

## Extracting Data from CSV file and saving it into another CSV file using Pandas:

```
import pandas as pd
sales_csv = ('C:/Users/jyoth/Downloads/Amazon Sale Report.csv,Amazon Sale Report.csv')
sales_df = pd.read_csv(sales_csv)
#print(sales_df)
display(sales_df)
display(sales_df.head(50))

# Save Data to Another CSV File
output_csv = 'C:/Users/jyoth/Downloads/Amazon_Sale_Report_Copy.csv'
sales_df.to_csv(output_csv, index=False, encoding='utf-8')

print(f"Data saved to {output_csv}")
```

[illegible]

### Extracting Data from CSV file and saving it into SQL Server Database:

```
import pandas as pd
import pyodbc
from sqlalchemy import create_engine

# Load Data from CSV
sales_csv = 'C:/Users/jyoth/Downloads/Amazon Sale Report.csv/Amazon Sale Report.csv'
sales_df = pd.read_csv(sales_csv, encoding='utf-8', low_memory=False)

# SQL Server Connection Details

db_server = 'JYOTHI\SQLEXPRESS' # SQL Server instance
db_name = 'dataExtraction' # SQL Server database name

# Connection string for SQL Server using pyodbc

conn_str = f'DRIVER={{SQL Server}};SERVER={db_server};DATABASE={db_name};Trusted_Connection=yes;

try:
    # Create SQLAlchemy engine using pyodbc for SQL Server

    engine = create_engine(f'mssql+pyodbc:///?odbc_connect={conn_str}')

    # Connect to SQL Server and insert the DataFrame

    with engine.connect() as conn:
        sales_df.to_sql('sales', conn, if_exists='replace', index=False)
        print("Data inserted into SQL Server database successfully!")
except Exception as e:
    print(f"Error inserting data into SQL Server: {e}")
```

Query: SELECT * FROM JDOFINDING [Status = x] WHERE [SELECT * FROM JDOFINDING].[Job].[Index]																			
100 %																			
Results 4 Messages																			
ID	Job	Index	Date	Status	Fulfillment	Status_Changed	Job-previous-ref	Date	SKU	Category	Size	ISBN	Counter	Qty	currency	Amount	shop	id	name
1	405-8767876-571545	10-3-22	Cancelled	Shipped	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-0P	Set	L	B09K0V7027	NALL	0	INR	6472	MUMBAI		
2	1774-010701-131748	10-3-22	Cancelled	Delivered to Buyer	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-000L	Set	3XL	B09K0V7027	NALL	0	INR	406	BENGALURU		
3	404-4043634-62261718	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	NALL	0	INR	520	INDIA		
4	407-00870-724522	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	J0341	J0341-048L	Women Dress	L	B09K0V7027	NALL	0	INR	753.13	PUNE		CHENNAI
5	407-00870-724522	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-7X-000L	Top	3XL	B09K0V7429	Shipped	1	INR	574	CHENNAI		
6	405-8767876-571545	10-3-22	Cancelled	Shipped	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	NALL	0	INR	524	CHENNAI		
7	409-746848-405855	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	J0095	J0095-087L	Set	L	B09K0V7468	Shipped	1	INR	653	CHENNAI		
8	405-8767876-571545	10-3-22	Cancelled	Delivered to Buyer	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-000L	Set	3XL	B09K0V7468	Shipped	1	INR	309	HYDERABAD		
9	405-8767876-571545	10-3-22	Cancelled	Shipped	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	NALL	0	INR	524	CHENNAI		
10	402-006767-021120	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	363	CHENNAI		
11	407-010422-021018	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	305	CHENNAI		
12	404-4043634-62261718	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	524	CHENNAI		
13	405-8767876-571545	10-3-22	Cancelled	Delivered to Buyer	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	309	HYDERABAD		
14	405-769808-030434	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	607	MUMBAI		
15	405-8767876-571545	10-3-22	Cancelled	Shipped	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	320	HYDERABAD		
16	405-8767876-571545	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	309	HYDERABAD		
17	405-8767876-571545	10-3-22	Cancelled	Delivered to Buyer	Merchant	Amazon.in	Standard	J0341	J0341-048L	Set	L	B09K0V7027	Shipped	1	INR	771	MUMBAI		
18	402-006767-021120	10-3-22	Cancelled	Delivered to Buyer	Merchant	Amazon.in	Standard	P46E3	P46E3-087L	Set	XL	B09K0V7027	Shipped	1	INR	644	HYDERABAD		
19	402-006767-021120	10-3-22	Cancelled	Delivered to Buyer	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	402	CHENNAI		
20	405-8767876-571545	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	606	BENGALURU		
21	404-4043634-62261718	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	524	CHENNAI		
22	404-4043634-62261718	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	L	B09K0V7027	Shipped	1	INR	576	HYDERABAD		
23	1774-010701-131748	10-3-22	Cancelled	Shipped	Merchant	Amazon.in	Standard	9ET339	9ET339-4N-0P	Set	M	B09K0V7027	NALL	0	INR	5748	CHENNAI		
24	407-010422-021018	10-3-22	Cancelled	Shipped	Amazon	Amazon.in	Expedited	9ET339	9ET339-4N-000L	Set	XL	B09K0V7027	Shipped	1	INR	400	HYDERABAD		

Extracting Data from SQL Server database  
table and saving it as CSV file:

```
import pyodbc
import pandas as pd

# SQL Server connection details
server = 'JYOTHISQLEXPRESS'
database = 'dataExtraction'
driver = '{ODBC Driver 17 for SQL Server}'

# Establish connection with Windows Authentication
conn = pyodbc.connect(f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;')

# SQL Query to extract data
query = "SELECT * FROM sales"

# Execute query and load data into a Pandas DataFrame
df = pd.read_sql(query, conn)

# Save extracted data to a CSV file
df.to_csv("C:/Users/jyoth/Data-Extraction-Project/data/extracted_data.csv", index=False)

# Close the connection
conn.close()

print("Data successfully extracted and saved to extracted_data.csv")
```

Extracting a result set by querying the table  
and saving the result set as CSV file:

```
import pyodbc
import pandas as pd

# SQL Server connection details
server = 'JYOTHISQLEXPRESS'
database = 'dataExtraction'
driver = '{ODBC Driver 17 for SQL Server}' # Ensure you have the right driver installed

# Establish connection with Windows Authentication
conn = pyodbc.connect(f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;')

# SQL Query to extract data
query = """Select * from dbo.sales WHERE
Status = 'Cancelled'
AND Sales_Channel = 'Amazon.in'
AND ship_city = 'MUMBAI'
ORDER BY
Date DESC;"""

# Execute query and load data into a Pandas DataFrame
df = pd.read_sql(query, conn)

# Save extracted data to a CSV file
df.to_csv("C:/Users/jyoth/Data-Extraction-Project/data/extracted_data1.csv", index=False)

# Close the connection
conn.close()

print("Data successfully extracted and saved to extracted_data.csv")
```

Result Set:

A1	A	B	C	D	E	F	G	H	I
index	Order ID	Date	Status	Fulfillment	Sales Channel	ship-service-level	Style	SKU	
1	0 405-8078784-5731545	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET389	SET389-KR-NP-S	
2	1 171-9198151-1101146	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3781	JNE3781-KR-XXXL	
3	2 404-0687676-7273146	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3371	JNE3371-KR-XL	
4	3 403-9615377-8133951	04-30-22	Cancelled	Merchant	Amazon.in	Standard	J0341	J0341-DR-L	
5	4 407-1080790-7240320	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3671	JNE3671-TU-XXXL	
6	5 404-1490984-4578765	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET254	SET254-KR-NP-XL	
7	6 408-5748499-6859555	04-30-22	Shipped	Amazon	Amazon.in	Expedited	J0095	J0095-SET-L	
8	7 406-7807733-3785945	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-S	
9	8 407-5443024-5233168	04-30-22	Cancelled	Amazon	Amazon.in	Expedited	SET200	SET200-KR-NP-A-XXXL	
10	9 402-4393761-0311520	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3461	JNE3461-KR-XXL	
11	10 407-5633625-6970741	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3160	JNE3160-KR-G-S	
12	11 171-4638481-6326716	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3500	JNE3500-KR-XS	
13	12 405-5513694-8146768	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3405	JNE3405-KR-XS	
14	13 408-7955685-3063534	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET182	SET182-KR-DH-XS	
15	16 406-9379318-6555504	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3721	JNE3721-KR-XXL	
16	17 405-9013853-8009018	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3405	JNE3405-KR-XL	
17	18 408-1286370-1920302	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	J0351	J0351-SET-L	
18	15 403-4965581-9520319	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	PINEX368	PINEX368-KR-6XL	
19	18 402-4003038-5835511	04-30-22	Shipped - Delivered to Buyer	Merchant	Amazon.in	Standard	JNE3697	JNE3697-KR-XXL	
20	19 405-5957889-1051546	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET254	SET254-KR-NP-XS	
21	20 405-0607789-0716360	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3795	JNE3795-KR-XXL	
22	21 404-8484550-5860325	04-30-22	Shipped	Amazon	Amazon.in	Expedited	SET345	SET345-KR-NP-M	
23	22 171-1305077-2813934	04-30-22	Shipped	Amazon	Amazon.in	Expedited	JNE3373	JNE3373-KR-L	
24	23 404-6019946-2909948	04-30-22	Cancelled	Merchant	Amazon.in	Standard	SET291	SET291-KR-PP-M	
25	24 402-3384087-4005164	04-30-22	Shipped	Amazon	Amazon.in	Expedited	MEN5002	MEN5002-KR-L	

Filtered result set :

A1	A	B	C	D	E	F	G	H	I
index	Order ID	Date	Status	Fulfillment	Sales Channel	ship-service-level	Style	SKU	
1	91877 406-6712400-6972359	06-28-22	Cancelled	Merchant	Amazon.in	Standard	JNE3801	JNE3801-KR-XS	
2	91832 406-9549929-4979542	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3761	JNE3761-KR-XL	
3	92270 405-3638899-0456309	06-28-22	Cancelled	Merchant	Amazon.in	Standard	JNE2009	JNE2009-KR-310-L	
4	92453 407-3557879-2436304	06-28-22	Cancelled	Merchant	Amazon.in	Standard	SET357	SET357-KR-NP-XL	
5	92458 406-9626309-1007560	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	SET357	SET357-KR-NP-XXXL	
6	92655 405-6127894-9861144	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	JNE2009	JNE2009-KR-310-L	
7	92733 404-0615388-6009964	06-28-22	Cancelled	Merchant	Amazon.in	Standard	SET144	SET144-KR-NP-M	
8	92785 405-4034297-6023559	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	J0011	J0011-LCD-M	
9	92806 405-0278233-6386749	06-28-22	Cancelled	Amazon	Amazon.in	Expedited	J0011	J0011-LCD-M	
10	93150 407-9683313-2113110	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	SET305	SET305-KR-DPT-L	
11	93191 407-3703598-0813902	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3500	JNE3500-KR-XS	
12	93594 404-9483397-5401156	06-27-22	Cancelled	Merchant	Amazon.in	Standard	J0230	J0230-SKD-XXXL	
13	93757 405-5954187-9969122	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3487	JNE3487-KR-XXL	
14	93847 404-0715658-3889120	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	SET400	SET400-KR-NP-XL	
15	93924 402-7612561-4811527	06-27-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3518	JNE3518-KR-M	
16	94390 406-5487952-2361935	06-26-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3613	JNE3613-KR-L	
17	94512 405-8618104-5388729	06-26-22	Cancelled	Amazon	Amazon.in	Expedited	J0391	J0391-TP-S	
18	94866 406-1538451-4421913	06-26-22	Cancelled	Merchant	Amazon.in	Standard	JNE3359	JNE3359-KR-XL	
19	94701 406-9086955-3385132	06-26-22	Cancelled	Merchant	Amazon.in	Standard	SET375	SET375-KR-NP-L	
20	95308 407-6887961-2531509	06-26-22	Cancelled	Merchant	Amazon.in	Standard	JNE3797	JNE3797-KR-S	
21	95876 407-0310640-3383509	06-25-22	Cancelled	Merchant	Amazon.in	Standard	J0151	J0151-KR-XS	
22	96364 405-2581875-4881140	06-25-22	Cancelled	Amazon	Amazon.in	Expedited	J0285	J0285-SKD-XXXL	
23	95589 405-3150486-9020338	06-25-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3634	JNE3634-KR-L	
24	97325 407-5573427-2716328	06-24-22	Cancelled	Merchant	Amazon.in	Standard	JNE3476	JNE3476-KR-XXL	
25	96632 171-7393804-1209152	06-24-22	Cancelled	Amazon	Amazon.in	Expedited	JNE3797	JNE3797-KR-XXL	

## Extracting data from Web API and saving the data as a table in SQL Server database and as a CSV file:

```
import requests
import pyodbc
import pandas as pd

# Database connection details
server = 'JYOTHI\\SQLEXPRESS'
database = 'dataExtraction'
driver = '{ODBC Driver 17 for SQL Server}'

# Establish SQL Server Connection
conn = pyodbc.connect(f'DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;')
cursor = conn.cursor()

# Fetch full data from API
api_url = "https://dummyjson.com/users"
response = requests.get(api_url)

if response.status_code == 200:
    users = response.json()["users"] # Extract full users list
else:
    print("Failed to fetch data from API")
    exit()
```

```
# Create Table for ALL Fields (Dynamically)
cursor.execute("""
IF NOT EXISTS (SELECT * FROM sysobjects WHERE name='Users' AND xtype='U')
CREATE TABLE Users (
    id INT PRIMARY KEY,
    firstName NVARCHAR(100),
    lastName NVARCHAR(100),
    maidenName NVARCHAR(100),
    age INT,
    gender NVARCHAR(20),
    email NVARCHAR(255),
    phone NVARCHAR(50),
    username NVARCHAR(100),
    password NVARCHAR(255),
    birthDate DATE,
    image NVARCHAR(255),
    bloodGroup NVARCHAR(10),
    height FLOAT,
    weight FLOAT,
    eyeColor NVARCHAR(50),
    hairColor NVARCHAR(50),
    hairType NVARCHAR(50),
    ip NVARCHAR(50),
    macAddress NVARCHAR(50),
    university NVARCHAR(255),
    city NVARCHAR(100),
    state NVARCHAR(100),
    stateCode NVARCHAR(10),
    postalCode NVARCHAR(20),
    country NVARCHAR(100),
    cardNumber NVARCHAR(50),
    cardType NVARCHAR(50),
    currency NVARCHAR(10),
    iban NVARCHAR(50),
    company NVARCHAR(255),
```

```
    jobTitle NVARCHAR(255),
    ein NVARCHAR(50),
    ssn NVARCHAR(50),
    coin NVARCHAR(50),
    wallet NVARCHAR(255),
    network NVARCHAR(50),
    role NVARCHAR(50)
)
""")
conn.commit()

# Insert Full API Data into SQL Server
for user in users:
    cursor.execute("""
    INSERT INTO Users (
        id, firstName, lastName, maidenName, age, gender, email, phone,
        username, password, birthDate, image, bloodGroup, height, weight,
        eyeColor, hairColor, hairType, ip, macAddress, university,
        city, state, stateCode, postalCode, country, cardNumber,
        cardType, currency, iban, company, jobTitle, ein, ssn, coin,
        wallet, network, role
    )
    VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
    """, (
        user["id"], user["firstName"], user["lastName"], user["maidenName"], user["age"],
        user["gender"], user["email"], user["phone"], user["username"], user["password"],
        user["birthDate"], user["image"], user["bloodGroup"], user["height"], user["weight"],
        user["eyeColor"], user["hair"]["color"], user["hair"]["type"], user["ip"],
        user["macAddress"], user["university"], user["address"]["city"],
        user["address"]["state"], user["address"]["stateCode"], user["address"]["postalCode"],
        user["address"]["country"], user["bank"]["cardNumber"], user["bank"]["cardType"],
        user["bank"]["currency"], user["bank"]["iban"], user["company"]["name"],
        user["company"]["title"], user["ein"], user["ssn"],
        user["crypto"]["coin"], user["crypto"]["wallet"], user["crypto"]["network"],
        user["role"]
    ))
```

```
conn.commit()

# Step 4: Fetch Data and Save to CSV
df = pd.read_sql("SELECT * FROM Users", conn)
df.to_csv("C:/Users/jyoth/Data-Extraction-Project/data/users_API_data.csv", index=False)

# Close connection
conn.close()

print("Data successfully extracted and stored in SQL Server & CSV!")
```

Data successfully extracted and stored in SQL Server & CSV!

Result Set saved in SQL Server database:

SQL Query 2sql - JO...[NOTH]jyoth (60%)

SQL Query 1sql - JO...[NOTH]jyoth (51%)

SELECT \* FROM [dataExtraction].[dbo].[Users]

100 %

Results Messages

	id	firstName	lastName	maidenName	age	gender	email	phone	username	password	birthDate	image	bloodGroup	height	weight	eyeColor	hairColor
1	1	Emily	Johnson	Smith	28	female	emilyjohnson@x.dummyjson.com	+1 895-431-3024	emilye	emilypass	1996-05-30	https://dummyjson.com/icon/emilye128	O-	193.24	63.16	Green	Brown
2	2	Michael			35	male	michael.williams@x.dummyjson.com	+49 258-627-6644	michaiele	micheelpass	1989-05-10	https://dummyjson.com/icon/michaiele128	B+	186.22	76.32	Red	White
3	3	Sophia	Brown		42	female	sophia.brown@x.dummyjson.com	+91 210-652-7265	sophiab	sophiabpass	1982-11-06	https://dummyjson.com/icon/sophiab128	O-	177.72	52.6	Hazel	White
4	4	James	Davis		42	male	james.davis@x.dummyjson.com	+61 814-956-3964	jamesdavis	jamespass	1979-05-04	https://dummyjson.com/icon/jamesd128	AB-	193.31	62.1	Amber	Bronde
5	5	Emma	Miller	Johnson	30	female	emma.miller@x.dummyjson.com	+91 789-778-1814	emmaj	emmapass	1984-05-13	https://dummyjson.com/icon/emmaj128	AB	102.8	63.62	Green	White
6	6	Olivia	Wilson		22	female	olivia.wilson@x.dummyjson.com	+91 607-295-4485	oliviaw	oliviawpass	2002-04-20	https://dummyjson.com/icon/oliviaw128	B+	162.51	58	Hazel	Gray
7	7	Alexander	Jones		38	male	alexander.jones@x.dummyjson.com	+61 260-424-4886	alexanderj	alexanderpass	1986-10-20	https://dummyjson.com/icon/alexanderj128	AB-	153.89	77.42	Blue	White
8	8	Ava	Taylor		27	female	ava.taylor@x.dummyjson.com	+1 458-853-7877	avata	avatapass	1997-08-25	https://dummyjson.com/icon/avata128	AB-	166.47	57.08	Hazel	Red
9	9	Ethan	Martinez		33	male	ethan.martinez@x.dummyjson.com	+92 630-698-5081	ethanm	ethanmpass	1991-02-12	https://dummyjson.com/icon/ethanm128	AB+	159.19	68.81	Hazel	Purple
10	10	Isabella	Anderson	Davis	31	female	isabella.anderson@x.dummyjson.com	+49 770-658-4855	isabellad	isabelladpass	1993-06-10	https://dummyjson.com/icon/isabellad128	A-	150.56	50.1	Brown	Bronde
11	11	Liam	Garcia		29	male	liam.garcia@x.dummyjson.com	+92 870-217-6201	liamg	liamgpas	1995-06-06	https://dummyjson.com/icon/liamg128	AB-	162.32	93.16	Violet	Red
12	12	Mia	Rodriguez		24	female	mia.rodriguez@x.dummyjson.com	+49 989-481-9403	miaarp	miaarpass	2000-06-04	https://dummyjson.com/icon/miaarp128	O-	148.08	56.6	Blue	Purple
13	13	Noah	Hernandez		40	male	noah.hernandez@x.dummyjson.com	+49 334-056-6968	noahh	noahpass	1934-06-05	https://dummyjson.com/icon/noahh128	AB-	186.52	69.49	Brown	Red
14	14	Charlotte	Lee	Martinez	36	female	charlotte.lee@x.dummyjson.com	+44 373-853-8022	charlotem	charlotempass	1988-08-08	https://dummyjson.com/icon/charlotem128	AB	178.92	82.46	Brown	Gray
15	15	William	Gonzalez		32	male	william.gonzalez@x.dummyjson.com	+91 905-252-719	williamg	williamgpas	1992-02-27	https://dummyjson.com/icon/williamg128	B-	173.21	82.41	Hazel	Gray
16	16	Avery	Periz		25	female	avery.periz@x.dummyjson.com	+61 731-431-3457	avewp	avewppass	1999-03-30	https://dummyjson.com/icon/avewp128	O-	172.63	89.3	Brown	Green
17	17	Evelyn	Sanchez		37	female	evelyn.sanchez@x.dummyjson.com	+1 623-880-6871	evelyns	evelynspass	1987-10-13	https://dummyjson.com/icon/evelyns128	B+	184.08	83.15	Violet	Blue
18	18	Logan	Tones		31	male	logan.tones@x.dummyjson.com	+91 507-344-9733	logant	logantpass	1993-10-26	https://dummyjson.com/icon/logant128	A+	190.04	72.43	Green	Green
19	19	Abigail	Rivera		28	female	abigail.rivera@x.dummyjson.com	+91 228-363-7006	abigailr	abigailrpass	1996-10-11	https://dummyjson.com/icon/abigailr128	B+	186.39	74.61	Violet	Blue
20	20	Jackson	Evans		34	male	jackson.evans@x.dummyjson.com	+44 682-428-6668	jacksonc	jacksoncpass	1990-11-30	https://dummyjson.com/icon/jacksonc128	O-	162.57	74.32	Green	Red
21	21	Madison	Collins		26	female	madison.collins@x.dummyjson.com	+61 269-957-5711	madisonc	madisoncpass	1998-03-07	https://dummyjson.com/icon/madisonc128	B-	189.28	96.96	Red	Gray
22	22	Elijah	Stewart		33	male	elijah.stewart@x.dummyjson.com	+44 650-367-7872	elijahs	elijahspass	1991-11-22	https://dummyjson.com/icon/elijahs128	A-	195.33	81.64	Blue	Purple
23	23	Olivia	Moses		38	female	olivia.moses@x.dummyjson.com	+92 486-541-7123	oliviam	oliviampass	1985-04-21	https://dummyjson.com/icon/oliviam128	O-	155.07	63.87	Brown	Red
24	24	Mateo	Nguyen		30	male	mateo.nguyen@x.dummyjson.com	+91 507-57-6694	mateom	mateompass	1994-06-02	https://dummyjson.com/icon/mateom128	A+	174.29	59.98	Red	Purple
25	25	Harper	Kelly	Evans	27	female	harper.kelly@x.dummyjson.com	+92 518-883-2863	harperk	harperpass	1997-03-03	https://dummyjson.com/icon/harperk128	AB-	184.32	81.69	Amber	Red
26	26	Evelyn	Gonzalez		35	female	evelyn.gonzalez@x.dummyjson.com	+61 708-508-4886	evelyns	evelynspass	1989-02-05	https://dummyjson.com/icon/evelyns128	O-	169.94	58.47	Red	Black
27	27	Daniel	Cook		41	male	daniel.cook@x.dummyjson.com	+44 254-761-6843	danielc	danielcpass	1983-12-25	https://dummyjson.com/icon/danielc128	AB+	186.21	83.72	Brown	Purple
28	28	Lily	Lee	Brown	29	female	lily.lee@x.dummyjson.com	+1 808-757-9887	lilyl	lilylpass	1995-12-03	https://dummyjson.com/icon/lilyl128	AB-	181.42	51.49	Gray	Blue
29	29	Henry	Hill		38	male	henry.hill@x.dummyjson.com	+1 240-833-4680	henryh	henryhpass	1986-08-19	https://dummyjson.com/icon/henryh128	O-	180.25	95.84	Gray	Black
30	30	Adison	Wright		32	female	adison.wright@x.dummyjson.com	+1 514-384-3300	adisonw	adisonwpass	1992-01-03	https://dummyjson.com/icon/adisonw128	B+	179.32	76.93	Gray	Blue

Query executed successfully.

NOTH\SQLSERVER (16.0 RTM) NOTH\jyoth (60) dataExtraction 000000 30 rows

Result Set saved in CSV File:

E20	[dropdown] [dropdown] [dropdown] [dropdown] [dropdown] [dropdown] 28										
	A	B	C	D	E	F	G	H	I	J	K
1	id	firstName	lastName	maidenName	age	gender	email	phone	username	password	birthDate
2	1	Emily	Johnson	Smith	28	female	emily.johnson@x.dummyjson.com	+81 965-431-3024	emilys	emilypass	30-05-1998
3	2	Michael	Williams		35	male	michael.williams@x.dummyjson.com	+49 258-627-6644	michaelw	michaelwpass	10-08-1988
4	3	Sophia	Brown		42	female	sophia.brown@x.dummyjson.com	+81 210-652-2785	sophiab	sophiabpass	06-11-1988
5	4	James	Davis		45	male	james.davis@x.dummyjson.com	+49 614-958-9364	jamesd	jamesdpass	04-05-1978
6	5	Emma	Miller		30	female	emma.miller@x.dummyjson.com	+91 759-776-1614	emmaj	emmajpass	13-06-1998
7	6	Olivia	Wilson	Jones	22	female	olivia.wilson@x.dummyjson.com	+91 607-295-6448	oliviaw	oliviawpass	20-04-2000
8	7	Alexander	Jones		38	male	alexander.jones@x.dummyjson.com	+61 260-824-4986	alexanderj	alexanderjpass	20-10-1988
9	8	Ava	Taylor		27	female	ava.taylor@x.dummyjson.com	+1 458-853-7877	avat	avatpass	25-08-1998
10	9	Ethan	Martinez		33	male	ethan.martinez@x.dummyjson.com	+92 933-608-5081	ethanm	ethanmpass	12-02-1998
11	10	Isabella	Anderson		31	female	isabella.anderson@x.dummyjson.com	+49 770-658-4885	isabellad	isabelladpass	10-06-1998
12	11	Liam	Garcia	Davis	29	male	liam.garcia@x.dummyjson.com	+92 870-217-6201	liamg	liamgpass	06-09-1999
13	12	Mia	Rodriguez		24	female	mia.rodriguez@x.dummyjson.com	+49 989-461-8403	miar	miarpass	04-08-2000
14	13	Noah	Hernandez		40	male	noah.hernandez@x.dummyjson.com	+49 393-605-6968	noahh	noahhpass	05-08-1998
15	14	Charlotte	Lopez	Martinez	36	female	charlotte.lopez@x.dummyjson.com	+44 373-953-5028	charlottom	charlottompass	08-06-1988
16	15	William	Gonzalez		32	male	william.gonzalez@x.dummyjson.com	+81 905-252-7319	williamg	williamgpass	27-03-1999
17	16	Avery	Perez		25	female	avery.perez@x.dummyjson.com	+61 731-431-3457	averyp	averyppass	10-03-1999
18	17	Evelyn	Sanchez		37	female	evelyn.sanchez@x.dummyjson.com	+1 623-880-6871	evelyns	evelynspass	13-10-1988
19	18	Logan	Torres		31	male	logan.torres@x.dummyjson.com	+81 507-434-8733	logant	logantpass	26-10-1998
20	19	Abigail	Rivera		28	female	abigail.rivera@x.dummyjson.com	+91 228-363-7806	abigailr	abigailrpass	11-10-1999
21	20	Jackson	Evans		34	male	jackson.evans@x.dummyjson.com	+44 468-628-6686	jacksonc	jacksoncpass	30-11-1999
22	21	Madison	Collins		26	female	madison.collins@x.dummyjson.com	+81 259-957-5711	madisonc	madisoncpass	07-03-1999
23	22	Elijah	Stewart		33	male	elijah.stewart@x.dummyjson.com	+44 468-357-7872	elijahs	elijahspass	22-10-1998
24	23	Chloe	Morales		39	female	chloe.morales@x.dummyjson.com	+92 468-541-7133	chloem	chloempass	21-04-1998
25	24	Mateo	Nguyen		30	male	mateo.nguyen@x.dummyjson.com	+1 341-597-6694	mateon	mateonpass	02-06-1999
26	25	Harper	Kelly	Evans	27	female	harper.kelly@x.dummyjson.com	+92 518-863-2863	harpere	harperepass	03-03-1999
	<	>	users_API_data		+						

## Extracting data from a web page through Web scraping and saving the data as a table in SQL Server database and as a CSV file:

```
from bs4 import BeautifulSoup
import requests
import pandas as pd
import numpy as np
```

```
# Function to extract Product Title
def get_title(soup):

    try:
        # Outer Tag Object
        title = soup.find("span", attrs={"id": 'productTitle'})

        # Inner NavigableString Object
        title_value = title.text

        # Title as a string value
        title_string = title_value.strip()

    except AttributeError:
        title_string = ""

    return title_string

# Function to extract Product Rating
def get_rating(soup):

    try:
        rating = soup.find("i", attrs={'class': 'a-icon a-icon-star a-star-4-5'}).string.strip()

    except AttributeError:
        try:
            rating = soup.find("span", attrs={'class': 'a-icon-alt'}).string.strip()
        except:
            rating = ""

    return rating
```

```
# Function to extract Number of User Reviews
def get_review_count(soup):

    try:
        review_count = soup.find("span", attrs={'id': 'acrCustomerReviewText'}).string.strip()

    except AttributeError:
        review_count = ""

    return review_count

# Function to extract Availability Status
def get_availability(soup):

    try:
        available = soup.find("div", attrs={'id': 'availability'})
        available = available.find("span").string.strip()

    except AttributeError:
        available = "Not Available"

    return available
```

```
if __name__ == '__main__':

    # add your user agent
    HEADERS = ({'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/132.0.0.0'})

    # The webpage URL
    URL = "https://www.amazon.ca/s?k=Electronic+devices&crid=18Y0R7TERHJ0J&prefix=electronic+devices%2Caps%2C195&ref=nb_sb_ss_123_1"

    # HTTP Request
    webpage = requests.get(URL, headers=HEADERS)

    # Soup Object containing all data
    soup = BeautifulSoup(webpage.content, "html.parser")
```

```
# Fetch Links as List of Tag Objects
links = soup.find_all("a", attrs={'class': 'a-link-normal s-line-clamp-4 s-link-style a-text-normal'})

# Store the Links
links_list = []

# Loop for extracting Links from Tag Objects
for link in links:
    links_list.append(link.get('href'))

d = {"title": [], "rating": [], "reviews": [], "availability": []}

# Loop for extracting product details from each link
for link in links_list:
    new_webpage = requests.get("https://www.amazon.com" + link, headers=HEADERS)

    new_soup = BeautifulSoup(new_webpage.content, "html.parser")

    # Function calls to display all necessary product information
    d['title'].append(get_title(new_soup))
    d['rating'].append(get_rating(new_soup))
    d['reviews'].append(get_review_count(new_soup))
    d['availability'].append(get_availability(new_soup))

amazon_df = pd.DataFrame.from_dict(d)
amazon_df['title'].replace('', np.nan, inplace=True)
amazon_df = amazon_df.dropna(subset=['title'])
amazon_df.to_csv("amazon_data.csv", header=True, index=False)
```

amazon\_df

Result Set:

A2

Vasco V4 Language Translator Device | 108 Languages | Free Lifetime Internet in Almost 200 Cou

	A	B	C	D
1	title	rating	reviews	availability
2	Vasco V4 Language Translator Device   108 La	3.8 out of 5 stars	1,241 ratings	Only 15 left in stock - order soon.
3	Language Translator Device 144 Languages an	3.5 out of 5 stars	287 ratings	In Stock
4	Apple AirTag 4 Pack	4.7 out of 5 stars	136,677 ratings	Not Available
5	VICHYIE 20 in 1 Multifunctional Cleaner Kit for I	4.3 out of 5 stars	1,383 ratings	In Stock
6	Electronic Devices (Conventional Current Vers	4.3 out of 5 stars	166 ratings	Not Available
7	Electronic Devices: Conventional Current Vers	4.6 out of 5 stars	26 ratings	Not Available
8	4 Port USB Charging Station for Multiple Device	4.3 out of 5 stars	1,269 ratings	Not Available
9	Roku Express 4K+   Streaming Player HD/4K/H	4.6 out of 5 stars	2,440 ratings	Not Available
10	Amazon Echo Pop (newest model), Our small	4.7 out of 5 stars	77,241 ratings	Not Available
11	70M Laser Measure Device, Mileseeey Digital L	4.4 out of 5 stars	1,708 ratings	Not Available
12	Solid-State Electronic Devices: An Introduc	3.7 out of 5 stars	4 ratings	In Stock
13	Electronic Devices (Conventional Current Vers	4.3 out of 5 stars	90 ratings	Not Available
14	Offline Language Voice Translator Device W09	3.9 out of 5 stars	155 ratings	In Stock
15	M9 AI Language Translator Earbuds Open-Ear I	4.0 out of 5 stars	41 ratings	In Stock
16	Translator Earbuds, 3-in-1 Language Translat	2.0 out of 5 stars	1 rating	In Stock
17	Language Translator Device No WiFi Needed I	3.8 out of 5 stars	77 ratings	In Stock
18	SooPii 60W 6-Port Charging Station for Multiple	4.7 out of 5 stars	4,414 ratings	In Stock
19	ATUVOS Air Tag-1 Pack, Bluetooth Luggage Tra	4.3 out of 5 stars	5,858 ratings	In Stock
20	Unitek Fast Charging Station for Multiple Devi	4.3 out of 5 stars	235 ratings	In Stock
21	Skylight Calendar: 15 inch Digital Calendar & C	4.4 out of 5 stars	1,769 ratings	In Stock
22	Amazon Echo (newest model), Alexa speaker v	4.7 out of 5 stars	149,562 ratings	Not Available
23	Power-Bank-Portable-Charger - 40000mAh Pc	4.3 out of 5 stars	165 ratings	Not Available
24	ATUVOS Air Tag-1 Pack, Bluetooth Luggage Tra	4.3 out of 5 stars	5,858 ratings	In Stock
25	Electronic Devices and Circuit Theory	4.4 out of 5 stars	712 ratings	Not Available
26	320W Charging Station for Multiple Devices, Zc	4.6 out of 5 stars	430 ratings	In Stock

<

>

amazon\_data

+