

PROJECT REPORT ON – “FLIPKART REVIEW SCRAPER”

TITLE -

[“Flipkart Review Scrapper : From Web Scraping to Cloud Deployment”]
by

Vijay Gurung - (ID : MST03-0070)

Submitted to – [Meta_Scifor_Technologies_Pvt.Ltd]

Script. Sculpt. Socialize

www.YouTube.com/@SciforTechnologies

The image shows a YouTube channel page for 'Meta Scifor Technologies Pvt. Ltd.'. The channel has a blue logo featuring a stylized white bird or flame design. A 'Subscribed' button with a bell icon is visible. The channel name is displayed prominently in white text against a dark background. To the right, there is a 'Links' section with icons for LinkedIn, Instagram, Facebook, Twitter, Website, Telegram, and WhatsApp, each with a corresponding URL.

Links	
	LinkedIn linkedin.com/company/scifortechnologies
	Instagram instagram.com/scifortechnologies
	Facebook facebook.com/scifortechnologies
	Twitter twitter.com/scifortechn?l=z4KKnjQzEWpobUxp1gmNiw&s=03
	Website scifortechnologies.com
	Telegram t.me/SciforTechnologies
	WhatsApp wa.me/ 917676646007

UNDER THE GUIDANCE:

Trainer: Urooj Khan

Designation: AI Engineer | Data Scientist

Bangalore, India

SUBMITTED BY:

Name: Vijay Gurung

Batch: 03 (Data Science Intern)

Siliguri, India, Aug 2024

TABLE OF CONTENTS

1. Abstract	03
2. Introduction	04-05
3. Technology Used	06-08
4. Dataset Information	09
5. Methodology	10-12
6. Code Snippet	13-21
7. Deployment with AWS, Heroku and Azure	22-25
8. Final Results and Discussion	26
9. Conclusion	27
10. References	28

ABSTRACT

This project focuses on developing a "**Review Scraper Application**" aimed at extracting and analyzing customer reviews from the e-commerce platform "**Flipkart**". The application is built using Python and leverages libraries such as **BeautifulSoup** for web scraping and Flask for creating a web-based interface. The scraper is designed to automatically gather reviews for specified products, filtering out valuable information such as ratings, review titles and review content.

Once the scraping process is complete, the application presents the data in a structured format on a web interface, enabling users to view, download, or further analyze the reviews. The application is deployed on platforms like [**AWS, Heroku and Azure**] ensuring accessibility and scalability. This project showcases the seamless integration of web scraping techniques with modern web development practices, emphasizing its utility in fields such as market research, consumer behavior analysis, and e-commerce strategy development.

Keywords:

Web Scraping, Python, Flask, BeautifulSoup, AWS, Heroku, Azure, E-commerce, Data Analysis.

Introduction

In the digital era, data has become the cornerstone of decision-making processes, and businesses are increasingly relying on data-driven insights to gain a competitive edge. One of the critical sources of valuable data is customer feedback, often found in the form of online reviews. These reviews offer insights into consumer behavior, product satisfaction, and market trends. However, manually sifting through vast amounts of data can be time-consuming and inefficient. This is where web scraping and automated data analysis come into play.

1. Web Scraping for E-Commerce:

Web scraping is a technique used to extract data from websites, enabling the automated collection of large volumes of information. In the context of e-commerce, web scraping allows businesses to gather customer reviews, product details, and pricing information from online platforms like Flipkart. This information can then be analyzed to derive actionable insights, such as identifying popular products, understanding customer sentiment, and monitoring competitors.

2. Flask Framework for Web Development: To make these insights accessible, a web application can be developed using the Flask framework. Flask is a lightweight web framework in Python that facilitates the creation of web applications with minimal overhead. By integrating Flask with web scraping tools, we can build an application that not only scrapes and processes reviews but also presents the results through a user-friendly interface. This combination of web scraping and web development offers a powerful solution for businesses looking to harness the potential of online data.

3. Project Objective: The primary objective of this project is to develop a web-based application capable of scraping and analyzing customer reviews from Flipkart. Utilizing libraries such as BeautifulSoup for web scraping and Flask for web development, the application will automate the process of collecting, processing, and displaying reviews. The project also aims to deploy the application on cloud platforms such as AWS, Heroku, and Azure, ensuring scalability and accessibility.

4. Application in Business Intelligence: By automating the extraction and analysis of customer reviews, businesses can gain deep insights into consumer preferences and market trends. This information is invaluable for making informed decisions in product development, marketing strategies, and customer service. The deployment of this application not only showcases the technical integration of web scraping and web development but also emphasizes its practical application in enhancing business intelligence.

Summary

This project endeavors to create an automated solution for extracting and analyzing customer reviews from Flipkart. By leveraging Python's web scraping capabilities and Flask's web development framework, the project aims to deliver a scalable, accessible tool for businesses to derive meaningful insights from online customer feedback. The successful deployment of this application on cloud platforms further underscores the project's commitment to practical, real-world implementation of advanced data processing techniques.

Technology Used

The development, deployment, and management of the Review Scrapper Project (Flipkart) involve various technologies and tools. Each technology plays a critical role in different stages of the project, from web scraping and data processing to web application development and cloud deployment. Below is an overview of the technologies utilized in this project.

1. **Python Libraries:** Python is a versatile programming language widely used in web development, data analysis, and automation. The project leverages several Python libraries that are essential for web scraping, data handling, and building the web application.
2. **BeautifulSoup:** BeautifulSoup is a powerful library used for web scraping in Python. It enables the extraction of data from HTML and XML files by providing Pythonic idioms for navigating, searching, and modifying the parse tree.
 - **Role in the Project:** BeautifulSoup is employed to scrape customer reviews from Flipkart. It parses the HTML content of product pages, extracting relevant data such as review text, ratings, and user details. This data is then processed and stored for further analysis.
3. **Requests:** Requests is a simple and elegant HTTP library for Python, allowing users to send HTTP requests to access web content.
 - **Role in the Project:** Requests is used to send GET requests to Flipkart's web pages. It retrieves the HTML content that BeautifulSoup processes, making it a crucial component of the web scraping process.
4. **Flask Framework:** Flask is a lightweight web framework that is easy to set up and use. It provides the tools needed to create web applications in Python, making it a popular choice for developing scalable and maintainable applications.
 - **Role in the Project:** Flask serves as the backbone of the web application, managing the server-side logic. It handles routing, user requests, and the integration of scraped data into the web interface, allowing users to interact with the application seamlessly.
5. **PyMongo:** PyMongo is a Python driver for MongoDB, a NoSQL database used for storing and retrieving data.
 - **Role in the Project:** PyMongo connects the Flask application to a MongoDB database, where scraped reviews are stored. It allows the application to efficiently manage large volumes of data and perform operations such as querying and updating records.

6. **Gunicorn:** Gunicorn (Green Unicorn) is a Python WSGI HTTP server for UNIX. It is used to run Python web applications in a production environment.
 - **Role in the Project:** Gunicorn is employed to serve the Flask application in a production environment. It handles multiple requests concurrently, ensuring that the application remains responsive under heavy load.
7. **Jinja2:** Jinja2 is a templating engine for Python, used to create dynamic web pages by embedding Python expressions within HTML.
 - **Role in the Project:** Jinja2 is integrated with Flask to render HTML templates, allowing the web application to dynamically display content, such as customer reviews, based on user input and database queries.
8. **PyCharm IDE:** PyCharm is a professional IDE for Python development, offering advanced features for code editing, debugging, and project management.
 - **Role in the Project:** PyCharm is the primary development environment used for coding the project. Its features like intelligent code completion, syntax highlighting, and integrated version control facilitate efficient development and collaboration.
9. **Deployment Tools:** The project is deployed across multiple cloud platforms, each requiring specific tools and configurations for successful deployment.
 - **AWS Elastic Beanstalk:** AWS Elastic Beanstalk is used to deploy the application on Amazon Web Services, providing scalability and easy management of the deployment environment.
 - **Heroku:** Heroku is another platform used to deploy the application, offering a simple and effective way to host the web app.
 - **Azure:** The project is also deployed on Microsoft Azure, taking advantage of its cloud computing capabilities for running the application.
10. **Streamlit (Deployment and Development Tool):** Streamlit is an open-source app framework that allows for the rapid development of web applications. It is particularly useful for deploying data science and machine learning models as interactive web apps.
 - **Role in the Project:** Although primarily used for deployment, Streamlit could also be utilized to create an intuitive user interface for displaying scraped reviews or providing additional interactive features. This potential addition highlights Streamlit's flexibility in both deployment and front-end development.

Summary:

The Review Scrapper Project (Flipkart) is built on a robust technology stack that integrates Python libraries, web scraping tools, web frameworks, and cloud deployment services. This combination of technologies ensures that the project is scalable, efficient, and accessible, allowing users to extract and analyze valuable customer feedback from Flipkart with ease. The successful implementation of these technologies demonstrates the power of modern web development and data processing tools in solving real-world problems.

Dataset Information

Dataset: Flipkart Customer Reviews Dataset

Source: Scrapped from Flipkart website using **BeautifulSoup** and **Requests** libraries.

Details:

1. **Total Number of Reviews Scraped:** Varies based on the product category and availability.
2. **Data Points:**
 - **Review Text:** Customer's written review of the product.
 - **Rating:** Numerical rating (typically on a scale of 1 to 5) provided by the customer.
 - **User Name:** The name or identifier of the reviewer.
 - **Date of Review:** The date when the review was posted.
 - **Product Name:** The name of the product being reviewed.
 - **Additional Features:** Information such as the purchase verification status, helpful votes, and product specifications (if available).

This dataset is dynamically generated based on the product category chosen for scraping. It is used to train and test the review analysis and sentiment classification components of the project. The diversity and richness of the dataset make it a valuable resource for understanding customer sentiment and feedback trends on Flipkart.

Methodology

The methodology section provides a comprehensive overview of the systematic approach taken to develop and deploy the Flipkart Review Scrapper project. The process is broken down into distinct phases, each playing a crucial role in ensuring the effectiveness and accuracy of the web scraping, sentiment analysis, and deployment. This section details the data scraping techniques, Flask application structure, and deployment strategies employed in the project.

1. Data Scraping:

Data scraping is a critical step in collecting raw data for input into the review analysis model. It involves extracting relevant information from Flipkart's website to build a robust dataset for further processing.

A. Scraping Process:

- **Objective:** To gather customer reviews, ratings, and other relevant data for analysis.
- **Tools Used:** BeautifulSoup and Requests libraries in Python.
- **Process:**
 - ❖ **Sending HTTP Requests:** The project uses the Requests library to send GET requests to Flipkart product pages, retrieving the HTML content.
 - ❖ **Parsing HTML:** The BeautifulSoup library is employed to parse the HTML content and extract specific elements such as review texts, ratings, user names, and review dates.
 - ❖ **Data Extraction:** The parsed data is filtered and stored in a structured format, typically in a Pandas DataFrame, for further processing and analysis.
 - ❖ **Handling Pagination:** The scraper handles multiple pages of reviews by iterating through pagination links, ensuring comprehensive data collection.

2. Data Preprocessing:

Data preprocessing is essential to clean and prepare the scraped data for sentiment analysis and further processing.

B. Text Cleaning:

- **Objective:** To remove noise and irrelevant information from the review text.
- **Process:**
 - ❖ **Lowercasing:** All text is converted to lowercase to maintain consistency.
 - ❖ **Removing Punctuation and Special Characters:** Punctuation, special characters, and stopwords are removed to focus on the core content of the reviews.

- ❖ **Tokenization and Lemmatization:** The text is split into individual tokens (words), and lemmatization is applied to reduce words to their base forms, aiding in accurate sentiment analysis.

3. Flask Application Architecture:

The project utilizes Flask, a lightweight web framework, to build and deploy the review analysis web application. The architecture is designed for efficiency and scalability.

C. Application Structure:

- **Objective:** To create a user-friendly web interface for interacting with the review scraper and sentiment analysis tool.
- **Components:**
 - ❖ **Routes:** Flask routes are defined to handle user requests and display the web pages. Key routes include:
 - ❖ **Home Route:** Displays the landing page with options to input the Flipkart product URL.
 - ❖ **Scrape Route:** Initiates the scraping process and displays the collected reviews.
 - ❖ **Analyze Route:** Conducts sentiment analysis on the scraped reviews and presents the results to the user.
 - ❖ **Templates:** HTML templates are used to render web pages, ensuring a cohesive and user-friendly interface.
 - ❖ **Error Handling:** The application includes error handling mechanisms to manage issues such as invalid URLs, failed HTTP requests, or missing data.

4. Deployment:

Deploying the Flask application is a crucial step to make the project accessible to users via the web.

D. Deployment Strategies:

- **Platform:** The project is deployed on AWS, Heroku, and Azure, utilizing their cloud infrastructure for hosting the application.
- **Process:**
 - ❖ **Containerization:** The application is containerized using Docker, ensuring consistency across different deployment environments.
 - ❖ **Environment Configuration:** Environment variables and dependencies are managed using a requirements.txt file, ensuring the application is correctly configured during deployment.
 - ❖ **CI/CD Pipeline:** For continuous integration and deployment, services like GitHub Actions or AWS CodePipeline are employed, automating the deployment process and ensuring smooth updates.

Conclusion:

The methodology outlined above highlights the systematic approach taken to ensure the successful implementation of the Flipkart Review Scrapper project. From meticulous data scraping to the design of a robust Flask application, every step contributes to the project's ability to extract, analyze, and present customer reviews effectively. The deployment strategy further emphasizes usability and accessibility, allowing users to interact with the application seamlessly across multiple platforms.

CODE SNIPPET

Line by Line Explained codes :

The screenshot shows a Jupyter Notebook interface with the title "Review_Scrapper_Project". The code cell [1] contains the following Python script:

```
# Import Important libraries
from flask import Flask , render_template , request , jsonify
from bs4 import BeautifulSoup as bs
from urllib.request import urlopen as urReq
import requests
flipcart_url = "https://www.flipkart.com/search?q=+"+iphone11"
flipcart_url
'response_website = urReq(flipcart_url)
data_flipcart = response_website.read()
bs(data_flipcart,"html.parser")
beautifyed_html
beautifyed_html.find_all("div", {"class": "cPHDOP col-12-12"})
```



```

[46]: bigbox[0].div.div

[46]: <div class="_30Wiq+ "><div class="rgHxCQ"><span>Filters</span></div></div>

[48]: bigbox[6].div.div

[48]: <div data-id="MOBFKCTSAYWFJ5" style="width:100%"><div class="tUxFRH"><a class="CgtC98" href="/apple-iphone-11-black-128-gb-includes-earpods-power-adapt
er/p/item06bac28995200?pid=MOBFKCTSAYWFJ5&lid=LSTMOBFKCTSAYWFJ5D45UTB&marketplace=FLIPKART&q=iphone11&srno=s_1_5&
mp;otracker=search&fm=organic&iid=a8ccc544-dfb9-404d-a9bf-9e08e33b4ce6.MOBFKCTSAYWFJ5.SEARCH&ppt=None&ppn=None&ssid=318sodcdsw00000
01724177823760&qh=d6db477051465f9a" rel="noopener noreferrer" target="_blank"><div class="Otbg5D"><div class="yPq5Io"><div><div class=".4WE1SP" style
="height:20px; width:20px"></div><
div><div class="DShtpz"><span class="vfSpSs">Currently unavailable</span></div><div class="A8uQd"><span class="Ln197G"><label
class="tJjCVx"><input class="vn9L2C" readonly="" type="checkbox"/><div class="XqNaEv"></div></label><span class="uu79Xy"><span>Add to Compare</span></
label></div></div><div class="oUss6M ssUU08"><div class="#E521"><svg class="N1bADF" height="16" viewBox="0 20 16" width="16" xmlns="http://www.w
3.org/2000/svg"><path class="x1UMqG" d="M8.695 16.682C4.06 12.382 1 9.536 1 6.065 1 3.219 3.178 1 5.95 1c.566 0 3.069.746 4.05 1.915C10.981 1.745 12.484
14.05 1 16.822 1 19.606C0 3.471-3.06 6.316-7.695 10.61L10 17.8971-1.305-1.215z" fill="#2874F0" fill-rule="evenodd" opacity=".9" stroke="#FFF"/></
path></svg></div></div><div class="KfJk" row><div class="col col-7-12"><div class="KzDHZ">Apple iPhone 11 (Black, 128 GB) (Includes EarPod
s, Power Adapter)</div></div><div class="_50esEi"><span class="Y1Hw08" id="productRating_LSTMOBFKCTSAYWFJ5D45UTB_MOBFKCTSAYWFJ5" ><div class="XQDdHH">4.6i
mg class="Rza2QY" src="
RiIgZD0i1TTuNSA5LjQz0WwMy42NzQgM14yMy45NC00LjI2LTMuMjEtM140DmGNC4yNTQtLjQwEw2LjuMTEybDEuNjkjgNC4wMSA0LjI2eiIvPjwvc3Np
g==" /></div><span><span class="Wphh3N"><span><span>2,05,188 Ratings </span><span class="hG7V4" >&#8226;</span><span>11,441 Reviews</span></span></span></div><
div><div class="6NEsgJ"><ul class="G4BRas"><li class="J+igdf">128 GB ROM</li><li class="J+igdf">15.49 cm (6.1 inch) Liquid Retina HD Display</li><li cla
ss="J+igdf">12MP + 12MP Front Camera</li><li class="J+igdf">A13 Bionic Chip Processor</li><li class="J+igdf">Brand Warranty for 1 Year</li><li cla
ss="J+igdf">12MP + 12MP Front Camera</li><li class="J+igdf">A13 Bionic Chip Processor</li><li class="J+igdf">Brand Warranty for 1 Year</li><li>
</li></ul></div></div><div class="cNyY0"><div class="h105eU"><div class="N9bqj_4b5Dlr" >₹29,999</div><div class="yRaYj_ZYyWLA">
₹<!-- -->54,900</div><div class="UkUFWk"><span>45% off</span></div></div><div class="0CSTHy"></div><div class="M4DNwV"><div class="h5v9c" style="padding-top:4px;padding-right:4px;padding-bottom:4px;padding-left:
4px;border-radius:2px;background-color:#E7F8EC"><div class="yiggsN_05Fpg8" style="color:#000000;font-size:14px;font-style:normal;font-weight:40
0" >upto </div><div class="yiggsN_05Fpg8" style="color:#000000;font-size:14px;font-style:normal;font-weight:700" >₹19,800</div><div class="yiggsN_05Fpg8" s
tyle="color:#000000;font-size:14px;font-style:normal;font-weight:400" Off on Exchange</div></div></div></div></div></div>
```

```

[50]: bigbox[6].div.div.div

[50]: <div class="tUxFRH"><a class="CgtC98" href="/apple-iphone-11-black-128-gb-includes-earpods-power-adapter/p/item06bac28995200?pid=MOBFKCTSAYWFJ5&lid=
LSTMOBFKCTSAYWFJ5D45UTB&marketplace=FLIPKART&q=iphone11&srno=s_1_5&otracker=search&fm=organic&iid=a8ccc544-dfb9-404d-a9bf-9e08e
33b4ce6.MOBFKCTSAYWFJ5.SEARCH&ppt=None&ppn=None&ssid=318sodcdsw000001724177823760&qh=d6db477051465f9a" rel="noopener noreferrer" target
=_blank"><div class="Otbg5D"><div class="yPq5Io"><div><div class="4WE1SP" style="height:20px; width:20px"><div class="tJjCVx"><input class="vn9L2C" readonly="" type="checkbox"/><div class="XqNaEv"></div></label><span class="uu79Xy"><span>Add to Compare</span></label></div><div class="oUss6M ssUU08"><div class="#E521"><svg class="x1UMqG" d="M8.695 16.682C4.06 12.382 1
9.536 1 6.065 1 3.219 3.178 1 5.95 1c.566 0 3.069.746 4.05 1.915C10.981 1.745 12.484 14.05 1 16.822 1 19.606C0 3.471-3.06 6.316-7.695 10.61
L10 17.8971-1.305-1.215z" fill="#2874F0" fill-rule="evenodd" opacity=".9" stroke="#FFF"/></div><span><span class="6NEsgJ"><ul class="G4BRas">
<li class="J+igdf">12MP + 12MP Front Camera</li><li class="J+igdf">A13 Bionic Chip Processor</li><li class="J+igdf">Brand Warranty for 1 Year</li><li>
</li></ul></div></div><div class="cNyY0"><div class="h105eU"><div class="N9bqj_4b5Dlr" >₹29,999</div><div class="yRaYj_ZYyWLA">
₹<!-- -->54,900</div><div class="UkUFWk"><span>45% off</span></div></div><div class="0CSTHy"></div><div class="M4DNwV"><div class="h5v9c" style="padding-top:4px;padding-right:4px;padding-bottom:4px;padding-left:
4px;border-radius:2px;background-color:#E7F8EC"><div class="yiggsN_05Fpg8" style="color:#000000;font-size:14px;font-style:normal;font-weight:40
0" >upto </div><div class="yiggsN_05Fpg8" style="color:#000000;font-size:14px;font-style:normal;font-weight:700" >₹19,800</div><div class="yiggsN_05Fpg8" s
tyle="color:#000000;font-size:14px;font-style:normal;font-weight:400" Off on Exchange</div></div></div></div></div></div>
```

```

[56]: bigbox[6].div.div.div.a['href']

[56]: '/apple-iphone-11-black-128-gb-includes-earpods-power-adapter/p/item06bac28995200?pid=MOBFKCTSAYWFJ5&lid=LSTMOBFKCTSAYWFJ5D45UTB&marketplace=FLIPKART&
q=iphone11&store=tty%2F4io&srno=s_1_5&otracker=search&fm=organic&iid=a8ccc544-dfb9-404d-a9bf-9e08e33b4ce6.MOBFKCTSAYWFJ5.SEARCH&ppt=None&ppn=None&ssid=
318sodcdsw000001724177823760&qh=d6db477051465f9a'

[74]: import requests

[96]: "https://www.flipkart.com" + bigbox[6].div.div.div.a['href']

[96]: 'https://www.flipkart.com/apple-iphone-11-black-128-gb-includes-earpods-power-adapter/p/item06bac28995200?pid=MOBFKCTSAYWFJ5&lid=LSTMOBFKCTSAYWFJ5D45U
TB&marketplace=FLIPKART&n=inhone11&store=tty%2F4io&srno=s_1_5&otracker=search&fm=organic&iid=a8ccc544-dfb9-404d-a9bf-9e08e33b4ce6.MOBFKCTSAYWFJ5.SEARCH'

```

```
[98]: product6 = "https://www.flipkart.com" + bigbox[6].div.div.div.a['href']

[100]: # Add headers

[102]: headers = {
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.5735.110 Safari/537.36",
    "Referer": "https://www.flipkart.com/",
    "Accept-Language": "en-US,en;q=0.9",
    "Connection": "keep-alive"
}

[104]: # Send the request with headers

[112]: product66 = requests.get(product6, headers=headers)

[114]: # Check the status code
       # This should return 200

[116]: print(response.status_code)
       200

[118]: product6

[118]: 'https://www.flipkart.com/apple-iphone-11-black-128-gb-includes-earpods-power-adapter/p/item06bac28995200?pid=MOBFKCTSYAPWYFJ5&lid=LSTMOBFKCTSYAPWYFJ5D45U
TB&marketplace=FLIPKART&q=iphone11&store=tty%2F4io&srno=s_1_5&otracker=search&fm=organic&iid=a8ccc544-dfb9-404d-a9bf-9e08e33b4ce6.MOBFKCTSYAPWYFJ5.SEARCH
&ppt=None&ppn=None&ssid=318sodcdsw000001724177823760&qH=d6db477051465f9a'

[120]: product66.encoding = 'utf-8'

[126]: bs(product66.text, "html.parser")

[126]: <!DOCTYPE html>
<html lang="en"><head><link href="https://rukminim2.flixcart.com" rel="preconnect"/><link href="/static-assets-web.flixcart.com/fk-p-linchpin-web/fk-
cp-zion/css/atlas.chunk.f9cc90.css" rel="stylesheet"/><link href="/static-assets-web.flixcart.com/fk-p-linchpin-web/fk-cp-zion/css/app_modules.chunk.
c48a12.css" rel="stylesheet"/><link href="/static-assets-web.flixcart.com/fk-p-linchpin-web/fk-cp-zion/css/app.chunk.67c888.css" rel="stylesheet"/></m
eta content="text/html; charset=utf-8" http-equiv="Content-type"/><meta content="IE=Edge" http-equiv="X-UA-Compatible"/><meta content="102988293558" n
```

Next Step:

```
[129]: product6_page = bs(product66.text, "html.parser")

[131]: product6_page.find_all("div", {"class": "RcXBOT"})

K6mk aQymJL"><svg class="U6FW-N aQymJL" height="15" width="20" xmlns="http://www.w3.org/2000/svg"><path class="kX6Hbt" d="M9.58.006c-.41.043-.794.32-1.01.278-.277.557-2.334 4.693-2.74 5.1-.41.407-.944.6-1.544.6v.572h.75.45 0 .835-.28 1.007-.665 0 2.207-6.234 2.207-6.834 0-.6-.47-1.072-1.07-1.072h-3.216c-.6 0-1.07-.535-1.07-1.07 0-.537.835-3.387 1.006-3.944.17-.557-.107-1.157-.664-1.35-.15-.043-.257-.086-.407-.064zM6 6.434v.572h.143V6.434h0z" fill="#fff" fill-rule="evenodd"></path></svg><span class="t19VpF">312</span></div><div class="23B1I2"><svg class="gouvhg" height="11" viewbox="0 16 27" width="6" xmlns="http://www.w3.org/2000/svg"><path class="wRjFe4" d="M16 23.207l6.11 13.161 16 3.093 12.955 0 0 13.161l12.955 13.161z" fill="#fff"></path></svg><div class="Pgd+O_0je3Z"><div class="NsXBt8"><a href="https://reviews.MOBFW06BKRYBPSX8:286" reviewId=27d5ecd9-94c3-4cbe-8caf-df608ab28e6"><span>Permalink</span></a></div><div class="NsXBt8_1F9SL9"><span>Report Abuse</span></div></div></div></div></div>, <div class="RcXBOT" style="background-image: url(<https://rukminim1.flixcart.com/blobio/124/124/imr-201911_bblobio-imr-201911_84e3b4c3c3184fa9bec65713e4b3b60c.jpeg?q=90>)"><div class="col EPCmJX"><div class="row"><div class="XQdDH Ga3i8K"></div><p class="z9E0IG">Best in the market!</p></div><div class="ZmyHeo"><div class="Good Camera"><div class="WYmpv"><span>READ MORE</span></div></div></div><div class="xmAgz5 pWVA7t"><div class="Be4x5X d517go" style="background-image: url(<https://rukminim1.flixcart.com/blobio/124/124/imr-201911_bblobio-imr-201911_84e3b4c3c3184fa9bec65713e4b3b60c.jpeg?q=90>)"><url( wIDt1MSAyLTJWNMwLltM5t0ylT1mtptCAXNS45MmLtjAyLjAzLS4wNi4Wn10uMdgudM1lyLjA4TDiuMDggmgxNy44M2MuMDuMD1uMDyMuDgudMh2MTMuQIMjB6Ii8+pHbdGggZ D0iTEWeidEyLjxtDcuNSA5LjUgjNcAxNggxNgwntC41LTZ6Ii8+C9nPjwvc3ZnPg==);width:62px;height:62px;"></div></div><div class="row ghqwab"><div class="row" style="background-image: url(<https://rukminim1.flixcart.com/blobio/124/124/imr-201911_bblobio-imr-201911_84e3b4c3c3184fa9bec65713e4b3b60c.jpeg?q=90>)"><p class="2NsDsF AwS1Ca">Steewan Ratra </p><div class="VjlQyz" height="14" viewBox="0 0 12 12" width="14" style="background-image: url(<https://rukminim1.flixcart.com/blobio/124/124/imr-201911_bblobio-imr-201911_84e3b4c3c3184fa9bec65713e4b3b60c.jpeg?q=90>)"><circle cx="6" cy="6" fill="#878787" r="6"></circle><path d="M3 6l 2 2 4-4" fill="#878787" stroke="#FFF" stroke-width="1.5"></path></div><div class="MztJPV" style="background-image: url(<https://rukminim1.flixcart.com/blobio/124/124/imr-201911_bblobio-imr-201911_84e3b4c3c3184fa9bec65713e4b3b60c.jpeg?q=90>)"><span>Read More</span></div></div></div></div>

[133]: all_review = product6_page.find_all("div", {"class": "RcXBOT"})

[188]: len(all_review)

[188]: 11

[192]: all_review[5]

[192]: <div class="RcXBOT d7A196"><div class="col"><div class="col EPCmJX"><div class="row"><div class="XQdDH Ga3i8K"></div><p class="z9E0IG">Perfect product!</p></div><div class="ZmyHeo"><div><div class="Good all"><div><span>READ MORE</span></div></div></div><div class="xmAgz5 pWVA7t"><div class="Rza2QY d517go" style="background-image: url(<https://rukminim1.flixcart.com/blobio/124/124/imr-201911_bblobio-imr-201911_84e3b4c3c3184fa9bec65713e4b3b60c.jpeg?q=90>)"><url( wIDt1MSAyLTJWNMwLltM5t0ylT1mtptCAXNS45MmLtjAyLjAzLS4wNi4Wn10uMdgudM1lyLjA4TDiuMDggmgxNy44M2MuMDuMD1uMDyMuDgudMh2MTMuQIMjB6Ii8+pHbdGggZ D0iTEWeidEyLjxtDcuNSA5LjUgjNcAxNggxNgwntC41LTZ6Ii8+C9nPjwvc3ZnPg==);width:62px;height:62px;"></div></div></div></div>
```

Rating_For_Particular_product

```
[143]: all_review[5].div.div.div.div

[143]: <div class="XQDdhH Ga3i8K">5<div class="row"><div class="XQDdHh Ga3i8K"><5>>Perfect product!</p></div><div class="row"><div class="S="ZnyHeo"><div><div class="WtYmpv"><span>READ MORE</span></span></div></div><div class="xmAgz5 pVVA7"><div class="Be4x5s d517go" style="background-image:url(&lt;https://rukminim1.flixcart.com/blob/o/124/124/1m-/202104_596d455b3784d4af8495b34142e5ef.jpeg?q=90,&gt;, url( g=>);width:16px; height:62px;"></div></div><div class="row ghQhw8"><div class="row"><p class="2NsdsF AwS1CA">Kuldeep Kumar</p><div class="Vj1Qyz" height="14" viewbox="0 0 12 12" width="14" xmlns="http://www.w3.org/2000/svg"><g><circle cx="6" cy="6" fill="#878787" r="6"></circle><path d="M3 6l2 2 4-4" fill="#878787" stroke="#FFF" stroke-width="1.5"></path></g><svg><p class="MzJtPv" id="review-243d4e3-06a3-4e70-a593-9fe6d9e195f0"><span>Certified Buyer</span><span>Kalka</span></p><div><div class="2NsdsF" Apr, 2021</div><div class="MTK1V"><div class="row" xmlns="http://www.w3.org/2000/svg"><path class="kX6Hbt" d="M9.58.006c-.41.043-.794.32-1.01.728-.277.557-2.334.4.693-2.74.5.1-41.487-.944.6-1.544.68.572h.5c.45.0.835-.281.007-.665.0.0.2207.6.2342.207.6.834.0-.6-.47-1.072-1.07-1.072h-3.216c.6.0-1.07-.535-1.07-1.07.0-.537,.835-3.3871.006-3.944.17-.557-.107-1.157-.664-1.35-.15-.043-.257-.086-.407-.064zM6.434v6.572h2.143v6.434Hz" fill="#ffff" fill-rule="evenodd"></path></div><div class="t19VpF" xmlns="http://www.w3.org/2000/svg"><path class="kX6Hbt" d="M9.58.006c-.41.043-.794.32-1.01.728-.277.557-.2344.693-2.74.51-.41.407-.944.6-1.544.68.572h.5c.45.0.835-.281.007-.665.0.0.2207.6.2342.207.6.834.0-.6-.47-1.072-1.07-1.072h-3.216c.6.0-1.07-.535-1.07-1.07.0-.537,.835-3.3871.006-3.944.17-.557-.107-1.157-.664-1.35-.15-.043-.257-.086-.407-.064zM6.434v6.572h2.143v6.434Hz" fill="#ffff" fill-rule="evenodd"></path></div><div class="t19VpF" xmlns="http://www.w3.org/2000/svg"><path class="kX6Hbt" d="M9.58.006c-.41.043-.794.32-1.01.728-.277.557-.2344.693-2.74.51-.41.407-.944.6-1.544.68.572h.5c.45.0.835-.281.007-.665.0.0.2207.6.2342.207.6.834.0-.6-.47-1.072-1.07-1.072h-3.216c.6.0-1.07-.535-1.07-1.07.0-.537,.835-3.3871.006-3.944.17-.557-.107-1.157-.664-1.35-.15-.043-.257-.086-.407-.064zM6.434v6.572h2.143v6.434Hz" fill="#ffff" fill-rule="evenodd"></path></div>
```

```
[153]: all_review[5].div.div.find_all("p", {"class": "_2NsDsF AwS1CA"})  
[153]: [

Kuldeep Kumar

]  
  
[155]: all_review[5].div.div.find_all("p", {"class": "_2NsDsF AwS1CA"})[0]  
[155]: <p class=" _2NsDsF AwS1CA">Kuldeep Kumar</p>  
  
[274]: name = all_review[5].div.div.find_all("p", {"class": "_2NsDsF AwS1CA"})[0].text  
[276]: name  
[276]: 'Kuldeep Kumar'
```

Comment Header

```
[164]: all_review[5].div.div.div  
  
[164]: <div class="row"><div class="XQDdMH Ga3i8K">5</div><p class="z9E0IG">Perfect product!</p></div>  
  
[166]: all_review[5].div.div.div.p  
  
[166]: <p class="z9E0IG">Perfect product!</p>  
  
[170]: comment_header = all_review[5].div.div.div.p.text  
  
[172]: comment_header  
  
[172]: 'Perfect product!'
```

Long_review_Extract

Long_review_Extract

```
[229]: all_review[5].div.find_all('div' , {'class': ""})  
  
[229]: [<div><div class="">V Good all</div><span class="wTYmpw"><span>READ MORE</span></span></div>, <div class="">V Good all</div>, <div></div>]  
  
[231]: all_review[5].div.find_all('div' , {'class': ""})[0].div  
  
[231]: <div class="">V Good all</div>  
  
[238]: Long_comment = all_review[5].div.find_all('div' , {'class': ""})[0].div.text  
  
[240]: Long_comment  
  
[240]: 'V Good all'
```

How to find just a [Date]

```
[242]: all_review[5].div.find_all("p", {"class": "_2NsDsF"})  
[242]: [<p class=_2NsDsF AwS1CA>Kuldeep Kumar</p>, <p class=_2NsDsF>Apr, 2021</p>]  
  
[244]: all_review[5].div.find_all("p", {"class": "_2NsDsF"})[1]  
[244]: <p class=_2NsDsF>Apr, 2021</p>  
  
[246]: date = all_review[5].div.find_all("p", {"class": "_2NsDsF"})[1].text  
[248]: date  
[248]: 'Apr, 2021'
```

Extract_The_prices

```
[253]: product6_page.find_all("div", {"class": "Nx9bqj CxhGGd"})  
  
[253]: [

₹29,999

]  
  
[255]: product6_page.find_all("div", {"class": "Nx9bqj CxhGGd"})[0]  
  
[255]: <div class="Nx9bqj CxhGGd">₹29,999</div>  
  
[259]: price = product6_page.find_all("div", {"class": "Nx9bqj CxhGGd"})[0].text  
  
[261]: price  
  
[261]: '₹29,999'
```

- what is offer available for this particular product.

- How to Load the imageses, everything possible
 - what is the Highling in this product
 - what is extra discount in this product
 - In web_Scrapping only things required is that, from where to where I am suppose to go that the only requirement for web_Scrapping

Now i am looking for to extract every review, for every reviews

- who has given comment, what was the comment
 - Here ["all_review"] extracted is basically a ["List"]

Now i am looking for to extract every review, for every reviews

- who has given comment, what was the comment
 - Here ["all review"] extracted is basically a ["List"]

[265]: all_review

```
b28e6a">><span>Permalink</span></a></div><div class="NsX8t8 _1F9SL9"><span>Report Abuse</span></div></div></div></div></div></div><div class="RcxBOT"><div class="col"><div class="col EPCMjX"><div class="row"><div class="XQDdHm Ga3i8K"></div><p class="z9F0IG">Best in the market!</p></div><div class="row"><div class="ZmyHeo"><div class=""><Good Camera</div><span class="TymVpp"><span>READ MORE</span></span></div></div><div class="xmAgz5 pVVA7t"><div class="B4e4x5 D517go" style="background-image: url(https://rukminim1.flixcart.com/blob/124/124/imr-201911/bloboi-imr-201911_84e3b4c3c1384fa9b6ec5713e4b36b0c.jpg?q=90), url(/image/svg+xml); base64,PHN2ZyB3wRA0dD0iMjIIHtg5zPSjodHrw0i8vd3d3lnczL9yZy8DwALn3LzvZIy+PggZmlsB0D1lzeU0Q5RCZgImzlslk1yWkL5PSJub256ZJvI48GFoBkPsKJnMjAgMeQyZeqCwgMAwIDegMCaydje0YzgM54LjkyM1AyD0iMthJMSA wIDITMSAyLTJWMmMwLTETMS0yLTItMnpMtCAxsNS45MmMtLjAyLjA2LS4wNi4wNi0uMDguMDhIMlYljA4TDIUmdggMnxNy44M2MuMDMuMDuMDyUyDyMuDgudh2MTMuODRIMjB6Ii8+PHBhdGggZ D0iTTWeDiyJlxUdxCSA5LjyUNCaNxGgxNgwtNrNc41lT26Ii8+PcN9pVbz3NpP==); width: 62px; height: 62px;"></div></div><div class="row ghqkwab"><div class="row"><p class="NsDsF Aw51CA">Steeven Ratra</p><div class="YlV0qz" height="14" viewBox="0 0 12 12" width="14" xmlns="http://www.w3.org/2000/svg"><g><circle cx="6" cy="6" fill="#878787" r="6"/><circle><path d="M3 6l2 4-4" fill="#878787" stroke="#FFF" stroke-width="1.5"/></path></g><svg class="MztJPv" id="review-f07b07-27db-4a6b-a15-45b16518ab8a0"><span>Certified Buyer</span><span>Ludhiana</span></div><div><div class="vMTK1V"><div class="row"><div class="qkmf-k"><div class="6k6Km6"><div class="U6FW-N" height="15" width="20" xmlns="http://www.w3.org/2000/svg"><path class="kX6HBt" d="M9.58 .006c-.41.843-.794.32-1.01.728-.277.557-2.334 4.693-.274 5.1-.41.407-.944.6-1.544.6v8.572h.75.45 0 .835-.28 1.0-.07-.665 0 .207-6.234 2.207-6.834 0 -.6-.47-1.072-1.07-1.072h-3.216c-.6 0-.07-.535-.1.07-1.07 0-.537.835-.3.387 1.006-.3.944.17-.557-.107-1.157-.664-1.35-.15-.043-.257-.086-.406zM0 6.434v8.572h2.143V6.434H0z" fill="#fff" fill-rule="evenodd"/></div></div><div class="6k6Km6 a0yMlm"><div class="t19Vpf" href="https://www.w3.org/2000/svg"><path class="kX6HBt" d="M9.58 .006c-.41.843-.794.32-1.01.728-.277.557-2.334 4.693-.274 5.1-.41.407-.944.6-1.544.6v8.572h.75.45 0 .835-.28 1.0-.07-.665 0 .207-6.234 2.207-6.834 0 -.6-.47-1.072-1.07-1.072h-3.216c-.6 0-.07-.535-.1.07-1.07 0-.537.835-.3.387 1.006-.3.944.17-.557-.107-1.157-.664-1.35-.15-.043-.257-.086-.406zM0 6.434v8.572h2.143V6.434H0z" fill="#fff" fill-rule="evenodd"/></div></div>
```

"The code below is from the Flask application inside PyCharm"

The screenshot shows the PyCharm IDE interface with the project 'ReviewFlask1' open. The left sidebar displays the project structure, including a 'templates' folder containing 'base.html', 'index.html', and 'results.html'. The main editor window shows the 'app.py' file with the following code:

```
from flask import Flask, render_template, request, jsonify
from flask_cors import CORS, cross_origin
import requests
from bs4 import BeautifulSoup as bs
from urllib.request import urlopen as uReq

app = Flask(__name__)

# Vijay Gurung
@app.route('/', methods=['GET']) # Route to display the home page
@cross_origin()
def home_page():
    return render_template("index.html")

# Vijay Gurung
@app.route('/review', methods=['POST', 'GET']) # Route to show the review comments in a web UI
@cross_origin()
def review():
    if request.method == 'POST':
        try:
            search_string = request.form['content'].replace(" ", "")
            flipkart_url = "https://www.flipkart.com/search?q=" + search_string
            u_client = uReq(flipkart_url)
            flipkart_page = u_client.read()

            flipkart_html = bs(flipkart_page, features="html.parser")
            bigboxes = flipkart_html.find_all(name="div", attrs={"class": "cPHDOP col-12-12"})
            del bigboxes[3]

            if not bigboxes:
                return 'No products found.'

            box = bigboxes[0]

```

The screenshot continues the 'app.py' file from the previous screen. The code block starts at line 19 and continues to line 87, handling the 'review' route's POST method to extract product reviews from the Flipkart page.

```
def review():
    box = bigboxes[0]
    product_link = f"https://www.flipkart.com{box.div.div.div.a['href']}"
    prod_res = requests.get(product_link)
    prod_res.encoding = 'utf-8'
    prod_html = bs(prod_res.text, features="html.parser")

    commentboxes = prod_html.find_all(name="div", attrs={'class': "RcXB0T"})

    filename = f"{search_string}.csv"
    with open(filename, "w", encoding='utf-8') as fw:
        headers = "Product, Customer Name, Rating, Heading, Comment\n"
        fw.write(headers)
        reviews = []

        for commentbox in commentboxes:
            try:
                price = prod_html.find_all(name="div", attrs={"class": "Nx9bqj_CxhGGd"})[0].text
                name = commentbox.div.div.find_all('p', {'class': '_2NsDsF_AwS1CA'})[0].text
            except (IndexError, AttributeError):
                name = 'No Name'

            try:
                rating = commentbox.div.div.div.div.text
            except AttributeError:
                rating = 'No Rating'

            try:
                comment_head = commentbox.div.div.div.p.text
            except AttributeError:
                comment_head = 'No Comment Heading'

            try:
                comtag = commentbox.div.div.find_all('div', {'class': ''})
                cust_comment = comtag[0].div.text

```

The screenshot shows a code editor interface with two tabs open: `app.py` and `requirements.txt`. The `app.py` file contains Python code for a Flask application. The `requirements.txt` file lists the dependencies required for the project.

```
app.py
19     def review():
20         with open(filename, "w", encoding='utf-8') as fw:
21             try:
22                 comtag = commentbox.div.div.find_all('div', {'class': ''})
23                 cust_comment = comtag[0].div.text
24             except (IndexError, AttributeError):
25                 cust_comment = 'No Comment'
26
27             my_dict = {
28                 "Price": price,
29                 "Product": search_string,
30                 "Name": name,
31                 "Rating": rating,
32                 "CommentHead": comment_head,
33                 "Comment": cust_comment
34             }
35
36             reviews.append(my_dict)
37
38         return render_template('results.html', reviews=reviews)
39
40     except Exception as e:
41         print(f'The Exception message is: {e}')
42         return 'Something went wrong'
43
44
45     return render_template('index.html')
46
47
48 if __name__ == "__main__":
49     app.run(debug=True)
```

```
requirements.txt
1 beautifulsoup4==4.9.1
2 bs4==0.0.1
3 certifi==2020.6.20
4 chardet==3.0.4
5 click==7.1.2
6 Flask==1.1.2
7 Flask-Cors==3.0.9
8 gunicorn==20.0.4
9 idna==2.10
10 itsdangerous==1.1.0
11 Jinja2==2.11.2
12 MarkupSafe==1.1.1
13 requests==2.31.0
14 six==1.15.0
15 soupsieve==2.0.1
16 urllib3==1.26.14
17 Werkzeug==1.0.1
18 pymongo==4.6.0
```

The screenshot shows a code editor interface with two tabs open: `app.py` and `requirements.txt`. The `app.py` file contains Python code for a Flask application. The `requirements.txt` file lists the dependencies required for the project.

```
app.py
19     def review():
20         with open(filename, "w", encoding='utf-8') as fw:
21             try:
22                 comtag = commentbox.div.div.find_all('div', {'class': ''})
23                 cust_comment = comtag[0].div.text
24             except (IndexError, AttributeError):
25                 cust_comment = 'No Comment'
26
27             my_dict = {
28                 "Price": price,
29                 "Product": search_string,
30                 "Name": name,
31                 "Rating": rating,
32                 "CommentHead": comment_head,
33                 "Comment": cust_comment
34             }
35
36             reviews.append(my_dict)
37
38         return render_template('results.html', reviews=reviews)
39
40     except Exception as e:
41         print(f'The Exception message is: {e}')
42         return 'Something went wrong'
43
44
45     return render_template('index.html')
46
47
48 if __name__ == "__main__":
49     app.run(debug=True)
```

```
requirements.txt
1 beautifulsoup4==4.9.1
2 bs4==0.0.1
3 certifi==2020.6.20
4 chardet==3.0.4
5 click==7.1.2
6 Flask==1.1.2
7 Flask-Cors==3.0.9
8 gunicorn==20.0.4
9 idna==2.10
10 itsdangerous==1.1.0
11 Jinja2==2.11.2
12 MarkupSafe==1.1.1
13 requests==2.31.0
14 six==1.15.0
15 soupsieve==2.0.1
16 urllib3==1.26.14
17 Werkzeug==1.0.1
18 pymongo==4.6.0
```

REVIEWS				
Product	Name	Rating	Comment Heading	Comments
iphone15	Arunji Govindaraju	5	Excellent	Awesome product very happy to hold this. Better In hand feel,matte finish. Good camera, Brighter screen with Dynamic Island, USB-C, great battery life.60Hz display is a dealbreaker for some. Its not much noticeable.Thank u Flipkart for the best deal and quick delivery.
iphone15	Ajin V	5	Mind-blowing purchase	High quality camera 😍
iphone15	Siddharth Malhotra	5	Perfect product!	Nice product with all features specially its looks and camera battery backup is good You should buy it without any issues
iphone15	Mousam Guha Roy	4	Good choice	Very nice
iphone15	mohit yadav	5	Classy product	Nice ❤️
iphone15	Prithivi Boruah	5	Just wow!	Camera Quality Is Improved Loving It
iphone15	Sheetla Prasad Maurya	4	Worth the money	Best mobile phone Camera quality is very nice Battery backup is very good Sound quality is amazing.
				Switch from OnePlus to iPhone I am stunned with

REVIEWS

Product	Name	Rating	Comment Heading	Comments
iphone16	Priteshkumar Rana	5	Terrific	Nice phone with great camera. My 1st iPhone and I m loving it
iphone16	TAPAL SADDAM	5	Brilliant	Super phone . camera is super 🤩
iphone16	Nagorao Ghagare Male	5	Brilliant	SUPERBGo for it guys without hesitation, great in hand feel ,, superb camera , awesome battery backup , Starlight Colour looks premium, 6.7 inch crisp display, Camera 5/5 performance 5/5 Very happy to have apple 14 plus,Thanks flipcart for genuine product delivered on time
iphone16	Rizwan malik	5	Highly recommended	Good quality and service amezing phone
iphone16	A K M Abdullah	5	Classy product	Excellent iPhone 14 plus ❤️ 🌟
iphone16	Anil Kumar Bore Gowda	5	Terrific purchase	Camera is really good and battery performance also too good. Now apple has inbuilt photo edit options which is really nice to make your photos much better one.
iphone16	subham kumar Kachhap	5	Simply awesome	Just wow I cannot express my feelings ❤️ 😍
iphone16	Faiz Khan	5	Perfect product!	Its good phone, value for the

AWS Deployed :

The screenshot shows the AWS CodePipeline console with the following details:

- Pipeline Name:** review_scrap
- Pipeline type:** V2
- Execution mode:** QUEUED
- Source Stage:** Succeeded (GitHub (Version 2))
 - Last successful run: 14d39502 (1 minute ago)
 - View details button
- Deploy Stage:** Succeeded (AWS Elastic Beanstalk)
 - Last successful run: Just now
 - View details button
- Pipeline execution ID:** dc265614-2a99-4c4d-a380-348dc252ae50

Final Results and Discussion: [AWS, Heroku, and Azure]

The project involves two main components: a web scraping script for extracting product reviews from Flipkart and a Flask application to present these reviews via a web interface. The results of these components are summarized below:

1. Web Scraping (Part 1):

- **Functionality:** The web scraping script successfully extracts product reviews for a specific product ("iphone11") from Flipkart. It captures details such as reviewer names, ratings, comments, review dates, and long reviews.
- **Accuracy:** The script demonstrated high accuracy in extracting the desired review elements, with precise extraction of reviewer names, ratings, comments, and other relevant details from the HTML content.
- **Challenges:**
 - ❖ Handling dynamic content and ensuring that the data extraction is resilient to changes in the website's HTML structure.
 - ❖ Managing large volumes of data and ensuring the robustness of the scraping logic against different types of reviews and page structures.

2. Flask Application (Part 2):

- **Functionality:** The Flask application serves as a web interface for users to search for products and view reviews. It integrates the web scraping logic to dynamically fetch and display product reviews based on user input.
- **Deployment:** The application is designed to be hosted on cloud platforms like AWS, Heroku, and Azure, making it accessible for users to interact with the model and view real-time results.
- **Challenges:**
 - ❖ Ensuring seamless integration between the front-end (HTML templates) and the back-end scraping logic.
 - ❖ Handling user inputs robustly and providing informative error messages when issues arise during data fetching or processing.

Discussion:

The integration of the web scraping script with the Flask application represents a practical approach to creating a user-friendly interface for accessing and analyzing product reviews. The project effectively demonstrates several key aspects:

1. Efficiency and Accuracy:

- The web scraping script is efficient in retrieving and processing product reviews. It successfully parses HTML and extracts relevant data, ensuring that the information is accurate and comprehensive.
- The Flask application efficiently renders this data, providing a responsive and interactive user experience.

2. Scalability and Deployment:

- By deploying the application across multiple cloud platforms (AWS, Heroku, and Azure), the project showcases its scalability and versatility. Each platform offers different benefits, such as AWS's superior scalability and Heroku's simplicity, while Azure provides a balanced approach.
- The choice of cloud platforms allows for flexibility in deployment and demonstrates the application's ability to handle various operational environments.

3. User Experience:

- The web interface created with Flask is intuitive and easy to navigate, allowing users to search for products and view reviews seamlessly.
- The application's ability to handle different user inputs and provide feedback on the review retrieval process enhances the overall user experience.

4. Future Improvements:

- **Enhancing Data Extraction:** Future improvements could include more advanced parsing techniques to handle dynamic content and JavaScript-rendered elements, potentially incorporating tools like Selenium.
- **User Interface:** Further enhancements to the Flask front-end could include interactive features such as search filters, sorting options, and improved visualizations of review data.
- **Error Handling:** Improving error handling and providing more detailed feedback to users in case of failures or missing data could enhance the robustness of the application.

In conclusion, the project successfully integrates web scraping with a web application framework to deliver a practical tool for extracting and presenting product reviews. The results highlight the effectiveness of the solution and its deployment across various cloud platforms, providing a solid foundation for further development and enhancements

Conclusion

In summary, the "Review Scraper Project" effectively demonstrates the application of web scraping and web development technologies to create a practical tool for automated review collection and presentation. The project's successful deployment across various cloud platforms underscores its scalability and adaptability. By providing a seamless and interactive experience for users, the project not only showcases technical proficiency but also offers valuable insights and functionalities that can benefit a wide range of users, from businesses seeking customer feedback to individuals making informed purchasing decisions.

References

- Flask Documentation: <https://flask.palletsprojects.com/>
- BeautifulSoup Documentation: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- Jinja2 Documentation: <https://jinja.palletsprojects.com/>
- Deployment References:

AWS Elastic Beanstalk: <http://awsreview-env-1.eba-dsbfwvem.eu-north-1.elasticbeanstalk.com/>

Heroku: <https://reviewscraping-32f3c51ba0bb.herokuapp.com/>

Azure App Services:

Github Link : [https://github.com/Vijaygurung1/Review_Scraping]