**Project Report**

**Amazon Product Data Extraction**

**Name : Nivitha P**

**Team : 2**

**1. Introduction**

The purpose of this project is to extract product details such as **Product Name**, **Price**, and **Reviews** from an Amazon webpage.  
The project uses **BeautifulSoup**, a powerful Python library for web scraping, to parse the HTML content and **Pandas** to organize and export the data into a clean Excel report.

This approach can help in:

* **Price analysis**
* **Tracking reviews and ratings**
* **Market research**
* **Competitor product comparison**

**2. Objectives**

The main objectives of this project are:

1. To read and parse Amazon product data from an HTML file.
2. To extract relevant product details such as:
   * Product Name
   * Product Price
   * Product Reviews
3. To organize the extracted data into a structured table using Pandas.
4. To save the organized data into an Excel file for reporting and analysis.
5. To create an efficient and reusable Python script for future scraping tasks.

**3. Tools and Technologies Used**

| **Tool/Library** | **Purpose** |
| --- | --- |
| **Python 3.x** | Programming language used for development |
| **BeautifulSoup** | Parsing and extracting data from HTML |
| **Pandas** | Structuring data into tables and exporting to Excel |
| **OpenPyXL** | Excel file handling and data export |
| **Amazon.html** | Input file containing saved Amazon webpage |

**4. Methodology**

The project follows these steps for extracting and storing data:

**Step 1: Reading HTML File**

The Amazon product webpage is saved as Amazon.html.  
The script opens and reads this file using Python's built-in file handling methods.

**Step 2: Parsing HTML**

BeautifulSoup is used to parse the HTML content into a tree-like structure for easy navigation and data extraction.

**Step 3: Finding Product Containers**

Amazon product details are grouped inside <div> elements with a unique class name.  
The script locates these containers using the find\_all() function.

**Step 4: Extracting Details**

For each product:

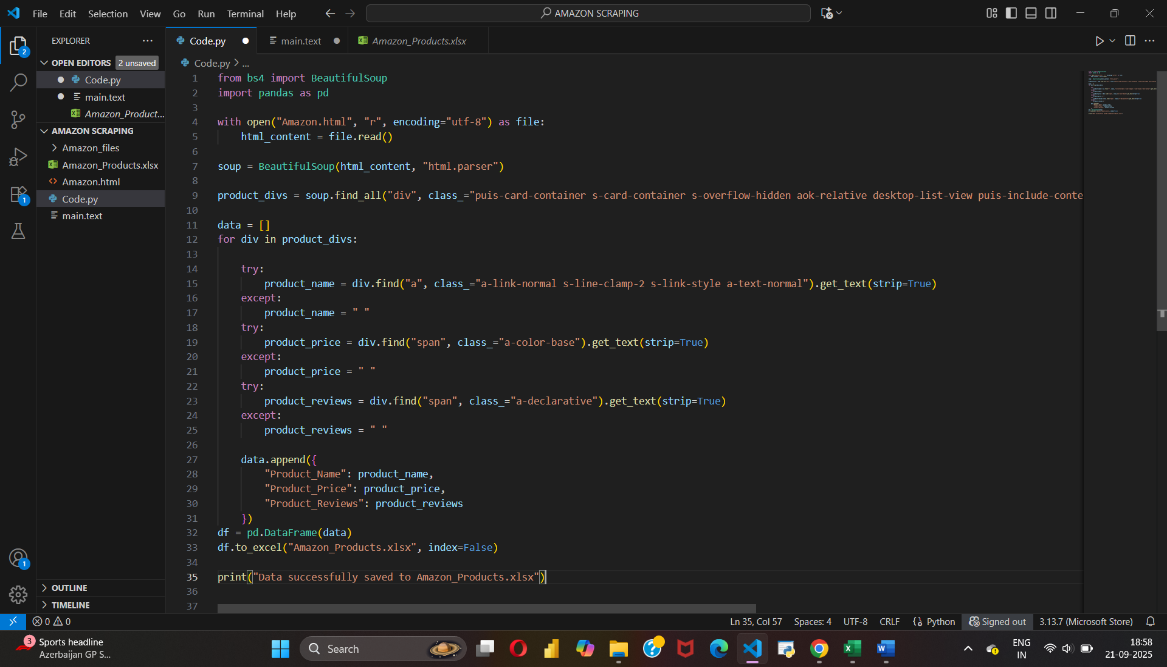
* **Product Name** is extracted from a <span> tag.
* **Product Price** is extracted from a <span> tag with class a-price-whole.
* **Product Reviews** are extracted from a <span> tag with class a-size-base.

**Step 5: Storing Data in Pandas DataFrame**

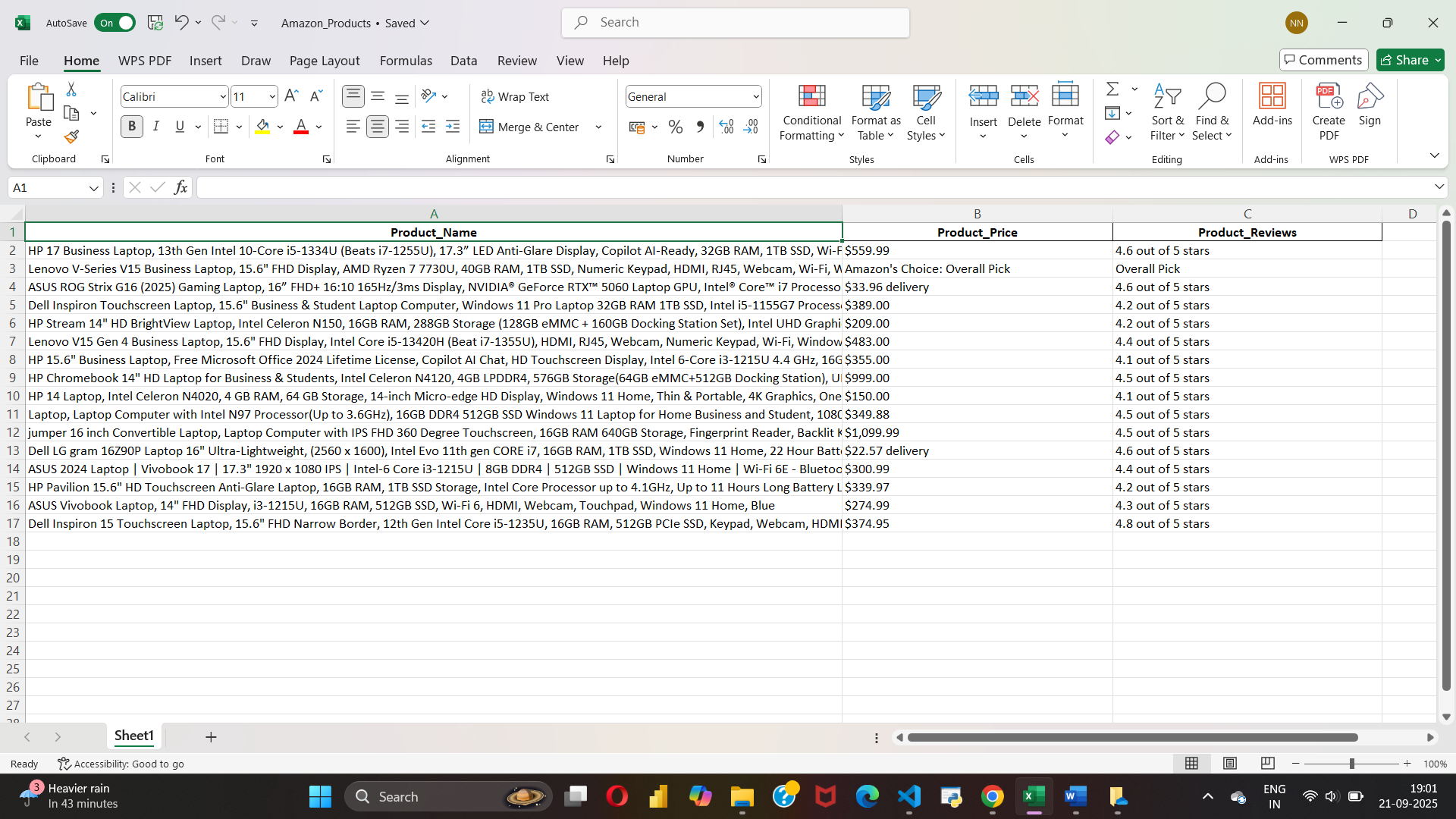
The extracted data is stored in a structured format (rows and columns) using Pandas.

**Step 6: Exporting to Excel**

The final dataset is exported to an Excel file named Amazon\_Products.xlsx.

1. **Code **

**6. Results**

The output an Excel file named **Amazon\_Products.xlsx** that contains a structured list of Amazon products.

|  |  |  |
| --- | --- | --- |
|  |  |  |

**7. Conclusion**

This project demonstrates how **web scraping and data processing** can automate the collection of product information from e-commerce websites.  
By using Python libraries like **BeautifulSoup** and **Pandas**, the script efficiently extracts and organizes data, saving time and reducing manual effort.

The method can be extended to:

* Track competitor products
* Monitor price changes
* Analyze customer reviews
* Generate real-time business insights