



UI Path RPA Hackathon

Topic: Web Scrapping

Team Name: TechBloom

Team Members: Vishal Sharma (Leader)

Abhishek Pandey

Amit Kumar Prasad

Web Scrapping:

Web scraping is an application of Robotic Process Automation which is used in almost all the industries. Either it be a stock trading websites, e-commerce websites, commodities trading websites, etc, you can scrape the data from any of them based on your interest.

Now, the problem with performing web scraping manually is that, it is quite prone to errors and takes lot of time. Also, the data present on the websites is never static. It gets updated very frequently. So, the data that is stored at a point of instance might not be accurate always.

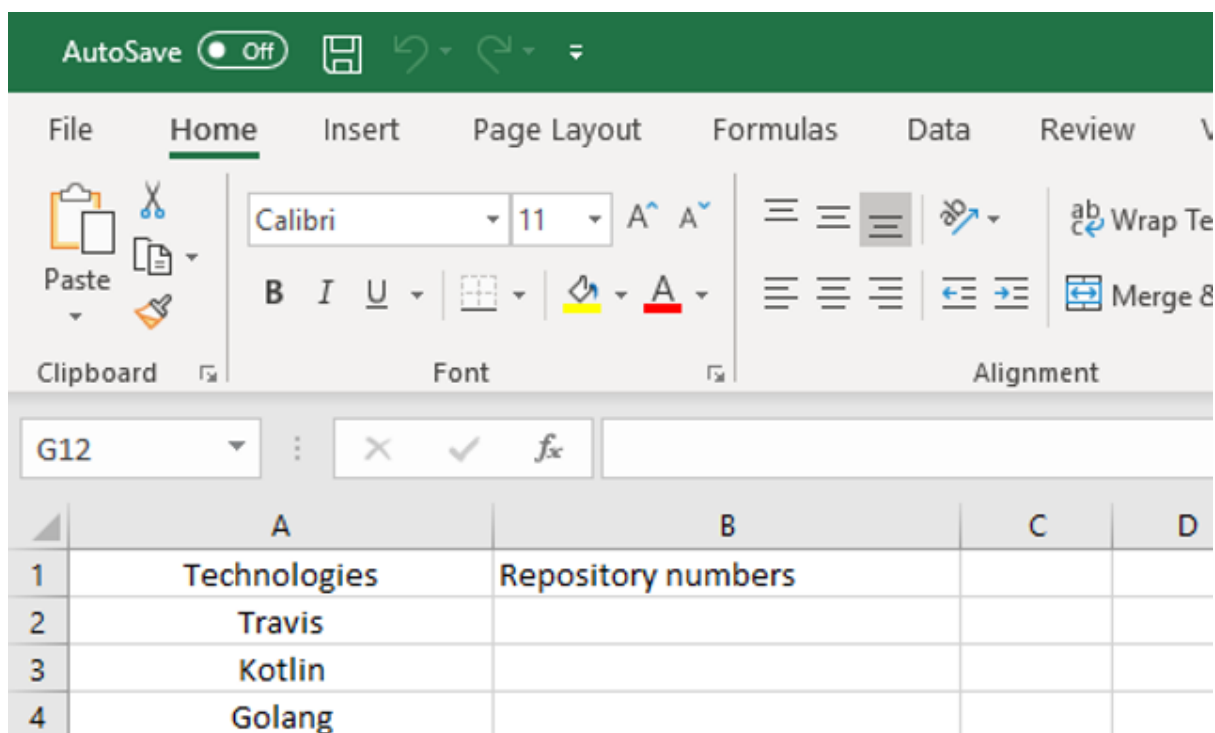
So, industries can simply automate this task. Below in this article I am going to show you, how to automate this task using UiPath.

Problem Statement: Task is to scrape the number of GitHub repositories for the top technologies in today's market.

How will you automate this task?

Solution:

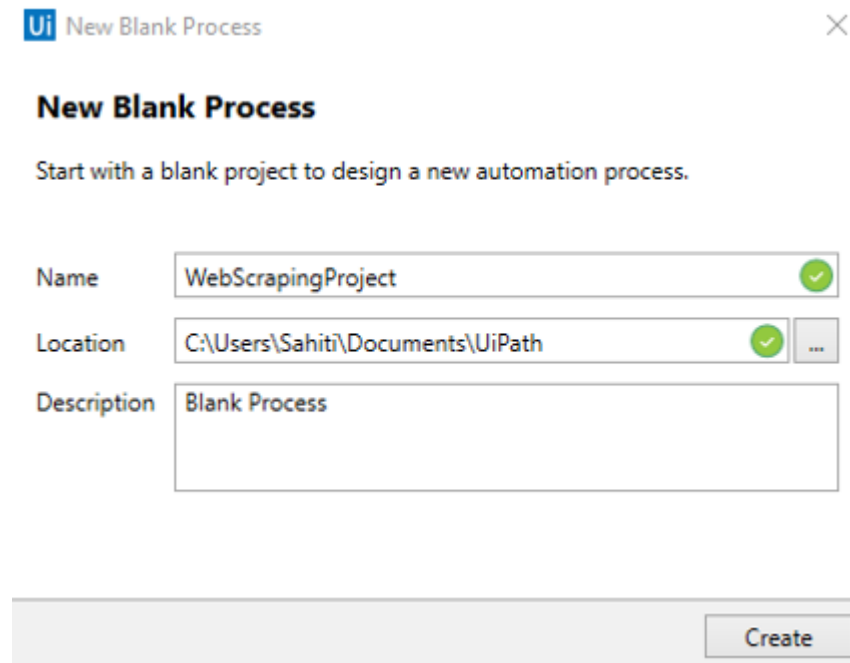
Step 1: Store the technologies list in an excel sheet with the *column name Technologies and Repository Count* as you can see below.



The screenshot shows the Microsoft Excel interface with the 'Home' tab selected. The ribbon includes options for File, Home, Insert, Page Layout, Formulas, Data, and Review. The 'Font' section shows 'Calibri' font and size '11'. The 'Alignment' section shows various alignment options. The worksheet grid shows columns A, B, C, and D, and rows 1 through 4. The data is as follows:

	A	B	C	D
1	Technologies	Repository numbers		
2	Travis			
3	Kotlin			
4	Golang			

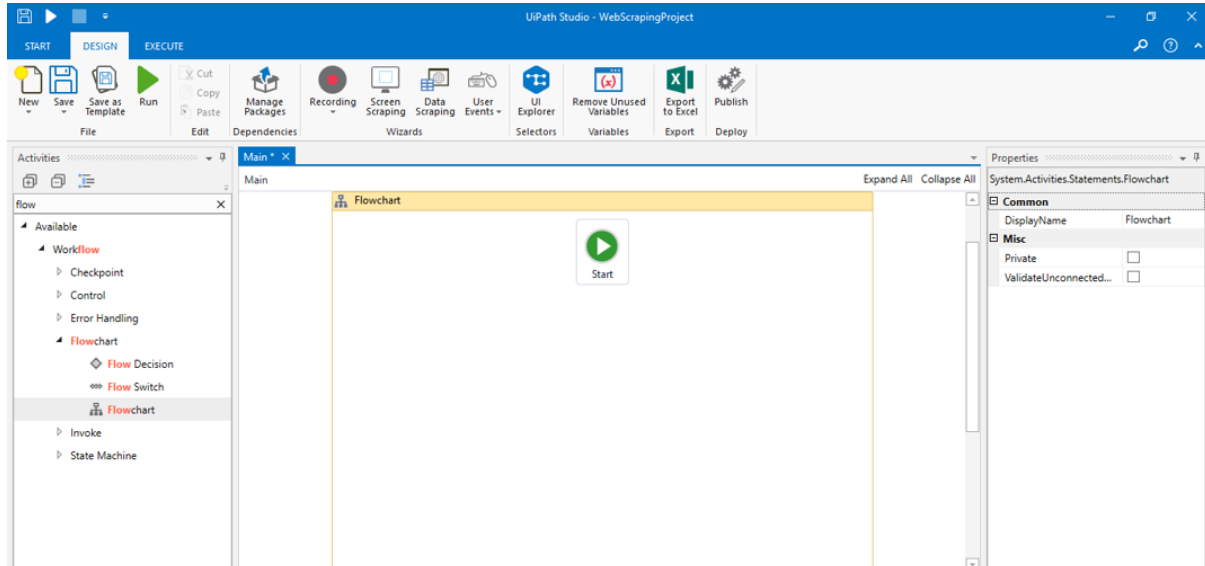
Step 2: Open **UiPath Studio** and create a **Blank Project**. Mention the Project Name, Location and Description. Then click on **Create**. Refer below.



The image shows the 'New Blank Process' dialog box in UiPath Studio. It has a title bar with the UiPath logo and a close button. The main title is 'New Blank Process'. Below it is a subtitle: 'Start with a blank project to design a new automation process.' There are three input fields: 'Name' with the value 'WebScrapingProject', 'Location' with the value 'C:\Users\Sahiti\Documents\UiPath', and 'Description' with the value 'Blank Process'. Each of the first two fields has a green checkmark icon to its right. At the bottom right is a 'Create' button.

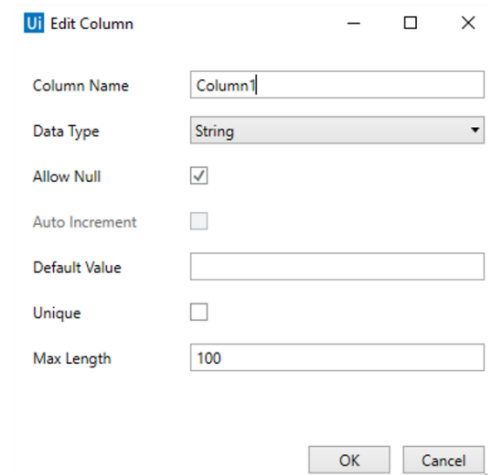
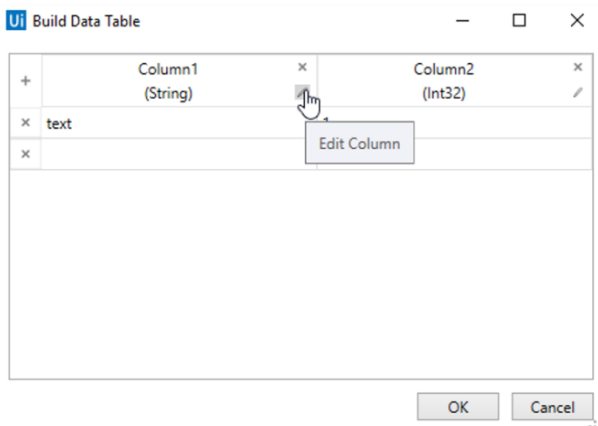
Name	WebScrapingProject
Location	C:\Users\Sahiti\Documents\UiPath
Description	Blank Process

Step 3: Once your dashboard opens, search for the **Flowchart** activity in the **Activity Pane** and drag it to the work space. *We are dragging the flowchart to ensure a proper workflow of the complete automation.* Refer below.

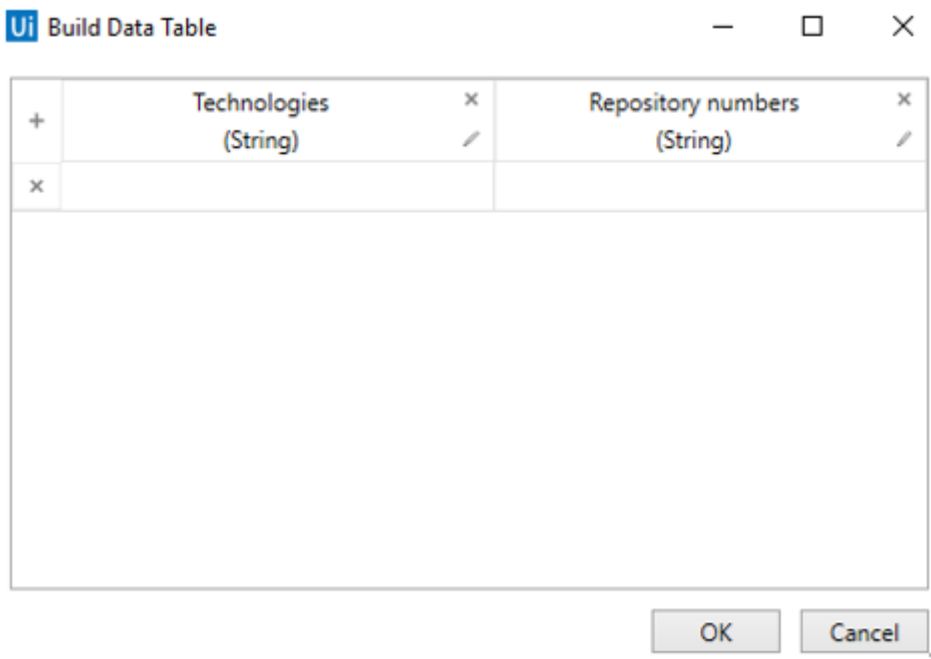


Step 4: Now, drag a **Build Data Table** activity from the **Activity Pane**. Connect it with the start point of the flowchart.

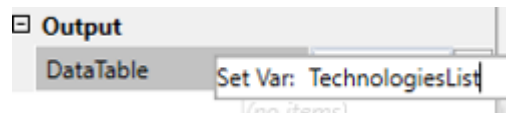
Step 4.1: Double click on the activity and click on the **Data Table** option. Then you have to mention the column names. Since we had only two columns in the excel sheet, we will mention the same column names in the Data Table. To do that **click on the edit column option** and mention the details. Refer below.



Step 4.2: After filling the details click on **OK**. This will create a Data Table. A Data Table is a table which will be used by UiPath to read the data present in the excel file and store the retrieved data in an excel file. Refer below.

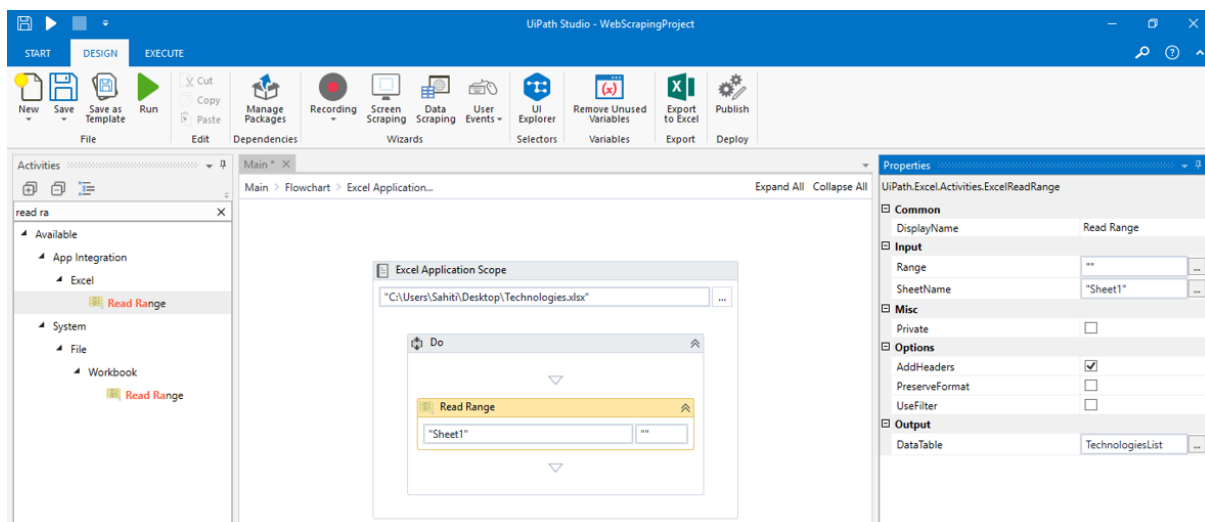


Step 4.3: Next, in the output section of the Data Table activity **mention a variable to store the output of the Data Table**. Here I have mentioned it as *TechnologiesList*. Refer below.

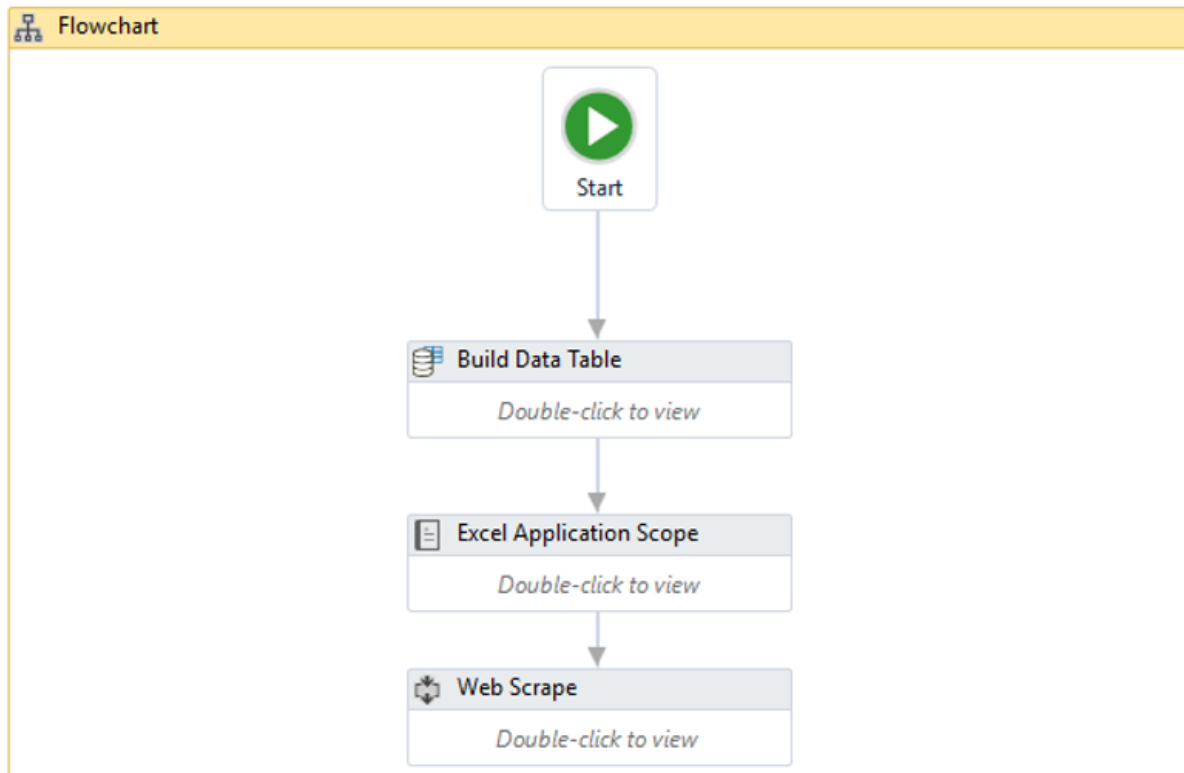


Step 5: Now go back to the **Flowchart** and add the **Excel Application Scope** activity from the **Activity Pane** to perform actions related to the Excel file. Then connect the **Build Data Table** activity to this activity in the flowchart.

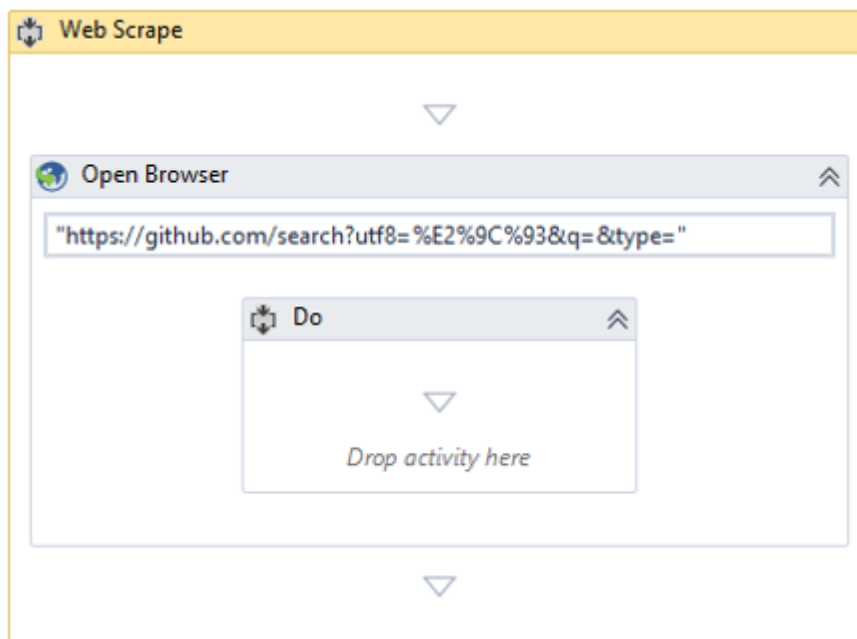
Step 5.1: Double click the **Excel Application scope** activity and mention the **path of the excel sheet**. Then, in the **Do** section of this activity drag the **Read Range** activity from the Activity pane and mention the **Sheet name and the Range**. Also, in the **output section** of the **Read Range** activity mention the **name of the Data Table variable** you created before i.e. *TechnologiesList*. Refer below.



Step 6: Now our next step is to extract the elements from the Web pages. To do that, **go back to the flowchart** and drag a **Sequence** from the **Activity Pane**. Then, connect the **Excel Application Scope** activity to this Sequence in the flowchart and rename the sequence as **Web Scrape** for better understanding as below.

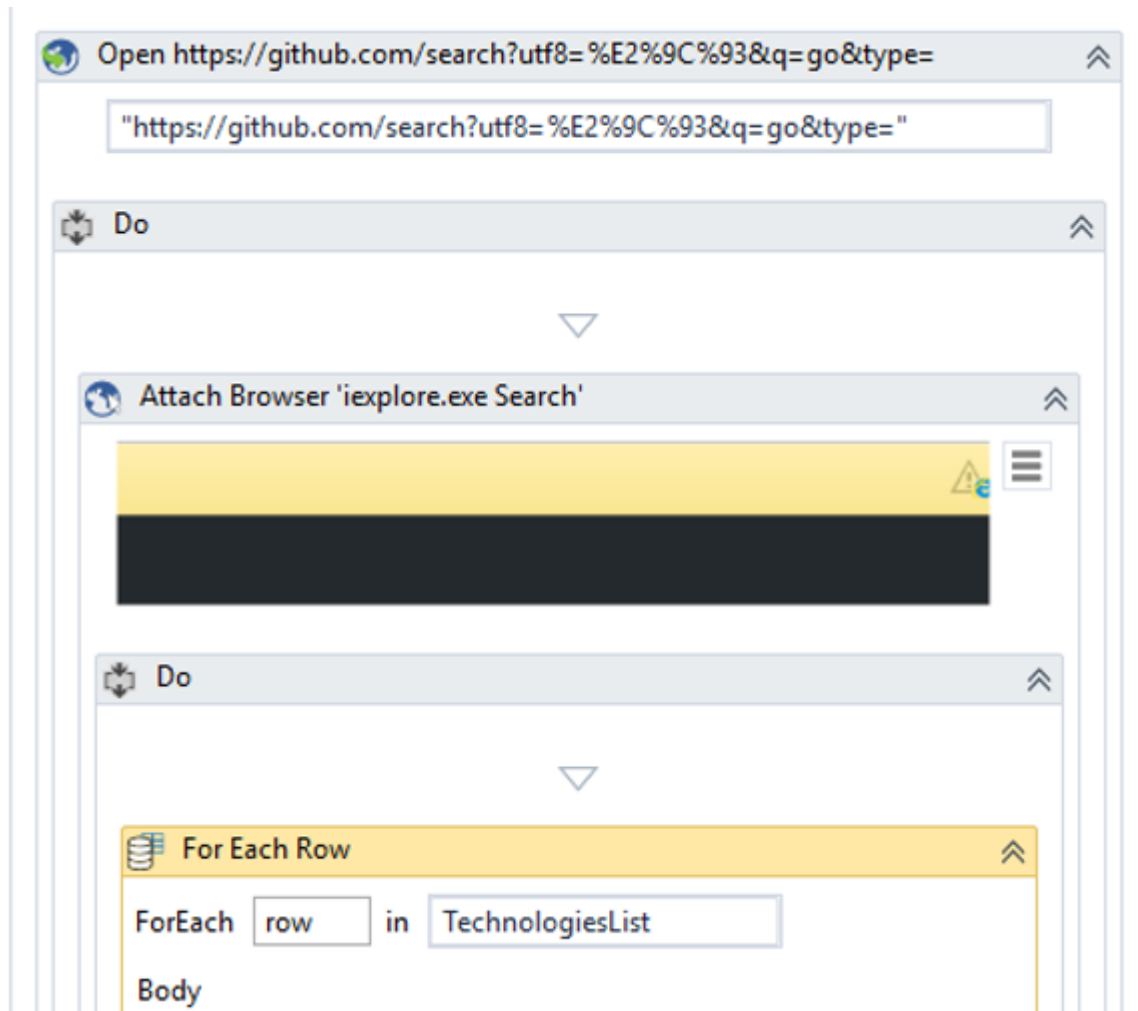


Step 6.1: Now, double click the Web Scrape sequence and drag the **Open Browser Activity**. In this activity mention the URL on which you wish to scrape the data. I will mention the **GitHub search URL** in double quotes as below.



Step 6.2: In the **Do** section of this activity drag the **Attach Browser** activity from the activity pane. Then just indicate on the browser or the screen. This is to make sure that all the activities have to occur on this specific web page.

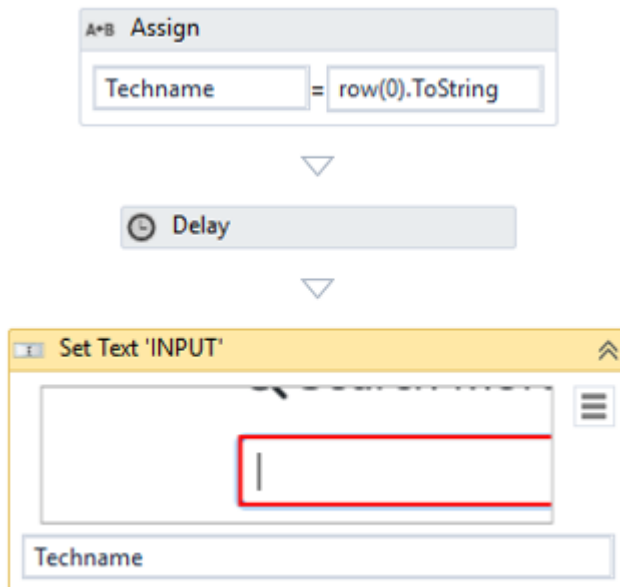
Step 6.3: Now, the **Do** section of the **Attach Browser** activity drag the **For Each Row** activity. In this activity mention the **Data Table variable** i.e. the *TechnologiesList* to start a loop for each row value in the Data Table. Refer below.



Step 6.4: In the **Body** section of the above activity, drag the **Assign** activity and mention the **Techname** variable in the **To** section and **row(0).ToString** in the **Value** section as below. This is to take the each and every technology name from the excel sheet and store it in the variable Techname.

Step 6.5: Then drag a **Delay** activity and mention a Delay of around *10-30 seconds*.

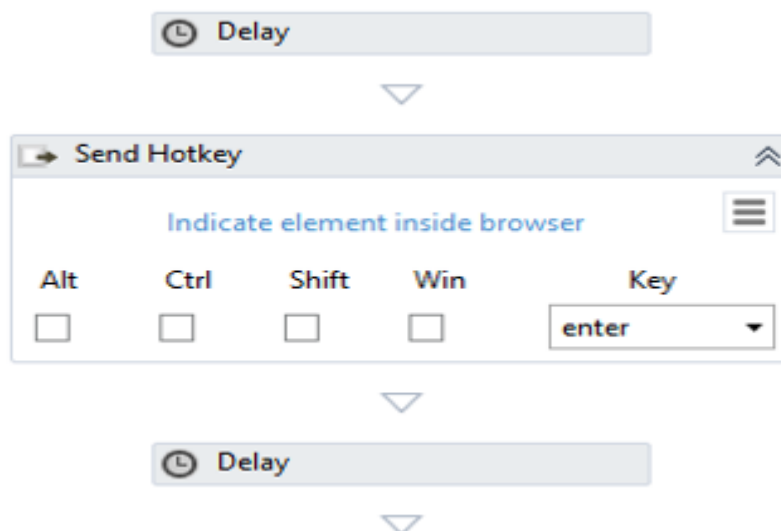
Step 6.6: Now, our next task is to type the technology name automatically. To do that you have to **Set Text activity** from the activity pane. Then you have to indicate on the screen, where the text should be automatically typed in. Here I will indicate it on the search bar. In the Text section off this activity, I will mention the *Techname* variable. Refer below.



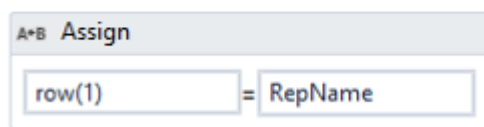
Step 6.7: Then drag a **Delay activity** and mention a delay of around 5-10 seconds.

Step 6.8: Next, drag the **Send Hotkey activity** and mention the key to be **enter**. This will help you automatically click on Enter on the web page.

Step 6.9: Now, again add delay to avoid any errors of around 10-30 seconds with the help of **Delay activity**. Refer below.

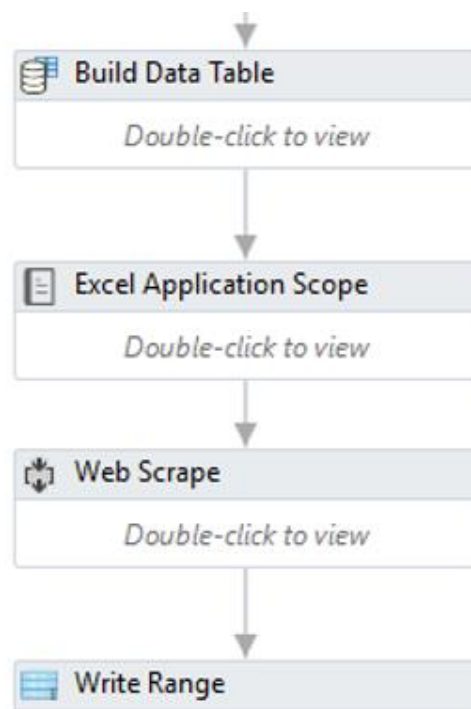


Step 6.10: Once you are done with the above steps, you have to next drag the **Get Text** activity from the activity pane and indicate on the browser, from where you wish to extract data. Here I will indicate on the screen where repositories are shown. Also, you have to mention an output variable in the output section of the properties pane this activity. Here I will mention the variable RepName. Refer below.

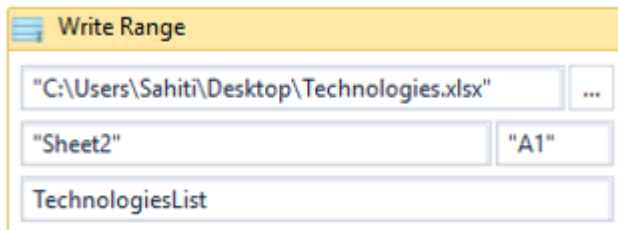


Step 6.11: Finally, you have to drag an **Assign** activity and mention row(1) in the To section and a variable to store Repository count. i.e. RepName. Refer above image.

Step 7: Now, you have to store the values back into the excel file. To do that, **go back to the flowchart** and add the **Write Range** activity from the activity pane. Connect the Web Scrape sequence to this activity as below.



Step 7.1: Then, mention the path of the excel sheet in quotes. Also mention the name of the Data Table, Sheet Number and the cell value from which it has to start writing data. Here the Data Table name is *TechnologiesList*, sheet number is *Sheet 2* and cell value is *A1*. Refer below.

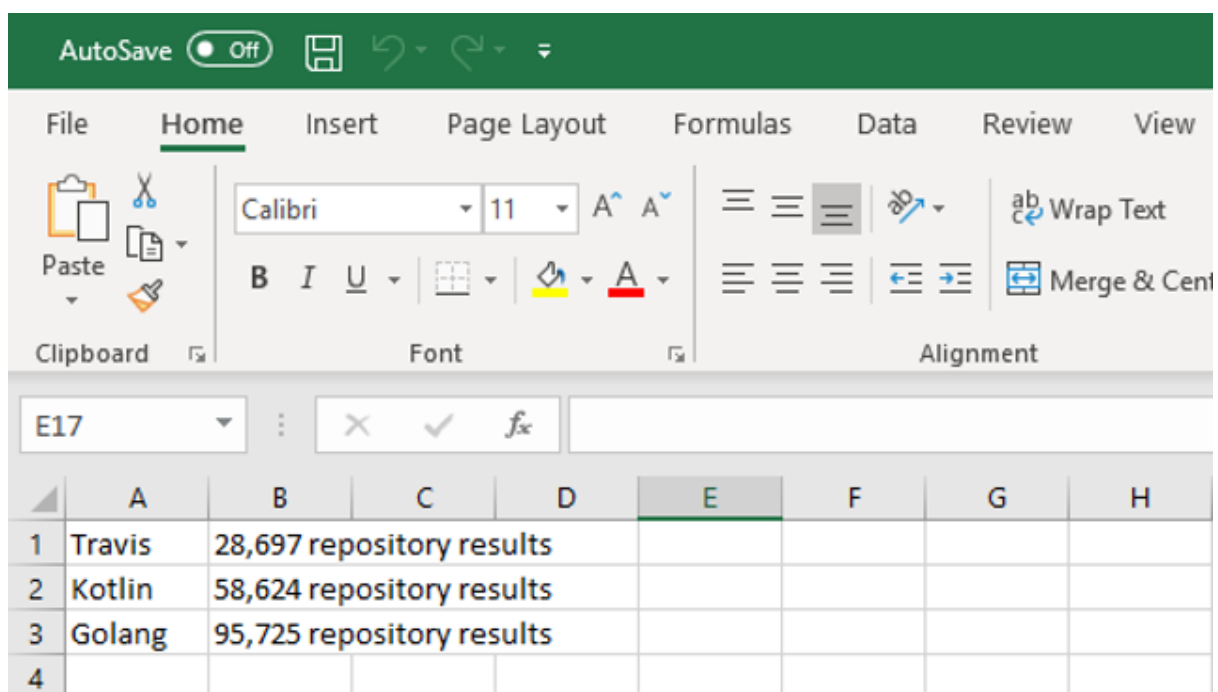


The 'Write Range' dialog box is shown with the following fields:

- File path: "C:\Users\Sahiti\Desktop\Technologies.xlsx"
- Sheet name: "Sheet2"
- Start cell: "A1"
- Table name: TechnologiesList

Step 8: Save and Execute the designed automation.

You will see the below output.



The screenshot shows the Excel interface with the 'Home' tab selected. The ribbon includes 'Clipboard', 'Font', and 'Alignment' groups. The formula bar shows 'E17'. The worksheet contains the following data:

	A	B	C	D	E	F	G	H
1	Travis	28,697 repository results						
2	Kotlin	58,624 repository results						
3	Golang	95,725 repository results						
4								

Now, that you know how you can automate tasks for web scraping.