#Import Packages

import configparser

import json

import asyncio

from datetime import date, datetime, timezone

from telethon import TelegramClient

from telethon.errors import SessionPasswordNeededError

from telethon.tl.functions.messages import (GetHistoryRequest)

from telethon.tl.types import (

PeerChannel

)

import pandas as pd

# Your credentials

api\_id = your\_id\_id (int)

api\_hash = 'your\_api\_hash'

#Password=958221

# Creating the client

client = TelegramClient('your\_bot\_name', api\_id, api\_hash)

async def extract\_channel\_messages(channel\_name, start\_time):

# Getting the channel entity

channel = await client.get\_entity(channel\_name)

# List to hold all messages

messages\_data = []

# Getting the messages

async for message in client.iter\_messages(channel):

# Check if the message is later than the start time

if message.date > start\_time.replace(tzinfo=timezone.utc): # Make start\_time offset-aware

messages\_data.append({

'channel': channel\_name,

'id': message.id,

'text': message.text,

'date': message.date,

'views': message.views

})

return messages\_data

async def main(start\_time\_str):

# Connecting and logging in

await client.start(phone='your\_phone\_number', password='your\_password')

# User-specified start time

start\_time = datetime.strptime(start\_time\_str, '%Y-%m-%d').replace(tzinfo=timezone.utc) # Make start\_time offset-aware

# Channels to extract from

channels = ['Channel\_1', 'channel\_2','Channel\_n']

all\_messages = []

for channel in channels:

messages = await extract\_channel\_messages(channel, start\_time)

all\_messages.extend(messages)

# Creating a DataFrame

df = pd.DataFrame(all\_messages)

# Save the DataFrame to a CSV file with channel name

df.to\_csv('telegram\_channels\_messages.csv', index=False, encoding='utf-8')

# Logout and disconnect

await client.log\_out()

await client.disconnect()

# User input for start time

user\_start\_time = input("Enter start time (YYYY-MM-DD): ")

# Running the client

async with client:

await main(user\_start\_time)