Q.1) Write a python program to display IMDB's Top rated 100 Indian movies' data

https://www.imdb.com/list/ls056092300/ (https://www.imdb.com/list/ls056092300/) (i.e. name, rating, year ofrelease) and make data frame.

In [114]:
```
from bs4 import BeautifulSoup
import requests
import pandas as pd
```

In [121]:
```
import requests

url = 'https://www.imdb.com/list/ls056092300/'  # Replace this with the target

headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
}

response = requests.get(url, headers=headers)

if response.status_code == 200:
    print(response.text)  # Print the content of the response
else:
    print(f'Request failed with status code: {response.status_code}')
```

le/tt0169102/","name":"Lagaan: Once Upon a Time in India","description":"The people of a small village in Victorian India stake their future on a game of cricket against their ruthless British rulers.","image":"https://m.media-amazon.com/images/M/MV5BYTU0MTQ1OWYtZTUxNi00N2QxLTlmY2ItOTI0Yzg4ODE2MjI3XkEyXkFqcGdeQXVyMTI5Njg5OTQz._V1_.jpg","aggregateRating":{"@type":"AggregateRating","bestRating":10,"worstRating":1,"ratingValue":8.1,"ratingCount":121135},"contentRating":"U","genre":"Drama, Musical, Sport","duration":"PT3H44M"}},{"@type":"ListItem","item":{"@type":"Movie","url":"https://www.imdb.com/title/tt0048473/","name":"Pather Panchali","description":"Impoverished priest Harihar Ray, dreaming of a better life for himself and his family, leaves his rural Bengal village in search of work.","image":"https://m.media-amazon.com/images/M/MV5BMmQ0MTE0NzktZTg4OC00OTgzLTg2NzQtNGE3MWIyYTBiM2NhXkEyXkFqcGdeQXVyNzI1NzMxNzM@._V1_.jpg","aggregateRating":{"@type":"AggregateRating","bestRating":10,"worstRating":1,"ratingValue":8.2,"ratingCount":38762},"contentRating":"U","genre":"Drama","duration":"PT2H5M"}},{"@type":"ListItem","item":{"@type":"Movie","url":"https://www.imdb.com/title/tt0057935/","name":"Charulata","description":"The lonely wife of a newspaper editor falls in love with her visiting cousin-in-law, who shares her love for literature.","image":"https://m.media-amazon.com/images/M/MV5BYjc4YjJiOTMtOTk5MS00NDQwLWJlZjYtNjlkNDgwOGIxNzRmXkEyXkFqcGdeQXVyMTA2ODMzMDU@._V1_.jpg","aggr

In [122]:
```
soup=BeautifulSoup(response.content)
soup
titles=[]
Ratings=[]
YearOfRelease=[]
for i in soup.find_all('div',class_="ipc-title ipc-title--base ipc-title--title
    titles.append(i.text)
for i in soup.find_all('div',class_="sc-e2dbc1a3-0 ajrIH sc-b189961a-2 fkPBP d
    Ratings.append(i.text)
for i in soup.find_all('div',class_="sc-b189961a-7 feoqjK dli-title-metadata")
    YearOfRelease.append(i.text)

df=pd.DataFrame({'Titles':titles,'Ratings':Ratings,'Year_of_Release':YearOfRele
df
```

Out[122]:

	Titles	Ratings	Year_of_Release
0	1. Ship of Theseus	8.0 (7.8K)Rate	20122h 19mUA
1	2. Iruvar	8.4 (7.4K)Rate	19972h 20mUA
2	3. Kaagaz Ke Phool	7.8 (2.8K)Rate	19592h 28mU
3	4. Lagaan: Once Upon a Time in India	8.1 (121K)Rate	20013h 44mU
4	5. Pather Panchali	8.2 (39K)Rate	19552h 5mU
5	6. Charulata	8.1 (7.3K)Rate	19641h 57mU
6	7. Rang De Basanti	8.1 (123K)Rate	20062h 47mUA
7	8. Dev.D	7.9 (32K)Rate	20092h 24mA
8	9. 3 Idiots	8.4 (437K)Rate	20092h 50mUA
9	10. Awaara	7.8 (4.5K)Rate	19513h 13mU
10	11. Nayakan	8.7 (24K)Rate	19872h 36mA
11	12. Aparajito	8.2 (16K)Rate	19561h 50mU
12	13. Pushpaka Vimana	8.6 (3.8K)Rate	19872h 5mU
13	14. Pyaasa	8.3 (7.3K)Rate	19572h 26mU
14	15. Ghatashraddha	7.5 (164)Rate	19772h 24mA
15	16. Sholay	8.1 (59K)Rate	19753h 24mU
16	17. Aradhana	7.6 (2K)Rate	19692h 49mU
17	18. Do Ankhen Barah Haath	8.4 (1.9K)Rate	19572h 23mU
18	19. Bombay	8.1 (14K)Rate	19952h 21mU
19	20. Neecha Nagar	6.6 (278)Rate	19462h 2mU
20	21. Do Bigha Zamin	8.3 (2.3K)Rate	19532h 11mU
21	22. Garm Hava	8.0 (838)Rate	19742h 26m(Banned)
22	23. Piravi	7.8 (297)Rate	19891h 50mUnrated
23	24. Mughal-E-Azam	8.1 (8.9K)Rate	19603h 17mPG
24	25. Amma Ariyan	7.4 (219)Rate	19861h 55m

Q.2) Write a python program to scrape details of all the posts from https://www.patreon.com/coreyms (https://www.patreon.com/coreyms) .Scrape the

heading, date, content and the likes for the video from the link for the youtube video from the post.

In [102]:
```
import requests
from bs4 import BeautifulSoup

url = 'https://www.patreon.com/coreyms'
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')

posts = soup.find_all('div', class_='post')

for post in posts:
    heading = post.find('h2').text
    date = post.find('time')['datetime']
    content = post.find('div', class_='post__content').text
    youtube_link = post.find('a', class_='post__youtube-link')['href']
    likes = post.find('span', class_='post__likes').text

print(f'Heading: {heading}')
print(f'Date: {date}')
print(f'Content: {content}')
print(f'Likes: {likes}')
print('\n')
```

```
Heading: []
Date: []
Content: ['I appreciate any support! Anyone who is a contributor through Patr
eon will have the option to be added to my Contributors page on my personal w
ebsite (coreyms.com)\xa0', 'All of my content is available for free on YouTub
e, but if you support at this tier then I will give you early access to video
s days before they go public. I will post links to Patreon and YouTube when a
n early access video is available.\xa0You will also have access to all of the
rewards for my lower tiers. In total, you will receive:Early Access to Videos
Be added to my Contributors page on my personal website (coreyms.com)', 'I wi
ll now be adding Patron credits to the end of each of my videos where I list
all of my current Patrons who support me at this level or higher. The names w
ill be listed with my biggest supporters towards the top.\xa0You will also ha
ve access to all of the rewards for my lower tiers. In total, you will receiv
e:Your Name Listed at the End of My VideosEarly Access to Videos\xa0Be added
to my Contributors page on my personal website (coreyms.com)', 'Be featured o
n every page of my personal website (coreyms.com) as a top contributor. You w
ill also have access to all of the rewards for my lower tiers. In total, you
will receive:Be Featured as a Top Contributor on\xa0my personal website (core
yms.com)Your Name Listed at the End of My VideosEarly Access to Videos']
Likes: []
```

Q.3 Write a python program to scrape house details from mentioned URL. It should include house title, location,

area, EMI and price from https://www.nobroker.in/ (https://www.nobroker.in/) .Enter three localities which are Indira Nagar, Jayanagar, Rajaji Nagar.

In [132]:
```
url = 'https://www.nobroker.in'
page=requests.get(url)

soup=BeautifulSoup(page.text, 'html.parser')
soup
house_title=[]
location=[]
area=[]
emi=[]
price=[]
for i in soup.find_all('span',class_="overflow-hidden overflow-ellipsis whitesp
    house_title.append(i.text)
for i in soup.find_all('div',class_="mt-0.5p overflow-hidden overflow-ellipsis
    location.append(i.text)
for i in soup.find_all('div',class_="p-1.5p flex border-b border-b-solid borde
    area.append(i.text.split("₹")[1].replace("sqftBuiltup",""))
for i in soup.find_all('div',class_="p-1.5p flex border-b border-b-solid borde
    emi.append(i.text.split("₹")[2].replace("/MonthEstimated EMI",""))
for i in soup.find_all('div',class_="p-1.5p flex border-b border-b-solid borde
    price.append(i.text.split("₹")[3])

df = pd.DataFrame({"Name of House":house_title,"Location":location,"Area":area
df
```

Out[132]:

	Name of House	Location	Area	EMI	Price

Q.4 Write a python program to scrape first 10 product details which include product name , price , Image URL from

https://www.bewakoof.com/bestseller?sort=popular (https://www.bewakoof.com/bestseller?sort=popular) .

In [58]:
```
import requests
from bs4 import BeautifulSoup

url='https://www.bewakoof.com/bestseller?sort=popular'
response=requests.get(url)

soup=BeautifulSoup(response.text,'html.parser')
products = []
product_elements = soup.find_all('div', class_='productCardBox', limit=10)

for product_element in product_elements:
    product = {}

        # Find the product name
    name = product_element.find('h3')
    product['name'] = name.text.strip() if name else 'No name'

        # Find the product price
    price = product_element.find('span', class_='discountedPriceText')
    product['price'] = price.text.strip() if price else 'No price'

        # Find the image URL
    image = product_element.find('img', class_='productImgTag')
    product['image_url'] = image['src'] if image else 'No image URL'

    products.append(product)
products
```

Out[58]:

```
[{'name': 'Bewakoof®',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-red-moon-rider-graphic
-printed-oversized-t-shirt-502017-1715257498-1.jpg'},
 {'name': 'Bewakoof®',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/women-s-purple-i-don-t-give-
a-sip-graphic-printed-oversized-t-shirt-516919-1715257908-1.jpg'},
 {'name': 'Bewakoof®',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/women-s-purple-music-is-my-e
scape-graphic-printed-t-shirt-519260-1715258076-1.jpg'},
 {'name': 'bewakoof x dc',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-red-the-batman-graphic
-printed-oversized-t-shirt-480316-1715257617-1.jpg'},
 {'name': 'Bewakoof®',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-purple-beast-within-gr
aphic-printed-oversized-t-shirt-581488-1715257463-1.jpg'},
 {'name': 'bewakoof x marvel',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-blue-moon-knight-graph
ic-printed-oversized-t-shirt-519226-1715257436-1.jpg'},
 {'name': 'Bewakoof®',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-white-keep-listening-t
ypography-oversized-t-shirt-519181-1708612763-1.jpg'},
 {'name': 'Bewakoof®',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-black-warriors-graphic
-printed-t-shirt-511263-1717060488-1.jpg'},
 {'name': 'bewakoof x dc',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-black-batman-riddle-bm
l-oversized-t-shirt-479765-1715257722-1.jpg'},
 {'name': 'Bewakoof®',
  'price': 'No price',
  'image_url': 'https://images.bewakoof.com/t640/men-s-blue-peace-not-war-gra
phic-printed-oversized-t-shirt-479634-1715257979-1.jpg'}]
```

Q.5 Please visit https://www.cnbc.com/world/?region=world (https://www.cnbc.com/world/?region=world) and scrap a) headings b) date c) News link

In [71]:
```
import requests
from bs4 import BeautifulSoup

url = 'https://www.cnbc.com/world/?region=world'
response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, 'html.parser')
news_items = []
article_elements = soup.find_all('div', class_='Card-standardBreakerCard')

for article in article_elements:

        # Find the heading
    heading = article.find('a', class_='Card-title')
    news['heading'] = heading.text.strip() if heading else 'No heading'

        # Find the date
    date = article.find('time', class_='Card-time')
    news['date'] = date['datetime'].strip() if date and 'datetime' in date.attr

        # Find the news link
    link = heading['href'] if heading and 'href' in heading.attrs else 'No link
    news['link'] = f"https://www.cnbc.com{link}" if link else 'No link'

    news_items.append(news)
news_items
```

Out[71]:

```
[{'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
```

```
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
```

```
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'},
 {'heading': 'What is the World Economic Forum?',
  'date': 'No date',
  'link': 'https://www.cnbc.comhttps://www.cnbc.com/video/2024/01/15/what-is-the-world-economic-forum.html'}]
```

Q.6 Please visit https://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/most-downloadedarticles/ (https://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/most-downloadedarticles/)

and scrapa) Paper title b) date c) Author

In [131]:
```
import requests
from bs4 import BeautifulSoup

url ="https://www.keaipublishing.com/en/journals/artificial-intelligence-in-agr

response = requests.get(url, headers=headers)
soup = BeautifulSoup(response.text, 'html.parser')
articles = []
article_elements = soup.find_all('div', class_='article-item')

for article in article_elements:
    article_details = {}

        # Find the paper title
    title = article.find('h2', class_='article-title')
    article_details['title'] = title.text.strip() if title else 'No title'

        # Find the date
    date = article.find('span', class_='article-date')
    article_details['date'] = date.text.strip() if date else 'No date'

        # Find the authors
    authors = article.find('span', class_='article-authors')
    article_details['authors'] = authors.text.strip() if authors else 'No autho

    articles.append(article_details)
articles
```

Out[131]:
```
[]
```

In [ ]:
```

```