NAME: shraddha Nitin Kadam

PROJECT NAME: Cryptocurrency Data Extraction

Project Guide: Prof. Sameer Warsolkar.

1. Aim:

The aim of this project is to extract and analyze key data points from the Coin Market Cap website, specifically for cryptocurrencies. By scraping columns such as Name, Symbol, Price, Market Cap, and Volume (24h), you can track and compare the performance and trends of different cryptocurrencies over time. This analysis can be useful for making informed investment decisions, conducting market research, and understanding the broader cryptocurrency market dynamics.

2.Objectives:

✓ Data Extraction:

- Scrape cryptocurrency data from Coin Market Cap for columns such as name, symbol, price, and market cap

✓ Data Cleaning and Formatting:

- Ensure the scraped data is accurate, complete, and formatted correctly.

✓ Trend Analysis:

- Analyze the price changes over different periods (1 hour, 24 hours) to identify trends.

✓ Market Insights:

- Assess market capitalization and trading volume to gauge market activity and liquidity.

✓ Comparative Analysis:

- Compare performance metrics across different cryptocurrencies.

✓ Visualization:

- Create visual representations (graphs, charts) to display trends and insights.

- ✓ Report Generation:
  - Summarize findings in a comprehensive report for stakeholders.

- ✓ Tool Development:
  - Develop tools or scripts for automated data scraping and analysis.

- ✓ Risk Assessment:
  - Evaluate the volatility and risks associated with different cryptocurrencies.

- ✓ Decision Support:
  - Provide actionable insights for investment decisions and market strategies.

3.OUTLINE:

From this site, we are going to grab the following information:

- ✓ Names
- ✓ Symbol
- ✓ Price
- ✓ One hour
- ✓ One Day
- ✓ Market Cap
- ✓ Volume
- ✓ Volume2
- ✓ Circulating Supply

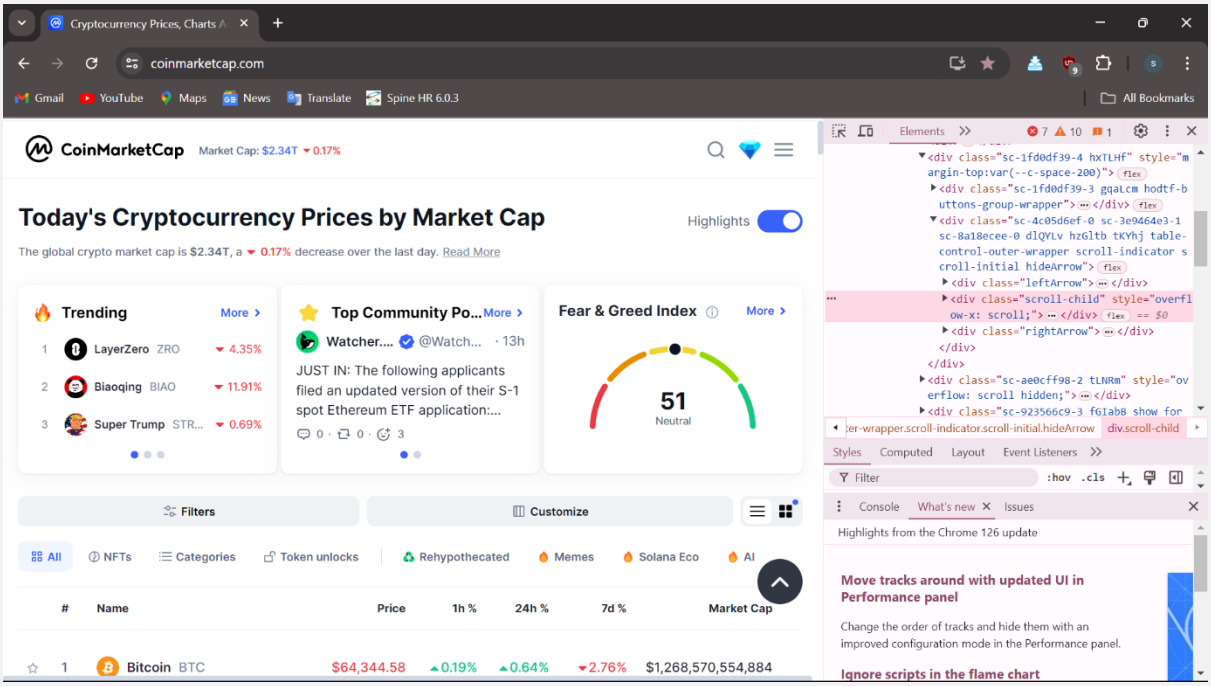
4. Steps:

Choose the Website and Webpage URL:

- The first step is to select the website you want to scrape. We will try to extract data of cryptocurrencies.

1. Inspect the website:

- Now the next step is to understand the website structure. Understand what the attributes of the elements that are of your interest are. Right click on the website to select "Inspect". This will open HTML code. Use the inspector tool to see the name of all the elements to use in the code.



2.Installing the important libraries:

Python has several web scrapping libraries. We will use the following libraries

- Requests – for making HTTP requests to website
- Beautiful Soup – for parsing the HTML code
- Pandas – for storing the scraped data in data frame

3. Write the Python source code:

We'll write the main python code. The code will perform the following steps:

- Using requests to send an HTTP GET requests
- Using Beautiful Soup to parse the HTML code
- Extracting the required data from the HTML code
- Store the information in a pandas Data Frame

4. Exporting the extracted data:

- We will export the data as a CSV file. We will use the pandas library. We'll use the pandas library.

5. Benefits:

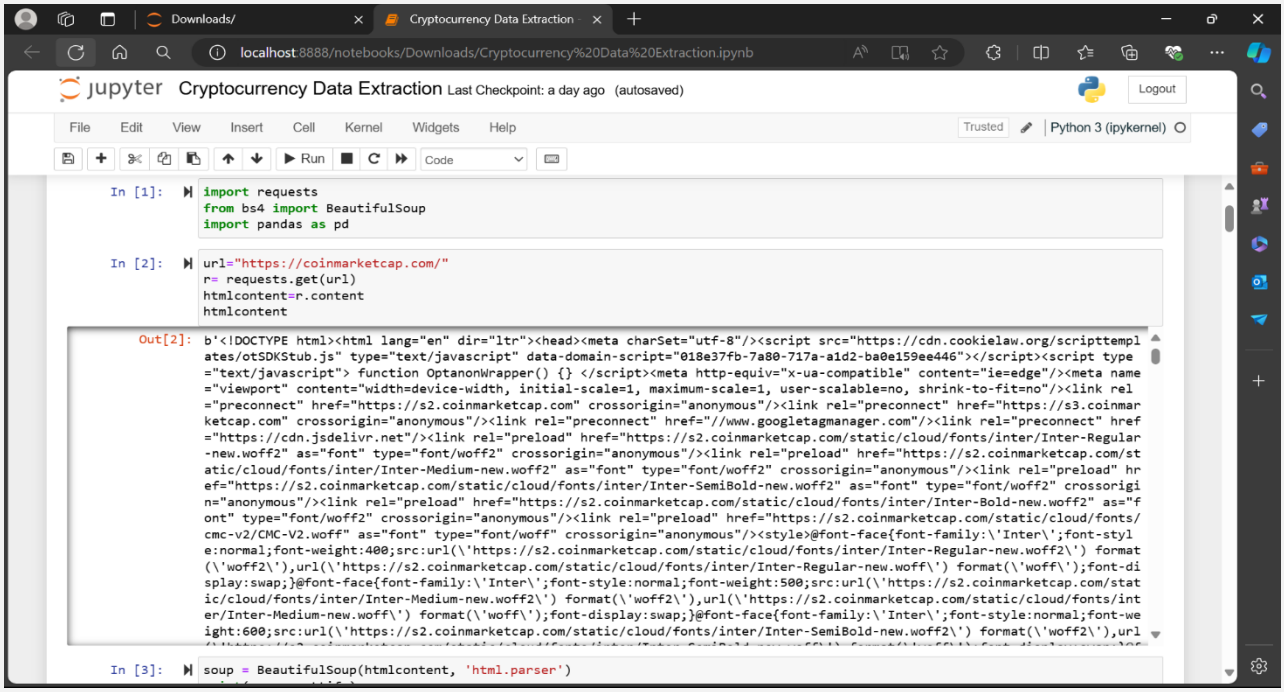
- Access to valuable data for analysis or research.
- Automation of data collection, saving time and effort.
- Stay up to date with change changes on the target websites.

6. Risk:

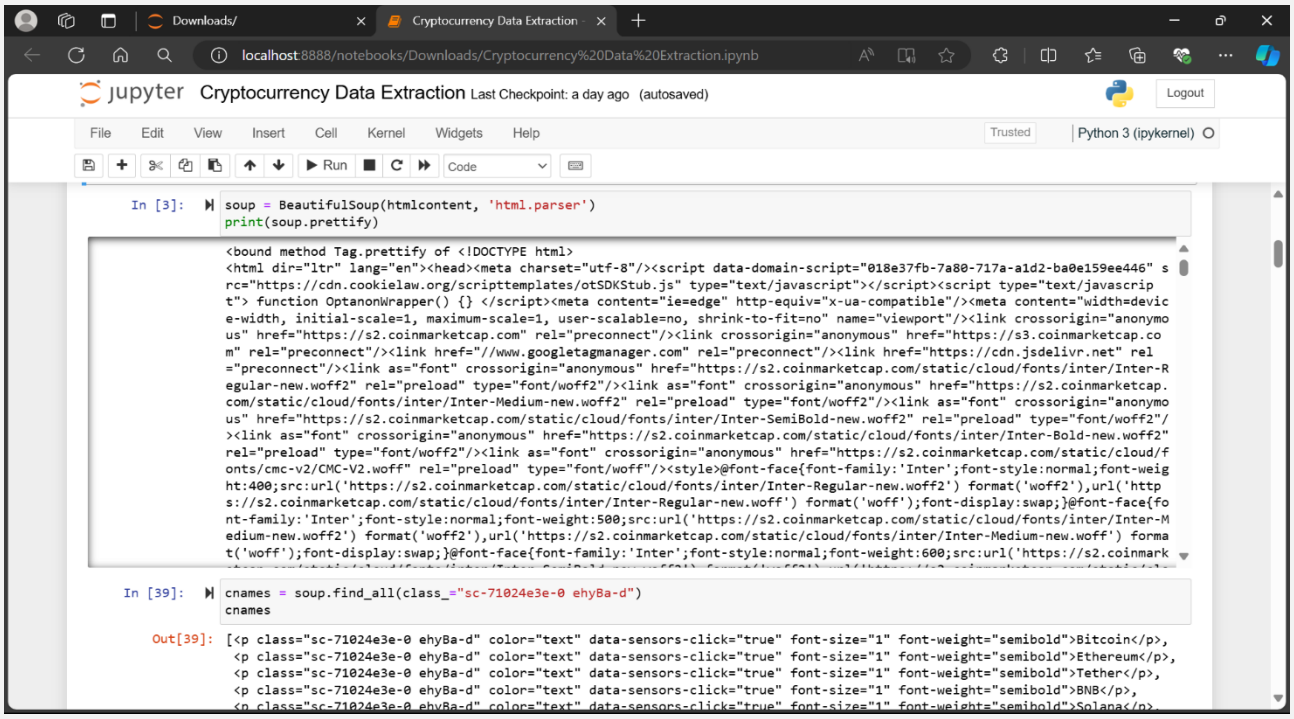
- Legal issues related to web scraping.
- Technical challenges due to website changes.

5. Cryptocurrency web scraping coding command steps

1. Accessing cryptocurrency website:



2. Using Beautiful Soup:



3. Accessing the cryptocurrency Names:

```
In [39]:  cnames = soup.find_all(class_="sc-71024e3e-0 ehyBa-d")
          cnames

Out[39]: [<p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">Bitcoin</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">Ethereum</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">Tether</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">BNB</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">Solana</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">USDC</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">XRP</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">Dogecoin</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">Toncoin</p>,
          <p class="sc-71024e3e-0 ehyBa-d" color="text" data-sensors-click="true" font-size="1" font-weight="semibold">Cardano</p>]

In [7]:   name=[]
          for i in cnames:
              name.append(i.get_text())

          print(name)

          len(name)

          ['Bitcoin', 'Ethereum', 'Tether', 'BNB', 'Solana', 'USDC', 'XRP', 'Dogecoin', 'Toncoin', 'Cardano']

Out[7]:  10
```

---

4. Accessing the cryptocurrency Symbols:

```
In [9]:   symb=soup.find_all(class_="sc-71024e3e-0 OqPKt coin-item-symbol")
          symb

Out[9]: [<p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">BTC</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">ETH</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">USDT</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">BNB</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">SOL</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">USDC</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">XRP</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">DOGE</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">TON</p>,
         <p class="sc-71024e3e-0 OqPKt coin-item-symbol" color="text3" data-sensors-click="true" font-size="1">ADA</p>]

In [10]:  symbol=[]
          for i in symb:
              symbol.append(i.get_text())

          print(symbol)

          len(symbol)

          ['BTC', 'ETH', 'USDT', 'BNB', 'SOL', 'USDC', 'XRP', 'DOGE', 'TON', 'ADA']

Out[10]: 10
```

5. Accessing cryptocurrencies price:

```
In [64]: rs=soup.find_all(class_="sc-a093f09c-0 gPTgRa")
         rs

Out[64]: [<div class="sc-a093f09c-0 gPTgRa"><span>$64,129.32</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$3,518.60</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$0.9993</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$585.59</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$132.19</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$1.00</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$0.4871</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$0.1247</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$7.11</span></div>,
          <div class="sc-a093f09c-0 gPTgRa"><span>$0.3856</span></div>]

In [65]: b=[]
         for i in rs:
             b.append(i.get_text())

         print(b)

         len(b)

         ['$64,129.32', '$3,518.60', '$0.9993', '$585.59', '$132.19', '$1.00', '$0.4871', '$0.1247', '$7.11', '$0.3856']

Out[65]: 10
```

6. Using Regex remove dollar sign from price list:

```
In [66]: import re
         price = [re.sub(r'[\$,]', '', value) for value in b]
         print(price)

         ['64129.32', '3518.60', '0.9993', '585.59', '132.19', '1.00', '0.4871', '0.1247', '7.11', '0.3856']
```

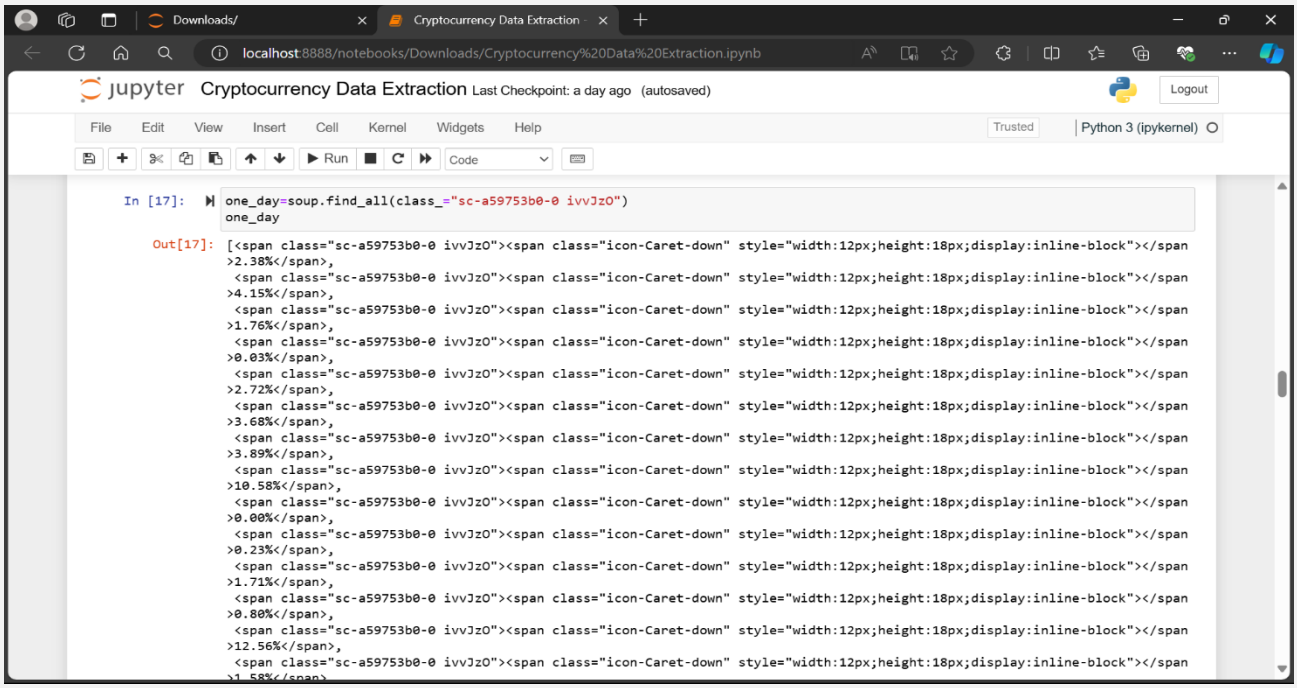
7. Accessing Percentage change in price over the last

```
In [14]: hour=soup.find_all(class_="sc-a59753b0-0 ivvJzO")
         hour

Out[14]: [<span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >2.38%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >4.15%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >1.76%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.03%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >2.72%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >3.68%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >3.89%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >10.58%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.00%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.23%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >1.71%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.80%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >12.56%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >1.58%</span>
```

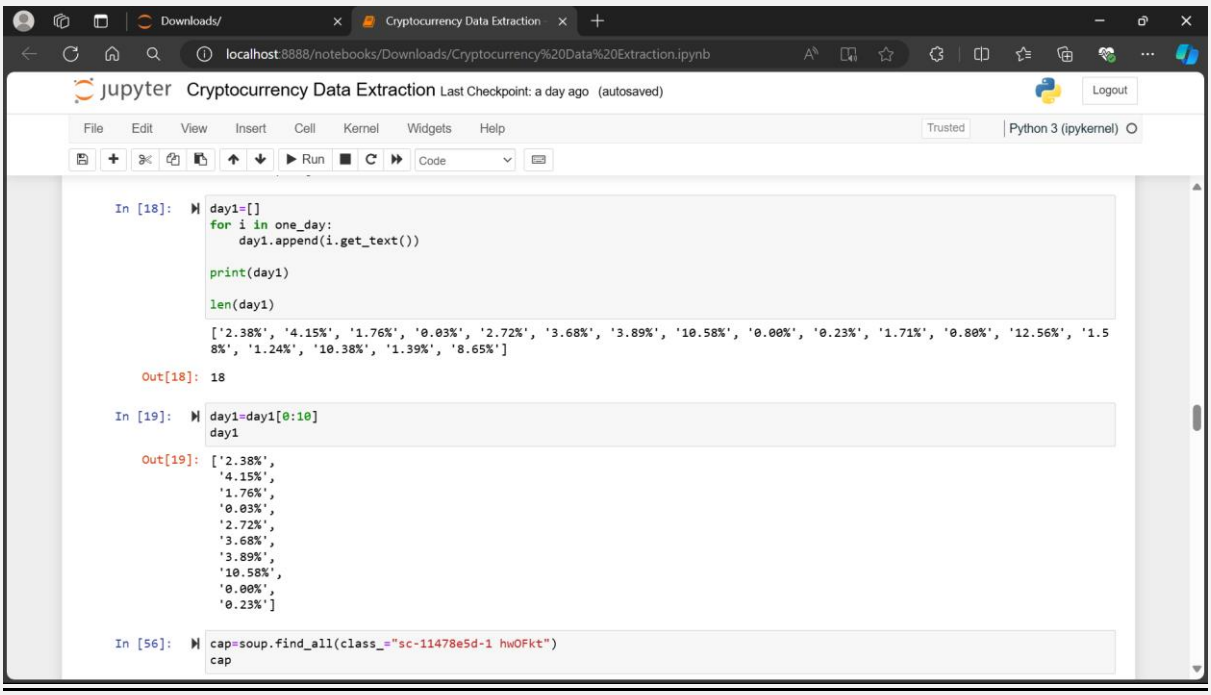
1hour

```
In [15]: one_h=[]
         for i in hour:
             one_h.append(i.get_text())

         print(one_h)

         len(one_h)

         ['2.38%', '4.15%', '1.76%', '0.03%', '2.72%', '3.68%', '3.89%', '10.58%', '0.00%', '0.23%', '1.71%', '0.80%', '12.56%', '1.5
         8%', '1.24%', '10.38%', '1.39%', '8.65%']

Out[15]: 18

In [16]: one_h=one_h[0:10]
         one_h

         len(one_h)

Out[16]: 10
```

8. Accessing Percentage change in price over the last 24 Hours



```
In [17]: one_day=soup.find_all(class_="sc-a59753b0-0 ivvJzO")
         one_day

Out[17]: [<span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >2.38%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >4.15%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >1.76%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.03%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >2.72%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >3.68%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >3.89%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >10.58%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.00%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.23%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >1.71%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >0.80%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >12.56%</span>,
          <span class="sc-a59753b0-0 ivvJzO"><span class="icon-Caret-down" style="width:12px;height:18px;display:inline-block"></span
         >1.58%</span>
```



```
In [18]: day1=[]
         for i in one_day:
             day1.append(i.get_text())

         print(day1)

         len(day1)

         ['2.38%', '4.15%', '1.76%', '0.03%', '2.72%', '3.68%', '3.89%', '10.58%', '0.00%', '0.23%', '1.71%', '0.80%', '12.56%', '1.5
         8%', '1.24%', '10.38%', '1.39%', '8.65%']

Out[18]: 18

In [19]: day1=day1[0:10]
         day1

Out[19]: ['2.38%',
          '4.15%',
          '1.76%',
          '0.03%',
          '2.72%',
          '3.68%',
          '3.89%',
          '10.58%',
          '0.00%',
          '0.23%']

In [56]: cap=soup.find_all(class_="sc-11478e5d-1 hwOFkt")
         cap
```

9. Accessing Market capitalization of the cryptocurrency

```
In [56]: cap=soup.find_all(class_="sc-11478e5d-1 hwOFkt")
         cap

Out[56]: [<span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$1,265,528,372,378</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$430,352,856,775</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$112,813,530,690</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$86,293,400,326</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$61,071,028,075</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$32,673,095,133</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$27,095,879,971</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$18,049,476,098</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$17,469,981,416</span>,
          <span class="sc-11478e5d-1 hwOFkt" data-nosnippet="true">$13,781,478,716</span>]

In [57]: a=[]
         for i in cap:
             a.append(i.get_text())

         print(a)

         len(a)

         ['$1,265,528,372,378', '$430,352,856,775', '$112,813,530,690', '$86,293,400,326', '$61,071,028,075', '$32,673,095,133', '$2
         7,095,879,971', '$18,049,476,098', '$17,469,981,416', '$13,781,478,716']

Out[57]: 10

In [58]: import re
         market_cap = [re.sub(r'[\$,]', '', value) for value in a]
         print(market_cap)
```

10. Accessing Trading volume in the last 24 hours

```
In [60]: vol=soup.find_all(class_="sc-71024e3e-0 bbHOdE font_weight_500")
         vol

Out[60]: [<p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$26,581,855,560</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$15,826,370,697</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$54,115,336,315</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$1,784,812,516</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$2,173,347,494</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$5,568,517,178</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$1,041,998,204</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$679,055,289</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$277,736,602</p>,
          <p class="sc-71024e3e-0 bbHOdE font_weight_500" color="text" data-sensors-click="true" font-size="1">$277,343,382</p>]

In [61]: x=[]
         for i in vol:
             x.append(i.get_text())

         print(x)

         len(x)

         ['$26,581,855,560', '$15,826,370,697', '$54,115,336,315', '$1,784,812,516', '$2,173,347,494', '$5,568,517,178', '$1,041,998,
         204', '$679,055,289', '$277,736,602', '$277,343,382']

Out[61]: 10

In [62]: import re
         volume = [re.sub(r'[\$,]', '', value) for value in x]
         print(volume)
```

11. Accessing Trading volume in the last 1 hour

```
In [26]: vol1=soup.find_all(class_="sc-71024e3e-0 hbcVUE")
         vol1

Out[26]: [<p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">414,100 BTC</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">4,496,768 ETH</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">54,152,667,539 USDT</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">3,052,485 BNB</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">16,441,659 SOL</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">5,568,024,311 USDC</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">2,138,850,992 XRP</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">5,447,611,363 DOGE</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">39,079,395 TON</p>,
          <p class="sc-71024e3e-0 hbcVUE" color="text2" data-sensors-click="true" font-size="0">719,163,502 ADA</p>]

In [27]: volume1=[]
         for i in vol1:
             volume1.append(i.get_text())

         print(volume1)

         len(volume1)

         ['414,100 BTC', '4,496,768 ETH', '54,152,667,539 USDT', '3,052,485 BNB', '16,441,659 SOL', '5,568,024,311 USDC', '2,138,850,
         992 XRP', '5,447,611,363 DOGE', '39,079,395 TON', '719,163,502 ADA']

Out[27]: 10
```

12. Accessing Circulating supply of the cryptocurrency

```
In [28]: cs=soup.find_all(class_="sc-71024e3e-0 hhmVNu")
         cs

Out[28]: [<p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">19,714,787 BTC</p
         >,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">122,276,728 ETH</
         p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">112,891,354,602 U
         SDT</p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">147,583,751 BNB</
         p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">462,010,348 SOL</
         p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">32,670,203,254 US
         DC</p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">55,618,185,850 XR
         P</p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">144,799,006,384 D
         OGE</p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">2,458,143,049 TON
         </p>,
          <p class="sc-71024e3e-0 hhmVNu" color="text" data-sensors-click="true" font-size="1" font-weight="medium">35,735,976,149 AD
         A</p>]

In [29]: supply=[]
         for i in cs:
             supply.append(i.get_text())

         print(supply)

         len(supply)
```

13. Importing Pandas and Create DataFrame

In [68]:
```
import pandas as pd
Cryptocurrency = pd.DataFrame({
    "Name": name,
    "Symbol": symbol,
    "Price": price,
    "One_hour":one_h,
    "One_Day":day1,
    "Market_Cap": market_cap,
    "Volume":volume,
    "Volume2":volume1,
    "Circulating_Supply":supply
})
Cryptocurrency
```

Out[68]:

	Name	Symbol	Price	One_hour	One_Day	Market_Cap	Volume	Volume2	Circulating_Supply
0	Bitcoin	BTC	64129.32	2.38%	2.38%	1265528372378	26581855560	414,100 BTC	19,714,787 BTC
1	Ethereum	ETH	3518.60	4.15%	4.15%	430352856775	15826370697	4,496,768 ETH	122,276,728 ETH
2	Tether	USDT	0.9993	1.76%	1.76%	112813530690	54115336315	54,152,667,539 USDT	112,891,354,602 USDT
3	BNB	BNB	585.59	0.03%	0.03%	86293400326	1784812516	3,052,485 BNB	147,583,751 BNB
4	Solana	SOL	132.19	2.72%	2.72%	61071028075	2173347494	16,441,659 SOL	462,010,348 SOL
5	USDC	USDC	1.00	3.68%	3.68%	32673095133	5568517178	5,568,024,311 USDC	32,670,203,254 USDC
6	XRP	XRP	0.4871	3.89%	3.89%	27095879971	1041998204	2,138,850,992 XRP	55,618,185,850 XRP
7	Dogecoin	DOGE	0.1247	10.58%	10.58%	18049476098	679055289	5,447,611,363 DOGE	144,799,006,384 DOGE
8	Toncoin	TON	7.11	0.00%	0.00%	17469981416	277736602	39,079,395 TON	2,458,143,049 TON
9	Cardano	ADA	0.3856	0.23%	0.23%	13781478716	277343382	719,163,502 ADA	35,735,976,149 ADA

14. converting and Storing DataFrame in the form of a CSV file and opening the file in application:

In [69]:
```
Cryptocurrency.to_csv("Cryptocurrency_Data_Extraction.csv",index=False)
```

In [70]:
```
sm=pd.read_csv("Cryptocurrency_Data_Extraction.csv")
sm
```

Out[70]:

	Name	Symbol	Price	One_hour	One_Day	Market_Cap	Volume	Volume2	Circulating_Supply
0	Bitcoin	BTC	64129.3200	2.38%	2.38%	1265528372378	26581855560	414,100 BTC	19,714,787 BTC
1	Ethereum	ETH	3518.6000	4.15%	4.15%	430352856775	15826370697	4,496,768 ETH	122,276,728 ETH
2	Tether	USDT	0.9993	1.76%	1.76%	112813530690	54115336315	54,152,667,539 USDT	112,891,354,602 USDT
3	BNB	BNB	585.5900	0.03%	0.03%	86293400326	1784812516	3,052,485 BNB	147,583,751 BNB
4	Solana	SOL	132.1900	2.72%	2.72%	61071028075	2173347494	16,441,659 SOL	462,010,348 SOL
5	USDC	USDC	1.0000	3.68%	3.68%	32673095133	5568517178	5,568,024,311 USDC	32,670,203,254 USDC
6	XRP	XRP	0.4871	3.89%	3.89%	27095879971	1041998204	2,138,850,992 XRP	55,618,185,850 XRP
7	Dogecoin	DOGE	0.1247	10.58%	10.58%	18049476098	679055289	5,447,611,363 DOGE	144,799,006,384 DOGE
8	Toncoin	TON	7.1100	0.00%	0.00%	17469981416	277736602	39,079,395 TON	2,458,143,049 TON
9	Cardano	ADA	0.3856	0.23%	0.23%	13781478716	277343382	719,163,502 ADA	35,735,976,149 ADA

15. Accessing first Five Rows of Data Frame Using head()

In [71]:
```
sm.head()
```

Out[71]:

	Name	Symbol	Price	One_hour	One_Day	Market_Cap	Volume	Volume2	Circulating_Supply
0	Bitcoin	BTC	64129.3200	2.38%	2.38%	1265528372378	26581855560	414,100 BTC	19,714,787 BTC
1	Ethereum	ETH	3518.6000	4.15%	4.15%	430352856775	15826370697	4,496,768 ETH	122,276,728 ETH
2	Tether	USDT	0.9993	1.76%	1.76%	112813530690	54115336315	54,152,667,539 USDT	112,891,354,602 USDT
3	BNB	BNB	585.5900	0.03%	0.03%	86293400326	1784812516	3,052,485 BNB	147,583,751 BNB
4	Solana	SOL	132.1900	2.72%	2.72%	61071028075	2173347494	16,441,659 SOL	462,010,348 SOL

16.Accessing last Five Rows of Data Frame Using tail ()

```
In [72]: sm.tail()
```

Out[72]:

	Name	Symbol	Price	One_hour	One_Day	Market_Cap	Volume	Volume2	Circulating_Supply
5	USDC	USDC	1.0000	3.68%	3.68%	32673095133	5568517178	5,568,024,311 USDC	32,670,203,254 USDC
6	XRP	XRP	0.4871	3.89%	3.89%	27095879971	1041998204	2,138,850,992 XRP	55,618,185,850 XRP
7	Dogecoin	DOGE	0.1247	10.58%	10.58%	18049476098	679055289	5,447,611,363 DOGE	144,799,006,384 DOGE
8	Toncoin	TON	7.1100	0.00%	0.00%	17469981416	277736602	39,079,395 TON	2,458,143,049 TON
9	Cardano	ADA	0.3856	0.23%	0.23%	13781478716	277343382	719,163,502 ADA	35,735,976,149 ADA

17.Using Describe() get information about numeric columns

```
In [74]: sm.describe()
```

Out[74]:

	Price	Market_Cap	Volume
count	10.000000	1.000000e+01	1.000000e+01
mean	6837.580670	2.065129e+11	1.083264e+10
std	20159.876483	3.925438e+11	1.748872e+10
min	0.124700	1.378148e+10	2.773434e+08
25%	0.615150	2.031108e+10	7.697910e+08
50%	4.055000	4.687206e+10	1.979080e+09
75%	472.240000	1.061835e+11	1.326191e+10
max	64129.320000	1.265528e+12	5.411534e+10

18.Using info() get information about Dataframe

```
In [73]: sm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Name                10 non-null     object
 1   Symbol              10 non-null     object
 2   Price               10 non-null     float64
 3   One_hour            10 non-null     object
 4   One_Day             10 non-null     object
 5   Market_Cap          10 non-null     int64
 6   Volume              10 non-null     int64
 7   Volume2             10 non-null     object
 8   Circulating_Supply  10 non-null     object
dtypes: float64(1), int64(2), object(6)
memory usage: 848.0+ bytes
```

6.Conclusion

In conclusion, this cryptocurrency data scraping and analysis project aims to provide comprehensive insights into the cryptocurrency market. By systematically extracting, cleaning, and analyzing data from Coin Market Cap, the project enables informed decision-making, trend identification, and risk assessment. The development of automated tools and visualizations further enhances the efficiency and clarity of market monitoring. Overall, this project equips stakeholders with the necessary information and tools to navigate the volatile cryptocurrency landscape effectively.