BAX-422: Data Design and Representation

Final Project: Amazon's Highly Rated Laptops & their Reviews

Table of Contents

Executive Summary

The objective of this project was to identify highly rated laptops on Amazon and understand why these laptops were rated highly. To accomplish this, web scraping techniques were used to collect data on product information, ratings, and reviews for five laptops in the highly rated section of Amazon. A Firefox headless browser was used to search for laptops, and the main page URLs of the top five laptops were identified and downloaded locally. The product name, price, overall rating, and product information were then scraped from these pages.

Next, the URLs for the reviews of each laptop were identified, and the review pages were dynamically loaded and downloaded locally. The ratings and reviews were scraped from these pages and saved as review_1, review_2, etc. for each laptop, without capturing any personally identifiable information (PII) such as the actual user name. The primary keys such as laptop_1, laptop_2, etc. were created to join the product information and review information if required. Finally, the product and review information were saved in MongoDB.

MongoDB was chosen for its flexibility in storing varying data structures, scalability in handling large amounts of unstructured data like user reviews, and powerful querying capabilities for efficient data retrieval and analysis, enabling efficient storage and retrieval of data for business decision-making purposes.

This project aimed to support a company's business decisions by providing data on highly rated laptops and understanding why these laptops are rated highly. The web scraping techniques used in this project enabled the collection and preparation of data for use in business decision making.

Background, Context and Domain Knowledge

The aim of this project was to identify Amazon's highly rated laptops and analyze what factors contributed to their high ratings. To achieve this, web scraping techniques were used to collect product information and user reviews from Amazon's website. The context for this project is the computer hardware industry, which is highly competitive and constantly evolving, with new products and technologies being released regularly.

To scrape the required data, Amazon's website was chosen as it is one of the most popular and widely-used e-commerce websites in the world, with a vast selection of laptops available for purchase. Relevant domain knowledge required for this project included familiarity with web scraping techniques, data cleaning and manipulation including Regular Expressions, and basic knowledge of computer hardware specifications and features.

By analyzing the scraped data, we aimed to gain insights into the key features and factors that contribute to highly-rated laptops, which could be useful for both consumers looking to purchase a laptop and businesses operating in the computer hardware industry.

Introduction to Data Sources

The data source for this project is Amazon's website. Specifically, we used the website's search feature to find products that were rated highly by users. We then extracted information from the product pages of these highly rated products. This information included the product name, rating, price, brand, and the number of customer reviews.

Refer Fig 1 in Appendix for Laptop Information data structure.

The columns here are dynamically captured, in other words few laptops may not have information on Graphics_Card_Description etc. You can refer to Fig 2 in Apendix for the code block of this. This is one of the reason we would be inclined to pick MongoDB as our database.

Refer Fig 3 in Appendix for Review Information data structure.

Description of the Web-Scraping Routine

Web scraping is the process of extracting data from websites automatically. This can be useful for a variety of purposes, such as market research or data analysis. In this case, we will be scraping Highly rated laptops section from Amazon.com and parse laptop information, laptop reviews(all reviews) into our **MongoDB** for further analysis. We use the **headless mode** of the Mozilla Firefox browser in web scraping because it allows us to run the browser in the background without any visible UI, which means that the browser window is not opened during the execution of the script. This can help to speed up the scraping process and reduce the memory usage of the script, as there is no need to render and display the web page on the screen. Additionally, headless mode provides better performance, improved stability, and higher security, as it does not require a GUI to run, making it less susceptible to hacking attempts. We used **regular expressions**, to extract the actual text and remove unwanted characters like emoticons and smiley characters which were observed in the reviews.

Below is the methodology and steps followed in our analysis.

1. Import the necessary libraries such as Selenium, requests, BeautifulSoup, Pandas, glob, and json.
2. Define the URL to scrape and the path of the GeckoDriver executable file.
3. Create a Firefox driver instance with headless mode enabled.
4. Load the URL using the driver instance and wait for the page to load.

5. Extract the HTML content of the page using BeautifulSoup.
6. Find the "Highly rated" section of the page and extract the URLs of the laptops.
7. Save the HTML content of each laptop page to a local file with the product number as the filename.
8. Extract the URLs of the reviews for each laptop and store them in a list.
9. Create a folder to store the HTML files of the reviews.
10. Scrape all review pages for each main review URL and save them as HTML files in the folder created in step 9.
11. Process each HTML file of the laptops and extract their information such as product title, brand, processor, storage, and display size etc. Store the extracted information in a dictionary and append it to the laptops_info list.
12. The laptops_info list is then processed to extract specific information and create a new list of dictionaries called Laptop_Information. This is done to format the information in a way that can be easily pushed into a MongoDB database later.
13. Process each HTML file of the review from step 10 and extract their information such as rating, text review, add product column to join with our laptop information. create review_number dynamically for each product. Store the extracted information in a dictionary and append it to the reviews list and then store it in a data frame. We also did regex transformations on the text review to remove all smiley and emoticon characters which are not useful. This is saved to a dataframe specifically to save it in a *csv file* for viewing purposes.
14. The reviews list is then processed to extract information and create a new list of dictionaries called Laptop_Reviews. This is done to format the information in a way that can be easily pushed into a MongoDB database later.

15. Next we connect to the MongoDB server using the MongoClient method and create a database called *amazon_highly_rated*.
16. Then we create two collections in the database called *Laptop_Information* and *Laptop_Reviews*.
17. We then Iterate over the laptop information and insert it into the laptop information collection in MongoDB
18. In the end we iterate over the laptop reviews and insert it into the laptop reviews collection in MongoDB
19. Finally, the laptop details and reviews are stored in the MongoDB server, which can be accessed and queried as per requirement.

Refer to Exhibit 1 for Product information data, Exhibit 2 for Reviews Information data and Exhibit 3 for Total reviews count in Appendix which contains Studio3T screenshots.

Explanation of the Database Design Choices

MongoDB was chosen as the database for this project due to its suitability for storing unstructured or semi-structured data. Since the laptops being scrapped from Amazon have varying information columns, a flexible document-based data model like MongoDB makes it easy to store and organize the data.

In addition, MongoDB's scalability and performance make it an ideal choice for handling large amounts of data, such as user reviews. The dynamic schema and indexing features of MongoDB also provide powerful querying capabilities, allowing for efficient data retrieval and analysis.

Compared to a traditional SQL database, MongoDB's flexible data model allows for a more intuitive and streamlined approach to data storage and retrieval. SQL databases typically require a pre-defined schema with fixed columns and data types, which can make it difficult to handle unstructured data like user reviews.

Furthermore, SQL databases may struggle with the performance and scalability required for handling large amounts of unstructured data. In contrast, MongoDB's distributed architecture and automatic sharding make it easy to scale horizontally to handle large amounts of data.

In summary, the decision to use MongoDB for this project was based on its ability to handle unstructured data, scalability, performance, and powerful querying capabilities. SQL databases may not have been a good fit for this project due to their rigid schema design and potential performance limitations.

Utilizing Laptop Data for Informed Business Decisions

The dataset generated from scraping the Amazon website for laptop information can provide valuable insights for businesses in the tech industry. By analyzing the dataset, businesses can gain a better understanding of the market trends and consumer preferences related to laptops.

Some of the questions that can be answered with the dataset include:

- What are the most popular brands of laptops in the market?
- What are the most common processor types used in laptops?
- What are the average prices for laptops with specific features such as a certain display size or storage capacity?
- What are the customer reviews for various laptop brands and models?

By answering these questions, businesses can gain a competitive advantage by better understanding the market and consumer preferences. For example, they can use the information to develop new products or improve existing ones, optimize pricing strategies, and improve customer satisfaction.

Additionally, the dataset can be used for efficient data analysis and retrieval. MongoDB's flexible data model and indexing features allow for efficient querying of the dataset, making it easier to extract specific information and insights. Furthermore, the dataset can be easily updated and expanded with new data as it becomes available, allowing for continuous analysis and improvement.

In summary, the dataset generated from scraping Amazon for laptop information provides valuable insights for businesses in the tech industry. By analyzing the data, businesses can gain a better understanding of the market and consumer preferences, allowing for improved decision-making and better positioning in the market.

Summary

This project involved scraping Amazon's website for information on highly-rated laptops and storing the data in a MongoDB database. The resulting dataset contains information on the laptops' brands, prices, specifications, and user reviews. The dataset can be used to answer business-relevant questions related to the laptop market, such as identifying popular brands, analyzing price trends, and assessing the strengths and weaknesses of different laptops based on user feedback. The use of MongoDB allows for efficient storage and retrieval of the data, making it easier to perform data analysis and generate insights for business decision-making.

Conclusion

In conclusion, this project has demonstrated the capability of web scraping as a valuable tool for collecting and organizing large amounts of data. The dataset collected and organized from Amazon's laptop section can be used for various business purposes, including analyzing product trends, conducting competitor analysis, and identifying consumer behavior patterns.

The use of MongoDB as the database to store the collected data has proven to be an effective solution, providing flexibility, scalability, and powerful querying capabilities. Additionally, the development of a web scraping routine using Selenium and Beautiful Soup has provided an efficient and reliable method for extracting the desired data.

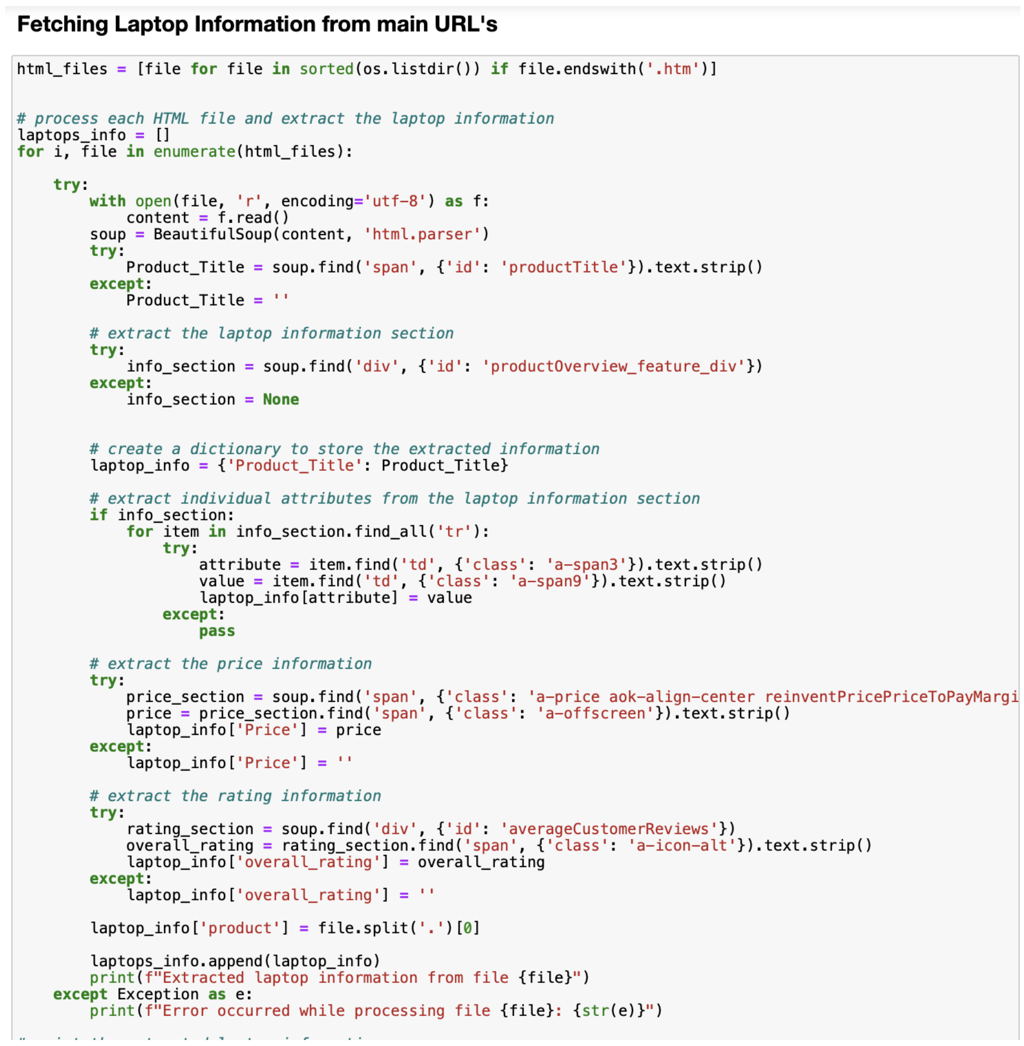
Overall, this project highlights the importance of data collection and analysis in business decision-making processes. The dataset and web scraping routine developed in this project can be further optimized and scaled to collect data from other e-commerce platforms, opening up new opportunities for data-driven insights and decision-making.

Appendix

Fig 1 - Laptop Information

S no	Column Name	Data Type	Sample Value
1	Product	Character	product_1
2	Product_Title	Character	HP 2022 Newest Pavilion Laptop..
3	Price	Character	$469.00
4	Brand	Character	HP
5	Screen_Size	Character	15.6 Inches
6	Hard_Disk_Size	Character	1 TB
7	CPU_Model	Character	Pentium N5000
8	RAM_memory_installed_size	Character	16 GB
9	Operating_System	Character	Windows 11
10	Graphics_Card_Description	Character	Integrated
11	Graphics_Coprocessor	Character	Intel UHD Graphics
12	CPU_speed	Character	2.7 GHz
13	Overall_Rating	Character	4.3 out of 5 stars

Fig 2 - Code for laptops

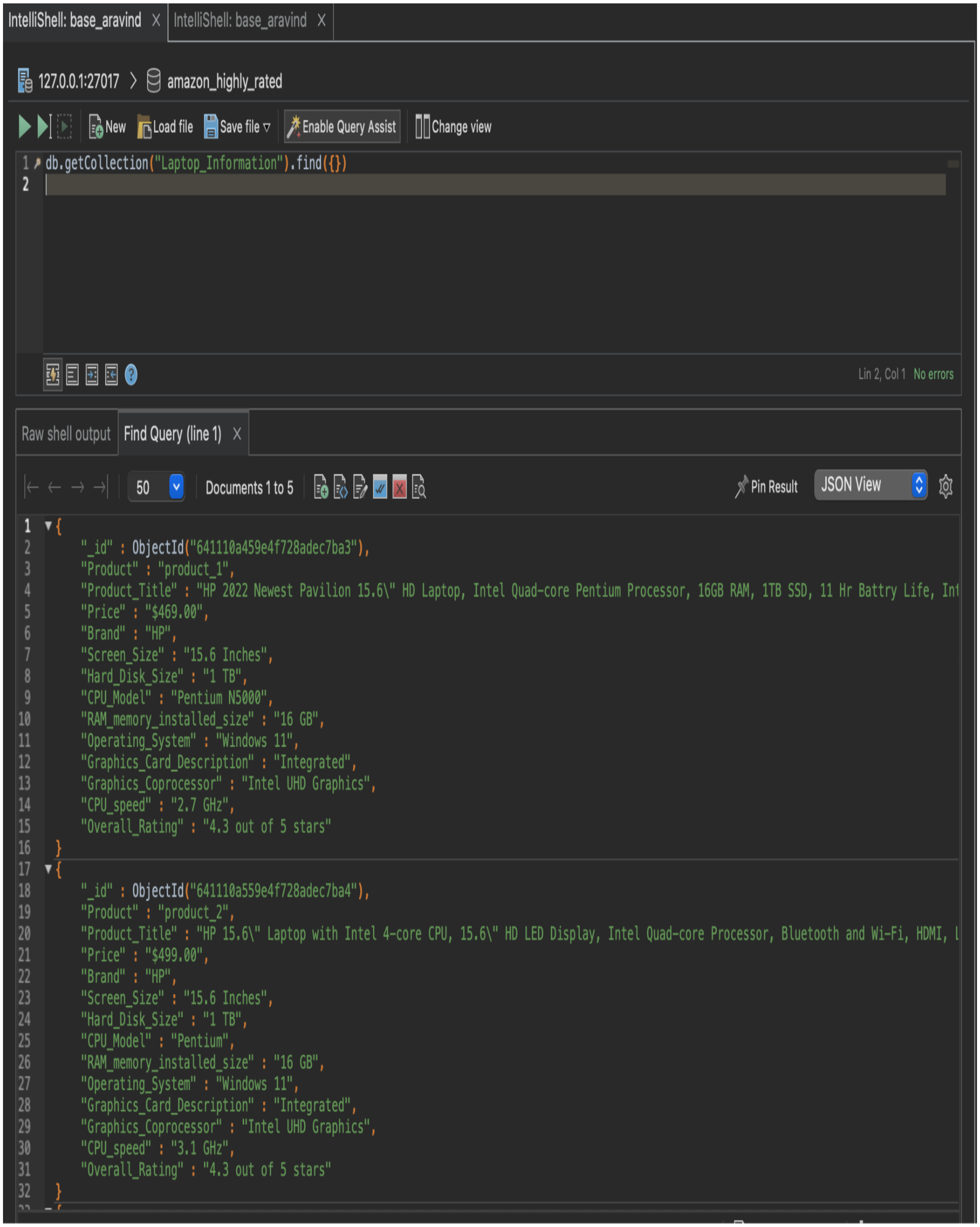


Fetching Laptop Information from main URL's

```
html_files = [file for file in sorted(os.listdir()) if file.endswith('.htm')]


# process each HTML file and extract the laptop information
laptops_info = []
for i, file in enumerate(html_files):

    try:
        with open(file, 'r', encoding='utf-8') as f:
            content = f.read()
        soup = BeautifulSoup(content, 'html.parser')
        try:
            Product_Title = soup.find('span', {'id': 'productTitle'}).text.strip()
        except:
            Product_Title = ''

        # extract the laptop information section
        try:
            info_section = soup.find('div', {'id': 'productOverview_feature_div'})
        except:
            info_section = None


        # create a dictionary to store the extracted information
        laptop_info = {'Product_Title': Product_Title}

        # extract individual attributes from the laptop information section
        if info_section:
            for item in info_section.find_all('tr'):
                try:
                    attribute = item.find('td', {'class': 'a-span3'}).text.strip()
                    value = item.find('td', {'class': 'a-span9'}).text.strip()
                    laptop_info[attribute] = value
                except:
                    pass

        # extract the price information
        try:
            price_section = soup.find('span', {'class': 'a-price aok-align-center reinventPricePriceToPayMargi
            price = price_section.find('span', {'class': 'a-offscreen'}).text.strip()
            laptop_info['Price'] = price
        except:
            laptop_info['Price'] = ''

        # extract the rating information
        try:
            rating_section = soup.find('div', {'id': 'averageCustomerReviews'})
            overall_rating = rating_section.find('span', {'class': 'a-icon-alt'}).text.strip()
            laptop_info['overall_rating'] = overall_rating
        except:
            laptop_info['overall_rating'] = ''

        laptop_info['product'] = file.split('.')[0]

        laptops_info.append(laptop_info)
        print(f"Extracted laptop information from file {file}")
    except Exception as e:
        print(f"Error occurred while processing file {file}: {str(e)}")
```

Fig 3- Review Information

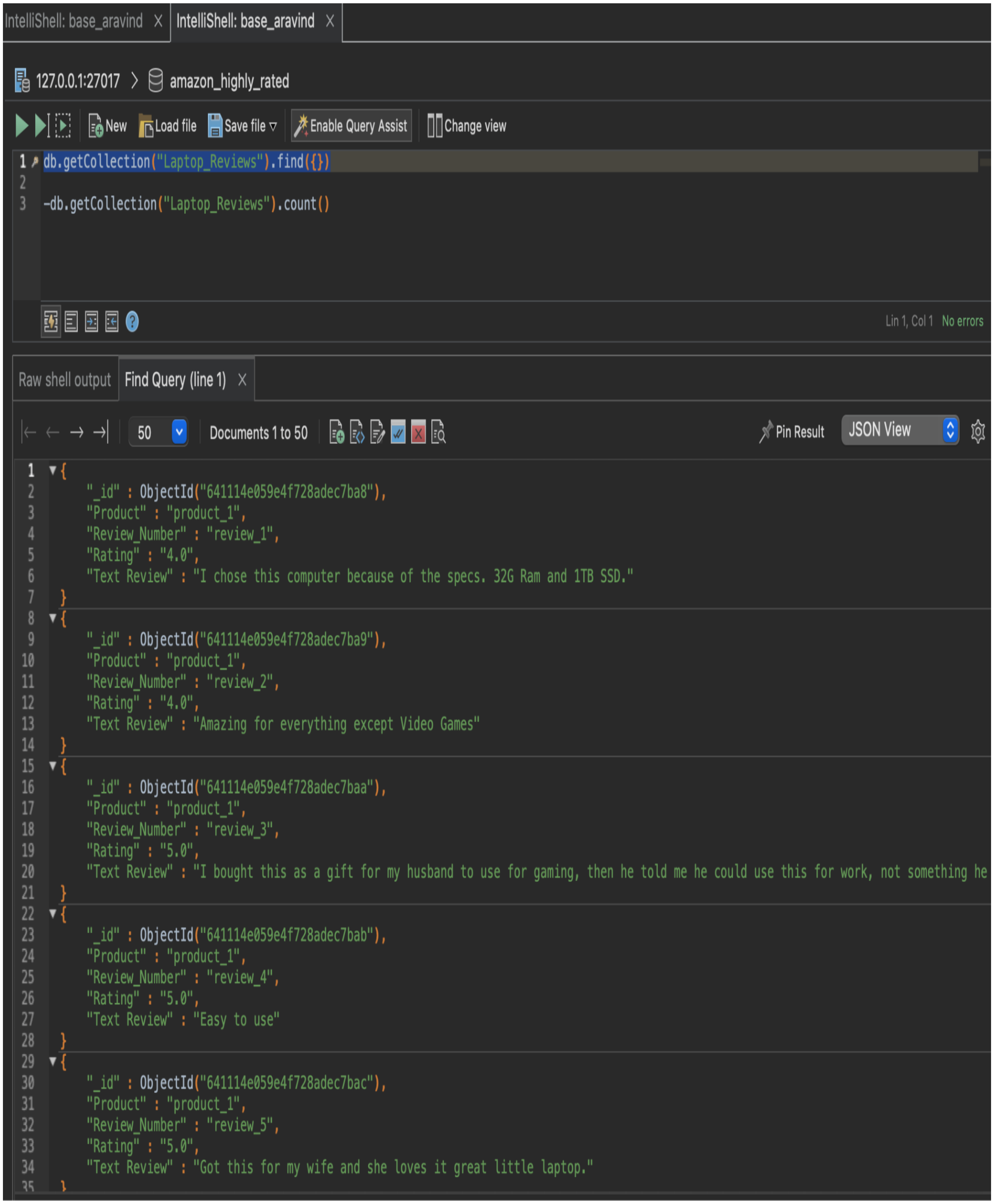
S no	Column Name	Data Type	Sample Value
1	Product	Character	product_1
2	Review_Number	Character	review_1
3	Rating	Character	4.0
4	Text Review	Character	Amazing for everything except Video Games



```
db.getCollection("Laptop_Information").find({})
```

```
{
    "_id" : ObjectId("641110a459e4f728adec7ba3"),
    "Product" : "product_1",
    "Product_Title" : "HP 2022 Newest Pavilion 15.6\" HD Laptop, Intel Quad-core Pentium Processor, 16GB RAM, 1TB SSD, 11 Hr Battry Life, Int
    "Price" : "$469.00",
    "Brand" : "HP",
    "Screen_Size" : "15.6 Inches",
    "Hard_Disk_Size" : "1 TB",
    "CPU_Model" : "Pentium N5000",
    "RAM_memory_installed_size" : "16 GB",
    "Operating_System" : "Windows 11",
    "Graphics_Card_Description" : "Integrated",
    "Graphics_Coprocessor" : "Intel UHD Graphics",
    "CPU_speed" : "2.7 GHz",
    "Overall_Rating" : "4.3 out of 5 stars"
}
{
    "_id" : ObjectId("641110a559e4f728adec7ba4"),
    "Product" : "product_2",
    "Product_Title" : "HP 15.6\" Laptop with Intel 4-core CPU, 15.6\" HD LED Display, Intel Quad-core Processor, Bluetooth and Wi-Fi, HDMI, L
    "Price" : "$499.00",
    "Brand" : "HP",
    "Screen_Size" : "15.6 Inches",
    "Hard_Disk_Size" : "1 TB",
    "CPU_Model" : "Pentium",
    "RAM_memory_installed_size" : "16 GB",
    "Operating_System" : "Windows 11",
    "Graphics_Card_Description" : "Integrated",
    "Graphics_Coprocessor" : "Intel UHD Graphics",
    "CPU_speed" : "3.1 GHz",
    "Overall_Rating" : "4.3 out of 5 stars"
}
```

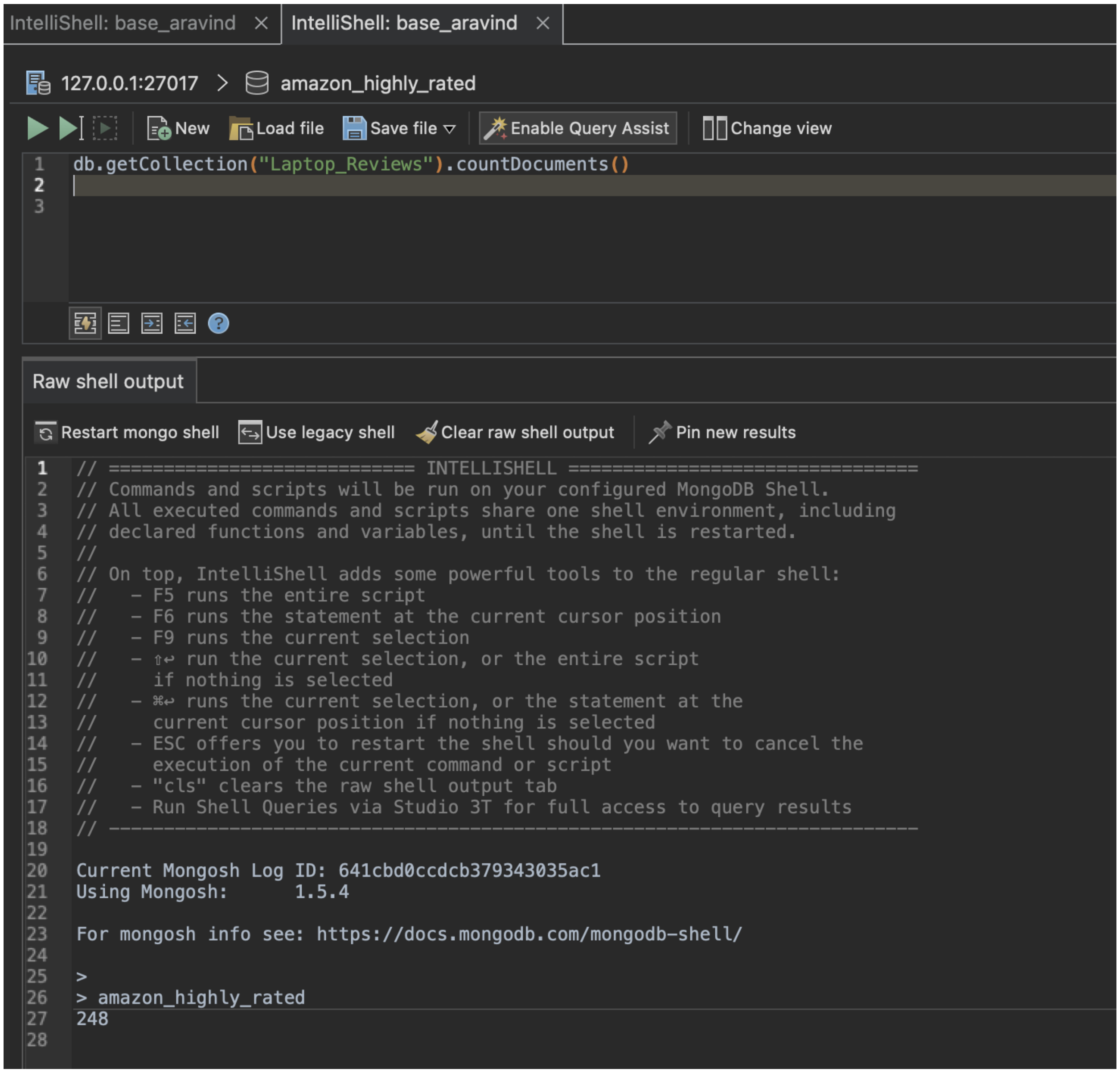
Exhibit 2-Reviews data(JSON view)



```
db.getCollection("Laptop_Reviews").find({})

-db.getCollection("Laptop_Reviews").count()
```

```
{
    "_id" : ObjectId("641114e059e4f728adec7ba8"),
    "Product" : "product_1",
    "Review_Number" : "review_1",
    "Rating" : "4.0",
    "Text Review" : "I chose this computer because of the specs. 32G Ram and 1TB SSD."
}
{
    "_id" : ObjectId("641114e059e4f728adec7ba9"),
    "Product" : "product_1",
    "Review_Number" : "review_2",
    "Rating" : "4.0",
    "Text Review" : "Amazing for everything except Video Games"
}
{
    "_id" : ObjectId("641114e059e4f728adec7baa"),
    "Product" : "product_1",
    "Review_Number" : "review_3",
    "Rating" : "5.0",
    "Text Review" : "I bought this as a gift for my husband to use for gaming, then he told me he could use this for work, not something he
}
{
    "_id" : ObjectId("641114e059e4f728adec7bab"),
    "Product" : "product_1",
    "Review_Number" : "review_4",
    "Rating" : "5.0",
    "Text Review" : "Easy to use"
}
{
    "_id" : ObjectId("641114e059e4f728adec7bac"),
    "Product" : "product_1",
    "Review_Number" : "review_5",
    "Rating" : "5.0",
    "Text Review" : "Got this for my wife and she loves it great little laptop."
}
```

Exhibit 3-Total Reviews Count



```
db.getCollection("Laptop_Reviews").countDocuments()
```

```
// ============================ INTELLISHELL ================================
// Commands and scripts will be run on your configured MongoDB Shell.
// All executed commands and scripts share one shell environment, including
// declared functions and variables, until the shell is restarted.
//
// On top, IntelliShell adds some powerful tools to the regular shell:
//   - F5 runs the entire script
//   - F6 runs the statement at the current cursor position
//   - F9 runs the current selection
//   - ⇧↩ run the current selection, or the entire script
//     if nothing is selected
//   - ⌘↩ runs the current selection, or the statement at the
//     current cursor position if nothing is selected
//   - ESC offers you to restart the shell should you want to cancel the
//     execution of the current command or script
//   - "cls" clears the raw shell output tab
//   - Run Shell Queries via Studio 3T for full access to query results
// --------------------------------------------------------------------------

Current Mongosh Log ID: 641cbd0ccdcb379343035ac1
Using Mongosh:      1.5.4

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

>
> amazon_highly_rated
248
```