PRICE COMPARISON WEBSITE USING WEB SCRAPING ALGORITHM

MINOR PROJECT REPORT

By

Sanyog Dani [Reg No.: RA2211031010087]
Arush Sirotiya [Reg No.: RA2211031010092]

Under the guidance of

Dr. M. Manickam

In partial fulfilment for the Course

of

21CSC203P – ADVANCED PROGRAMMING PRACTICE

in Department of Networking and Communications

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this minor project report for the course **21CSC203P ADVANCED PROGRAMMING PRACTICE** entitled in "**Price Comparison Website Using Web-scraping Algorithm**" is the bonafide work of **Sanyog Dani (RA2211031010087)** and **Arush Sirotiya (RA2211031010092)** who carried out the work under my supervision.

SIGNATURE

Dr. Manickam. M
Assistant Professor,
Department of
Networking and Communication,
School of Computing,
SRM Institute of Science and Technology
Kattankulathur

SIGNATURE

Dr. Annapurani. K
Head of the Department,
Department of
Networking and Communication,
School of Computing,
SRM Institute of Science and Technology
Kattankulathur

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Price comparison platforms are specifically designed to assess the cost of commodities and services from a variety of providers, aiding customers in their selection of products that offer the most savings when shopping online. Given the fast-paced lives of urban residents, a significant portion of consumers favor online purchases as a means to conserve time.

They can conveniently peruse these price comparison websites and determine the best source for the items they require. Optimal deals are prominently showcased. While not all consumers exclusively shop online, it remains an effective method for enhancing consumer price awareness.

Consequently, this benefits consumers who are consistently informed about the prevailing prices of particular items, safeguarding them from potentially misleading claims made by physical stores. In addition, this platform provides a valuable opportunity for grocers and retailers to promote their merchandise. This project seeks to empower consumers with the knowledge and tools needed to make informed purchasing decisions, ultimately improving their financial well-being and shopping experiences.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement.

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professors Dr. Vadivu. G , Professor, Department of Data Science and Business Systems and Dr. Sasikala. E Professor, Department of Data Science and Business Systems** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our Course project Faculty **Dr. Manickam. M, Assistant Professor, Department of Networking and Communications, School of Computing School of Computing,** for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr. Annapurani Panaiyappan .K (Networking And Communication)** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Motivation

In the contemporary digital landscape, it's become a common practice for every internet user to seek out the most advantageous bargains when contemplating a purchase. Among the paramount determinants that sway the decision to acquire a product is its price or cost. Prospective buyers habitually engage in a comprehensive price evaluation prior to making a purchase. Nevertheless, the process of scouring multiple price comparison websites for a specific item can be quite laborious. Hence, a compelling solution to streamline and automate this entire procedure is imperative.

Enter the necessity for an automated system that expedites this intricate task, saving users valuable time and effort in their pursuit of the best deals. This innovative solution is poised to revolutionize the online shopping experience, enabling consumers to effortlessly access and compare prices across a multitude of platforms, rendering their purchasing decisions more informed and economically advantageous. By harnessing the power of automation and cutting-edge technology, this project aims to redefine the way consumers engage with e-commerce and empower them to make savvy financial choices with ease and efficiency.

1.2 Objective

This platform serves as a valuable tool for comparing prices across diverse E-commerce websites. It caters specifically to the needs of frequent online shoppers, offering the convenience of centralized price comparisons from a multitude of online retailers. This system efficiently aggregates pricing data from various sellers, enabling users to pinpoint the most cost-effective source for their desired products. Once the data from these websites is collected, it is presented on the platform in the form of an easily navigable price comparison.

In the realm of E-commerce, applications consist of several essential components, including a database server, a web application server, and the Payment Gateway Interface (PGI) for facilitating online transactions. The pervasive influence of the internet has fundamentally transformed the way individuals and businesses approach their operations and decision-making processes. This project embodies the fusion of technology and consumer empowerment in the ever-evolving world of online shopping.

1.3 Problem Statement

An individual is seeking to purchase a Titan watch, but he's encountering varying price listings on different websites. Additionally, he's frustrated by intrusive ads and the risk of stumbling upon fraudulent websites, resulting in a significant waste of 15 to 30 minutes.

Develop an open-source software tool that helps users avoid intrusive ads, identify counterfeit and fraudulent websites, and obtain up-to-date, accurate prices for authentic and high-quality products from online retailers.

1.4 Challenges

Price comparison website developers encounter several challenges while creating and maintaining their platforms:

1. Data Aggregation: Gathering and updating links of different ecommerce and their classes can be challenging due to differences in data formats.
2. Data Accuracy: Ensuring the accuracy of price and product information is crucial to gain user trust, and this requires continuous data validation and cleansing.
3. Scalability: As the number of products and retailers increases, the platform needs to scale efficiently to handle the growing data volume and user traffic.
4. Website Performance: Maintaining a responsive and fast loading website, especially when dealing with large datasets, can be a technical challenge.
5. User Experience: Designing an intuitive and user-friendly interface is crucial for retaining and attracting users. This involves addressing navigation, search functionality, and overall usability.
6. Monetization: Finding a sustainable revenue model, such as affiliate marketing, sponsored listings, or premium subscriptions, is important to support the platform's operation.
7. Data Privacy and Security: Handling sensitive user data and adhering to privacy regulations is essential. Developers must implement strong security measures to protect both user data and the platform itself from cyber threats.
8. Competition: Staying competitive in the crowded price comparison website market and differentiating from other platforms can be challenging.
9. SEO and Marketing: Attracting users to the platform requires effective search engine optimization (SEO) and marketing strategies to increase visibility and drive organic traffic.

10. Legal Compliance: Adhering to copyright, trademark, and data usage laws when scraping data from third-party websites is essential to avoid legal issues.
11. Continuous Maintenance: Keeping the platform up-to-date with changing retailer websites, adding new features, and fixing bugs is an ongoing challenge.
12. Mobile Responsiveness: Ensuring the platform is mobile-friendly is increasingly important as more users access price comparison websites through smartphones and tablets.
13. Price Tracking: Implementing price tracking features to alert users when the price of a product drops or when a deal becomes available can be complex.
14. Handling User Reviews and Feedback: Incorporating user reviews and feedback effectively while maintaining their credibility and authenticity can be challenging.
15. Price Discrepancies: Dealing with situations where retailers provide different prices for the same product or have outdated information requires careful handling.
16. Web Scraping Challenges: Overcoming website structure changes and implementing efficient scraping techniques.
17. Database Management: Designing an optimized schema and maintaining data consistency.
18. User Experience: Ensuring the website is user-friendly and responsive.

Overcoming these challenges requires a combination of technical expertise, effective data management, user-centric design, and a commitment to maintaining data accuracy and user trust.

2. LITERATURE SURVEY

1. Intelligent Online Shopping using ML-based Product Comparison Engine

   This research work summarizes the product comparison website that implements intelligent web scraping. The website has a processing model that uses the Machine Learning (ML)-based product comparison engine. The simulation analysis for model training and testing uses the GitHub dataset and the Support Vector Machine (SVM) algorithm. The SVM classifies the product from various websites based on the rank value, assigned based on the selected features of the product. The proposed model is compared with other ML algorithms such as Decision Tree, Naïve Bayes, and Random Forest. The comparative analysis shows that the SVM algorithm can classify the best products with the highest accuracy ratio of 94.71%over other algorithms.

2. Comparative Study Of Various Scraping Tools: Pros And Cons

   In this paper, we cover the most powerful known web scraping tools available to date and study them. We have compared these tools and clarified their benefits as well as drawbacks concerning various applications.

3. Co-Mart - A Daily Necessity Price Comparison Application

   This paper focuses on an web application called Co-Mart, where one can compare prices of various products on different E-commerce websites and thus save one's Time, Efforts and Money. The methods that will be used for identifying the best deals will be web crawling and web scraping. The web scraping scripts will be written using python libraries and web crawling works on HTML labels. The framework will be designed using HTML (Hypertext markup language), CSS (Cascading style sheet), and JavaScript as front-end and Python and Django will be used for back-end support with SQLite as the Database Management System.

4. Web Scraping based Product Comparison Model for E-Commerce Websites

   The objective of this paper is to propose a web application that identifies those basic details for any product from different e-commerce websites. These details are compared, and the result is displayed to the user graphically for the final decision. The proposed web application uses the web-scraping methodology with selenium and is implemented on the python framework using various algorithms and techniques which are discussed in the paper. As an outcome, the system will give the simulation results, so that the user can get the recommendations on purchasing the relevant product with better user satisfaction and in minimum clicks and time.

3. REQUIREMENTS

Hardware Requirements:

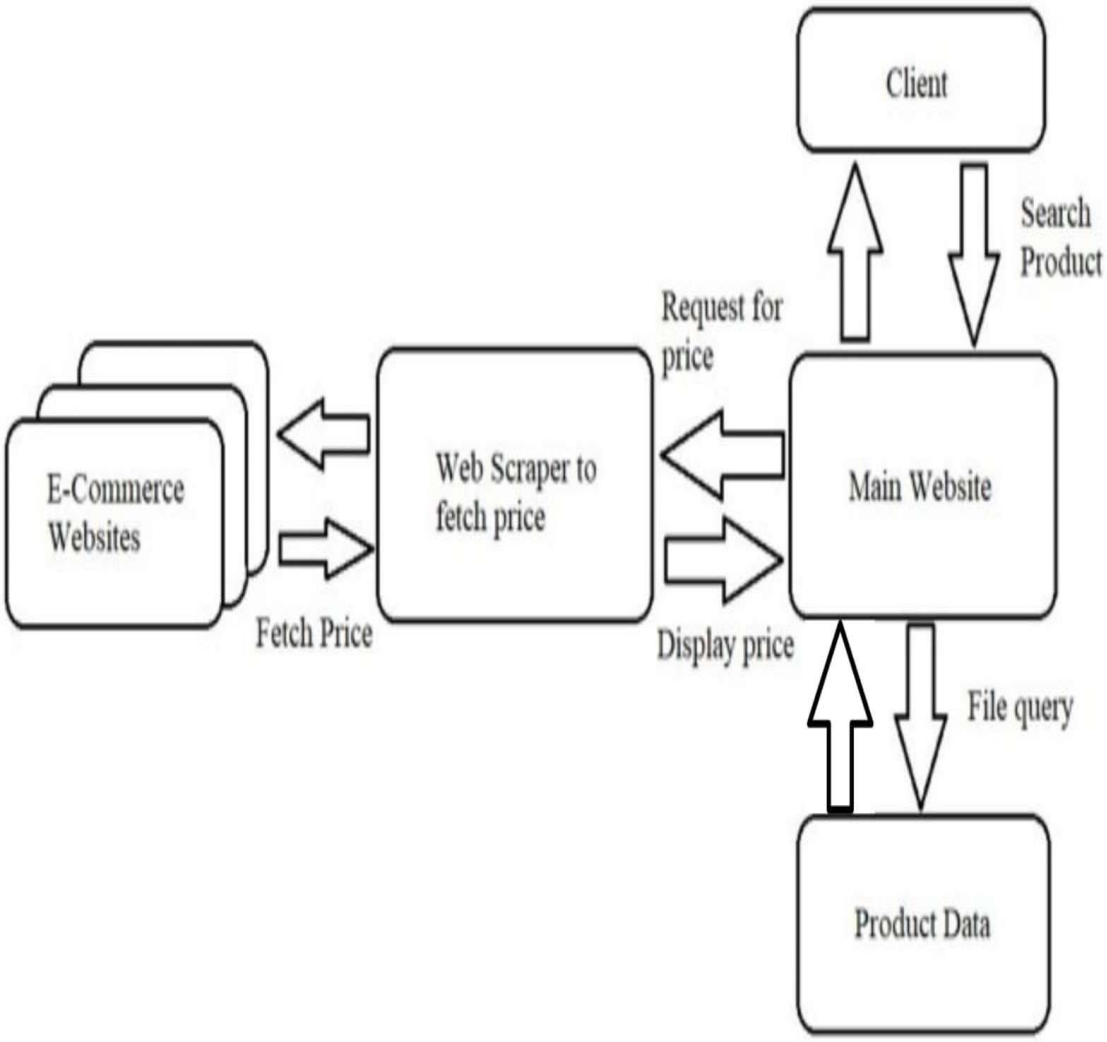
- A server with a fast processor and at least 8GB of RAM. This is necessary to handle the load of web scraping and serving the website to users.
- A large hard drive or SSD, depending on the amount of data you need to store.
- A reliable internet connection.
- A dedicated server from a reputable provider such as Amazon Web Services (AWS) or Google Cloud Platform (GCP).
- A Linux operating system such as Ubuntu or Debian.
- A reliable internet connection with at least 100 Mbps bandwidth.

Software Requirements:

- Python 3
- Beautiful Soup: A Python library for parsing HTML and XML documents.
- Requests: A Python library for making HTTP requests.
- A database management system (DBMS) such as MySQL or PostgreSQL to store the links and classes of a particular attribute.
- A web server such as Apache or Nginx to serve the website to users.

In addition to these requirements, it is also recommended to use a virtual private network (VPN) to protect your privacy when web scraping.

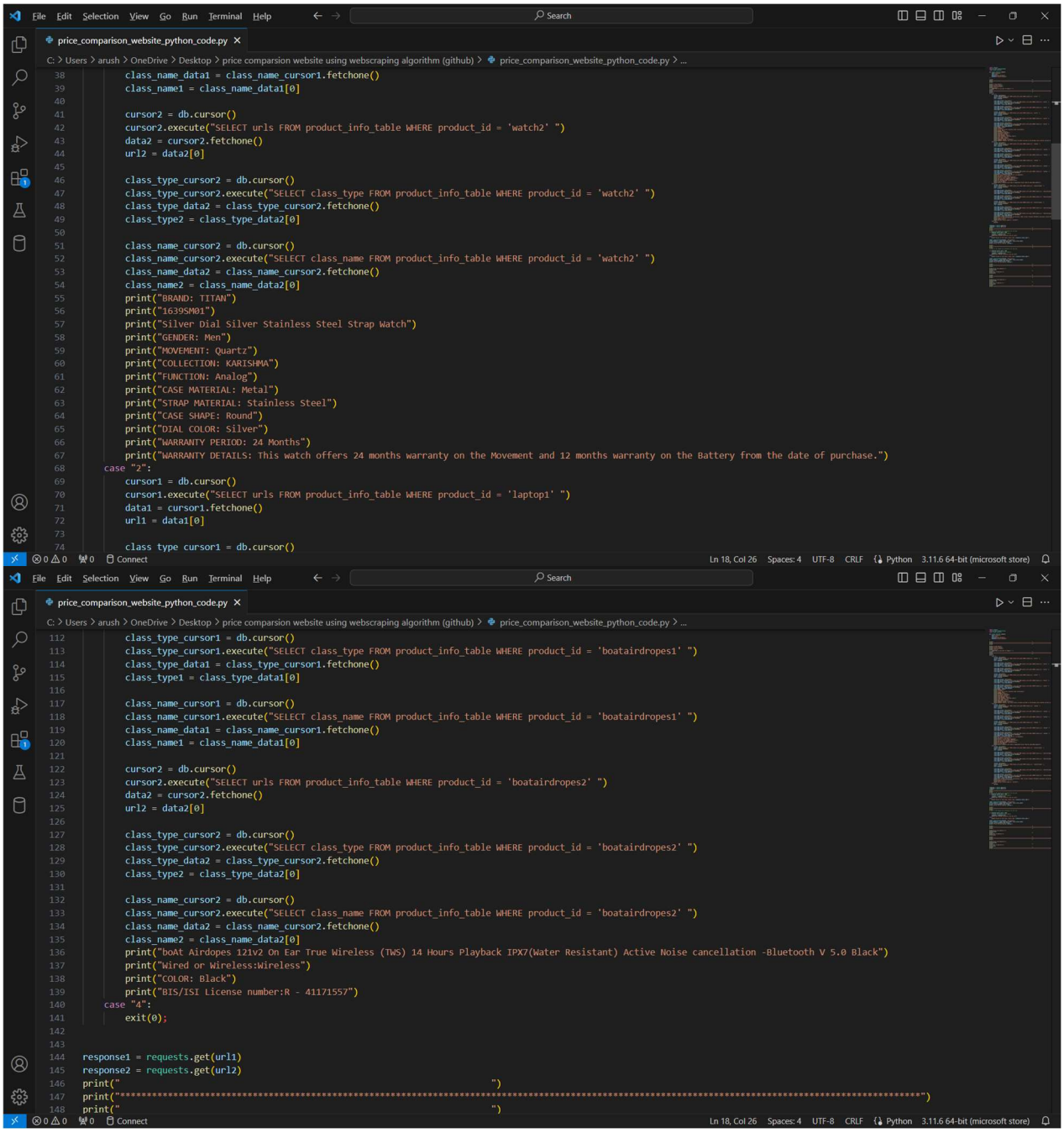
4. ARCHITECTURE AND DESIGN

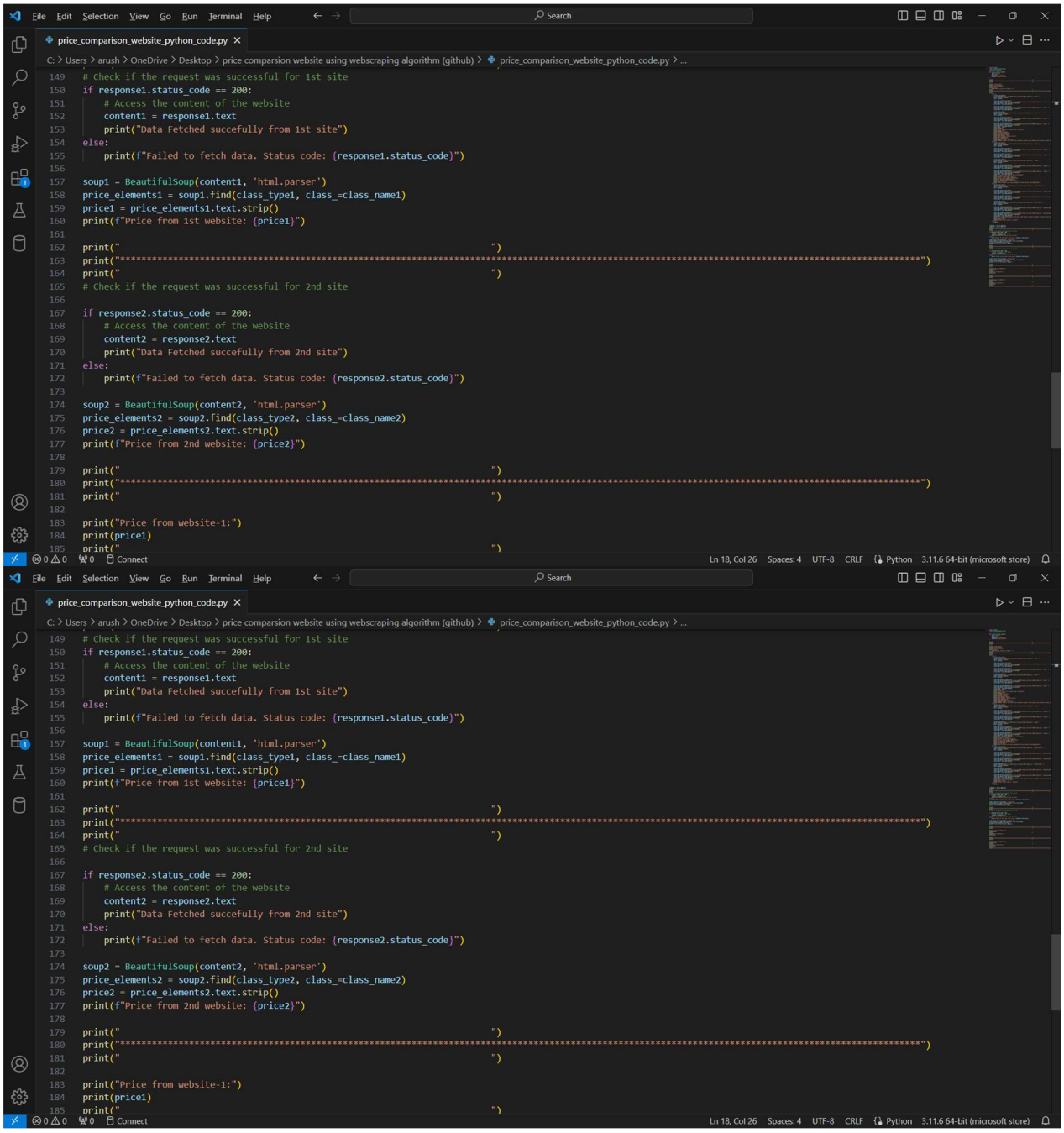


5. IMPLEMENTATION

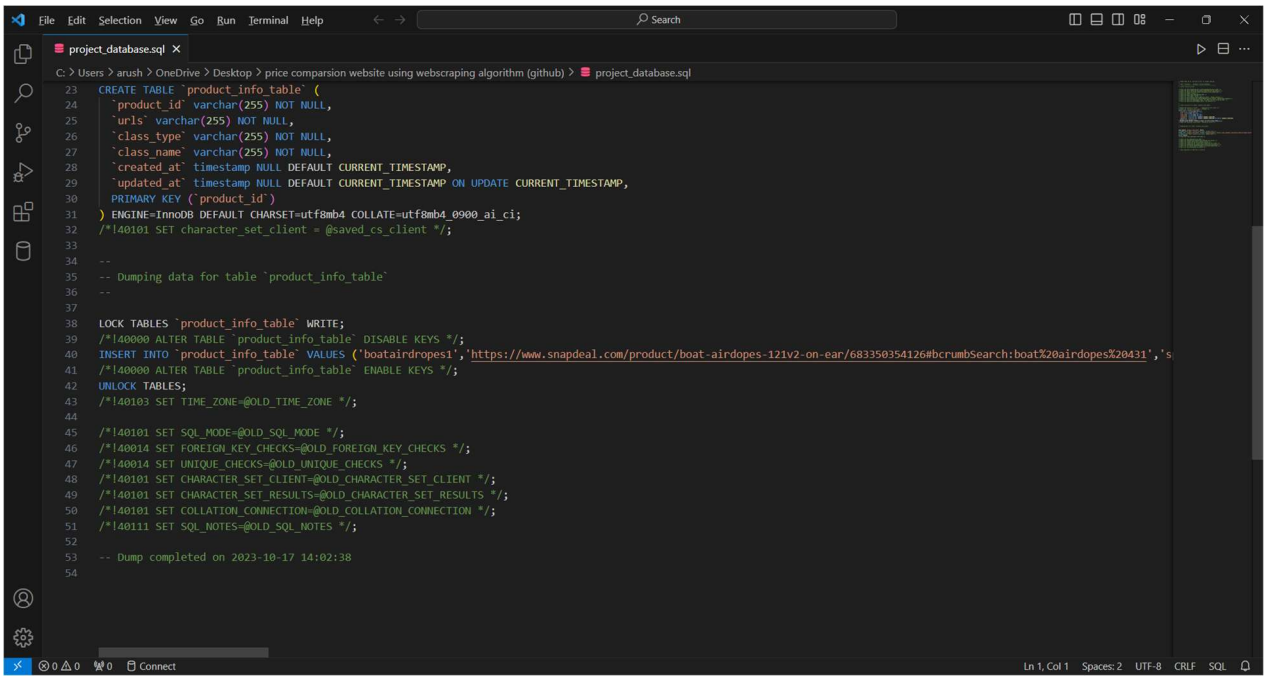
PYTHON CODE:

```python
import requests
from bs4 import BeautifulSoup
import mysql.connector

db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="your_password",
    database="project_database"
)

print("                                                                      ")
print("*******************************************************************************************************************************************************************")
print("                                                                      ")

print("1.Titan Watch")
print("2.Victus Laptop")
print("3.Boat Airdropes")
print("4.exit")
n = input("What do you want to compare ? ")
print("                                                                      ")
print("*******************************************************************************************************************************************************************")
print("                                                                      ")
match n:
    case "1":
        cursor1 = db.cursor()
        cursor1.execute("SELECT urls FROM product_info_table WHERE product_id = 'watch1' ")
        data1 = cursor1.fetchone()
        url1 = data1[0]

        class_type_cursor1 = db.cursor()
        class_type_cursor1.execute("SELECT class_type FROM product_info_table WHERE product_id = 'watch1' ")
        class_type_data1 = class_type_cursor1.fetchone()
        class_type1 = class_type_data1[0]

        class_name_cursor1 = db.cursor()
        class_name_cursor1.execute("SELECT class_name FROM product_info_table WHERE product_id = 'watch1' ")
        class_name_data1 = class_name_cursor1.fetchone()
        class_name1 = class_name_data1[0]

        cursor2 = db.cursor()
        cursor2.execute("SELECT urls FROM product_info_table WHERE product_id = 'watch2' ")
        data2 = cursor2.fetchone()
        url2 = data2[0]

        class_type_cursor2 = db.cursor()
        class_type_cursor2.execute("SELECT class_type FROM product_info_table WHERE product_id = 'watch2' ")
        class_type_data2 = class_type_cursor2.fetchone()
        class_type2 = class_type_data2[0]

        class_name_cursor2 = db.cursor()
        class_name_cursor2.execute("SELECT class_name FROM product_info_table WHERE product_id = 'watch2' ")
        class_name_data2 = class_name_cursor2.fetchone()
        class_name2 = class_name_data2[0]
        print("BRAND: TITAN")
        print("1639SM01")
        print("Silver Dial Silver Stainless Steel Strap Watch")
        print("GENDER: Men")
        print("MOVEMENT: Quartz")
        print("COLLECTION: KARISHMA")
        print("FUNCTION: Analog")
        print("CASE MATERIAL: Metal")
        print("STRAP MATERIAL: Stainless Steel")
        print("CASE SHAPE: Round")
        print("DIAL COLOR: Silver")
        print("WARRANTY PERIOD: 24 Months")
        print("WARRANTY DETAILS: This watch offers 24 months warranty on the Movement and 12 months warranty on the Battery from the date of purchase.")
    case "2":
        cursor1 = db.cursor()
        cursor1.execute("SELECT urls FROM product_info_table WHERE product_id = 'laptop1' ")
        data1 = cursor1.fetchone()
        url1 = data1[0]

        class_type_cursor1 = db.cursor()
```

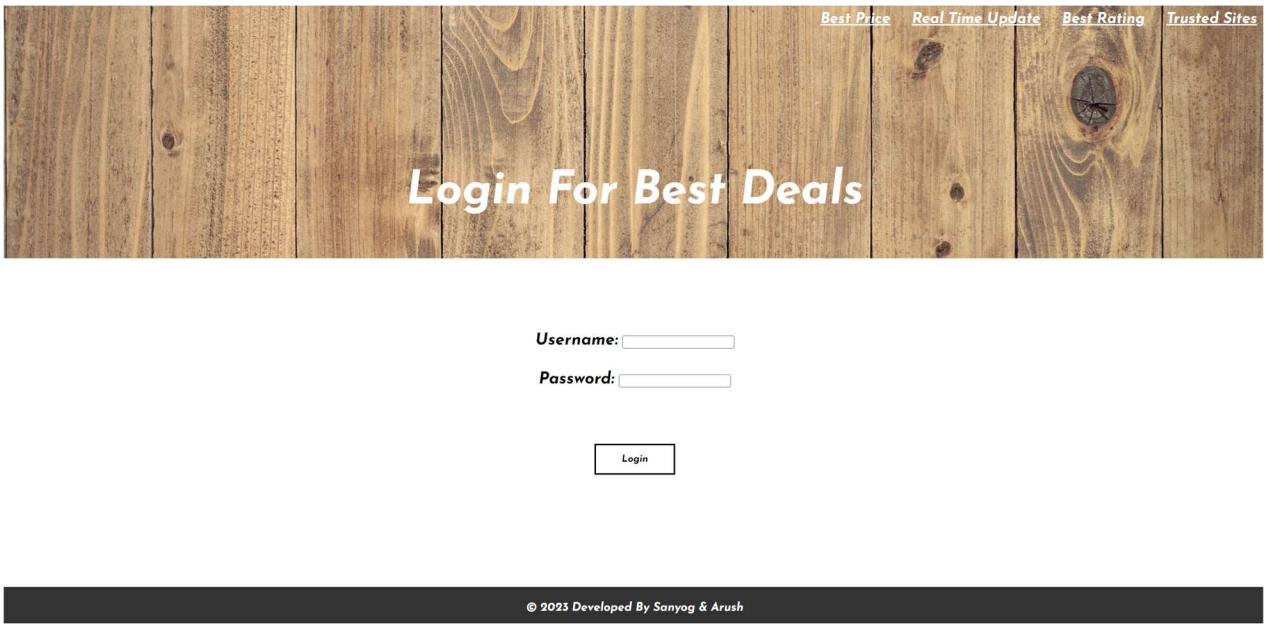

File Edit Selection View Go Run Terminal Help
Search
price_comparison_website_python_code.py
C: > Users > arush > OneDrive > Desktop > price comparsion website using webscraping algorithm (github) > price_comparison_website_python_code.py > ...
class_name_data1 = class_name_cursor1.fetchone()
class_name1 = class_name_data1[0]
cursor2 = db.cursor()
cursor2.execute("SELECT urls FROM product_info_table WHERE product_id = 'watch2' ")
data2 = cursor2.fetchone()
url2 = data2[0]
class_type_cursor2 = db.cursor()
class_type_cursor2.execute("SELECT class_type FROM product_info_table WHERE product_id = 'watch2' ")
class_type_data2 = class_type_cursor2.fetchone()
class_type2 = class_type_data2[0]
class_name_cursor2 = db.cursor()
class_name_cursor2.execute("SELECT class_name FROM product_info_table WHERE product_id = 'watch2' ")
class_name_data2 = class_name_cursor2.fetchone()
class_name2 = class_name_data2[0]
print("BRAND: TITAN")
print("1639SM01")
print("Silver Dial Silver Stainless Steel Strap Watch")
print("GENDER: Men")
print("MOVEMENT: Quartz")
print("COLLECTION: KARISHMA")
print("FUNCTION: Analog")
print("CASE MATERIAL: Metal")
print("STRAP MATERIAL: Stainless Steel")
print("CASE SHAPE: Round")
print("DIAL COLOR: Silver")
print("WARRANTY PERIOD: 24 Months")
print("WARRANTY DETAILS: This watch offers 24 months warranty on the Movement and 12 months warranty on the Battery from the date of purchase.")
case "2":
cursor1 = db.cursor()
cursor1.execute("SELECT urls FROM product_info_table WHERE product_id = 'laptop1' ")
data1 = cursor1.fetchone()
url1 = data1[0]
class type cursor1 = db.cursor()
Connect
Ln 18, Col 26 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)
File Edit Selection View Go Run Terminal Help
Search
price_comparison_website_python_code.py
C: > Users > arush > OneDrive > Desktop > price comparsion website using webscraping algorithm (github) > price_comparison_website_python_code.py > ...
class_type_cursor1 = db.cursor()
class_type_cursor1.execute("SELECT class_type FROM product_info_table WHERE product_id = 'boatairdropes1' ")
class_type_data1 = class_type_cursor1.fetchone()
class_type1 = class_type_data1[0]
class_name_cursor1 = db.cursor()
class_name_cursor1.execute("SELECT class_name FROM product_info_table WHERE product_id = 'boatairdropes1' ")
class_name_data1 = class_name_cursor1.fetchone()
class_name1 = class_name_data1[0]
cursor2 = db.cursor()
cursor2.execute("SELECT urls FROM product_info_table WHERE product_id = 'boatairdropes2' ")
data2 = cursor2.fetchone()
url2 = data2[0]
class_type_cursor2 = db.cursor()
class_type_cursor2.execute("SELECT class_type FROM product_info_table WHERE product_id = 'boatairdropes2' ")
class_type_data2 = class_type_cursor2.fetchone()
class_type2 = class_type_data2[0]
class_name_cursor2 = db.cursor()
class_name_cursor2.execute("SELECT class_name FROM product_info_table WHERE product_id = 'boatairdropes2' ")
class_name_data2 = class_name_cursor2.fetchone()
class_name2 = class_name_data2[0]
print("boAt Airdopes 121v2 On Ear True Wireless (TWS) 14 Hours Playback IPX7(Water Resistant) Active Noise cancellation -Bluetooth V 5.0 Black")
print("Wired or Wireless:Wireless")
print("COLOR: Black")
print("BIS/ISI License number:R - 41171557")
case "4":
exit(0);
response1 = requests.get(url1)
response2 = requests.get(url2)
print("
")
print("**********")
print("
")
Connect
Ln 18, Col 26 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)

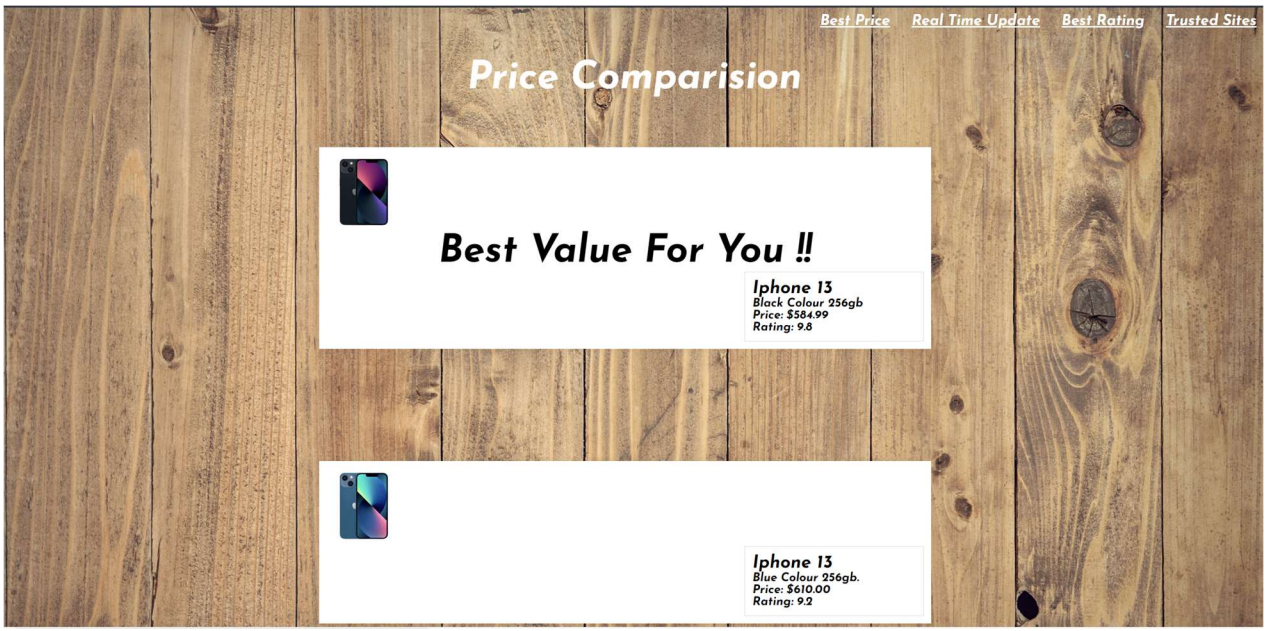

File Edit Selection View Go Run Terminal Help
Search
price_comparison_website_python_code.py
C: > Users > arush > OneDrive > Desktop > price comparsion website using webscraping algorithm (github) > price_comparison_website_python_code.py > ...
# Check if the request was successful for 1st site
if response1.status_code == 200:
# Access the content of the website
content1 = response1.text
print("Data Fetched succefully from 1st site")
else:
print(f"Failed to fetch data. Status code: {response1.status_code}")
soup1 = BeautifulSoup(content1, 'html.parser')
price_elements1 = soup1.find(class_type1, class_=class_name1)
price1 = price_elements1.text.strip()
print(f"Price from 1st website: {price1}")
print("
")
print("**********")
print("
")
# Check if the request was successful for 2nd site
if response2.status_code == 200:
# Access the content of the website
content2 = response2.text
print("Data Fetched succefully from 2nd site")
else:
print(f"Failed to fetch data. Status code: {response2.status_code}")
soup2 = BeautifulSoup(content2, 'html.parser')
price_elements2 = soup2.find(class_type2, class_=class_name2)
price2 = price_elements2.text.strip()
print(f"Price from 2nd website: {price2}")
print("
")
print("**********")
print("
")
print("Price from website-1:")
print(price1)
print("
")
Connect
Ln 18, Col 26 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)

MySQL DATABASE:

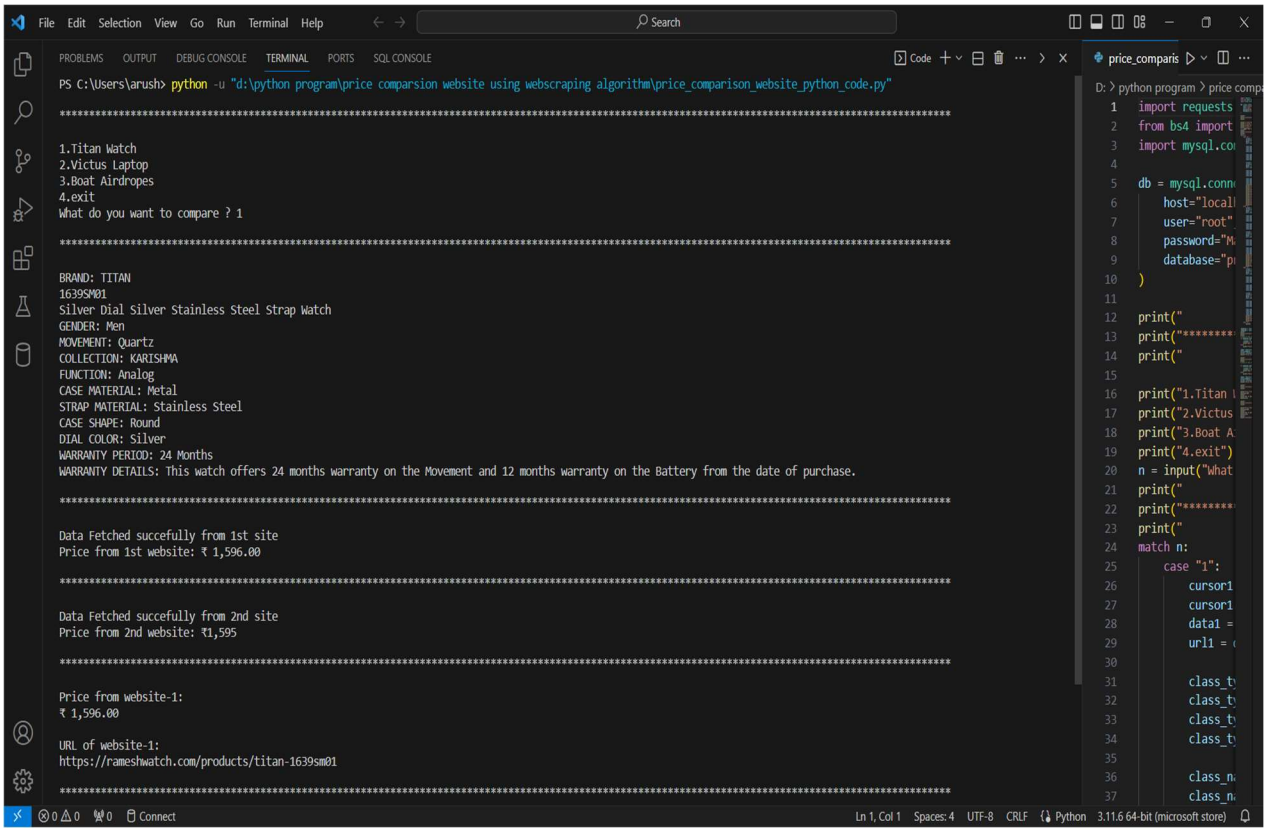


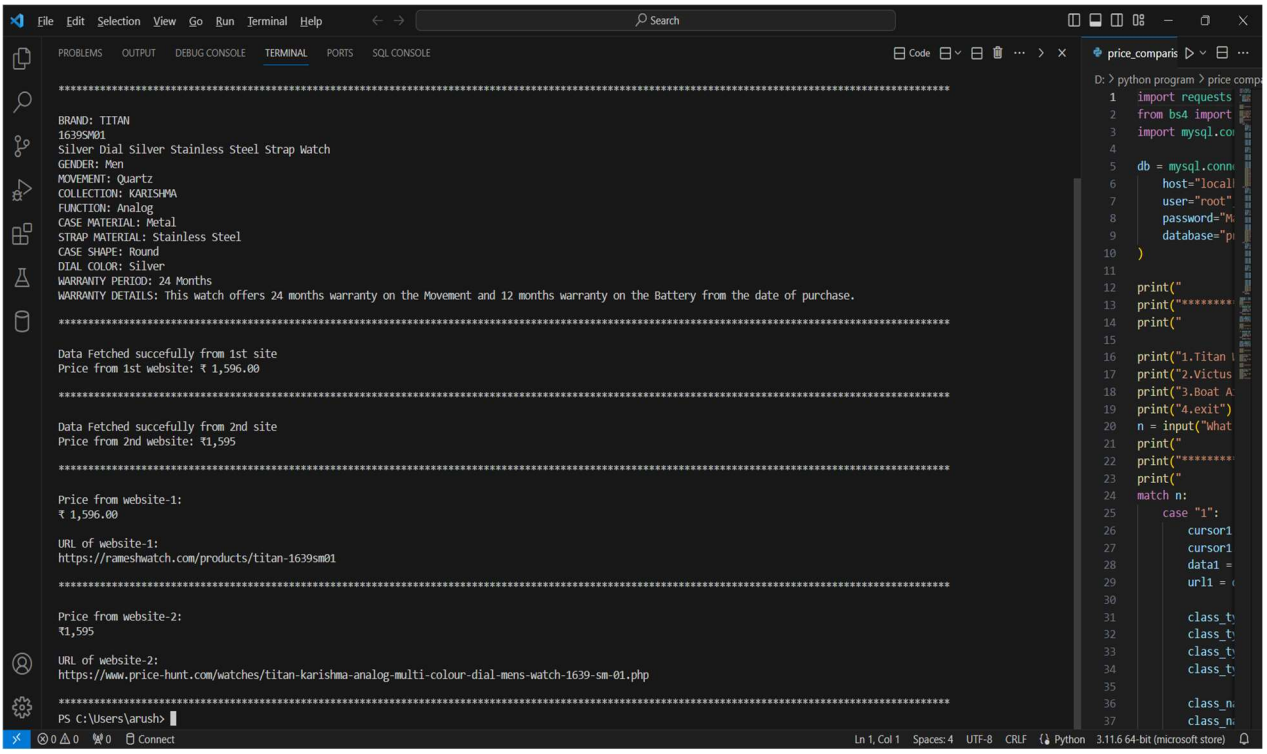
GUI:

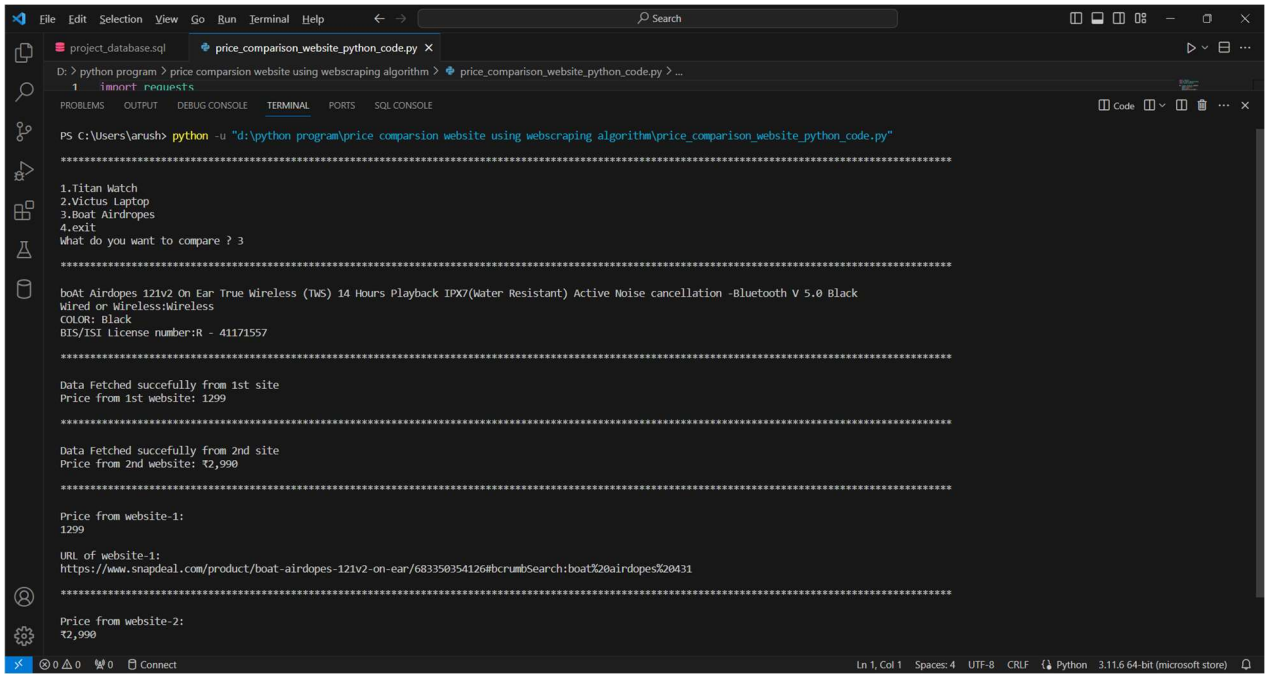

Best Price
Real Time Update
Best Rating
Trusted Sites
Login For Best Deals
Username:
Password:
Login
© 2023 Developed By Sanyog & Arush

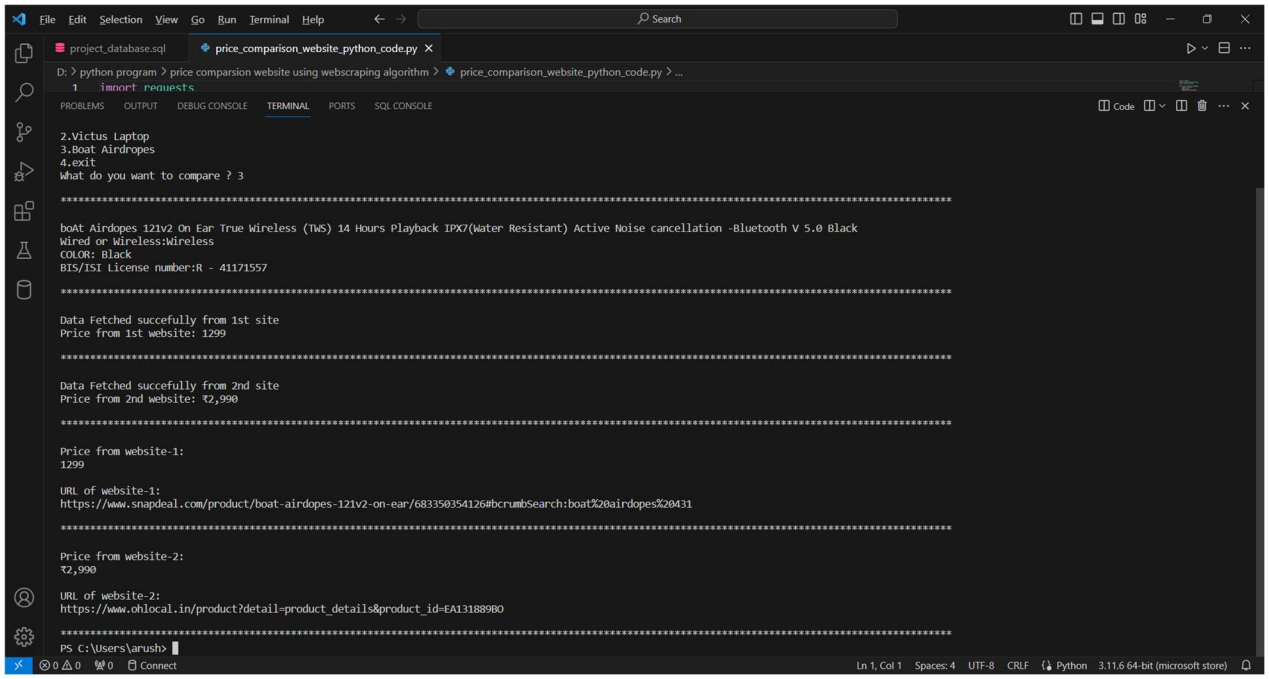

Best Price
Real Time Update
Best Rating
Trusted Sites
Price Comparision
Best Value For You !!
Iphone 13
Black Colour 256gb
Price: $584.99
Rating: 9.8
Iphone 13
Blue Colour 256gb.
Price: $610.00
Rating: 9.2

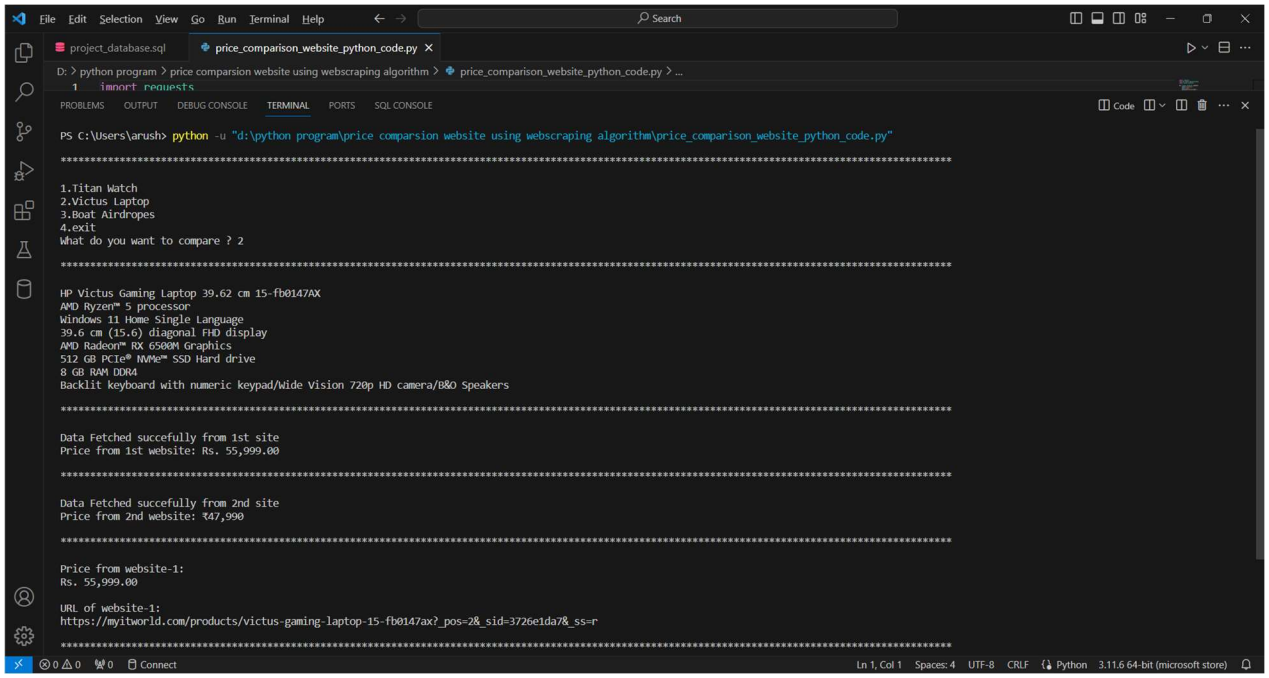
6. RESULTS AND DISCUSSION

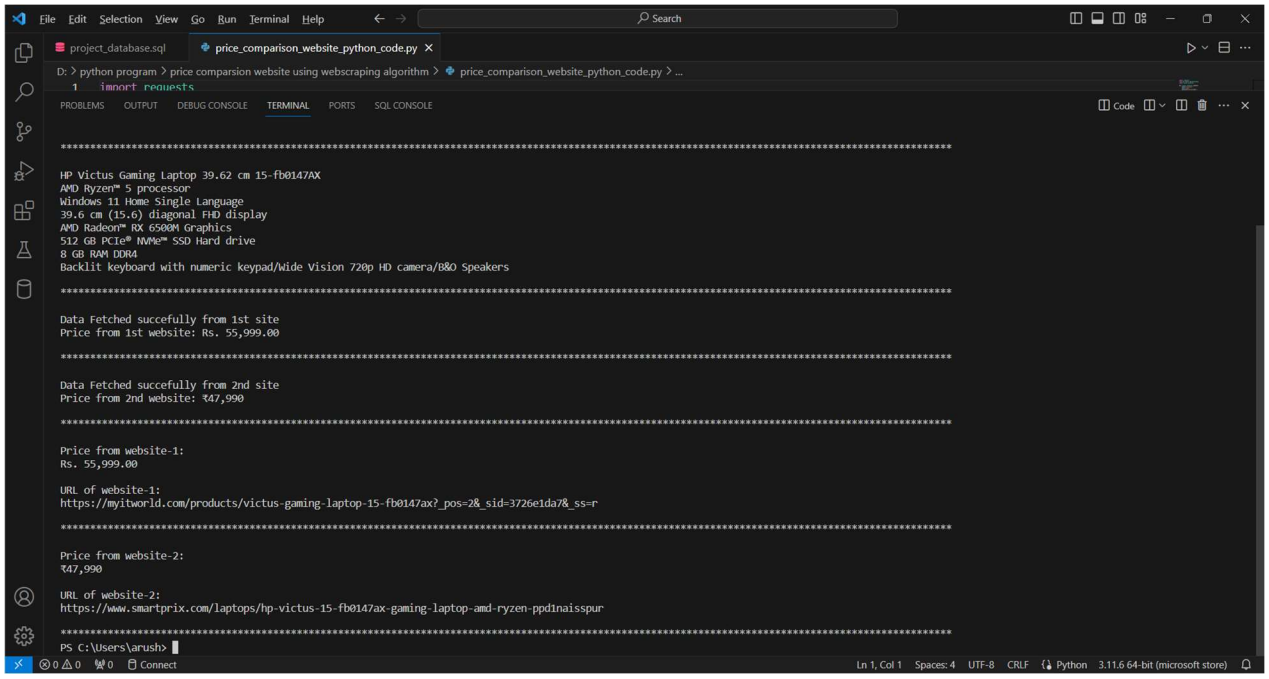





File Edit Selection View Go Run Terminal Help
Search
project_database.sql
price_comparison_website_python_code.py
D: > python program > price comparsion website using webscraping algorithm > price_comparison_website_python_code.py > ...
1 import requests
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
Code
PS C:\Users\arush> python -u "d:\python program\price comparsion website using webscraping algorithm\price_comparison_website_python_code.py"
****
1.Titan Watch
2.Victus Laptop
3.Boat Airdropes
4.exit
What do you want to compare ? 3
****
boAt Airdopes 121v2 On Ear True Wireless (TWS) 14 Hours Playback IPX7(Water Resistant) Active Noise cancellation -Bluetooth V 5.0 Black
Wired or Wireless:Wireless
COLOR: Black
BIS/ISI License number:R - 41171557
****
Data Fetched succefully from 1st site
Price from 1st website: 1299
****
Data Fetched succefully from 2nd site
Price from 2nd website: ₹2,990
****
Price from website-1:
1299
URL of website-1:
https://www.snapdeal.com/product/boat-airdopes-121v2-on-ear/683350354126#bcrumbSearch:boat%20airdopes%20431
****
Price from website-2:
₹2,990
Connect
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)


File Edit Selection View Go Run Terminal Help
Search
project_database.sql
price_comparison_website_python_code.py
D: > python program > price comparsion website using webscraping algorithm > price_comparison_website_python_code.py > ...
1 import requests
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
Code
2.Victus Laptop
3.Boat Airdropes
4.exit
What do you want to compare ? 3
****
boAt Airdopes 121v2 On Ear True Wireless (TWS) 14 Hours Playback IPX7(Water Resistant) Active Noise cancellation -Bluetooth V 5.0 Black
Wired or Wireless:Wireless
COLOR: Black
BIS/ISI License number:R - 41171557
****
Data Fetched succefully from 1st site
Price from 1st website: 1299
****
Data Fetched succefully from 2nd site
Price from 2nd website: ₹2,990
****
Price from website-1:
1299
URL of website-1:
https://www.snapdeal.com/product/boat-airdopes-121v2-on-ear/683350354126#bcrumbSearch:boat%20airdopes%20431
****
Price from website-2:
₹2,990
URL of website-2:
https://www.ohlocal.in/product?detail=product_details&product_id=EA131889BO
****
PS C:\Users\arush>
Connect
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)


File Edit Selection View Go Run Terminal Help
Search
project_database.sql
price_comparison_website_python_code.py
D: > python program > price comparsion website using webscraping algorithm > price_comparison_website_python_code.py > ...
1 import requests
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
Code
PS C:\Users\arush> python -u "d:\python program\price comparsion website using webscraping algorithm\price_comparison_website_python_code.py"
1.Titan Watch
2.Victus Laptop
3.Boat Airdropes
4.exit
What do you want to compare ? 2
HP Victus Gaming Laptop 39.62 cm 15-fb0147AX
AMD Ryzen™ 5 processor
Windows 11 Home Single Language
39.6 cm (15.6) diagonal FHD display
AMD Radeon™ RX 6500M Graphics
512 GB PCIe® NVMe™ SSD Hard drive
8 GB RAM DDR4
Backlit keyboard with numeric keypad/Wide Vision 720p HD camera/B&O Speakers
Data Fetched succefully from 1st site
Price from 1st website: Rs. 55,999.00
Data Fetched succefully from 2nd site
Price from 2nd website: ₹47,990
Price from website-1:
Rs. 55,999.00
URL of website-1:
https://myitworld.com/products/victus-gaming-laptop-15-fb0147ax?_pos=2&_sid=3726e1da7&_ss=r
Connect
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)


File Edit Selection View Go Run Terminal Help
Search
project_database.sql
price_comparison_website_python_code.py
D: > python program > price comparsion website using webscraping algorithm > price_comparison_website_python_code.py > ...
1 import requests
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
Code
HP Victus Gaming Laptop 39.62 cm 15-fb0147AX
AMD Ryzen™ 5 processor
Windows 11 Home Single Language
39.6 cm (15.6) diagonal FHD display
AMD Radeon™ RX 6500M Graphics
512 GB PCIe® NVMe™ SSD Hard drive
8 GB RAM DDR4
Backlit keyboard with numeric keypad/Wide Vision 720p HD camera/B&O Speakers
Data Fetched succefully from 1st site
Price from 1st website: Rs. 55,999.00
Data Fetched succefully from 2nd site
Price from 2nd website: ₹47,990
Price from website-1:
Rs. 55,999.00
URL of website-1:
https://myitworld.com/products/victus-gaming-laptop-15-fb0147ax?_pos=2&_sid=3726e1da7&_ss=r
Price from website-2:
₹47,990
URL of website-2:
https://www.smartprix.com/laptops/hp-victus-15-fb0147ax-gaming-laptop-amd-ryzen-ppd1naisspur
PS C:\Users\arush>
Connect
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.6 64-bit (microsoft store)

7. CONCLUSION

our price comparison website project is a significant achievement, demonstrating our skills in web scraping, database management, and web development. It provides a valuable service to users by simplifying their product research and saving them time and money. As we continue to work on this project and make improvements, it has the potential to become a popular destination for online shoppers seeking the best deals. It showcases our ability to create practical, data-driven solutions and opens the door to various opportunities for expansion and monetization. Keep refining and enhancing our project to stay competitive in the dynamic e-commerce landscape.

8. REFERENCES

- Comparative Study Of Various Scraping Tools: Pros And Cons

  Comparative Study Of Various Scraping Tools: Pros And Cons | IEEE Conference Publication | IEEE Xplore

- Intelligent Online Shopping using ML-based Product Comparison Engine

  Intelligent Online Shopping using ML-based Product Comparison Engine | IEEE Conference Publication | IEEE Xplore

- Co-Mart - A Daily Necessity Price Comparison Application

  Co-Mart - A Daily Necessity Price Comparison Application | IEEE Conference Publication | IEEE Xplore

- Web Scraping based Product Comparison Model for E-Commerce Websites

  Web Scraping based Product Comparison Model for E-Commerce Websites | IEEE Conference Publication | IEEE Xplore