE 1500 ___| City: [HOUSTON | State
: [TX | ZipCode: [77002 | NULL           | NULL               | BAKER     | CARSON      |
+----+--------------------+------------------------------------------+-----------------------------------------------+------------------------------------------------------------
---------------------+----------------+--------------------+-----------+-------------+
3 rows in set (0.00 sec)
```

Additional Web Scraped Information

Now we have all of the API numbers and well names. We can use them to make a search query on https://www.drillingedge.com/search



Search Our Oil & Gas Database

What Are You Searching For?

Wells

Operator Name: Operator Name

Well Name: Columbus Federal 3-16H

API # (eg 42-503-12345): 33-053-04856

Lease #: Lease #

State: Select State

County: County

Well Status: Nothing selected

Section:

Township:

Range:

Min Monthly BOE: Min BOE

Max Monthly BOE: Max BOE

Min Depth: Min Depth

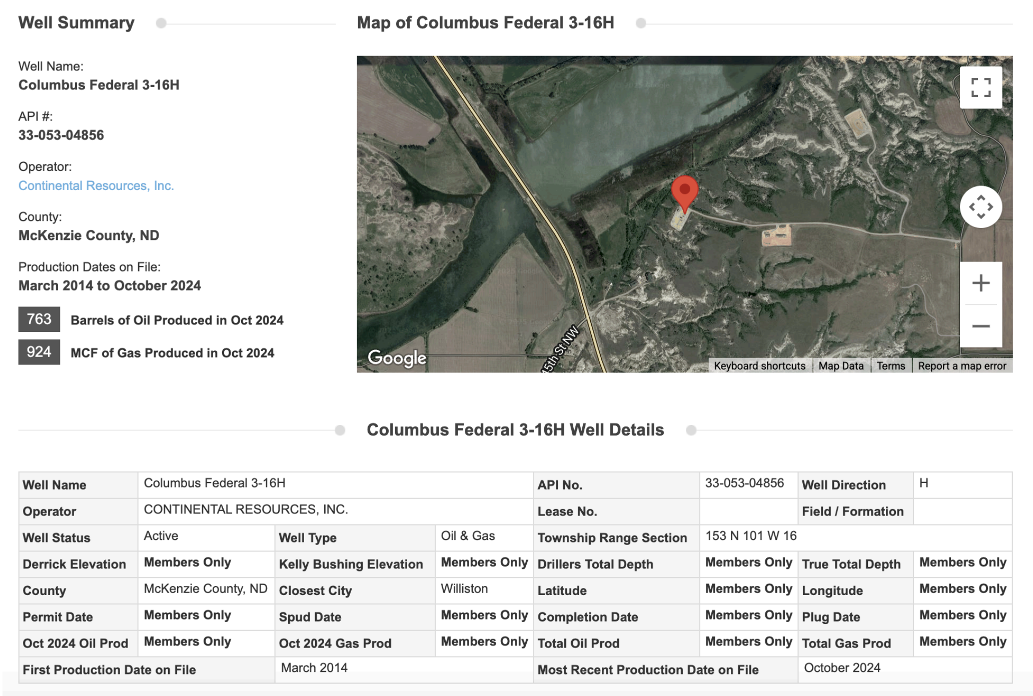
Max Depth: Max Depth

Field / Formation: Field/Formation

Search Results

Displaying Results 1 - 1 of 1

API #	Well Name	Lease Name	Location	Operator	Status
33-053-04856	COLUMBUS FEDERAL 3-16H		McKenzie County, ND	CONTINENTAL RESOURCES, INC.	Active



Well Summary

Well Name:
Columbus Federal 3-16H

API #:
33-053-04856

Operator:
Continental Resources, Inc.

County:
McKenzie County, ND

Production Dates on File:
March 2014 to October 2024

763 **Barrels of Oil Produced in Oct 2024**

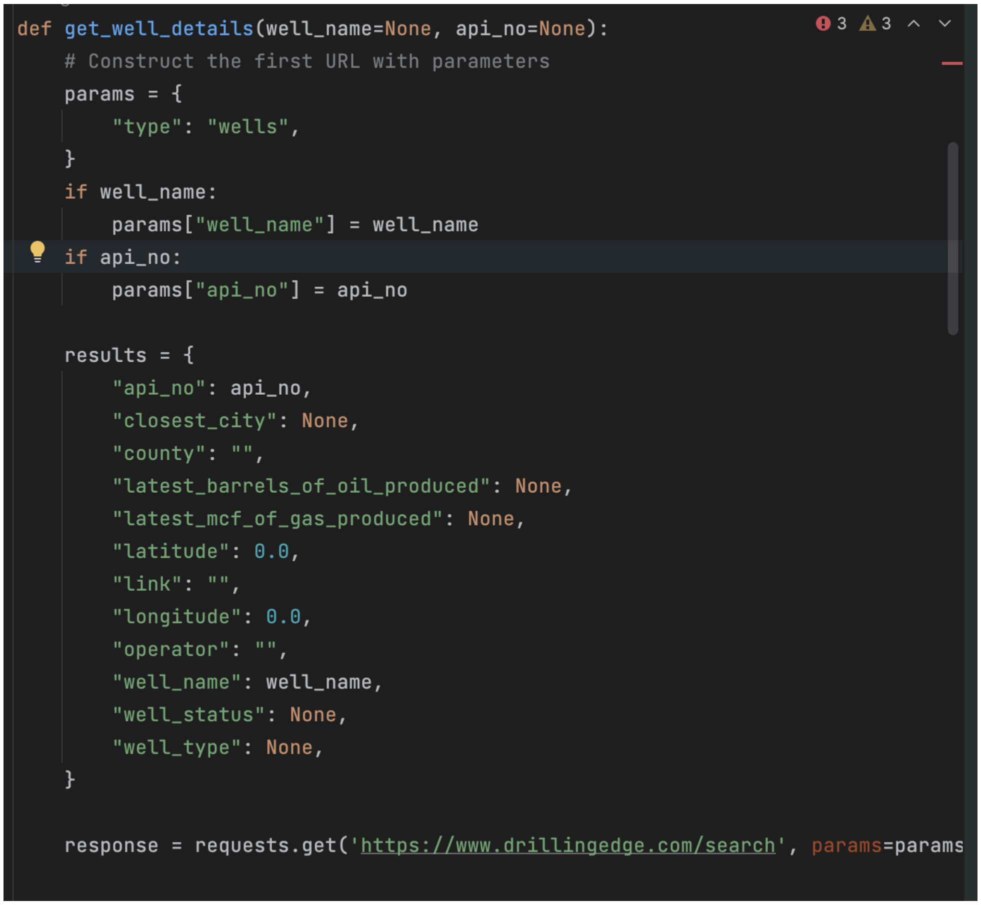
924 **MCF of Gas Produced in Oct 2024**

Map of Columbus Federal 3-16H

Columbus Federal 3-16H Well Details

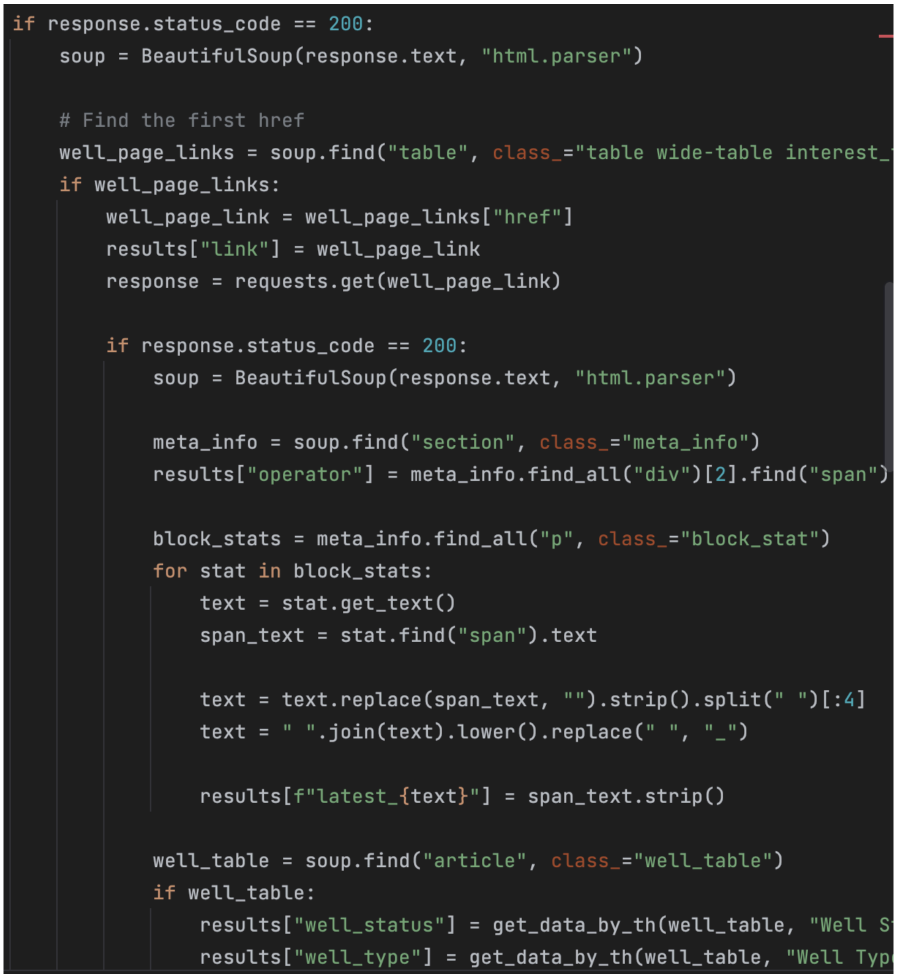
Well Name	Columbus Federal 3-16H			**API No.**	33-053-04856	**Well Direction**	H
Operator	CONTINENTAL RESOURCES, INC.			**Lease No.**		**Field / Formation**	
Well Status	Active	**Well Type**	Oil & Gas	**Township Range Section**	153 N 101 W 16		
Derrick Elevation	**Members Only**	**Kelly Bushing Elevation**	**Members Only**	**Drillers Total Depth**	**Members Only**	**True Total Depth**	**Members Only**
County	McKenzie County, ND	**Closest City**	Williston	**Latitude**	**Members Only**	**Longitude**	**Members Only**
Permit Date	**Members Only**	**Spud Date**	**Members Only**	**Completion Date**	**Members Only**	**Plug Date**	**Members Only**
Oct 2024 Oil Prod	**Members Only**	**Oct 2024 Gas Prod**	**Members Only**	**Total Oil Prod**	**Members Only**	**Total Gas Prod**	**Members Only**
First Production Date on File		March 2014		**Most Recent Production Date on File**		October 2024	

Then we use requests in Python to mimic this query and get the additional information we need.



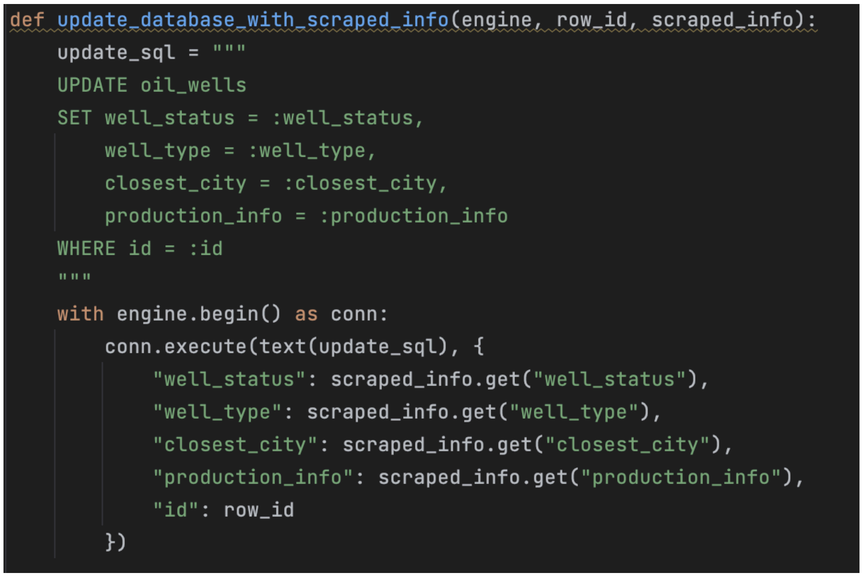
```
def get_well_details(well_name=None, api_no=None):
    # Construct the first URL with parameters
    params = {
        "type": "wells",
    }
    if well_name:
        params["well_name"] = well_name
    if api_no:
        params["api_no"] = api_no

    results = {
        "api_no": api_no,
        "closest_city": None,
        "county": "",
        "latest_barrels_of_oil_produced": None,
        "latest_mcf_of_gas_produced": None,
        "latitude": 0.0,
        "link": "",
        "longitude": 0.0,
        "operator": "",
        "well_name": well_name,
        "well_status": None,
        "well_type": None,
    }

    response = requests.get('https://www.drillingedge.com/search', params=params
```

And we use beautifulsoup to preprocess the data we fetched.

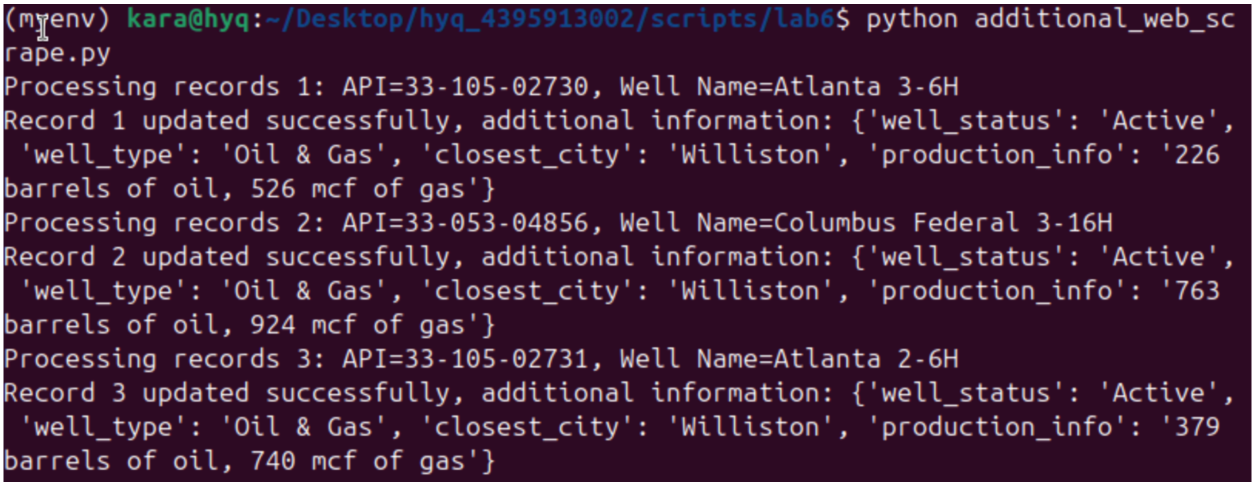


```
if response.status_code == 200:
    soup = BeautifulSoup(response.text, "html.parser")

    # Find the first href
    well_page_links = soup.find("table", class_="table wide-table interest_
    if well_page_links:
        well_page_link = well_page_links["href"]
        results["link"] = well_page_link
        response = requests.get(well_page_link)

        if response.status_code == 200:
            soup = BeautifulSoup(response.text, "html.parser")

            meta_info = soup.find("section", class_="meta_info")
            results["operator"] = meta_info.find_all("div")[2].find("span")

            block_stats = meta_info.find_all("p", class_="block_stat")
            for stat in block_stats:
                text = stat.get_text()
                span_text = stat.find("span").text

                text = text.replace(span_text, "").strip().split(" ")[:4]
                text = " ".join(text).lower().replace(" ", "_")

                results[f"latest_{text}"] = span_text.strip()

            well_table = soup.find("article", class_="well_table")
            if well_table:
                results["well_status"] = get_data_by_th(well_table, "Well S
                results["well_type"] = get_data_by_th(well_table, "Well Typ
```

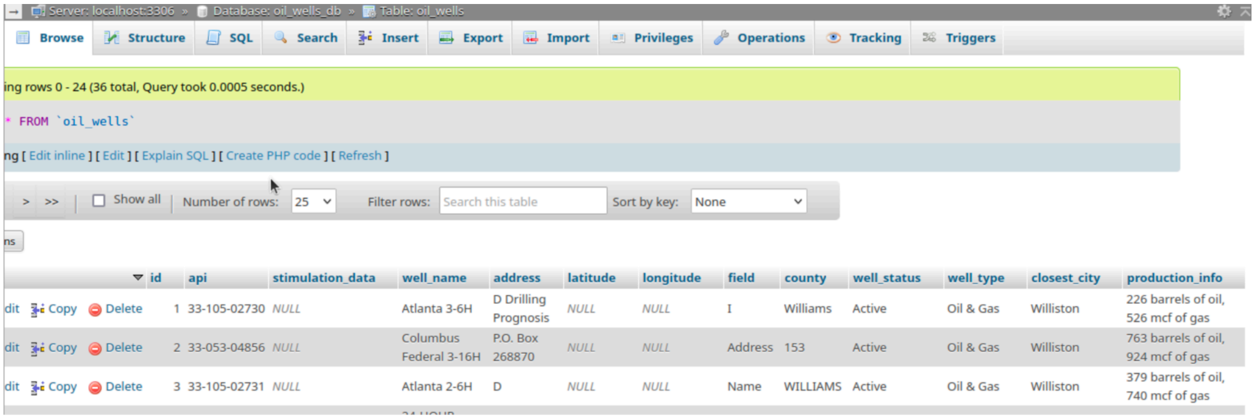
Finally, we update the mysql database with the new information we got.



```
def update_database_with_scraped_info(engine, row_id, scraped_info):
    update_sql = """
    UPDATE oil_wells
    SET well_status = :well_status,
        well_type = :well_type,
        closest_city = :closest_city,
        production_info = :production_info
    WHERE id = :id
    """
    with engine.begin() as conn:
        conn.execute(text(update_sql), {
            "well_status": scraped_info.get("well_status"),
            "well_type": scraped_info.get("well_type"),
            "closest_city": scraped_info.get("closest_city"),
            "production_info": scraped_info.get("production_info"),
            "id": row_id
        })
```



```
(myenv) kara@hyq:~/Desktop/hyq_4395913002/scripts/lab6$ python additional_web_sc
rape.py
Processing records 1: API=33-105-02730, Well Name=Atlanta 3-6H
Record 1 updated successfully, additional information: {'well_status': 'Active',
 'well_type': 'Oil & Gas', 'closest_city': 'Williston', 'production_info': '226
barrels of oil, 526 mcf of gas'}
Processing records 2: API=33-053-04856, Well Name=Columbus Federal 3-16H
Record 2 updated successfully, additional information: {'well_status': 'Active',
 'well_type': 'Oil & Gas', 'closest_city': 'Williston', 'production_info': '763
barrels of oil, 924 mcf of gas'}
Processing records 3: API=33-105-02731, Well Name=Atlanta 2-6H
Record 3 updated successfully, additional information: {'well_status': 'Active',
 'well_type': 'Oil & Gas', 'closest_city': 'Williston', 'production_info': '379
barrels of oil, 740 mcf of gas'}
```



ing rows 0 - 24 (36 total, Query took 0.0005 seconds.)

```
* FROM `oil_wells`
```

ng [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

> >> | □ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

ns

			id	api	stimulation_data	well_name	address	latitude	longitude	field	county	well_status	well_type	closest_city	production_info
dit	Copy	Delete	1	33-105-02730	NULL	Atlanta 3-6H	D Drilling Prognosis	NULL	NULL	I	Williams	Active	Oil & Gas	Williston	226 barrels of oil, 526 mcf of gas
dit	Copy	Delete	2	33-053-04856	NULL	Columbus Federal 3-16H	P.O. Box 268870	NULL	NULL	Address	153	Active	Oil & Gas	Williston	763 barrels of oil, 924 mcf of gas
dit	Copy	Delete	3	33-105-02731	NULL	Atlanta 2-6H	D	NULL	NULL	Name	WILLIAMS	Active	Oil & Gas	Williston	379 barrels of oil, 740 mcf of gas