In [24]:

```
from bs4 import BeautifulSoup as BS
import pandas as pd
import requests
import webbrowser
import urllib.request
def getReviews(url):
    req = urllib.request.Request(url)
    req.add_header('User-Agent', 'Mozilla/5.0 (Windows NT 10.0; Win64;
    req.add_header('Accept', 'text/html,application/xhtml+xml,applicati
    req.add_header('Accept-Language', 'en-US,en;q=0.5')
    response = urllib.request.urlopen(req)
    html_code = response.read().decode('utf-8')
    soup=BS(html_code,'html.parser')
    totalReviews=soup.find('span',class_='biGQs _P pZUbB KxBGd').text[:
    hotelName=soup.find('h1',id='HEADING').text
    reviewHeads=[i.text for i in soup.find_all('span',class_='JbGkU Cj'
    reviewContent=[i.text for i in soup.find_all('span',class_='orRIx C
    reviewerNdate = [i.text for i in soup.find_all('div', class_='tVWyV
    reviewerandDate=[]
    for i in reviewerNdate:
        reviewerandDate.append(i.split('wrote a review '))
    finalDate=[]
    finalreviewer=[]
    for i in reviewerandDate:
        finalDate.append(i[1])
        finalreviewer.append(i[0])
    ratings=[i.find('title').text[0:3] for i in soup.find_all('div',cla
    tempHotelLocation= [i.text for i in soup.find_all('span',class_='bi
    hotelLocation=tempHotelLocation[1]
    df = pd.DataFrame({
        'Hotel Name': hotelName,
        'Hotel Location': hotelLocation,
        'Reviewer': finalreviewer,
        'Rating':ratings,
        'Date of Review': finalDate,
        'Review':reviewHeads,
        'Complete Review':reviewContent
    })
    return [df,totalReviews,hotelName,hotelLocation]

def getPureReviews(url,hotelName,hotelLocation):
    req = urllib.request.Request(url)
    req.add_header('User-Agent', 'Mozilla/5.0 (Windows NT 10.0; Win64;
    req.add_header('Accept', 'text/html,application/xhtml+xml,applicati
    req.add_header('Accept-Language', 'en-US,en;q=0.5')
    response = urllib.request.urlopen(req)
    html_code = response.read().decode('utf-8')
    soup=BS(html_code,'html.parser')
    reviewHeads=[i.text for i in soup.find_all('span',class_='JbGkU Cj'
    reviewContent=[i.text for i in soup.find_all('span',class_='orRIx C
    reviewerNdate = [i.text for i in soup.find_all('div', class_='tVWyV
    reviewerandDate=[]
    for i in reviewerNdate:
        reviewerandDate.append(i.split('wrote a review '))
    finalDate=[]
    finalreviewer=[]
    for i in reviewerandDate:
        finalDate.append(i[1])
        finalreviewer.append(i[0])
    ratings=[i.find('title').text[0:3] for i in soup.find_all('div',cla
    df = pd.DataFrame({
```

```
        'Hotel Name': hotelName,
        'Hotel Location': hotelLocation,
        'Reviewer': finalreviewer,
        'Rating':ratings,
        'Date of Review': finalDate,
        'Review':reviewHeads,
        'Complete Review':reviewContent
    })
    return df
```

In [25]:

```
url='https://www.tripadvisor.in/Hotel_Review-g297698-d3633245-Reviews-M
split_url = url.split("Reviews-")
first_part = split_url[0] + "Reviews-or"
second_part = "-" + split_url[1]
result = getReviews(url)
df=result[0]
totalReviews=result[1]
hotelName=result[2]
hotelLocation=result[3]
urls = [f"{first_part}{i}{second_part}" for i in range(10, int(totalRev
for i in urls:
    newdf=getPureReviews(i,hotelName,hotelLocation)
    df = pd.concat([df, newdf], ignore_index=True)
df
```

```
---------------------------------------------------------------------
------
HTTPError                                 Traceback (most recent call
last)
Cell In [25], line 12
     10 urls = [f"{first_part}{i}{second_part}" for i in range(10, in
t(totalReviews)-10, 10)]
     11 for i in urls:
---> 12     newdf=getPureReviews(i,hotelName,hotelLocation)
     13     df = pd.concat([df, newdf], ignore_index=True)
     14 df

Cell In [24], line 46, in getPureReviews(url, hotelName, hotelLocatio
n)
     44 req.add_header('Accept', 'text/html,application/xhtml+xml,app
lication/xml;q=0.9,image/avif,image/webp,*/*;q=0.8')
     45 req.add_header('Accept-Language', 'en-US,en;q=0.5')
---> 46 response = urllib.request.urlopen(req)
     47 html_code = response.read().decode('utf-8')
     48 soup=BS(html_code,'html.parser')

File c:\users\viren\appdata\local\programs\python\python38\lib\urllib
\request.py:222, in urlopen(url, data, timeout, cafile, capath, cadef
ault, context)
    220 else:
    221     opener = _opener
--> 222 return opener.open(url, data, timeout)

File c:\users\viren\appdata\local\programs\python\python38\lib\urllib
\request.py:531, in OpenerDirector.open(self, fullurl, data, timeout)
    529 for processor in self.process_response.get(protocol, []):
    530     meth = getattr(processor, meth_name)
--> 531     response = meth(req, response)
    533 return response

File c:\users\viren\appdata\local\programs\python\python38\lib\urllib
\request.py:640, in HTTPErrorProcessor.http_response(self, request, r
esponse)
    637 # According to RFC 2616, "2xx" code indicates that the clien
t's
    638 # request was successfully received, understood, and accepte
d.
    639 if not (200 <= code < 300):
--> 640     response = self.parent.error(
    641         'http', request, response, code, msg, hdrs)
    643 return response

File c:\users\viren\appdata\local\programs\python\python38\lib\urllib
\request.py:569, in OpenerDirector.error(self, proto, *args)
    567 if http_err:
    568     args = (dict, 'default', 'http_error_default') + orig_arg
s
--> 569     return self._call_chain(*args)

File c:\users\viren\appdata\local\programs\python\python38\lib\urllib
\request.py:502, in OpenerDirector._call_chain(self, chain, kind, met
h_name, *args)
    500 for handler in handlers:
    501     func = getattr(handler, meth_name)
--> 502     result = func(*args)
    503     if result is not None:
```

```
    504         return result

File c:\users\viren\appdata\local\programs\python\python38\lib\urllib
\request.py:649, in HTTPDefaultErrorHandler.http_error_default(self,
req, fp, code, msg, hdrs)
    648 def http_error_default(self, req, fp, code, msg, hdrs):
--> 649     raise HTTPError(req.full_url, code, msg, hdrs, fp)

HTTPError: HTTP Error 403: Forbidden
```

In [26]:
```
df
```

Out[26]:

	Hotel Name	Hotel Location	Reviewer	Rating	Date of Review	Review	Compl Revi
0	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	aandsaustralia	5.0	Nov 2023	One word: incredible	We w recommend this resort my wife's
1	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8	Ozmandapanda	5.0	Jan 2020	You won't want to leave.....	We took three you kids to t resort,

In [ ]:
```
df.to_excel('trial1.xlsx', index=False, engine='openpyxl')
```

In [12]:

```
url='https://www.tripadvisor.in/Hotel_Review-g297698-d304528-Reviews-Th
df=getReviews(url)
df[0]
```

Out[12]:

	Hotel Name	Hotel Location	Reviewer	Rating	Date of Review	Review	Complete Review
0	The Westin Resort Nusa Dua Bali	Kawasan Pariwisata Nusa Dua, Itdc Heavenly Spa...	Valdis	5.0	Jul 2020	Long term experience!	I stayed in Westin Nusa Dua for almost 2 month...
1	The Westin Resort Nusa Dua Bali	Kawasan Pariwisata Nusa Dua, Itdc Heavenly Spa...	Benny L	5.0	Sept 2020	Amazing staycation	We had a chance to take a short break for 2 ni...
2	The Westin Resort Nusa Dua Bali	Kawasan Pariwisata Nusa Dua, Itdc Heavenly Spa...	tommydju	5.0	Aug 2020	Excellent Service and Asistance by Westin Hote...	This is my first vacation after pandemic ..I c...

In [11]:
```
df
```

Out[11]:

	Hotel Name	Hotel Location	Reviewer	Rating	Date of Review	Review	Complete Review
0	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	aandsaustralia	5.0	Nov 2023	One word: incredible	We were recommended this resort by my wife's f...
1	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	Ozmandapanda	5.0	Jan 2020	You won't want to leave.....	We took our three young kids to this resort, a...
2	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	FrequentTraveler9	5.0	Mar 2022	Calming place, good for vacation	The room we got was at a quiet location with o...
3	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	Sinta Jakarta	5.0	Mar 2022	Amazing time @Mulia Resort	We had the short break during covid time, so w...
4	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	Hian Ong	4.0	Feb 2020	Like Gulliver in Brobdingnag	If this review was just about the restaurants,...
...	...	...	...	...	...	...	...
325	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	Shayla	5.0	Mar 2023	Kids Beach Play	I took my son to play in the kids section on t...
326	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	kiwi_simonphillip	5.0	Mar 2023	Honeymoon retreat	This place is amazing. Much bigger than we tho...
327	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	Sameer P	5.0	Nov 2022	A piece of heaven on Earth	Boy oh Boy, a true paradise!!! Exceeded expect...

	Hotel Name	Hotel Location	Reviewer	Rating	Date of Review	Review	Complete Review
328	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	indra p	5.0	Feb 2024	Very pleasent exp	5 out of 5 exp. Everything is perfect, the fac...
329	Mulia Resort	Jalan Raya Nusa Dua Selatan, Nusa Dua, Benoa 8...	Stephanie C	5.0	May 2023	Honeymoon	We stayed in the Signature Lagoon Room and it ...

330 rows × 7 columns

In [2]:

```
import asyncio
import requests
import aiohttp
import pandas as pd
from bs4 import BeautifulSoup as BS

headers = {
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:106.0)
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,im
    'Accept-Language': 'en-US,en;q=0.5'
}

# Function to extract data from a single page
def parse_reviews(html, hotel_name, hotel_location):
    soup = BS(html, 'html.parser')

    review_heads = [i.text for i in soup.find_all('span', class_='JbGkU
    review_content = [i.text for i in soup.find_all('span', class_='orR
    reviewer_n_date = [i.text for i in soup.find_all('div', class_='tVW

    reviewer_and_date = [i.split('wrote a review ') for i in reviewer_r
    final_dates = [i[1] for i in reviewer_and_date]
    final_reviewers = [i[0] for i in reviewer_and_date]

    ratings = [i.find('title').text[:3] for i in soup.find_all('div', c

    # Create a DataFrame for each page
    df = pd.DataFrame({
        'Hotel Name': hotel_name,
        'Hotel Location': hotel_location,
        'Reviewer': final_reviewers,
        'Rating': ratings,
        'Date of Review': final_dates,
        'Review': review_heads,
        'Complete Review': review_content
    })

    return df

# Asynchronous function to fetch a single page
async def fetch_page(session, url, hotel_name, hotel_location):
    async with session.get(url) as response:
        html = await response.text()
        return parse_reviews(html, hotel_name, hotel_location)

# Asynchronous function to fetch all review pages
async def fetch_all_reviews(urls, hotel_name, hotel_location):
    async with aiohttp.ClientSession(headers=headers) as session:
        tasks = [fetch_page(session, url, hotel_name, hotel_location) f
        results = await asyncio.gather(*tasks)

    # Combine all DataFrames into one
    return pd.concat(results, ignore_index=True)

# Main function to get the total reviews and the first page
def get_initial_reviews(url):
    response = requests.get(url, headers=headers)
    soup = BS(response.text, 'html.parser')

    total_reviews = int(soup.find('span', class_='biGQs _P pZUbB KxBGd'
    hotel_name = soup.find('h1', id='HEADING').text
```

```
    temp_hotel_location = [i.text for i in soup.find_all('span', class_
    hotel_location = temp_hotel_location[1]

    # Parse the first page reviews
    first_page_df = parse_reviews(response.text, hotel_name, hotel_loca

    return first_page_df, total_reviews, hotel_name, hotel_location

# Entry point
async def main():
    url = 'https://www.tripadvisor.in/Hotel_Review-g297698-d3633245-Rev

    # Get initial data from the first page
    first_page_df, total_reviews, hotel_name, hotel_location = get_init

    # Generate URLs for all remaining pages
    split_url = url.split("Reviews-")
    first_part = split_url[0] + "Reviews-or"
    second_part = "-" + split_url[1]
    urls = [f"{first_part}{i}{second_part}" for i in range(10, total_re

    # Fetch all pages asynchronously
    additional_reviews_df = await fetch_all_reviews(urls, hotel_name, h

    # Combine the first page and the rest
    full_df = pd.concat([first_page_df, additional_reviews_df], ignore_

    print(full_df)
    return full_df

# Run the asynchronous code in Jupyter or an environment supporting asy
df = await main()
```

```
---------------------------------------------------------------------
------
AttributeError                            Traceback (most recent call
last)
Cell In [2], line 93
     90     return full_df
     92 # Run the asynchronous code in Jupyter or an environment supp
orting async
---> 93 df = await main()

Cell In [2], line 75, in main()
     72 url = 'https://www.tripadvisor.in/Hotel_Review-g297698-d36332
45-Reviews-Mulia_Resort-Nusa_Dua_Benoa_South_Kuta_Bali.html'
     74 # Get initial data from the first page
---> 75 first_page_df, total_reviews, hotel_name, hotel_location = ge
t_initial_reviews(url)
     77 # Generate URLs for all remaining pages
     78 split_url = url.split("Reviews-")

Cell In [2], line 60, in get_initial_reviews(url)
     57 response = requests.get(url, headers=headers)
     58 soup = BS(response.text, 'html.parser')
---> 60 total_reviews = int(soup.find('span', class_='biGQs _P pZUbB
KxBGd').text[:-8].replace(',', ''))
     61 hotel_name = soup.find('h1', id='HEADING').text
     62 temp_hotel_location = [i.text for i in soup.find_all('span',
class_='biGQs _P pZUbB KxBGd')]

AttributeError: 'NoneType' object has no attribute 'text'
```

In [ ]: