Intelligent Automation Project Report

Class: PE01

Student 1: Ray Leong

Student 2: Bryan Koh

1. Objectives

The objective of our project is to create a program that compiles and compares similar products from different ecommerce sites for the user to have an easier purchasing decision.

Firstly, the user will input the name of the product they want to search, along with the number of searches to get.

Secondly, we will use UiPath automation to scrape then chosen product from 4 popular ecommerce websites:

1. *Amazon*
2. *Lazada*
3. *Shoppee*
4. *AliExpress*

Thirdly, we will use UiPath functions to merge the datatables and sort them based either on Price, Reviews or Ratings according to the User's Input.

Therefore, our program helps users to easily find and compare products across ecommerce sites to make the best decision.

2. Functionalities & Description

Function	**Description**	**Student**
Input Product Name	Get user's chosen product	Ray
Input Number of search results	Get user's limit for the number of searches	Bryan
Data Scraping for Amazon	Get product details from Amazon: ● Product Name ● Price ● Original Price ● Rating ● Link to Product	Ray
Data Scraping for Lazada	Get product details from Lazada: (same as amazon)	Ray
Data Scraping for Shoppee	Get product details Shoppee: (same as amazon)	Bryan
Data Scraping for AliExpress	Get product details AliExpress: (same as amazon)	Bryan
Manual Scraping of Ratings	Some websites do not have ratings value so manual scraping is done	Ray
Merging Datatables	Merging all the	Ray

	scraped Datatables together into a single Datatable	
Parallel Data Scraping	Running all 4 Data Scraping as a single process	Bryan
Sorting products	Sorting the Datatables as Price, Reviews or Ratings according to the user’s input	Bryan
Displaying sorted table under input file name	Displaying final output into excel with name chosen by User	Ray
Displaying Website Data under an index	Display all original website data with name as website with index chosen by user, eg Lazada50	Bryan

3. Learning Journey & Challenges Overcome

Ray:

During this project there were many difficulties when scraping the data from Lazada.

Lazada would not show the star ratings on the home page, nor would it show the original price of the products.

Instead I had to research online to find out other methods of scraping data such as getting the “style” of the division instead of just the “text”.

Furthermore, there were many unwanted prefixes and formatting errors from the scraped data that led to errors when merging the datatables. Therefore, I also had to research several additional functions to remove certain characters from the rows in the data tables.

Overall, this project has led me to become far more comfortable with UiPath. With a greater understanding of the software, I now have greater proficiency with UiPath.

Bryan:

During the project, the basics were simple, scraping from websites were all running initially, but as integration and more sequences started to pile up, errors started appearing almost everywhere.

One example will be if the search results were too low, some products may not have certain columns such as reviews, so actions such as making it nullable must be taken.

Another example is upon opening a website through a string, all websites contain different ways of handling strings in URLs which prove to be a challenge after a number of results are tried, such as AliExpress having the search string in between the URL and Lazada not allowing capital letters on it.

Throughout the project, I’ve been constantly researching on the fixing of the errors, which also helped me understand the limitless potential on the different things that UIPath is able to achieve, helping me understand more about UIPath and also getting me more efficient at using it.

4. Conclusion

In conclusion, our program has proved itself to be a success.
It is able to scrape the data from various sites and compile the information in an easily digestible format. Being a practical and useful program that can help users make the best decisions when finding products.

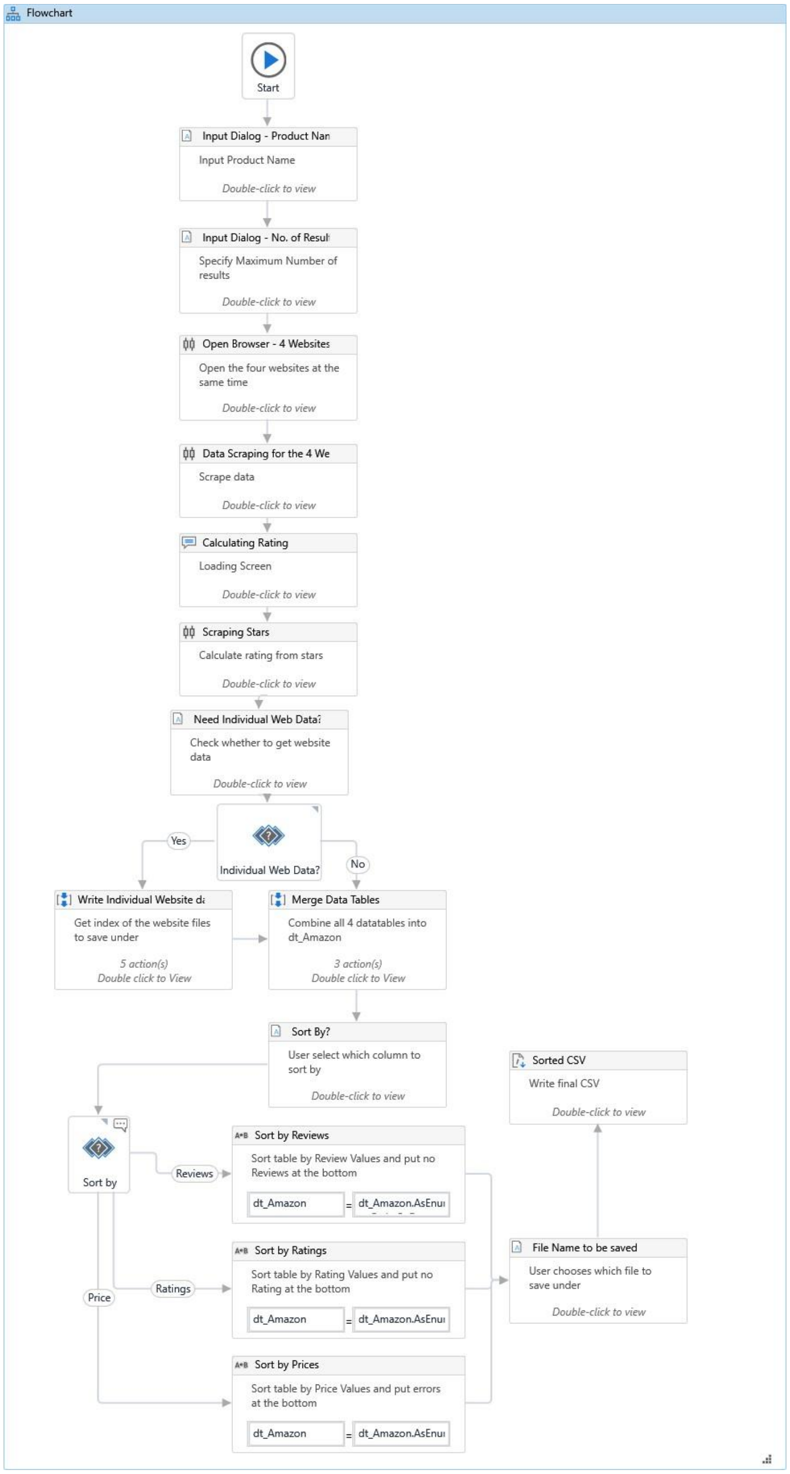
However, we also realize the program has several limitations.
For example, when scraping large amounts of search results the program might miss out some products due to scraping accuracy, which may affect the decision making of Users.

Another example would also be some websites slowing down the UiPath program from accessing the website due to it being a bot, therefore making the program run slower overall.

Overall, the program is very beneficial for users to make the best decisions through multiple websites after letting the program boot up and load through all websites with limited searches.

5. Appendix: UiPath scripts (screenshots of workflow and output results)

Main FlowChart:



Input Dialog - Product Nan
Input Product Name
Double-click to view

Input Dialog - No. of Resul
Specify Maximum Number of results
Double-click to view

Open Browser - 4 Websites
Open the four websites at the same time
Double-click to view

Data Scraping for the 4 We
Scrape data
Double-click to view

Calculating Rating
Loading Screen
Double-click to view

Scraping Stars
Calculate rating from stars
Double-click to view

Need Individual Web Data?
Check whether to get website data
Double-click to view

Write Individual Website da
Get index of the website files to save under
5 action(s)
Double click to View

Merge Data Tables
Combine all 4 datatables into dt_Amazon
3 action(s)
Double click to View

Sort By?
User select which column to sort by
Double-click to view

Sort by Reviews
Sort table by Review Values and put no Reviews at the bottom
dt_Amazon = dt_Amazon.AsEnu

Sort by Ratings
Sort table by Rating Values and put no Rating at the bottom
dt_Amazon = dt_Amazon.AsEnu

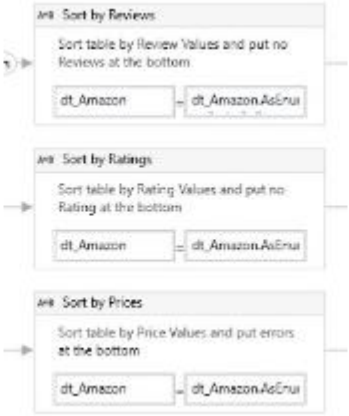
Sort by Prices
Sort table by Price Values and put errors at the bottom
dt_Amazon = dt_Amazon.AsEnu

File Name to be saved
User chooses which file to save under
Double-click to view

Sorted CSV
Write final CSV
Double-click to view

Calculating Ratings basically adds a message box to tell the user that Rating Calculation is undergoing.

Sort By:



Sort by Reviews

Sort table by Review Values and put no Reviews at the bottom

dt_Amazon = dt_Amazon.AsEnu

Sort by Ratings

Sort table by Rating Values and put no Rating at the bottom

dt_Amazon = dt_Amazon.AsEnu

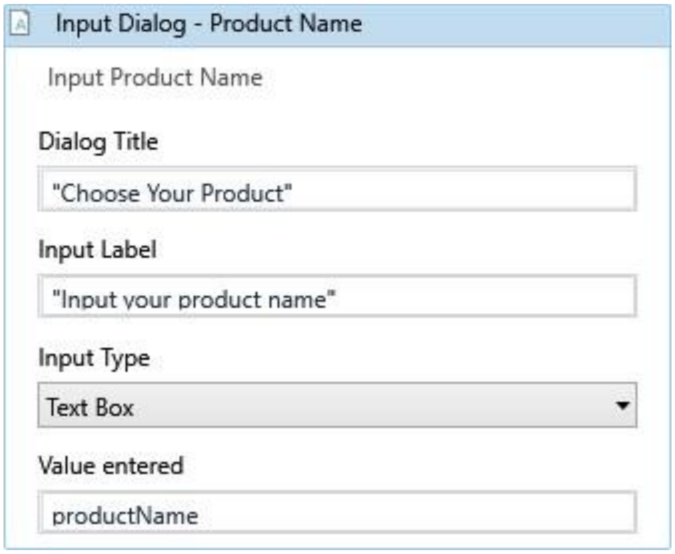
Sort by Prices

Sort table by Price Values and put errors at the bottom

dt_Amazon = dt_Amazon.AsEnu

Assign function sorts DataTable by Descending Order while moving all null or error values at the bottom of the list.

Input Product Name:



Input Dialog - Product Name

Input Product Name

Dialog Title

"Choose Your Product"

Input Label

"Input your product name"

Input Type

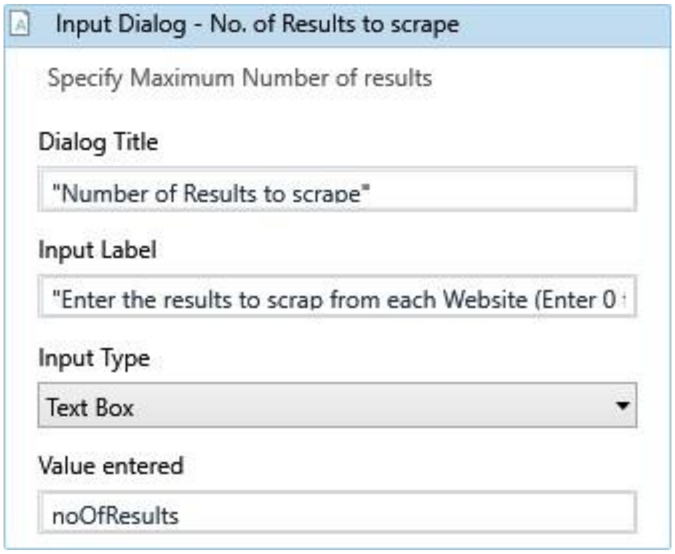
Text Box

Value entered

productName

Take user’s input of which product to scrape and save it as prodName

Input Search Results:



Input Dialog - No. of Results to scrape

Specify Maximum Number of results

Dialog Title

"Number of Results to scrape"

Input Label

"Enter the results to scrap from each Website (Enter 0

Input Type

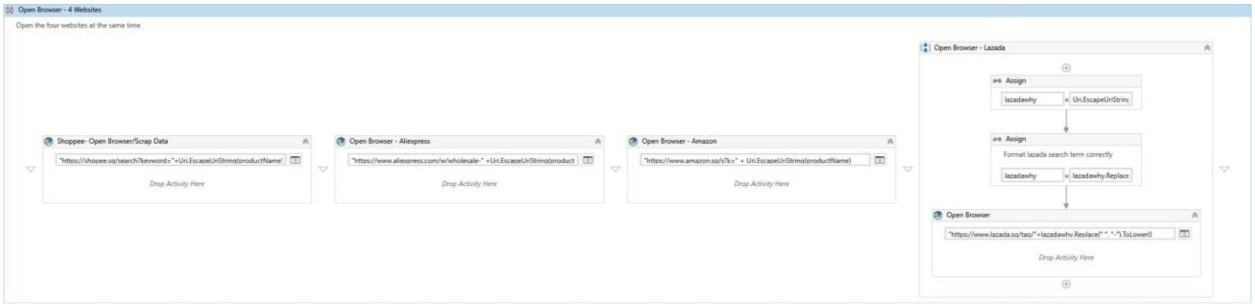
Text Box

Value entered

noOfResults

Check how many results does the user want from each website, saved under noOfResults.

Open all websites:



Open the four websites at the same time

"https://shopee.sg/search?keyword="+Uri.EscapeUriString(productName

"https://www.aliexpress.com/w/wholesale-" +Uri.EscapeUriString(product

"https://www.amazon.sg/s?k=" + Uri.EscapeUriString(productName)

lazadawhy = Uri.EscapeUriStrin

Format lazada search term correctly

lazadawhy = lazadawhy.Replace

"https://www.lazada.sg/tag/"+lazadawhy.Replace(" ", "-").ToLower()

Open:

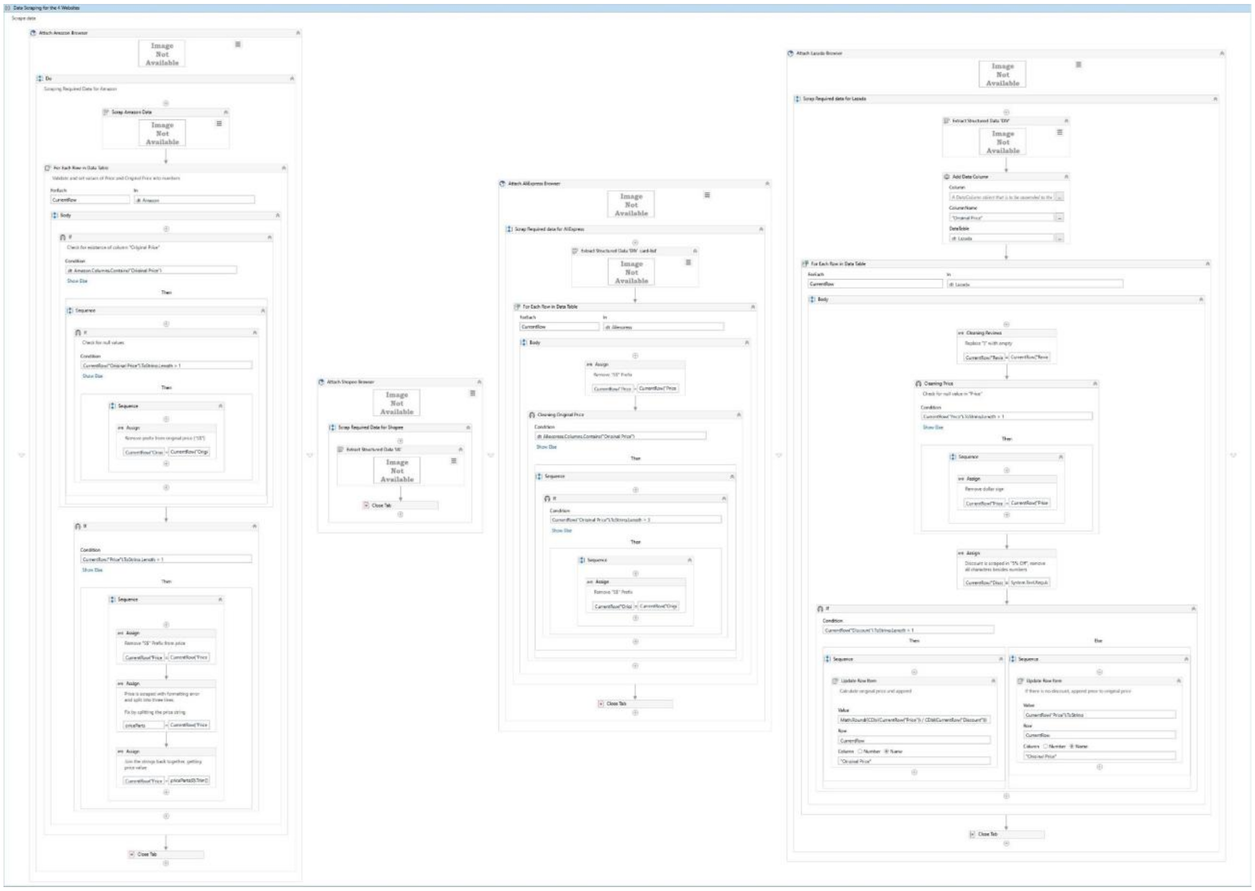
1. Amazon
2. Lazada
3. Shoppee
4. AliExpress

By using Uri.EscapeUriString(productName) to format spaces correctly for website links.

The exception is Lazada which requires an assign function to make the url link in the correct format. Changing from “%20” in between words to “-”.

Lazada also require all product names in links to be lower case, using an assign function to lower case it is assigned to the product name.

Data Scraping:



Data Scraping for the 4 Websites

Scrape data

Attach Amazon Browser

Do

Scraping Required Data for Amazon

Scrap Amazon Data

For Each Row in Data Table

Validate and set values of Price and Original Price into numbers

ForEach: CurrentRow In: dt Amazon

Body

If

Check for existence of column "Original Price"

Condition: dt Amazon.Columns.Contains("Original Price")

Show Else

Then

Sequence

If

Check for null values

Condition: CurrentRow("Original Price").ToString.Length > 1

Show Else

Then

Sequence

Assign

Remove prefix from original price ("S$")

CurrentRow("Origi = CurrentRow("Origi

If

Condition: CurrentRow("Price").ToString.Length > 1

Show Else

Then

Sequence

Assign

Remove "S$" Prefix from price

CurrentRow("Price = CurrentRow("Price

Assign

Price is scraped with formatting error and split into three lines.

Fix by splitting the price string

priceParts = CurrentRow("Price

Assign

Join the strings back together, getting price value

CurrentRow("Price = priceParts(0).Trim()

Close Tab

Attach Shopee Browser

Scrap Required Data for Shopee

Extract Structured Data 'UL'

Close Tab

Attach AliExpress Browser

Scrap Required data for AliExpress

Extract Structured Data 'DIV' card-list

For Each Row in Data Table

ForEach: CurrentRow In: dt Aliexpress

Body

Assign

Remove "S$" Prefix

CurrentRow("Price = CurrentRow("Price

Cleaning Original Price

Condition: dt Aliexpress.Columns.Contains("Original Price")

Show Else

Then

Sequence

If

Condition: CurrentRow("Original Price").ToString.Length > 1

Show Else

Then

Sequence

Assign

Remove "S$" Prefix

CurrentRow("Origi = CurrentRow("Origi

Close Tab

Attach Lazada Browser

Scrap Required data for Lazada

Extract Structured Data 'DIV'

Add Data Column

Column: A DataColumn object that is to be appended to the

ColumnName: "Original Price"

DataTable: dt Lazada

For Each Row in Data Table

ForEach: CurrentRow In: dt Lazada

Body

Cleaning Reviews

Replace "()" with empty

CurrentRow("Revie = CurrentRow("Revie

Cleaning Price

Check for null value in "Price"

Condition: CurrentRow("Price").ToString.Length > 1

Show Else

Then

Sequence

Assign

Remove dollar sign

CurrentRow("Price = CurrentRow("Price

Assign

Discount is scraped in "5% Off", remove all characters besides numbers

CurrentRow("Disc = System.Text.Regul

If

Condition: CurrentRow("Discount").ToString.Length > 1

Then

Sequence

Update Row Item

Calculate original price and append

Value: Math.Round(CDbl(CurrentRow("Price")) / CDbl(CurrentRow("Discount"))

Row: CurrentRow

Column ○ Number ● Name

"Original Price"

Else

Sequence

Update Row Item

If there is no discount, append price to original price

Value: CurrentRow("Price").ToString

Row: CurrentRow

Column ○ Number ● Name

"Original Price"

Close Tab

Data scraping searches for:

1. *Product name*
2. *Price*
3. *Original Price*
4. *Number of Reviews*
5. *Rating*
6. *Link*

After scraping the above features, data cleaning is done to format the string correctly for data table merging.

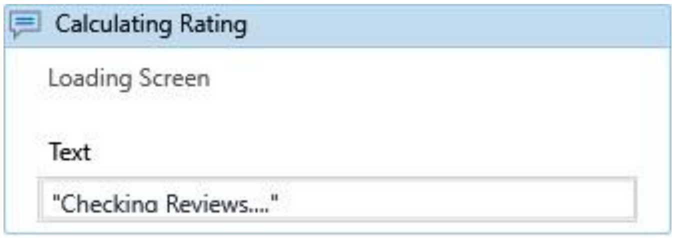
Amazon has prefixes in front of its “Price” and “Original Price” that need to be removed. Amazon “Price” is also split into 3 lines due to the styling in the web page. This is fixed by splitting the “Price” string into two before joining them back in the same line.

Lazada has a dollar sign prefix in front of “Price” that needs to be removed.
Lazada also has several unwanted characters in the number of reviews, therefore regular expression is used to replace all characters besides numerical values with null.

Lazada also has no direct component to get the original price, instead the discount percent is scraped and the original price is calculated based on that.

AliExpress has prefixes in front of “Price” that need to be removed.

Calculate Rating:


Calculating Rating

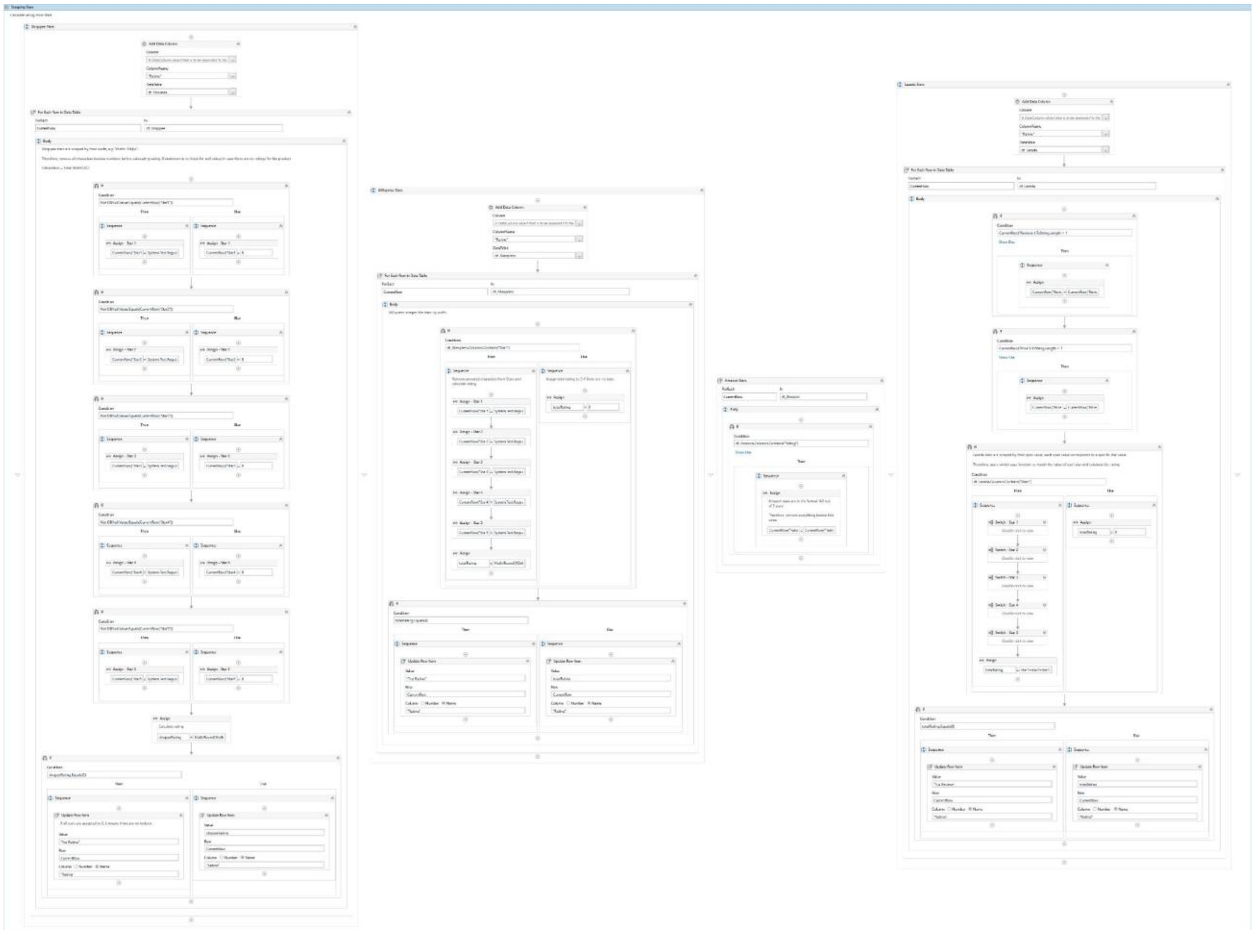
Loading Screen

Text

"Checking Reviews...."

Validate and show that Ratings calculations is running.

Scraping Stars:



Most websites do not show the rating value in their catalog page, therefore custom data definition and several calculations are needed to get the rating.

Amazon:
When scraping the stars Amazon returns it in the format “4.5 out of 5 stars”.
Therefore, all characters after the first few value is removed to get the rating.

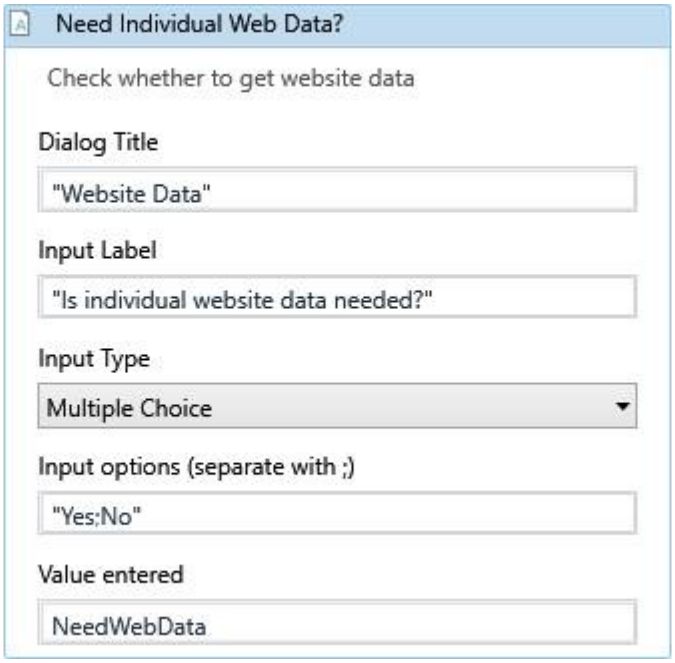
Lazada:
For Lazada, each individual star is scraped for its class, returning values such as “_9-ogB W1iJ5”.
In Lazada, every star rating from 0.1 to 1 has a unique class value.
Therefore, a switch case is used to match the class value with the rating value. After all 5 stars are assigned their values, the total rating is calculated.

Shoppee:
Similarly to Lazada, each individual star is scraped in Shoppee. However, instead of scraping the class, the <style> of the star is scraped, returning values such as “Width: 100px”.

So, the <style> is cleaned, removing all characters that are not numbers, returning a value ranging from 0 to 100.
The rating is then calculated by adding all 5 star values together and dividing by 500.

AliExpress:
Similarly to Shoppee, AliExpress individually scrapes each star’s <style> for the width.
Then the width is cleaned to get a value from 0 to 100, with the rating calculated by adding all 5 values together and dividing by 500.

Ask for individual web data:



Need Individual Web Data?

Check whether to get website data

Dialog Title

"Website Data"

Input Label

"Is individual website data needed?"

Input Type

Multiple Choice

Input options (separate with ;)

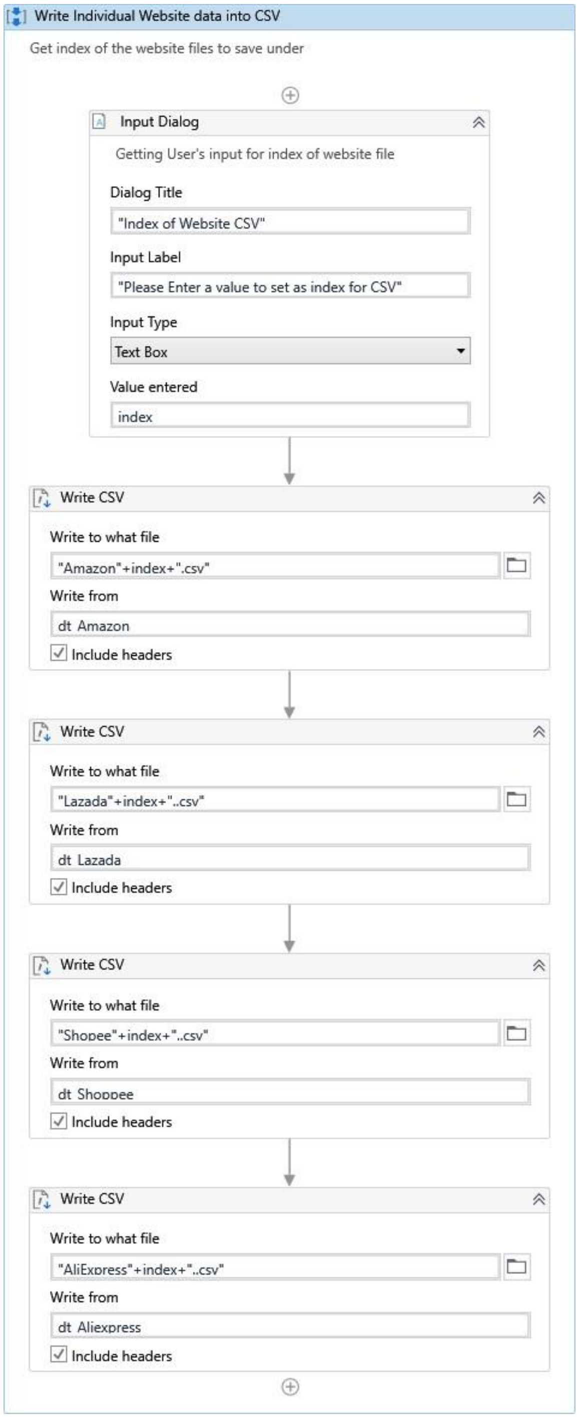
"Yes;No"

Value entered

NeedWebData

Asking if the user needs individual website data, handled by a flow switch.

Write individual website data:



Write Individual Website data into CSV

Get index of the website files to save under

Input Dialog

Getting User's input for index of website file

Dialog Title
"Index of Website CSV"

Input Label
"Please Enter a value to set as index for CSV"

Input Type
Text Box

Value entered
index

Write CSV

Write to what file
"Amazon"+index+".csv"

Write from
dt Amazon

☑ Include headers

Write CSV

Write to what file
"Lazada"+index+"..csv"

Write from
dt Lazada

☑ Include headers

Write CSV

Write to what file
"Shopee"+index+"..csv"

Write from
dt Shoppee

☑ Include headers

Write CSV

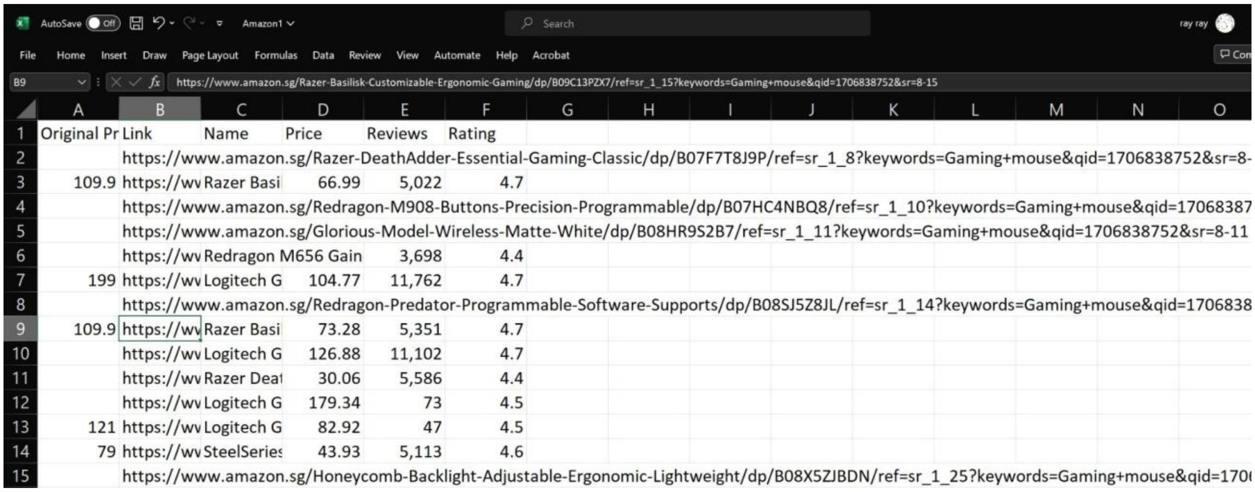
Write to what file
"AliExpress"+index+"..csv"

Write from
dt Aliexpress

☑ Include headers

Upon selecting “Yes” for saving individual website data, user will be prompt to add an index after each file name, such as Lazada50 or Shopee1.

One Example from Amazon after selecting “Yes”:

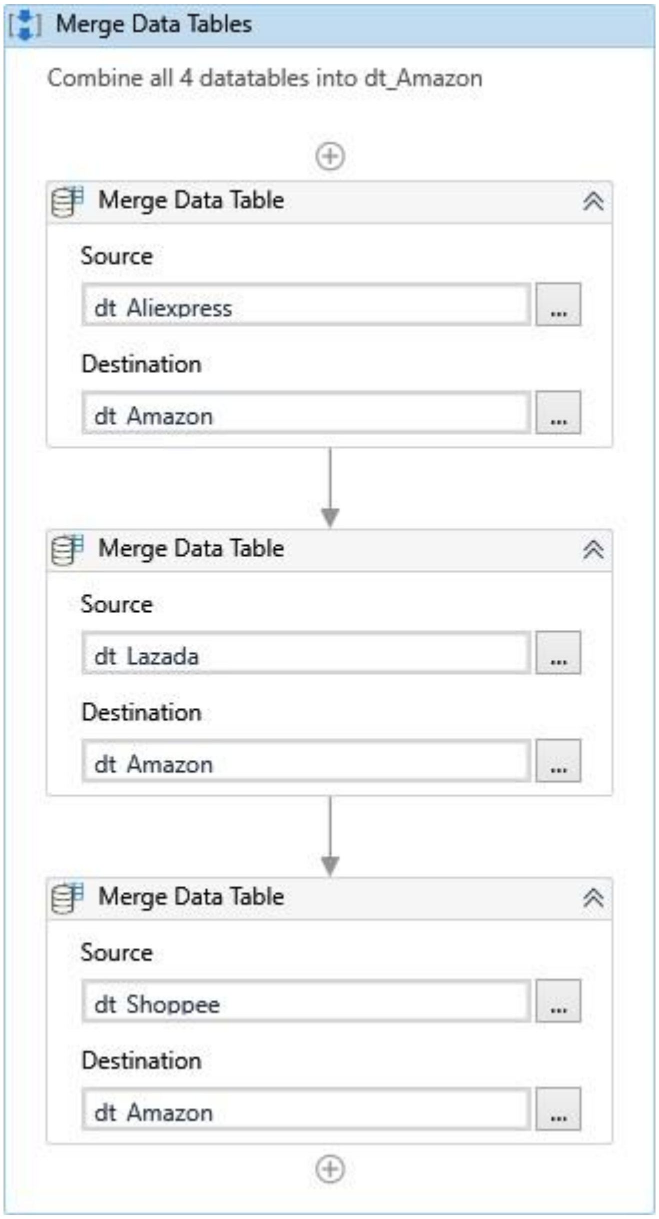


B9 https://www.amazon.sg/Razer-Basilisk-Customizable-Ergonomic-Gaming/dp/B09C13PZX7/ref=sr_1_15?keywords=Gaming+mouse&qid=1706838752&sr=8-15

	A	B	C	D	E	F
1	Original Pr	Link	Name	Price	Reviews	Rating
2		https://www.amazon.sg/Razer-DeathAdder-Essential-Gaming-Classic/dp/B07F7T8J9P/ref=sr_1_8?keywords=Gaming+mouse&qid=1706838752&sr=8-				
3	109.9	https://wv	Razer Basi	66.99	5,022	4.7
4		https://www.amazon.sg/Redragon-M908-Buttons-Precision-Programmable/dp/B07HC4NBQ8/ref=sr_1_10?keywords=Gaming+mouse&qid=17068387				
5		https://www.amazon.sg/Glorious-Model-Wireless-Matte-White/dp/B08HR9S2B7/ref=sr_1_11?keywords=Gaming+mouse&qid=1706838752&sr=8-11				
6		https://wv	Redragon M656 Gain		3,698	4.4
7	199	https://wv	Logitech G	104.77	11,762	4.7
8		https://www.amazon.sg/Redragon-Predator-Programmable-Software-Supports/dp/B08SJ5Z8JL/ref=sr_1_14?keywords=Gaming+mouse&qid=1706838				
9	109.9	https://wv	Razer Basi	73.28	5,351	4.7
10		https://wv	Logitech G	126.88	11,102	4.7
11		https://wv	Razer Deat	30.06	5,586	4.4
12		https://wv	Logitech G	179.34	73	4.5
13	121	https://wv	Logitech G	82.92	47	4.5
14	79	https://wv	SteelSerie:	43.93	5,113	4.6
15		https://www.amazon.sg/Honeycomb-Backlight-Adjustable-Ergonomic-Lightweight/dp/B08X5ZJBDN/ref=sr_1_25?keywords=Gaming+mouse&qid=170				

From Amazon, links are overshooting due to compressed but prices and the rest are all accurate. This is the search results of all the data scraped from Amazon into CSV. Other Websites produce similar results as well, but with manual star calculation columns, so Amazon DataTable will be used as final to remove manual calculation, showed later on.

Merge data tables:



Merge Data Tables

Combine all 4 datatables into dt_Amazon

Merge Data Table

Source: dt Aliexpress

Destination: dt Amazon

Merge Data Table

Source: dt Lazada

Destination: dt Amazon

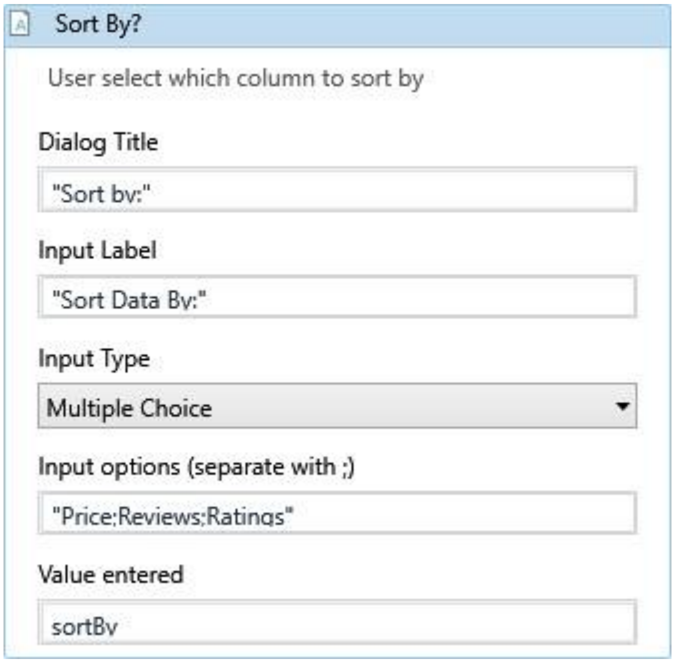
Merge Data Table

Source: dt Shoppee

Destination: dt Amazon

Merge all DataTables into Amazon DataTable as Amazon does not have the manually calculated Star Ratings, so only the final ratings will be merged.

Sort by input:



Sort By?

User select which column to sort by

Dialog Title

"Sort by:"

Input Label

"Sort Data By:"

Input Type

Multiple Choice

Input options (separate with ;)

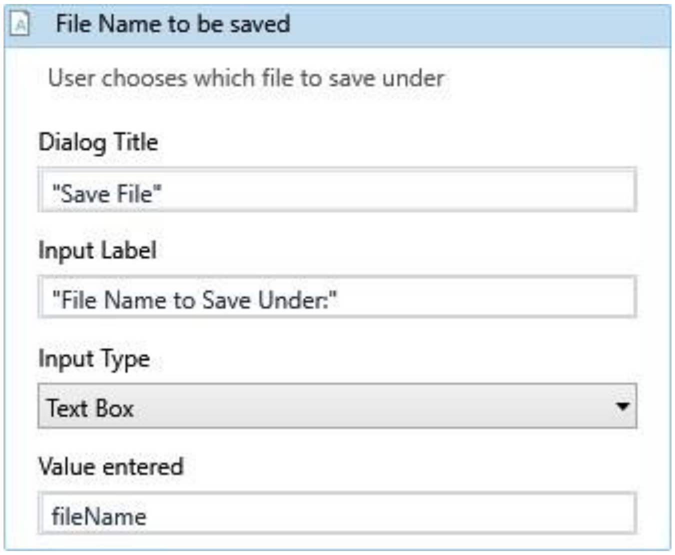
"Price;Reviews;Ratings"

Value entered

sortBy

Asking user if they want to sort by Price, Ratings or Reviews.

File name to be saved under:



File Name to be saved

User chooses which file to save under

Dialog Title: "Save File"

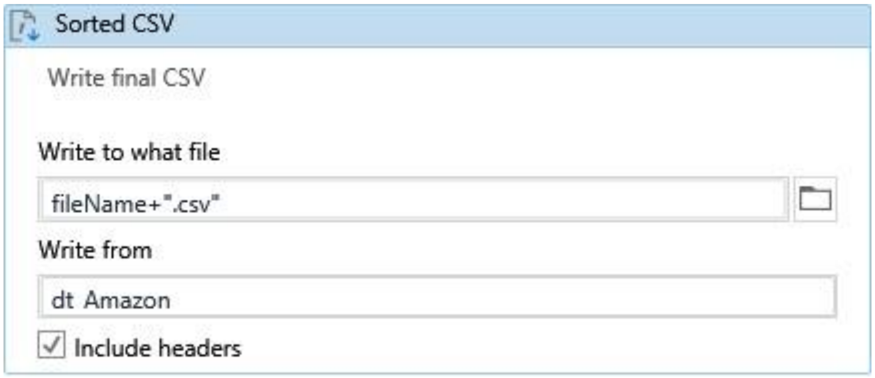
Input Label: "File Name to Save Under:"

Input Type: Text Box

Value entered: fileName

Ask the user what file name would the sorted data table be saved under.

Sorted CSV:



Sorted CSV

Write final CSV

Write to what file

fileName+".csv"

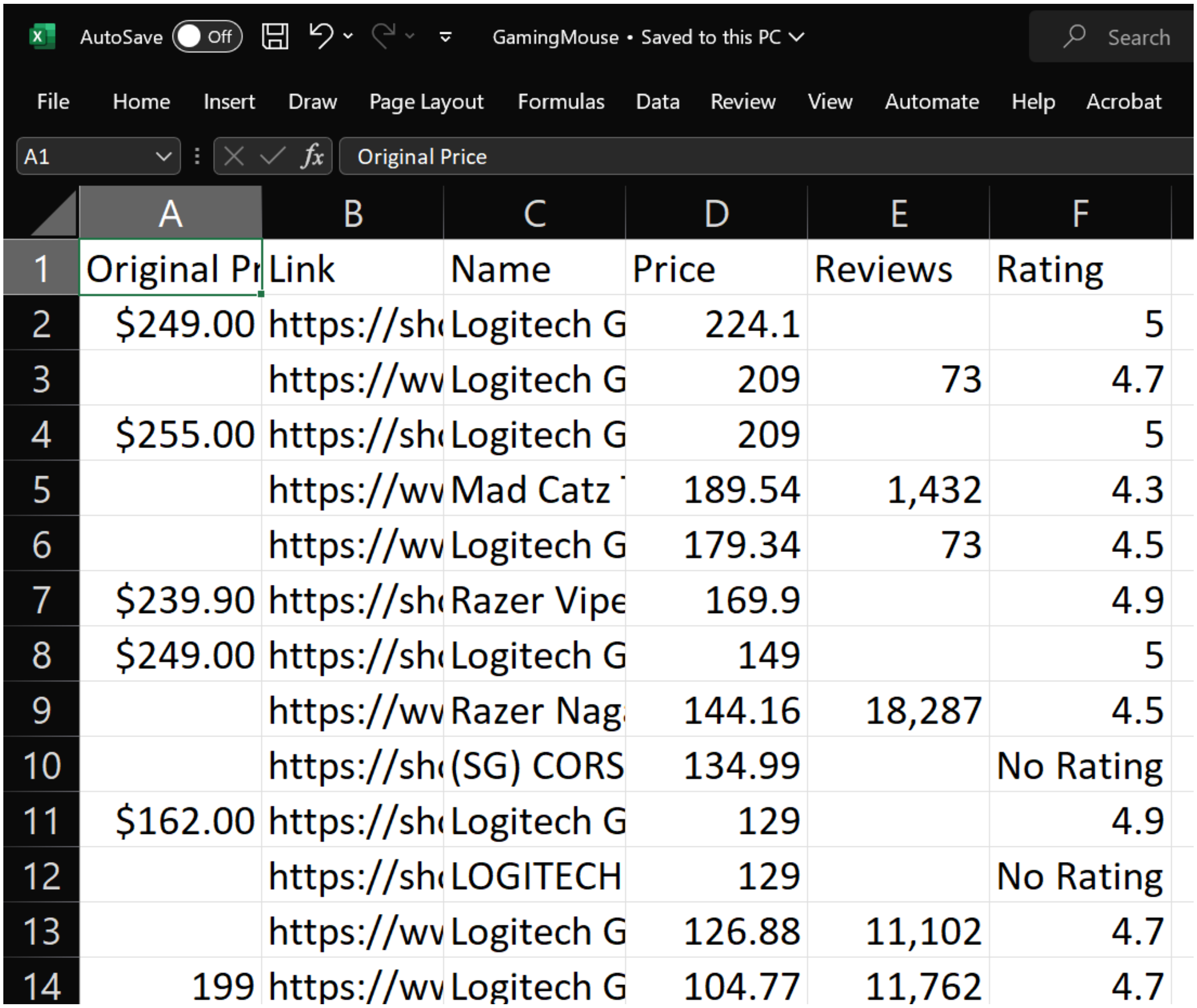
Write from

dt Amazon

☑ Include headers

Write into csv according to the user’s desired file name.

Final CSV:



	A	B	C	D	E	F
1	Original Pr	Link	Name	Price	Reviews	Rating
2	$249.00	https://sh	Logitech G	224.1		5
3		https://wv	Logitech G	209	73	4.7
4	$255.00	https://sh	Logitech G	209		5
5		https://wv	Mad Catz	189.54	1,432	4.3
6		https://wv	Logitech G	179.34	73	4.5
7	$239.90	https://sh	Razer Vipe	169.9		4.9
8	$249.00	https://sh	Logitech G	149		5
9		https://wv	Razer Nag	144.16	18,287	4.5
10		https://sh	(SG) CORS	134.99		No Rating
11	$162.00	https://sh	Logitech G	129		4.9
12		https://sh	LOGITECH	129		No Rating
13		https://wv	Logitech G	126.88	11,102	4.7
14	199	https://wv	Logitech G	104.77	11,762	4.7

All 4 data tables are merged and sorted according to user input, in this case its Price.