Web Scraping-

- -Data Analysis-

1.Business Problems - E-commerce- Whar data????

2.Data collection--API(App programming Interface), Databases -SQL

Web Scrapping-- Collect the data from Website, automatically

- Salenium.
- Python - (BeatufulSoup, Scrapy)

1.Ensure Ethical and Legal Compliance:

- Check the website's robots.txt file for allowed paths and restrictions.
- Respect the terms of service to avoid any legal issues.
- Use a Structured Scraping Process:

2.Identify the URL structure and relevant data fields.

- Implement a scraping script using Python and BeautifulSoup.
- Introduce delays between requests to avoid overloading the server.

Project Structure

- Introduction
- Web Scraping
- Data Cleaning and Preparation
- Data Analysis
- Data Visualization
- Conclusion

1. Introduction

Web Scraping and Data Analysis of Book Prices

In this project, we will scrape data from the `books.toscrape.com` website to analyze book prices. We will:

1. Collect data using web scraping.
2. Clean and prepare the data.
3. Analyze trends in book prices.
4. Visualize the data to uncover insights.

Web Scraping for Auction Data

- Step 1: Import Necessary Libraries

In [15]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import time
```

- Step 2: Define the URL and Headers

In [16]:
```
base_url = 'http://books.toscrape.com/catalogue/page-{}.html'
headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0
}
```

- Step 3: Function to Scrape Auction Data

In [17]:
```
def scrape_auction_data(page_number):
    url = base_url.format(page_number)
    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        soup = BeautifulSoup(response.content, 'html.parser')
        items = soup.find_all('article', class_='product_pod')
        data = []
        for item in items:
            title = item.h3.a['title']
            price = item.find('p', class_='price_color').text.strip()
            data.append({
                'title': title,
                'price': price
            })
        return data
    else:
        print('Failed to retrieve data')
        return []
```

- Step 4: Scrape Data with Rate Limiting

In [18]:
```
# Scrape multiple pages
all_auction_data = []
for page in range(1, 6):  # Example for first 5 pages
    all_auction_data.extend(scrape_auction_data(page))
    time.sleep(1)  # Adding a delay between requests to prevent overloading the server
```

- Step 5:Convert to DataFrame

In [19]:
```
auction_df = pd.DataFrame(all_auction_data)
auction_df['price'] = auction_df['price'].str.replace('£', '').astype(float)
print(auction_df.head())
```

```
                                title  price
0                A Light in the Attic  51.77
1                  Tipping the Velvet  53.74
2                        Soumission  50.10
3                     Sharp Objects  47.82
4  Sapiens: A Brief History of Humankind  54.23
```

- Step 5:Save the data to a CSV file

In [20]:
```
auction_df.to_csv('auction_data.csv', index=False)
```

To identify trends in buyer behavior, item valuation, and market demand for an auction house, I would:

Data Collection:

- Gather historical auction data, including buyer information, item details, and auction outcomes.

Data Analysis:

- Clean and preprocess the data.
- Use statistical methods and visualization techniques to identify trends.

In [21]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

- Step1.Load the data

In [22]:
```
auction_data = pd.read_csv('auction_data.csv')
```

In [23]:
```
auction_data
```

Out[23]:

	title	price
0	A Light in the Attic	51.77
1	Tipping the Velvet	53.74
2	Soumission	50.10
3	Sharp Objects	47.82
4	Sapiens: A Brief History of Humankind	54.23
...	...	...
95	Lumberjanes Vol. 3: A Terrible Plan (Lumberjan...	19.92
96	Layered: Baking, Building, and Styling Spectac...	40.11
97	Judo: Seven Steps to Black Belt (an Introducto...	53.90
98	Join	35.67
99	In the Country We Love: My Family Divided	22.00

100 rows × 2 columns

In [24]:
```
auction_data.head()
```

Out[24]:

	title	price
0	A Light in the Attic	51.77
1	Tipping the Velvet	53.74
2	Soumission	50.10
3	Sharp Objects	47.82
4	Sapiens: A Brief History of Humankind	54.23

In [25]:
```
auction_data.shape
```

Out[25]:
```
(100, 2)
```

- Step2.Data Cleaning

In [26]:
```
auction_data['price'] = auction_data['price'].astype(float)
```

- Step3.Identify trends in buyer behavior (e.g., average price by item)

In [28]:
```
average_price_by_item = auction_data.groupby('title')['price'].mean().sort_values(ascending=False)
print(average_price_by_item.head())
```

```
title
The Death of Humanity: and the Case for Life    58.11
Slow States of Collapse: Poems                  57.31
Our Band Could Be Your Life: Scenes from the American Indie Underground, 1981-1991    57.25
The Past Never Ends                             56.50
The Pioneer Woman Cooks: Dinnertime: Comfort Classics, Freezer Food, 16-Minute Meals, and Other Delicious Ways
to Solve Supper!    56.41
Name: price, dtype: float64
```

- Step4.Plotting the average price by item

In [31]:
```
plt.figure(figsize=(15, 10))
sns.barplot(x=average_price_by_item.values, y=average_price_by_item.index)
plt.title('Average Price by Auction Item')
plt.xlabel('Average Price (£)')
plt.ylabel('Item')
plt.show()
```


- Step5:Identify market demand trends (e.g., distribution of item prices)

In [32]:
```
plt.figure(figsize=(10, 6))
sns.histplot(auction_data['price'], bins=20, kde=True, color='blue')
plt.title('Distribution of Auction Item Prices')
plt.xlabel('Price (£)')
plt.ylabel('Frequency')
plt.show()
```


- Regards

--Tejas Patil--

In [ ]:
```

```