  
  
it took almost one hour to run the code script -1

C:\Users\CJP\PycharmProjects\PythonProject2\.venv\Scripts\python.exe C:\Users\CJP\PycharmProjects\PythonProject2\script2.py

2024-12-12 19:57:54,824 - INFO - Found credentials in shared credentials file: ~/.aws/credentials

2024-12-12 19:57:55,211 - INFO - Starting process...

2024-12-12 19:57:55,211 - INFO - Attempting to read S3 file Input\_data/Suspected\_Product\_Brand\_Name/drug\_names.txt from bucket cvp-2-bucket...

2024-12-12 19:57:56,541 - INFO - Successfully read S3 file Input\_data/Suspected\_Product\_Brand\_Name/drug\_names.txt from bucket cvp-2-bucket.

2024-12-12 19:57:56,543 - INFO - Parsing drug names...

2024-12-12 19:57:56,543 - INFO - Parsed 2 drug names.

2024-12-12 19:57:56,545 - INFO - Attempting to read S3 file Input\_data/report\_id\_database/report\_drug.txt from bucket cvp-2-bucket...

2024-12-12 19:57:56,888 - INFO - Successfully read S3 file Input\_data/report\_id\_database/report\_drug.txt from bucket cvp-2-bucket.

2024-12-12 20:25:07,731 - INFO - Finding REPORT\_IDs for 2 drug names...

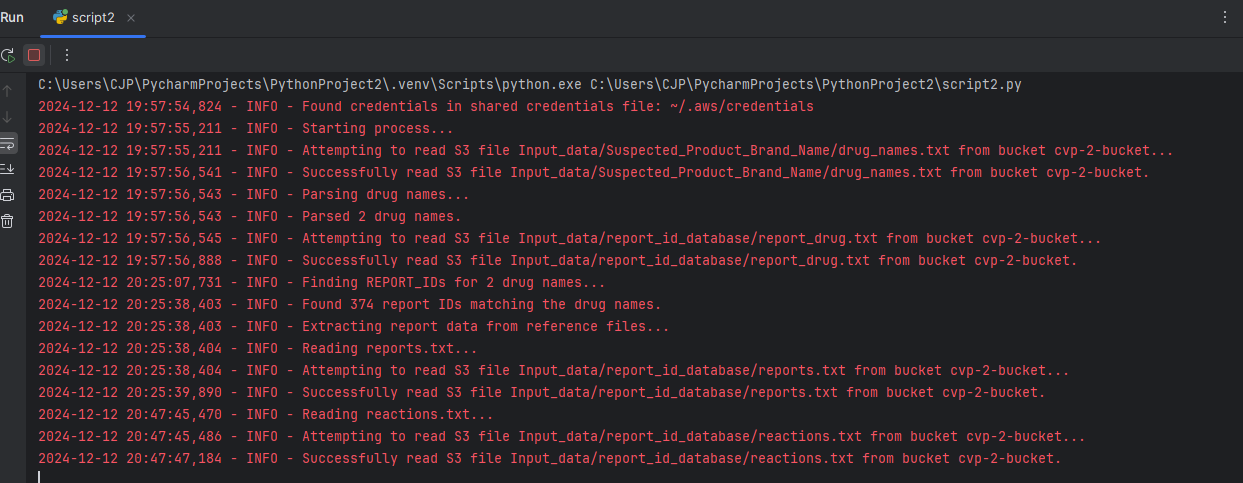
2024-12-12 20:25:38,403 - INFO - Found 374 report IDs matching the drug names.

2024-12-12 20:25:38,403 - INFO - Extracting report data from reference files...

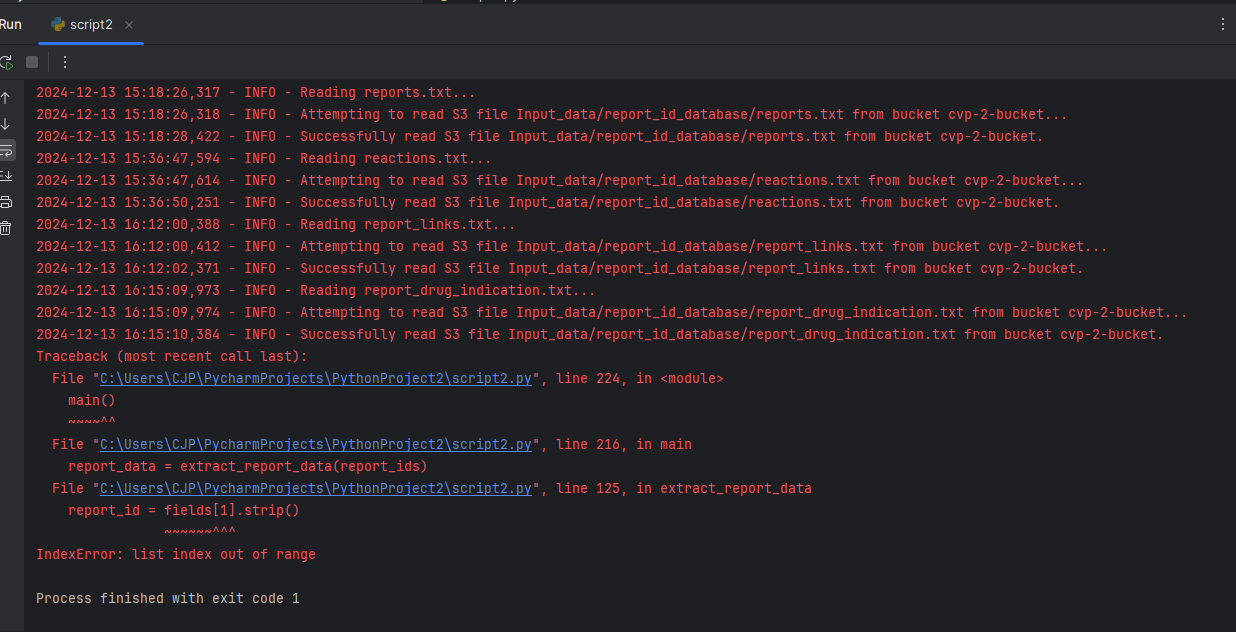
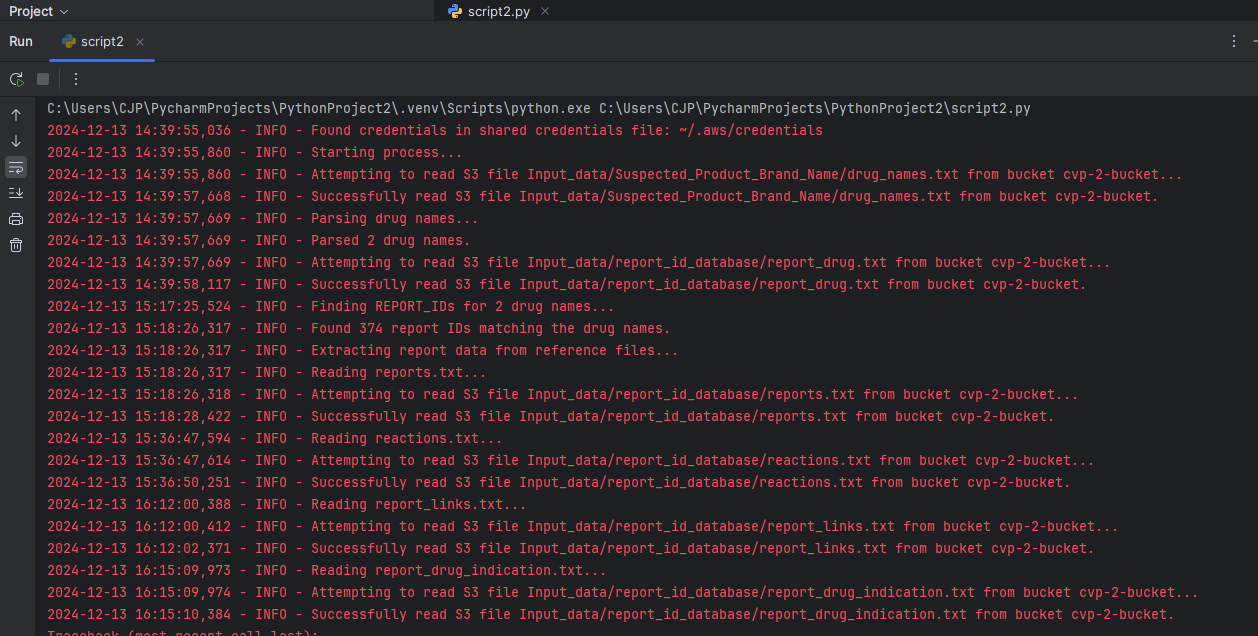
2024-12-12 20:25:38,404 - INFO - Reading reports.txt...

2024-12-12 20:25:38,404 - INFO - Attempting to read S3 file Input\_data/report\_id\_database/reports.txt from bucket cvp-2-bucket...

2024-12-12 20:25:39,890 - INFO - Successfully read S3 file Input\_data/report\_id\_database/reports.txt from bucket cvp-2-bucket.

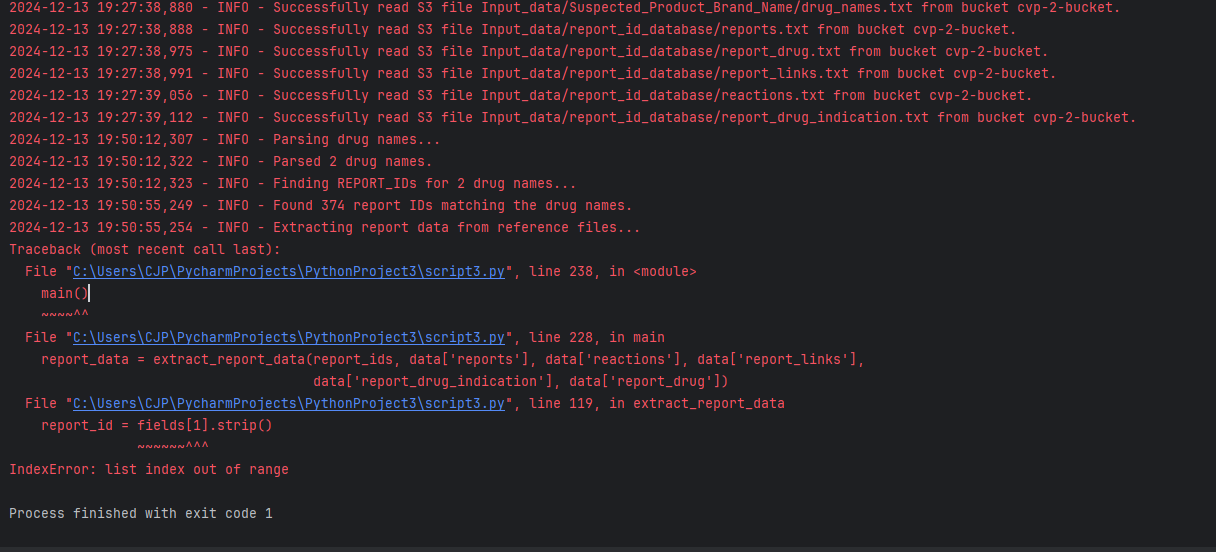


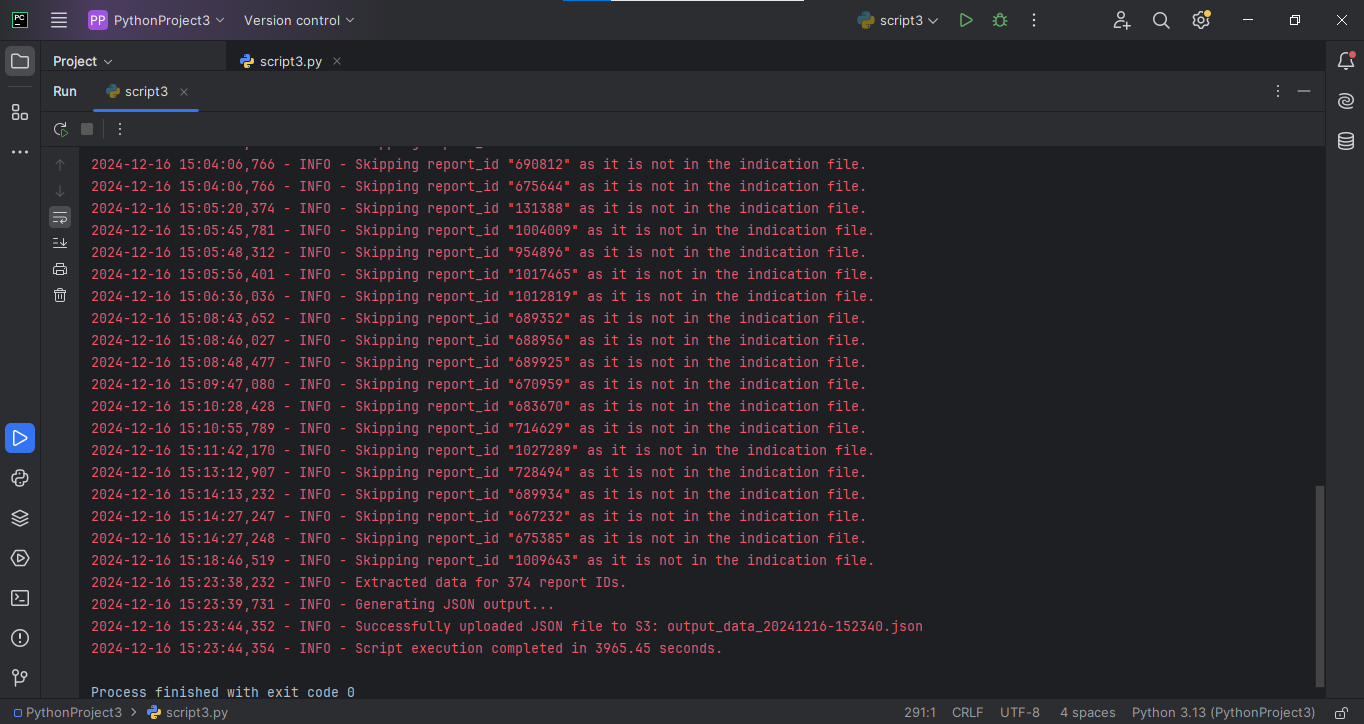
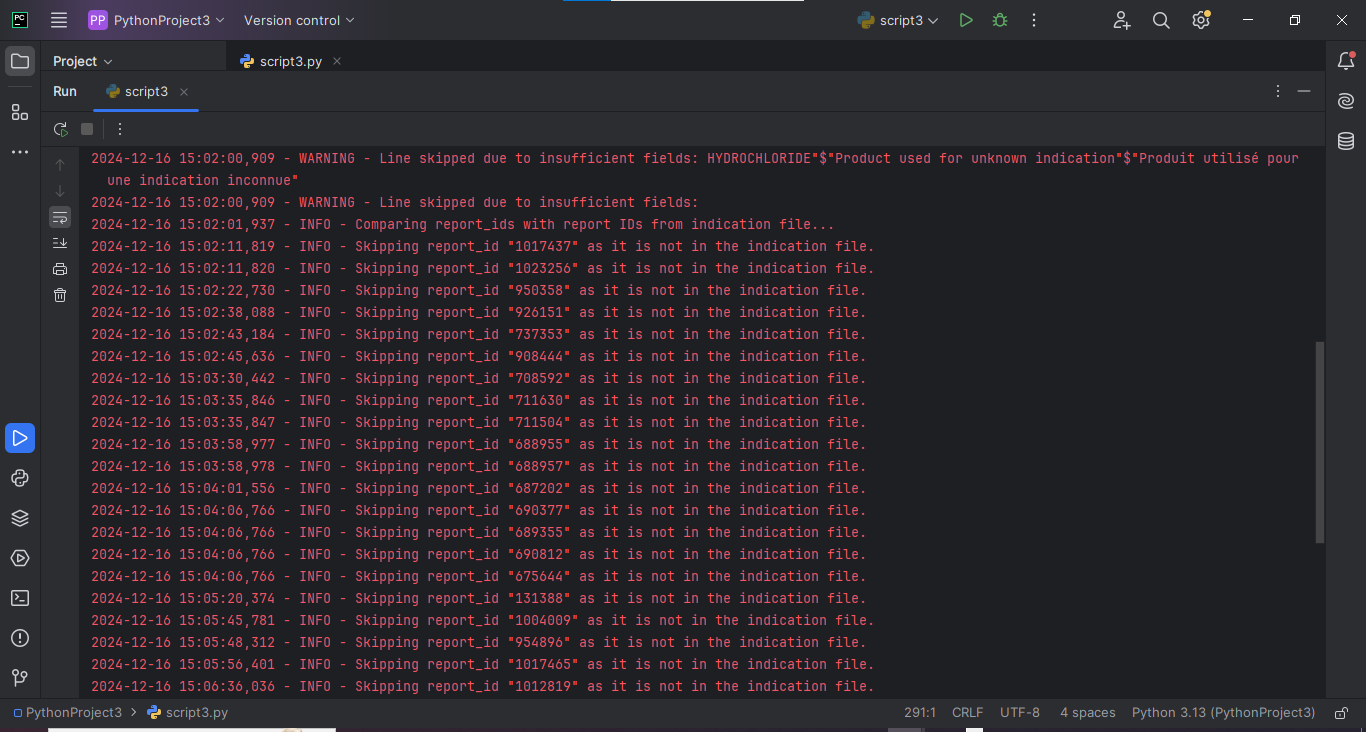
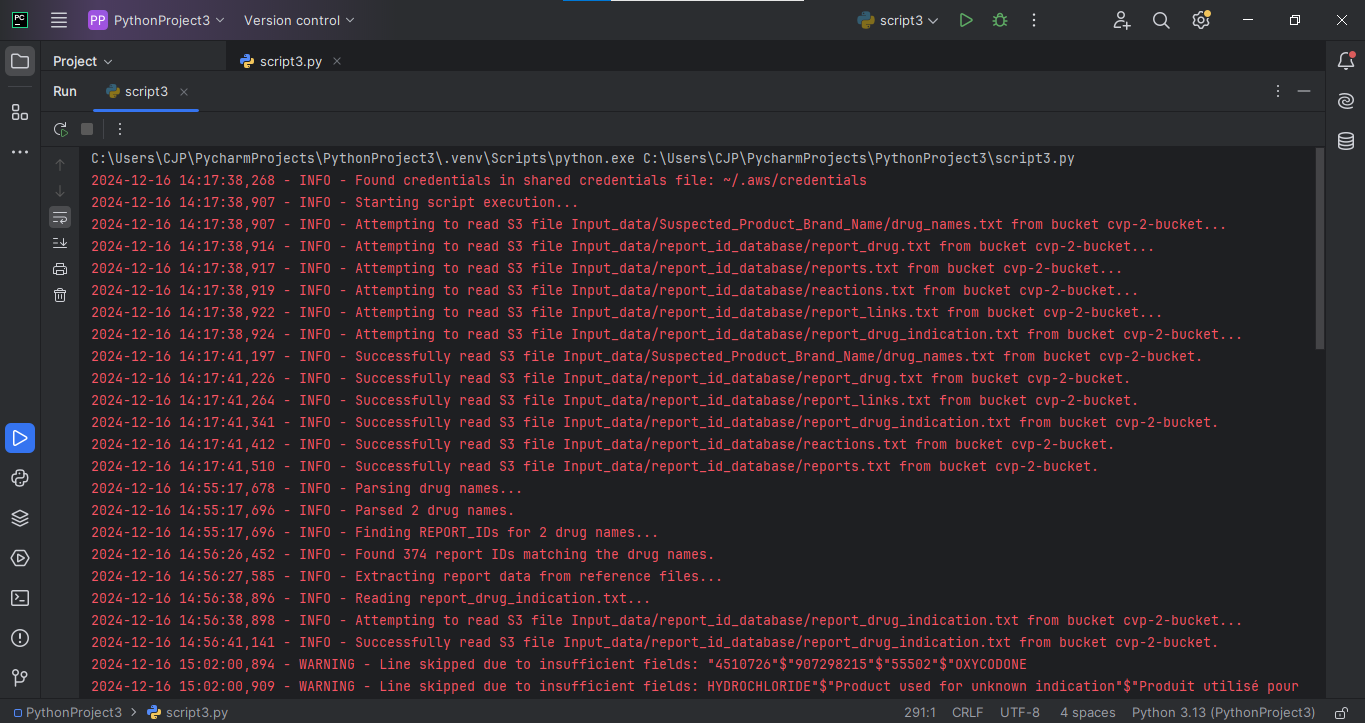
import boto3 import json import logging from collections import defaultdict import time  
  
# Initialize logging and S3 client  
logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')  
s3\_client = boto3.client('s3')  
  
# Input and output S3 buckets  
input\_bucket = 'cvp-2-bucket'  
output\_bucket = 'cvp-2-output'  
  
# File paths in S3  
drug\_names\_file = 'Input\_data/Suspected\_Product\_Brand\_Name/drug\_names.txt'  
report\_drug\_file = 'Input\_data/report\_id\_database/report\_drug.txt'  
reports\_file = 'Input\_data/report\_id\_database/reports.txt'  
reactions\_file = 'Input\_data/report\_id\_database/reactions.txt'  
report\_links\_file = 'Input\_data/report\_id\_database/report\_links.txt'  
report\_drug\_indication\_file = 'Input\_data/report\_id\_database/report\_drug\_indication.txt'  
  
  
# Function to read files from S3  
def read\_s3\_file(bucket, key):  
 try:  
 logging.info(f"Attempting to read S3 file {key} from bucket {bucket}...")  
 response = s3\_client.get\_object(Bucket=bucket, Key=key)  
 logging.info(f"Successfully read S3 file {key} from bucket {bucket}.")  
 return response['Body'].read().decode('utf-8').splitlines()  
 except Exception as e:  
 logging.error(f"Error reading S3 file {key} from bucket {bucket}: {e}")  
 return []  
  
  
# Step 1: Parse drug names from file  
def parse\_drug\_names(file\_content):  
 logging.info("Parsing drug names...")  
 drug\_names = []  
 for line in file\_content:  
 line = line.strip().lower() # Case-insensitive parsing  
 if line:  
 drug\_names.append(line)  
 logging.info(f"Parsed {len(drug\_names)} drug names.")  
 return drug\_names  
  
  
# Step 2: Locate REPORT\_IDs corresponding to drug names  
def find\_report\_ids(drug\_names, report\_drug\_content):  
 logging.info(f"Finding REPORT\_IDs for {len(drug\_names)} drug names...")  
 report\_ids = defaultdict(list)  
 for line in report\_drug\_content:  
 fields = line.split('$')  
 if len(fields) > 1:  
 drug\_name = fields[3].strip().lower()  
 report\_id = fields[1].strip()  
 if any(drug\_name in line.lower() for drug\_name in drug\_names):  
 report\_ids[report\_id].append(fields)  
 logging.info(f"Found {len(report\_ids)} report IDs matching the drug names.")  
 return report\_ids  
  
  
# Step 3: Extract data from reference files based on REPORT\_ID  
def extract\_report\_data(report\_ids):  
 logging.info("Extracting report data from reference files...")  
 report\_data = {}  
  
 # Read reports.txt  
 logging.info("Reading reports.txt...")  
 reports\_content = read\_s3\_file(input\_bucket, reports\_file)  
 for line in reports\_content:  
 fields = line.split('$')  
 if len(fields) > 1:  
 report\_id = fields[0].strip()  
 if report\_id in report\_ids:  
 report\_data[report\_id] = {  
 'report\_no': fields[1],  
 'version\_no': fields[2],  
 'datintreceived': fields[4],  
 'datreceived': fields[3],  
 'source\_eng': fields[37],  
 'mah\_no': fields[5],  
 'report\_type\_eng': fields[7],  
 'reporter\_type\_eng': fields[34],  
 'seriousness\_eng': fields[26],  
 'death': fields[28],  
 'disability': fields[29],  
 'congenital\_anomaly': fields[30],  
 'life\_threatening': fields[31],  
 'hospitalization': fields[32],  
 'other\_medically\_imp\_cond': fields[33],  
 'age': fields[12],  
 'gender\_eng': fields[10],  
 'height': fields[22],  
 'weight': fields[19],  
 'outcome\_eng': fields[17]  
 }  
  
 # Read reactions.txt  
 logging.info("Reading reactions.txt...")  
 reactions\_content = read\_s3\_file(input\_bucket, reactions\_file)  
 for line in reactions\_content:  
 fields = line.split('$')  
 report\_id = fields[1].strip()  
 if report\_id in report\_ids:  
 report\_data[report\_id]['reaction\_eng'] = fields[3]  
 report\_data[report\_id]['version'] = fields[10]  
 report\_data[report\_id]['duration'] = fields[9]  
  
 # Read report\_links.txt and add data to report\_data  
 logging.info("Reading report\_links.txt...")  
 report\_links\_content = read\_s3\_file(input\_bucket, report\_links\_file)  
 for line in report\_links\_content:  
 fields = line.split('$')  
 report\_id = fields[1].strip()  
 if report\_id in report\_ids:  
 report\_data[report\_id]['link\_type\_eng'] = fields[1]  
 report\_data[report\_id]['e2b\_report\_no'] = fields[2]  
  
 # Read report\_drug\_indication.txt and add data to report\_data  
 logging.info("Reading report\_drug\_indication.txt...")  
 report\_drug\_indication\_content = read\_s3\_file(input\_bucket, report\_drug\_indication\_file)  
 for line in report\_drug\_indication\_content:  
 fields = line.split('$')  
 report\_id = fields[1].strip()  
 if report\_id in report\_ids:  
 report\_data[report\_id]['indication\_eng'] = fields[8]  
  
 # Read report\_drug.txt and add data to report\_data  
 logging.info("Reading report\_drug.txt...")  
 report\_drug\_content = read\_s3\_file(input\_bucket, report\_drug\_file)  
 for line in report\_drug\_content:  
 fields = line.split('$')  
 if len(fields) > 1:  
 report\_id = fields[1].strip()  
 if report\_id in report\_ids:  
 # Add the new data fields from report\_drug.txt  
 report\_data[report\_id]['drug\_name\_eng'] = fields[3] # DRUG\_NAME\_ENG  
 report\_data[report\_id]['drug\_type\_eng'] = fields[5] # DRUG\_TYPE\_ENG  
 report\_data[report\_id]['dose\_unit\_eng'] = fields[8] # DOSE\_UNIT\_ENG  
 report\_data[report\_id]['route\_eng'] = fields[16] # ROUTE\_ENG  
 report\_data[report\_id]['dose'] = fields[7] # DOSE  
 report\_data[report\_id]['freq'] = fields[10] # FREQ  
 report\_data[report\_id]['therapy\_duration'] = fields[13] # DURATION  
  
 logging.info(f"Extracted data for {len(report\_data)} report IDs.")  
 return report\_data  
  
  
# Step 4: Generate the JSON structure and save it to the output S3 bucket  
def generate\_json\_output(report\_data):  
 logging.info("Generating JSON output...")  
 final\_data = []  
 for report\_id, data in report\_data.items():  
 final\_data.append({  
 "report\_id": report\_id,  
 "report\_no": data.get('report\_no', ''),  
 "version\_no": data.get('version\_no', ''),  
 "datintreceived": data.get('datintreceived', ''),  
 "datreceived": data.get('datreceived', ''),  
 "source\_eng": data.get('source\_eng', ''),  
 "mah\_no": data.get('mah\_no', ''),  
 "report\_type\_eng": data.get('report\_type\_eng', ''),  
 "reporter\_type\_eng": data.get('reporter\_type\_eng', ''),  
 "seriousness\_eng": data.get('seriousness\_eng', ''),  
 "death": data.get('death', ''),  
 "disability": data.get('disability', ''),  
 "congenital\_anomaly": data.get('congenital\_anomaly', ''),  
 "life\_threatening": data.get('life\_threatening', ''),  
 "hospitalization": data.get('hospitalization', ''),  
 "other\_medically\_imp\_cond": data.get('other\_medically\_imp\_cond', ''),  
 "age": data.get('age', ''),  
 "gender\_eng": data.get('gender\_eng', ''),  
 "height": data.get('height', ''),  
 "weight": data.get('weight', ''),  
 "outcome\_eng": data.get('outcome\_eng', ''),  
 "reaction\_eng": data.get('reaction\_eng', ''),  
 "version": data.get('version', ''),  
 "duration": data.get('duration', ''),  
 "link\_type\_eng": data.get('link\_type\_eng', ''),  
 "e2b\_report\_no": data.get('e2b\_report\_no', ''),  
 "drug\_name\_eng": data.get('drug\_name\_eng', ''),  
 "drug\_type\_eng": data.get('drug\_type\_eng', ''),  
 "dose\_unit\_eng": data.get('dose\_unit\_eng', ''),  
 "route\_eng": data.get('route\_eng', ''),  
 "dose": data.get('dose', ''),  
 "freq": data.get('freq', ''),  
 "therapy\_duration": data.get('therapy\_duration', ''),  
 "indication\_eng": data.get('indication\_eng', '')  
 })  
  
 try:  
 json\_data = json.dumps(final\_data, indent=4)  
 timestamp = time.strftime('%Y%m%d-%H%M%S')  
 output\_file\_key = f'output\_{timestamp}.json'  
 logging.info(f"Uploading JSON data to S3 as {output\_file\_key}...")  
 s3\_client.put\_object(Bucket=output\_bucket, Key=output\_file\_key, Body=json\_data)  
 logging.info(f"Successfully uploaded JSON data to S3 as {output\_file\_key}.")  
 except Exception as e:  
 logging.error(f"Error uploading JSON data to S3: {e}")  
  
  
# Main function  
def main():  
 logging.info("Starting process...")  
  
 # Step 1: Read and parse drug names  
 drug\_names\_content = read\_s3\_file(input\_bucket, drug\_names\_file)  
 drug\_names = parse\_drug\_names(drug\_names\_content)  
  
 # Step 2: Find report IDs for the parsed drug names  
 report\_drug\_content = read\_s3\_file(input\_bucket, report\_drug\_file)  
 report\_ids = find\_report\_ids(drug\_names, report\_drug\_content)  
  
 # Step 3: Extract data for each report ID  
 report\_data = extract\_report\_data(report\_ids)  
  
 # Step 4: Generate JSON and upload to S3  
 generate\_json\_output(report\_data)  
  
 logging.info("Process completed.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()



**Script3**

C:\Users\CJP\PycharmProjects\PythonProject3\.venv\Scripts\python.exe C:\Users\CJP\PycharmProjects\PythonProject3\script3.py 2024-12-13 15:48:09,873 - INFO - Found credentials in shared credentials file: ~/.aws/credentials 2024-12-13 15:48:10,246 - INFO - Attempting to read S3 file Input\_data/Suspected\_Product\_Brand\_Name/drug\_names.txt from bucket cvp-2-bucket... 2024-12-13 15:48:10,251 - INFO - Attempting to read S3 file Input\_data/report\_id\_database/report\_drug.txt from bucket cvp-2-bucket... 2024-12-13 15:48:10,254 - INFO - Attempting to read S3 file Input\_data/report\_id\_database/reports.txt from bucket cvp-2-bucket... 2024-12-13 15:48:10,257 - INFO - Attempting to read S3 file Input\_data/report\_id\_database/reactions.txt from bucket cvp-2-bucket... 2024-12-13 15:48:10,259 - INFO - Attempting to read S3 file Input\_data/report\_id\_database/report\_links.txt from bucket cvp-2-bucket... 2024-12-13 15:48:10,261 - INFO - Attempting to read S3 file Input\_data/report\_id\_database/report\_drug\_indication.txt from bucket cvp-2-bucket... 2024-12-13 15:48:12,939 - INFO - Successfully read S3 file Input\_data/Suspected\_Product\_Brand\_Name/drug\_names.txt from bucket cvp-2-bucket. 2024-12-13 15:48:12,940 - INFO - Finished reading drug\_names. 2024-12-13 15:48:12,959 - INFO - Successfully read S3 file Input\_data/report\_id\_database/reactions.txt from bucket cvp-2-bucket. 2024-12-13 15:48:13,002 - INFO - Successfully read S3 file Input\_data/report\_id\_database/report\_links.txt from bucket cvp-2-bucket. 2024-12-13 15:48:13,105 - INFO - Successfully read S3 file Input\_data/report\_id\_database/report\_drug\_indication.txt from bucket cvp-2-bucket. 2024-12-13 15:48:13,126 - INFO - Successfully read S3 file Input\_data/report\_id\_database/report\_drug.txt from bucket cvp-2-bucket. 2024-12-13 15:48:13,146 - INFO - Successfully read S3 file Input\_data/report\_id\_database/reports.txt from bucket cvp-2-bucket. 2024-12-13 16:36:49,476 - INFO - Finished reading report\_drug. 2024-12-13 16:36:49,515 - INFO - Finished reading reports. 2024-12-13 16:36:49,516 - INFO - Finished reading reactions. 2024-12-13 16:36:49,516 - INFO - Finished reading report\_links. 2024-12-13 16:36:49,516 - INFO - Finished reading report\_drug\_indication. 2024-12-13 16:36:49,516 - INFO - Drug names file size: 4 lines 2024-12-13 16:36:49,517 - INFO - Report drug file size: 4953547 lines 2024-12-13 16:36:49,517 - INFO - Reports file size: 1136110 lines 2024-12-13 16:36:49,517 - INFO - Reactions file size: 4392489 lines 2024-12-13 16:36:49,517 - INFO - Report links file size: 1294010 lines 2024-12-13 16:36:49,517 - INFO - Report drug indication file size: 3222924 lines 2024-12-13 16:36:49,518 - INFO - Parsing drug names... 2024-12-13 16:36:49,519 - INFO - Parsed 2 drug names. 2024-12-13 16:36:49,520 - INFO - Finding REPORT\_IDs for 2 drug names... 2024-12-13 16:38:03,102 - INFO - Found 374 report IDs matching the drug names. 2024-12-13 16:38:03,112 - INFO - Extracting report data from reference files... Traceback (most recent call last): File "C:\Users\CJP\PycharmProjects\PythonProject3\script3.py", line 255, in <module> main() ~~~~^^ File "C:\Users\CJP\PycharmProjects\PythonProject3\script3.py", line 242, in main report\_data = extract\_report\_data( report\_ids, ...<4 lines>... results['report\_drug'] ) File "C:\Users\CJP\PycharmProjects\PythonProject3\script3.py", line 110, in extract\_report\_data report\_id = fields[1].strip() ~~~~~~^^^ IndexError: list index out of range Process finished with exit code 1





Script2 15:38

