

Practical No: 1

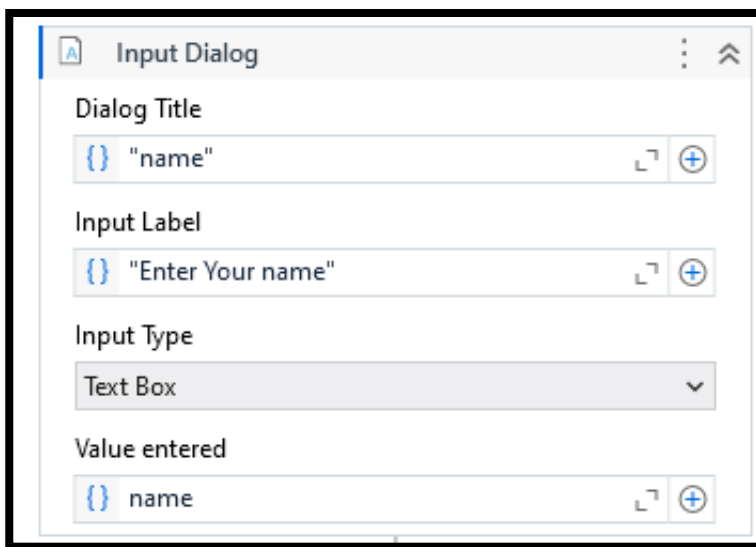
Sequence and Flowchart

AIM: A) Create a simple sequence- based project.

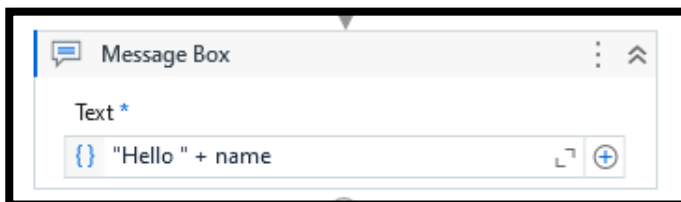
Steps with output

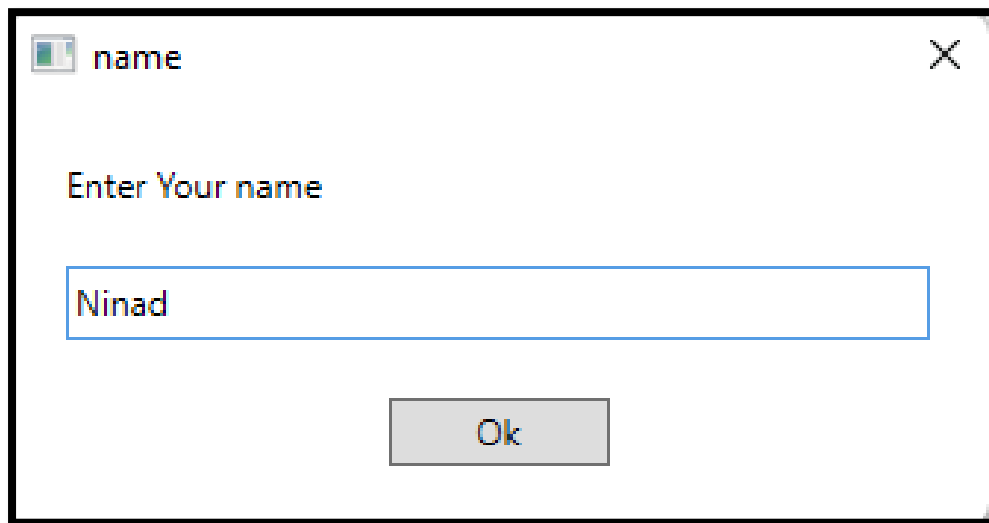
Step 1. Create a New Sequence:

Step 2. Add an Input Dialog Activity:

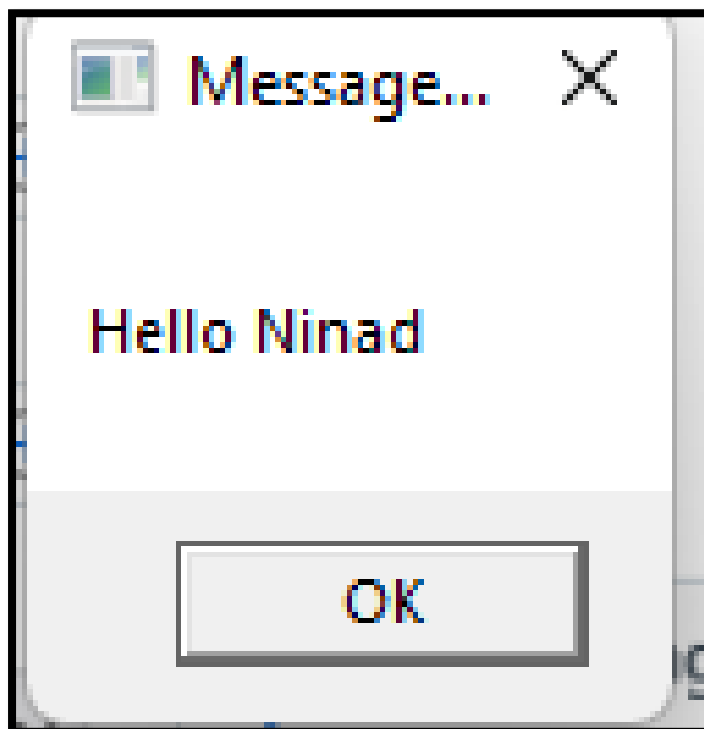


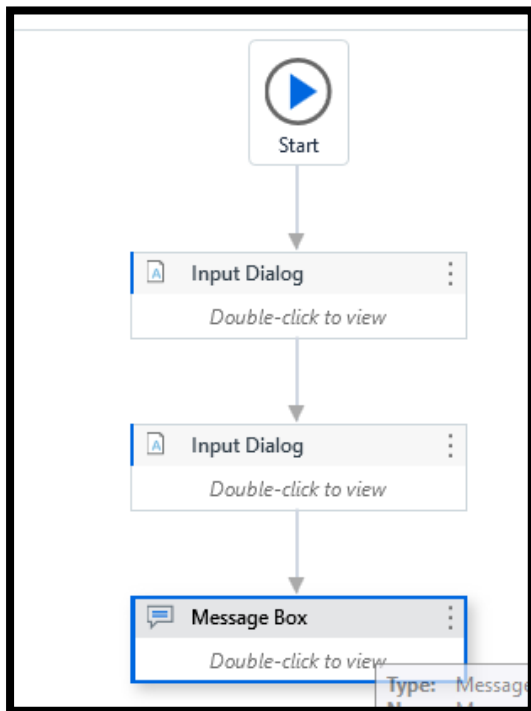
Step 3. Add a Message Box Activity:



OUTPUT:

A screenshot of a Windows-style dialog box titled "name". It features a standard window icon and a close button (X) in the top right corner. The main text inside the dialog is "Enter Your name". Below this text is a single-line text input field containing the name "Ninad". At the bottom center of the dialog is a button labeled "Ok".



AIM: B) Create a flowchart-based project.**Step 1.** Create a New Flowchart:**Step 2.** Add the Input Dialog:

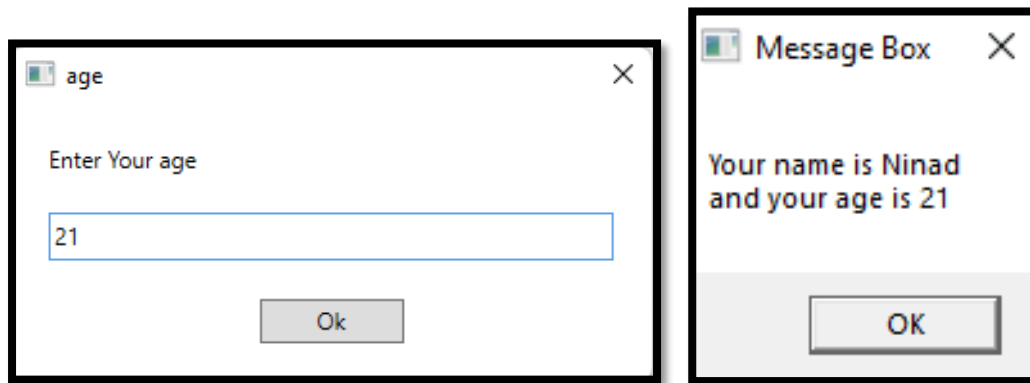
The screenshot shows the 'Input Dialog' configuration window. It contains the following fields and settings:

- Dialog Title:** A text box containing the value `"age"`.
- Input Label:** A text box containing the value `"Enter Your age"`.
- Input Type:** A dropdown menu currently set to `Text Box`.
- Value entered:** A text box containing the value `age`.

Step 3. Add a Message Box:

The screenshot shows the 'Message Box' configuration window. The title bar reads 'Message Box > Content (InArgument<Object>)'. Below the title bar, there is a section labeled 'Use Variables' with a dropdown arrow. The content area contains two lines of text:

```
1 "Your name is " + (name).ToString+Environment.NewLine +
2 "and your age is " + (age).ToString+Environment.NewLine
```

OUTPUT:**Learnings:****A. Create a Simple Sequence-based Project:**

Creating a new sequence, adding an Input Dialog, and a Message Box activity taught us the basics of building a linear workflow where user input is obtained and displayed in a MessageBox.

B. Create a Flowchart-based Project:

Building a new flowchart, incorporating an Input Dialog, and a Message Box activity helped us understand the flowchart structure, allowing for a more visual and branched project layout.

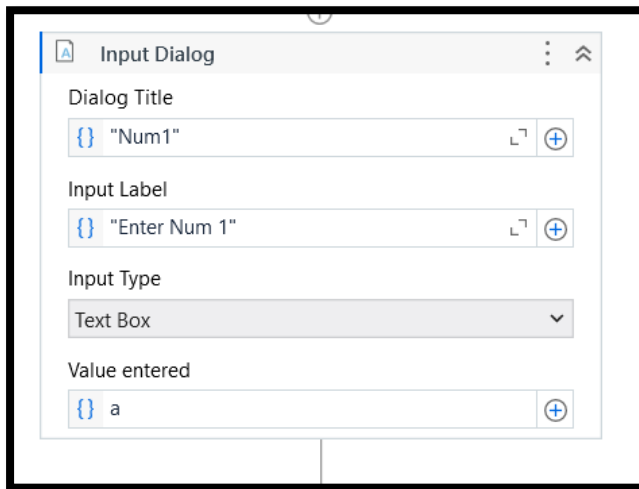
Practical No: 2

Calculator | Types of Variables

AIM: A) Automate UiPath Number Calculation (Subtraction, Multiplication, Division of numbers).

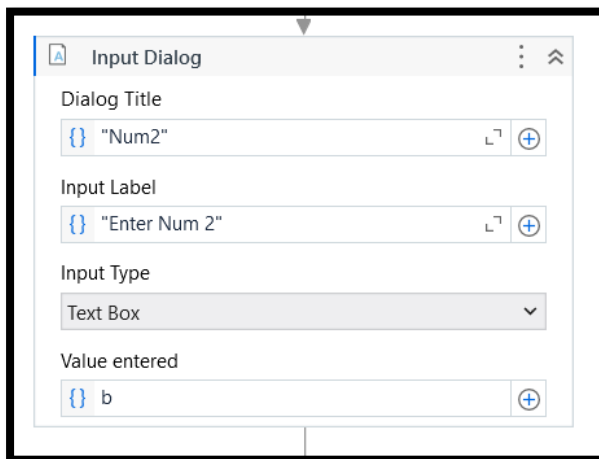
Step 1. Add Input Dialog for First Number:

- Drag and drop the "Input Dialog" activity into your sequence.
- Configure the input dialog to prompt the user for the first number.

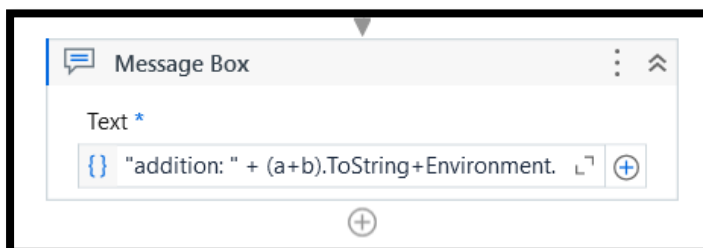


Step 2. Add Input Dialog for Second Number:

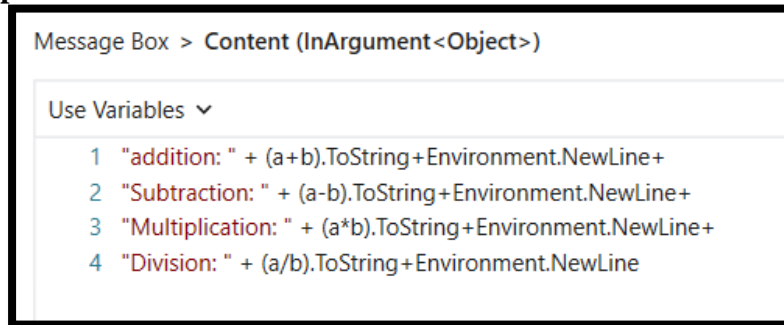
- Add another "Input Dialog" activity into your sequence.
- Configure this input dialog to prompt the user for the second number.



Step 3. Add a "Message Box" activity to your sequence.

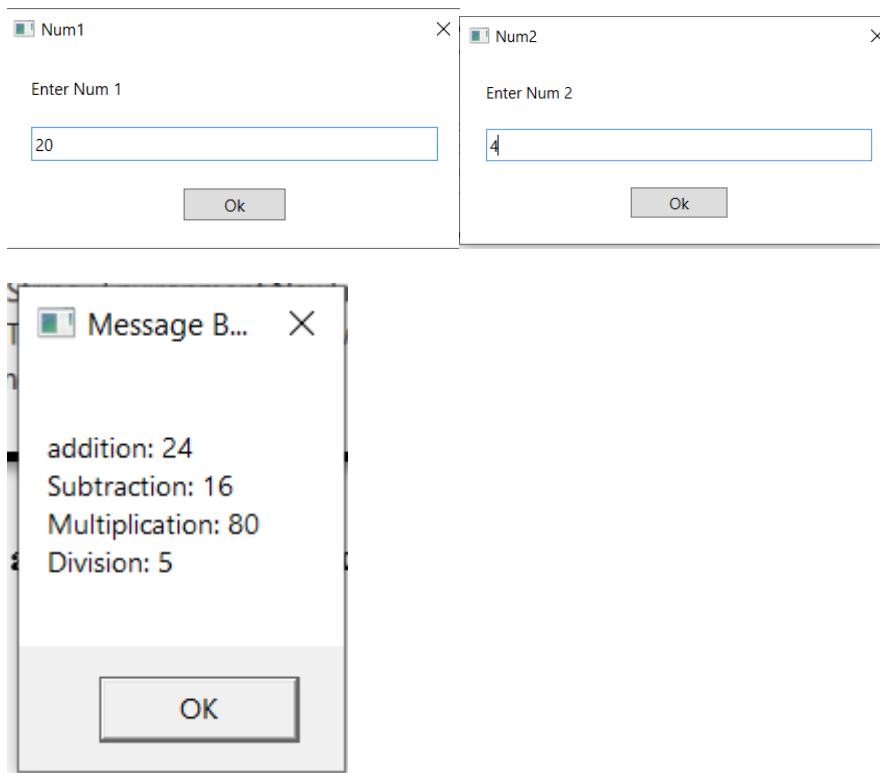


Step 4. Use expressions in the message box to display the results of arithmetic operations such as subtraction, multiplication, and division based on the user-provided numbers.



Step 5. Save your workflow and run the sequence.

OUTPUT:



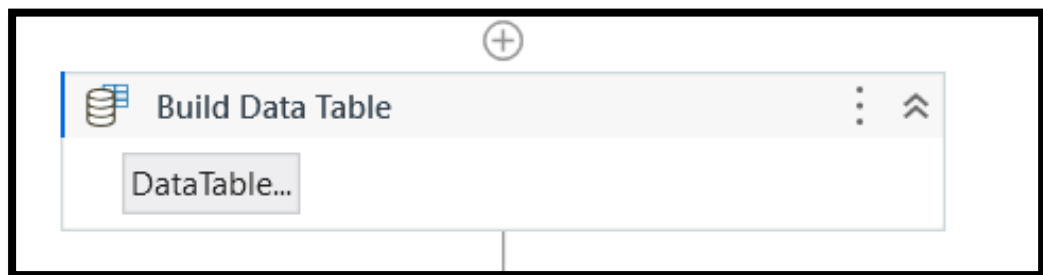
Learnings:

Using Input Dialogs, we took two inputs from the user for the first and second numbers. Subsequently, we performed addition, subtraction, multiplication, and division on those inputs and displayed the results individually in a Message Box. Additionally, we discovered how to incorporate a new line in a Message Box using `Environment.NewLine`.

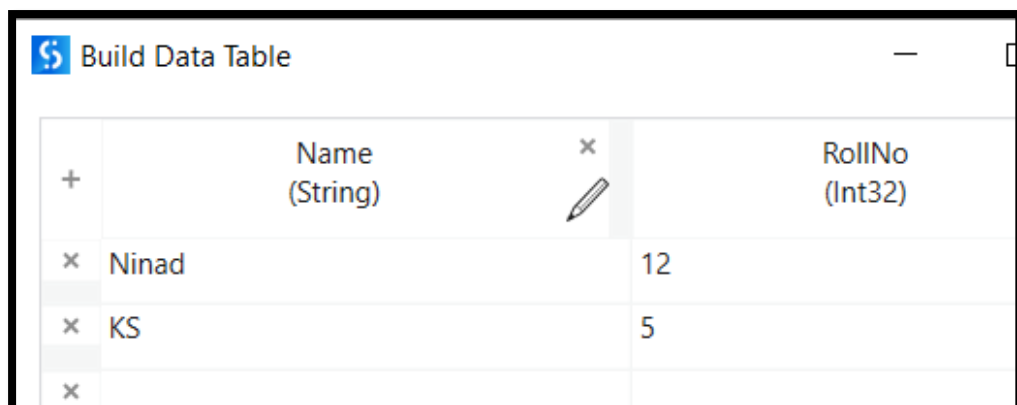
AIM: b) Create an automation UiPath project using different types of variables (number, datetime, Boolean, generic, array, data table)

Step 1. Build Data Table Activity:

- a. Use the "Build Data Table" activity to create a DataTable.

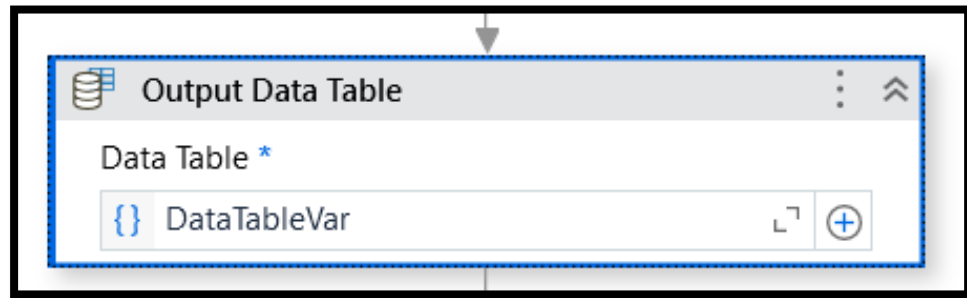


- b. Add columns and set their data types (e.g., "RollNo" as Int32, "Name" as String).



Step 2. Output Data Table Activity:

- a. Use the "Output Data Table" activity to convert the DataTable to a string.



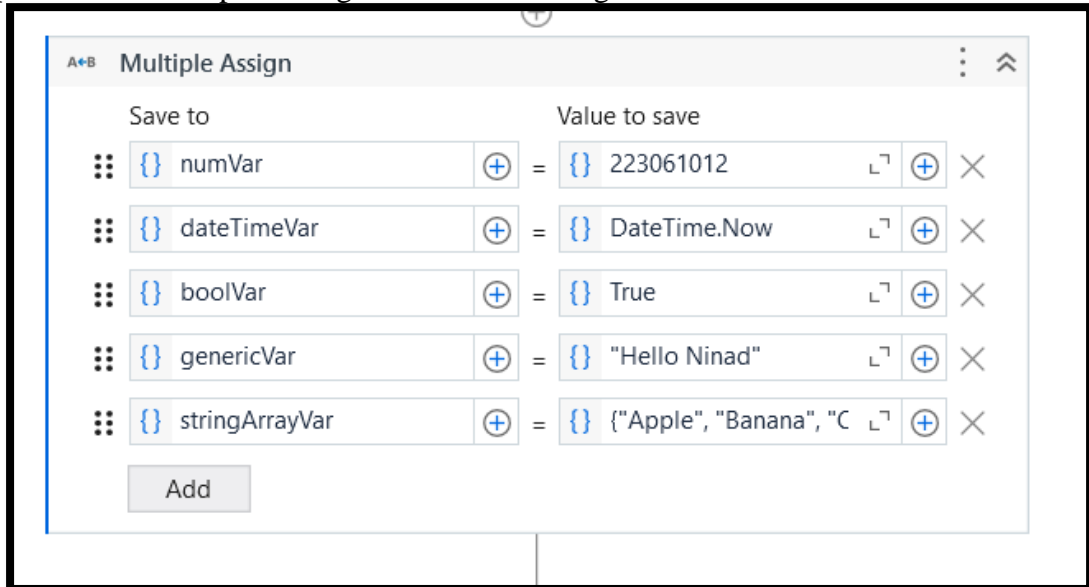
- b. Set the DataTableVar as the DataTable and create a new variable (e.g., TableVar) for the output.



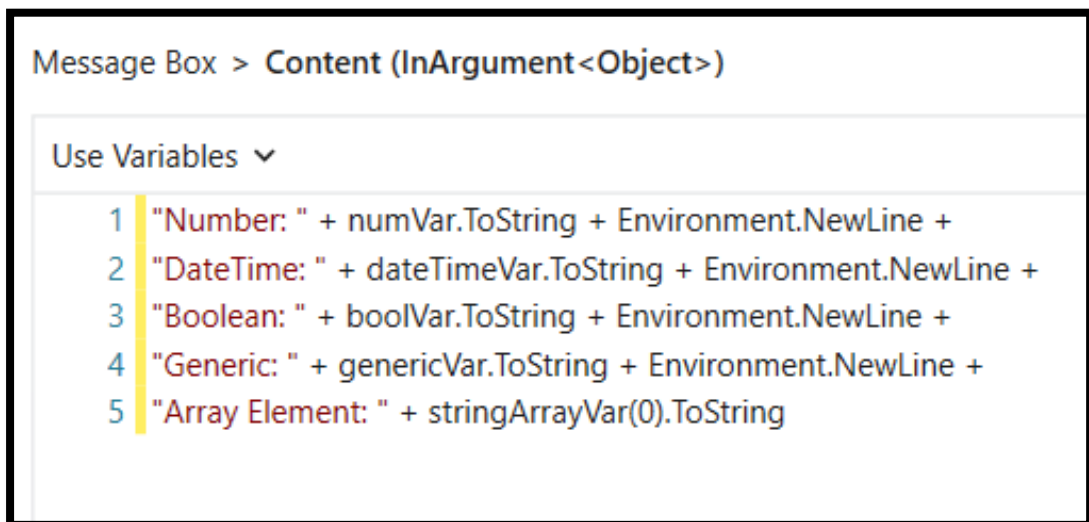
Step 3. From the Variables tab, create variables for Number (numVar), DateTime (dateTimeVar), Boolean (boolVar), Generic (genericVar), and Array (stringArrayVar).

Name	Variable type	Scope	De
numVar	Int32	TypesOfVariables	En
dateTimeVar	DateTime	TypesOfVariables	En
boolVar	Boolean	TypesOfVariables	En
genericVar	String	TypesOfVariables	En
stringArrayVar	String[]	TypesOfVariables	En
DataTableVar	DataTable	TypesOfVariables	En
TableVar	String	TypesOfVariables	En

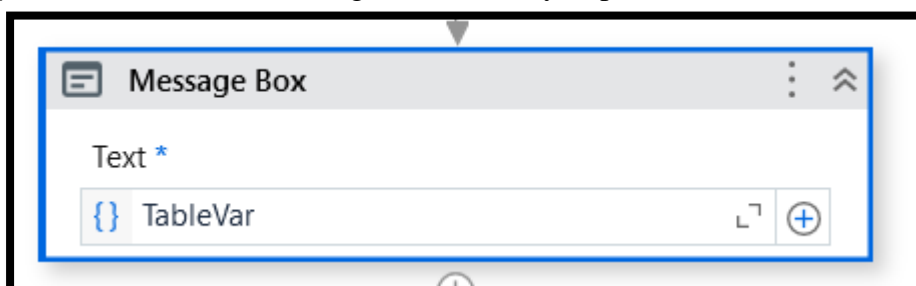
Step 4. Use multiple "Assign" activities to assign values to the variables.



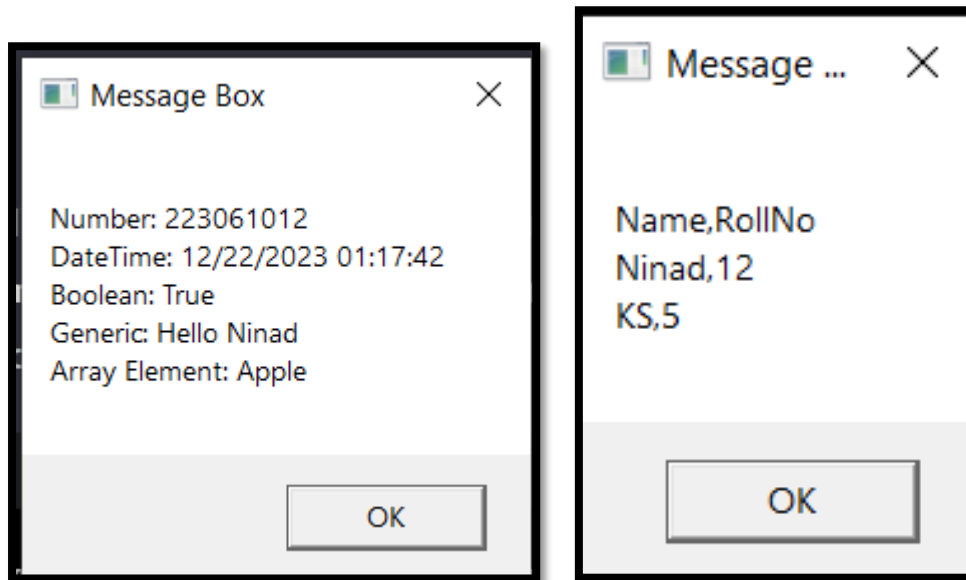
Step 5. Use the "Message Box" activity to print all variables.



Step 6. Use another "Message Box" activity to print datatable.



OUTPUT:



Learnings:

Understanding the usage of the "Build Data Table" activity to create and configure a DataTable, defining columns with specific data types.

Practical application of different variable types (Number, DateTime, Boolean, Generic, Array) and DataTable, utilizing "Assign" activities, and displaying their values using the "Message Box" activity in UiPath.

Practical No: 3

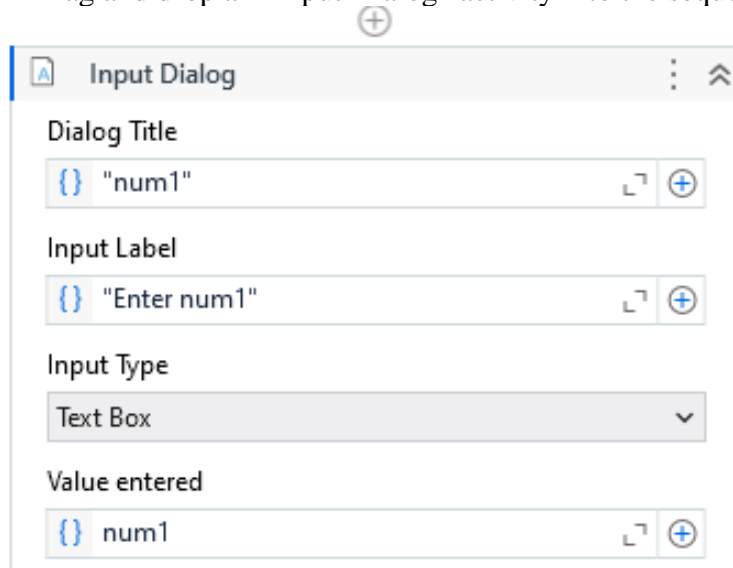
Decision Making and Looping statements.

AIM: A) Create an automation UiPath Project using Decision Making statements.

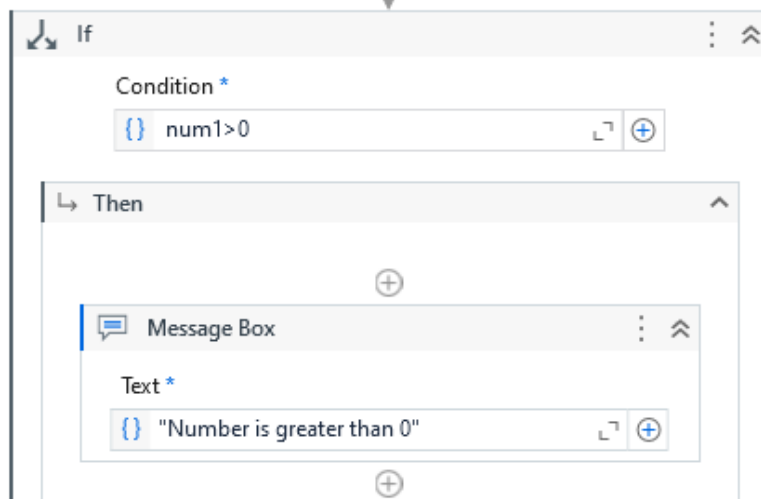
Steps with output

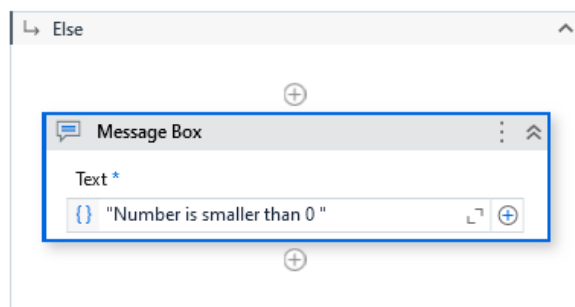
A. If Then

Step 1. Drag and drop an "Input Dialog" activity into the sequence.

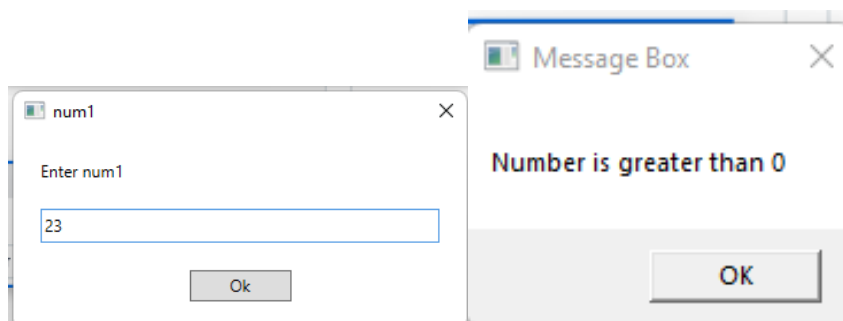
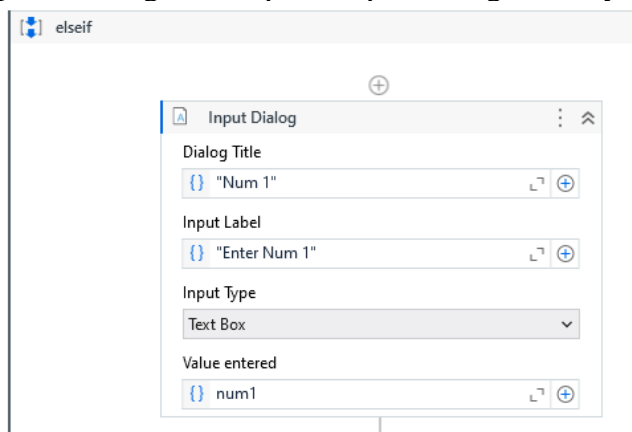


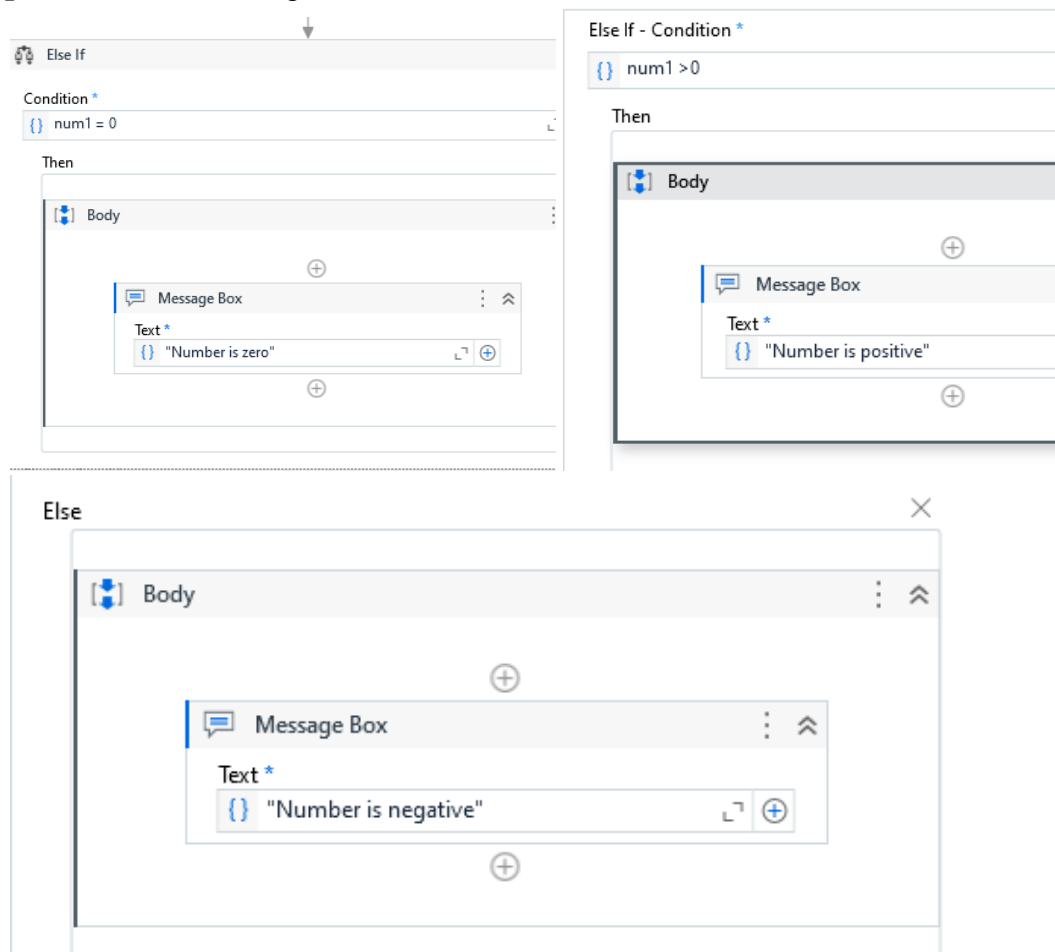
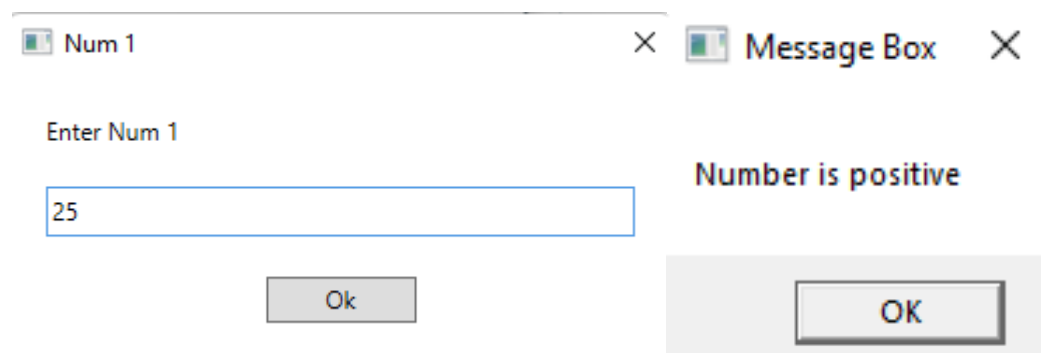
Step 2. Drag and drop an "If" activity below the "Input Dialog" activity.
Step 3. Add a "Message Box" Activity (Then Branch):



Step 4. Add a "Message Box" Activity (Else Branch)

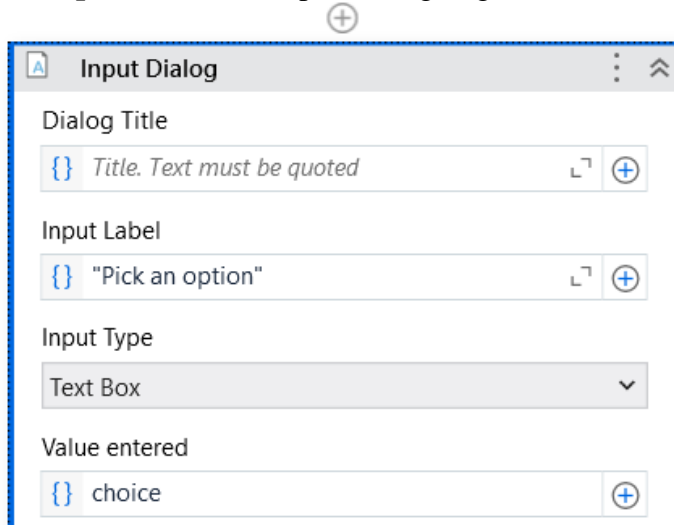
Output:

**B. Else IF****Step 1.** Drag and drop an "Input Dialog" activity into the sequence.

Step 2. Add a "Message Box"**OUTPUT:**

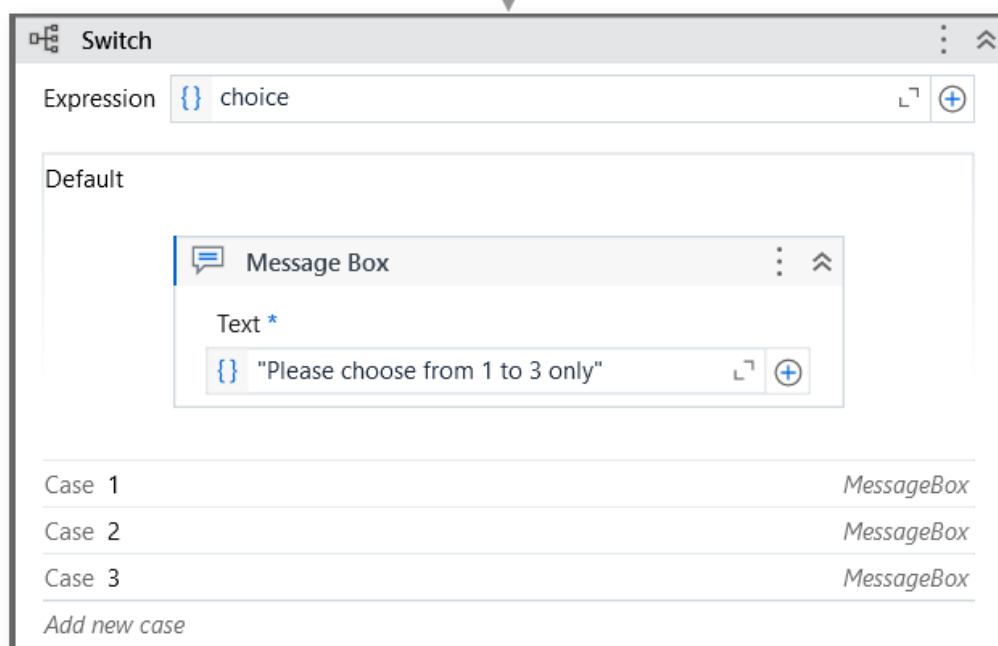
C. Switch

Step 1. Use an Input Dialog to get the user's choice.



The screenshot shows the 'Input Dialog' configuration window. It has four sections: 'Dialog Title' with a text box containing '{ } Title. Text must be quoted'; 'Input Label' with a text box containing '{ } "Pick an option"'; 'Input Type' with a dropdown menu set to 'Text Box'; and 'Value entered' with a text box containing '{ } choice'. Each text box has a small icon on the left and a '+' button on the right.

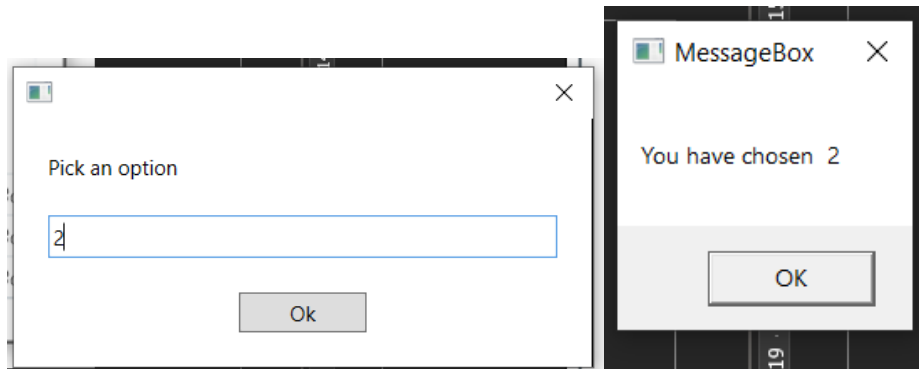
Step 2. Set up a Switch Activity. Input the variable for expression. Add cases with associated message boxes for each choice. Don't forget to include a Default case for any choices not covered.



The screenshot shows the 'Switch' configuration window. At the top, the 'Expression' field contains '{ } choice'. Below this is the 'Default' section, which contains a 'Message Box' activity with the text 'Please choose from 1 to 3 only'. At the bottom, there is a table with three cases, each associated with a 'MessageBox' activity.

Case	Associated Activity
Case 1	MessageBox
Case 2	MessageBox
Case 3	MessageBox

Below the table is a link that says 'Add new case'.

OUTPUT:**Learnings****A. If Then:**

By using an Input Dialog followed by an If activity, we learned to create a decision-making statement. If a certain condition (e.g., number > 0) is true, a MessageBox in the "Then" branch is executed; otherwise, the "Else" branch with another MessageBox is triggered.

B. Else If:

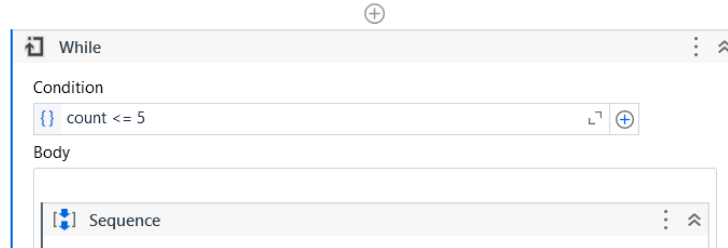
Employing an Input Dialog and a Message Box, we grasped the concept of an "Else If" statement. Depending on the condition, a specific message is displayed in the MessageBox, demonstrating an alternative path in the execution.

C. Switch:

Through an Input Dialog capturing user choice and a Switch activity, we learned to handle multiple cases efficiently. The Switch statement directs the flow based on the user's choice, with each case associated with a specific action in a MessageBox.

AIM: B) Create an automation UiPath Project using looping statements**Steps with output****A. While Loop**

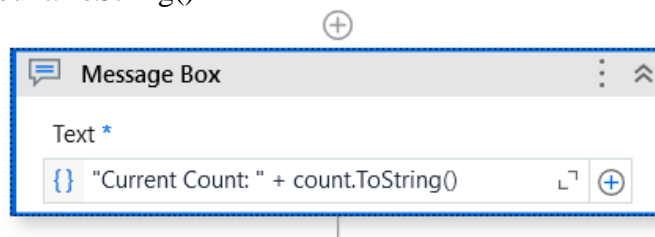
Step 1. Add while activity and set condition to -> `count <= 5`



Step 2. Create variable count and set variable type to int32

Name	Variable type	
count	Int32	R
Create Variable		

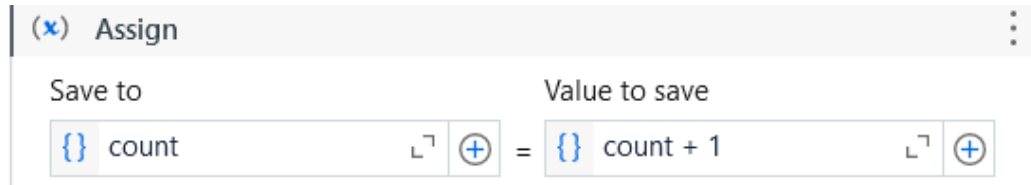
Step 3. Inside while body create message box and write message -> "Current Count: " + `count.ToString()`

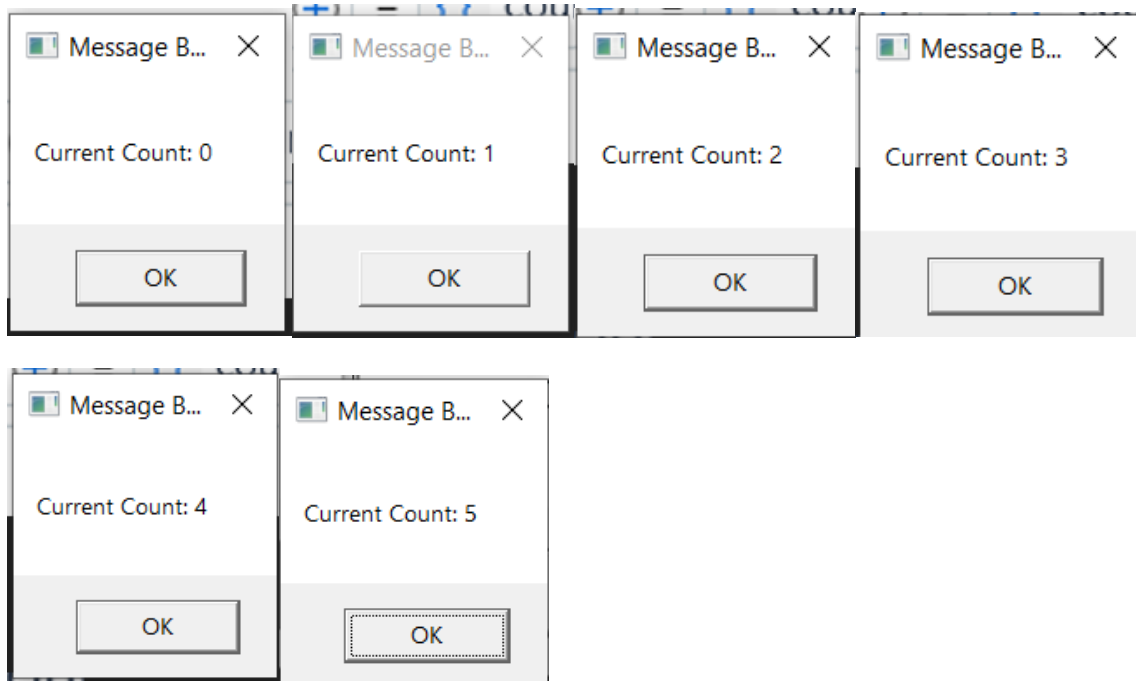


Step 4. Drag and drop assign activity inside while

Set the "To" field to Count.

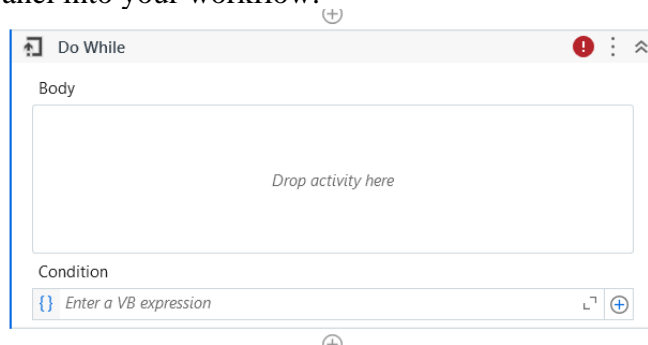
Set the "Value" field to Count + 1.

**OUTPUT**



B. Do While

Step 1. Do While Loop: Drag and drop a "Do While" activity from the "Activities" panel into your workflow.



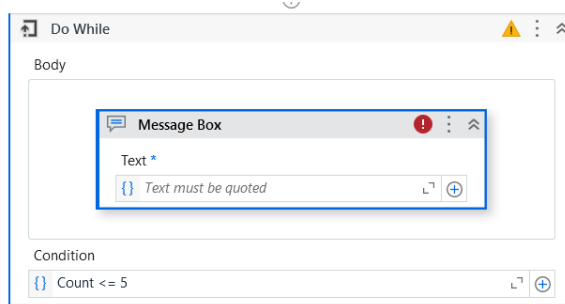
Step 2. Create a variable named 'Count' of type "Int32" to keep track of the current count.

Name	Variable type	Scope
Count	Int32	Do While

Create Variable

Step 3. we will use the condition: $\text{Count} \leq 5$. This means the loop will continue as long as the Count variable is less than or equal to 5.

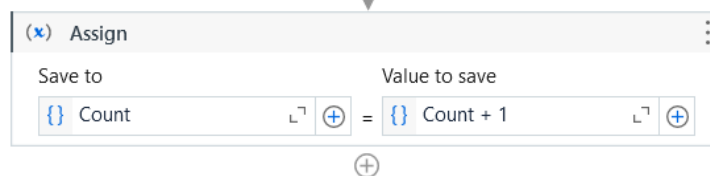
Step 4. Inside the "Do While" activity, drag and drop a "Message Box" activity.



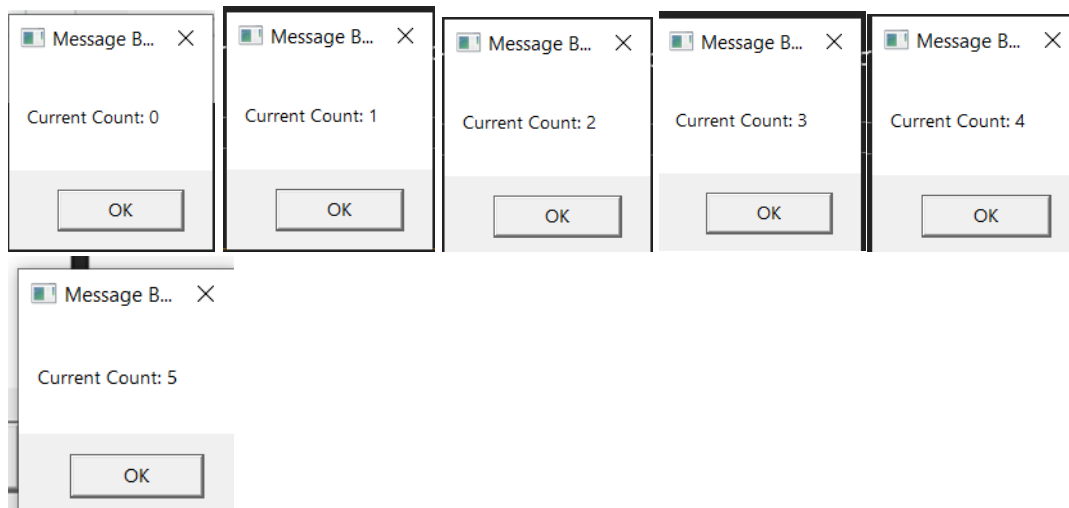
Step 5. Configure the Message Box Activity:



Step 6. Add an "Assign" activity inside the "Do While" loop:
Set the "To" field to Count.
Set the "Value" field to Count + 1.



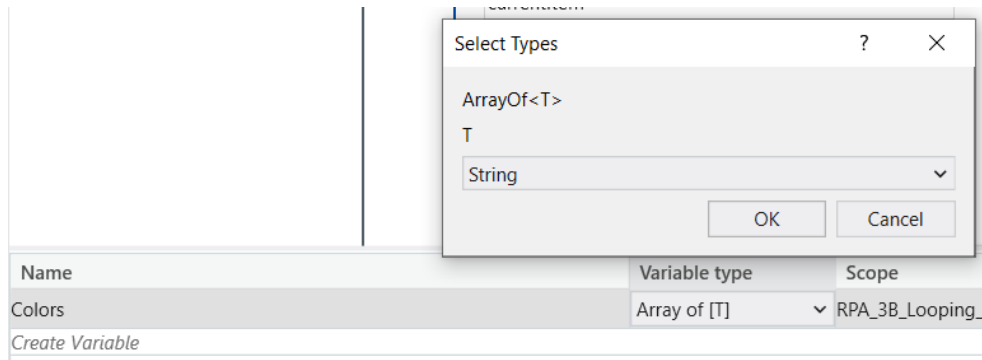
OUTPUT



C. FOR EACH

we'll create a "For Each" loop to iterate through a list of names and display each name using a message box.

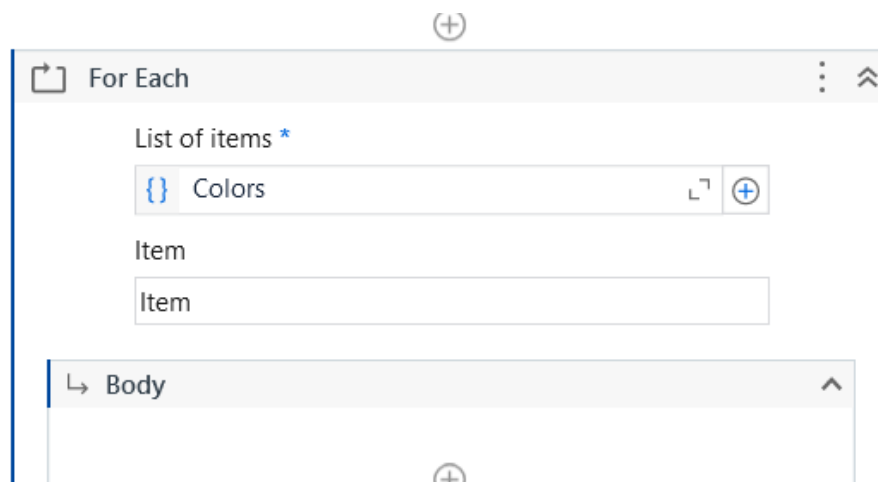
Step 1. Add a List of Names:



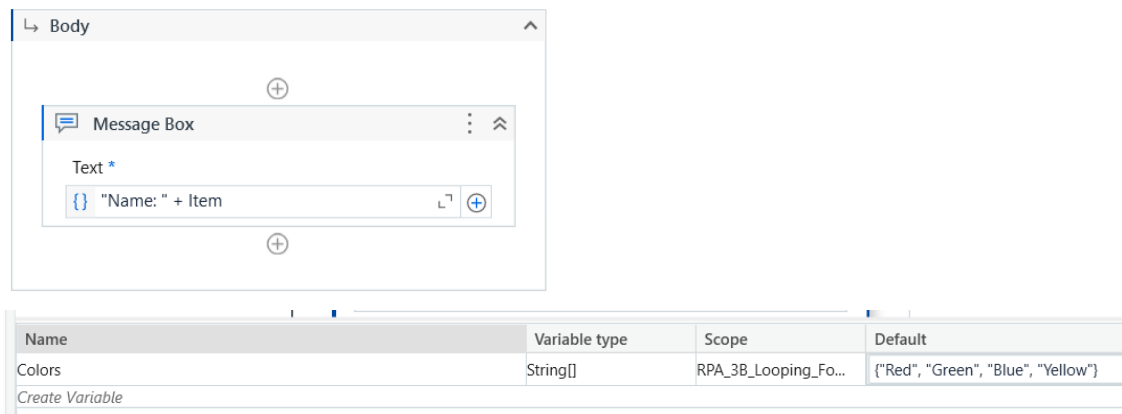
Step 2. In the "Default" value field of the variable, enter the list of colors enclosed in curly braces { } and separated by commas.



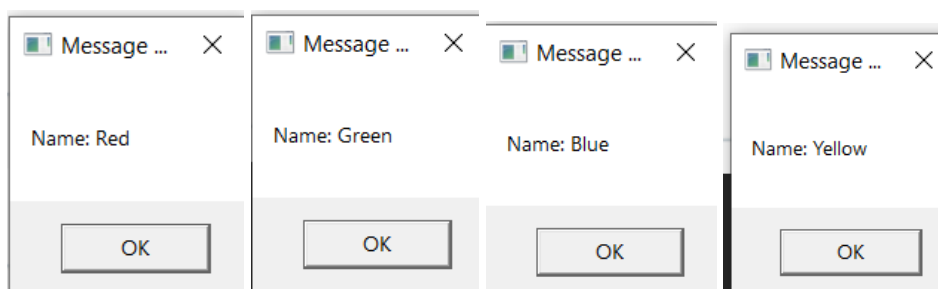
Step 3. Drag and drop a "For Each" activity from the "Activities" panel into your workflow.



Step 4. Inside the "For Each" activity, drag and drop a "Message Box" activity.



OUTPUT:



Learnings

A. While Loop:

The While Loop iterates as long as the count is less than or equal to 5, displaying the current count in a Message Box and incrementing the count in each iteration.

B. Do While Loop:

The Do While Loop continues executing as long as the count is less than or equal to 5, showing the current count in a Message Box and incrementing the count within the loop.

C. For Each Loop:

The For Each Loop iterates through a list of names, displaying each name in a Message Box, showcasing the functionality of iterating through collections using a loop.

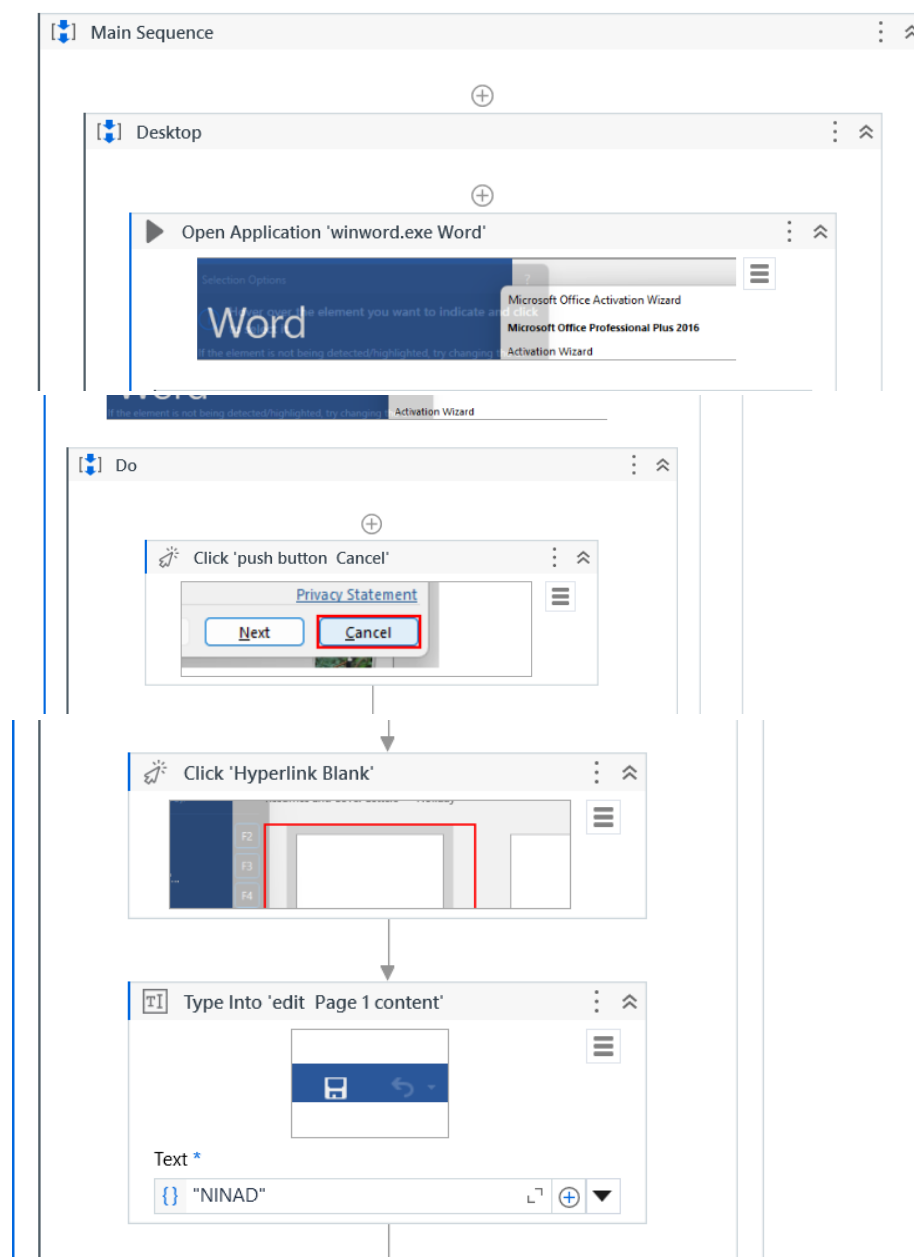
Practical No: 4

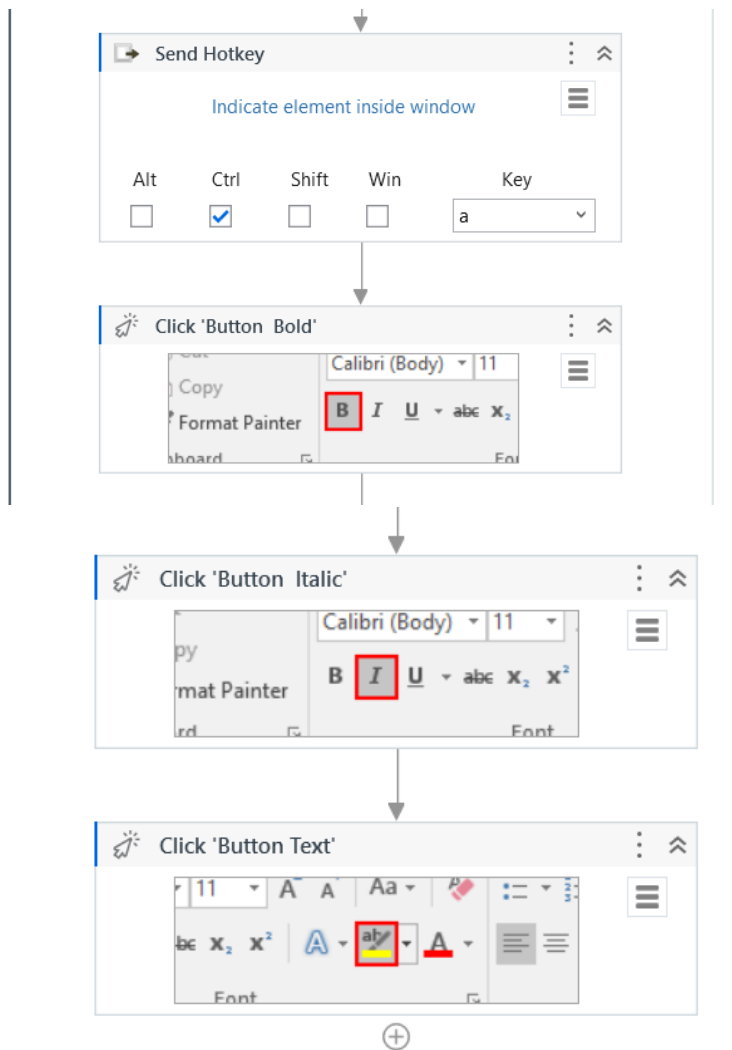
Desktop & Web Recording

AIM: A) Automate any process using desktop recording.

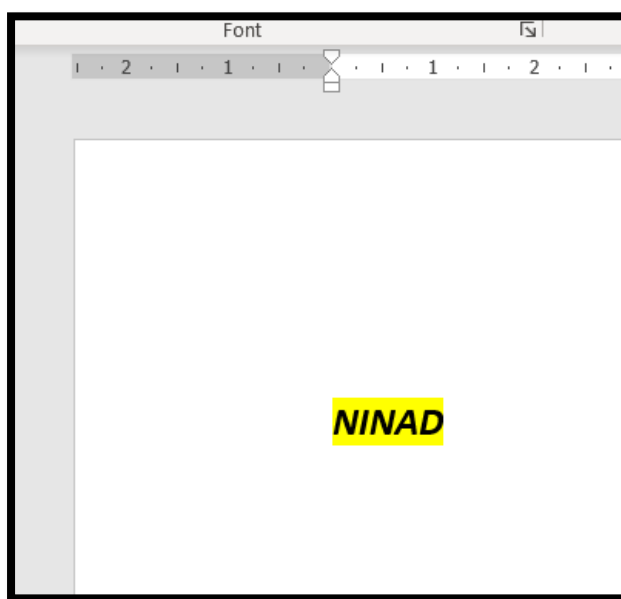
Steps with output

- Step 1.** Click on recording -> Desktop recording
- Step 2.** Start app -> choose word app
- Step 3.** Click on record



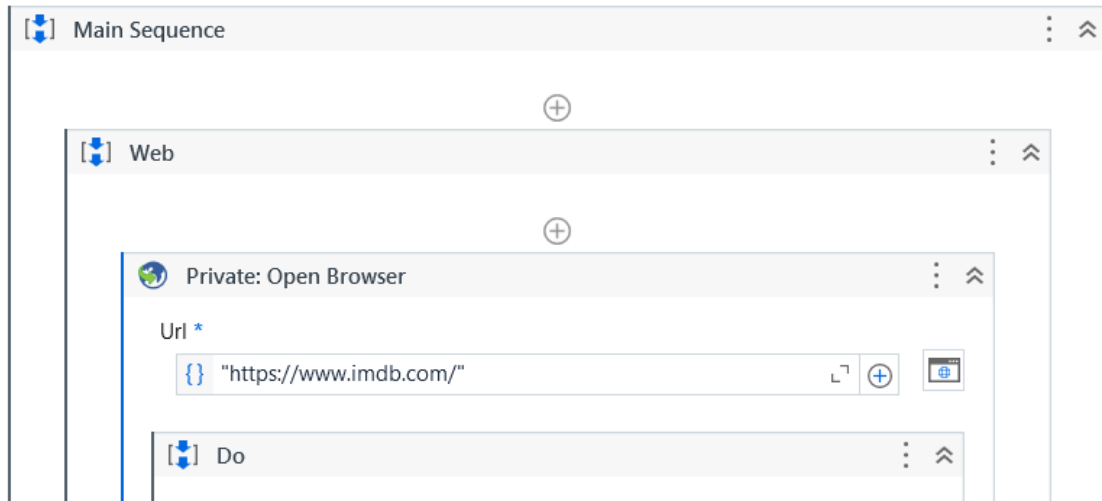


OUTPUT

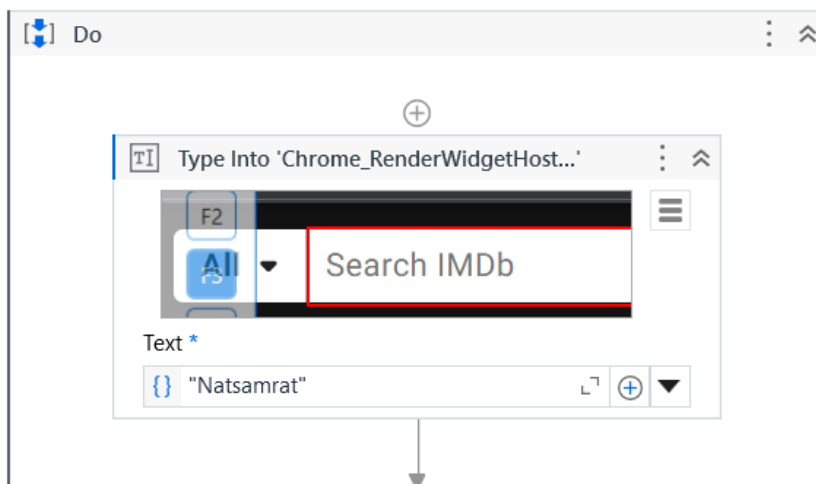


AIM: B) Automate any process using web recording.**Steps with output**

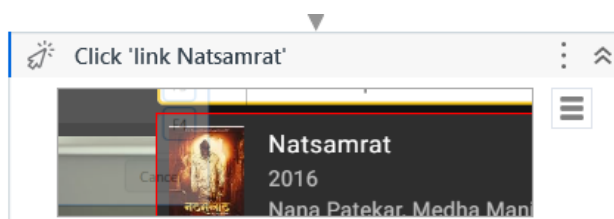
Step 1: Add the Open Browser activity to the sequence and input the URL [www.imdb.com]. Additionally, maximize or snap the window.



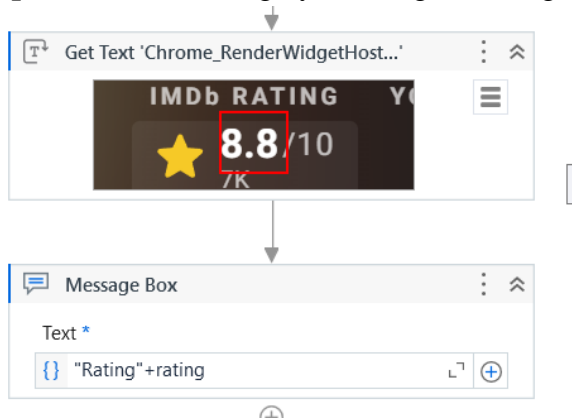
Step 2: Utilize the Click activity to select the search bar. Follow it up with the Type Into activity to input the movie name.



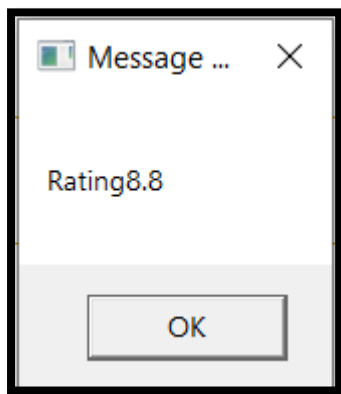
Step 3: Repeat the Click activity to select the movie panel. Proceed to use the Get Text activity to fetch the rating. [Generate a string variable and input it into the Get Text properties.]



Step 4: Show the rating by utilizing a Message Box.



OUTPUT:



Learnings:

A. Automate any process using Desktop Recording:

By initiating desktop recording and automating steps like starting the Word app, we learned the basics of capturing and automating desktop processes.

B. Automate any process using Web Recording:

Through web recording, we learned how to automate web-based processes, such as opening a browser, interacting with elements like search bars and movie panels, extracting information using the Get Text activity, and displaying the result in a Message Box.

Practical No: 5

ARRAY

AIM: A) Consider an array of names. We have to find out how many of them start with the letter "a". Create an automation where the number of names starting with "a" is counted and the result is displayed.

Steps with output

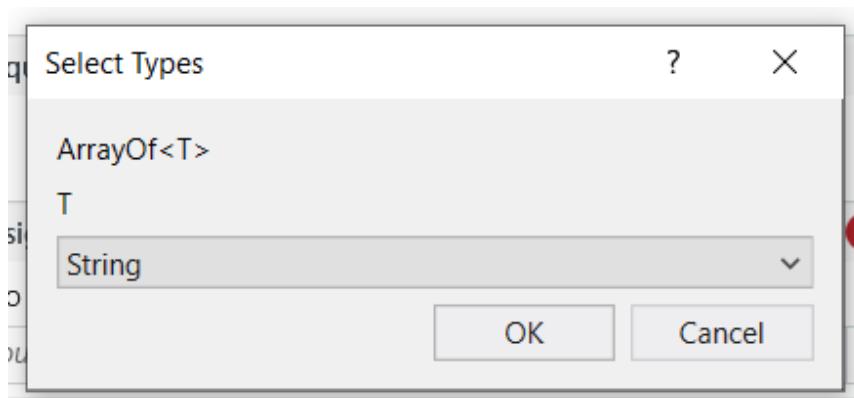
Step 1. Create the Counter Variable:

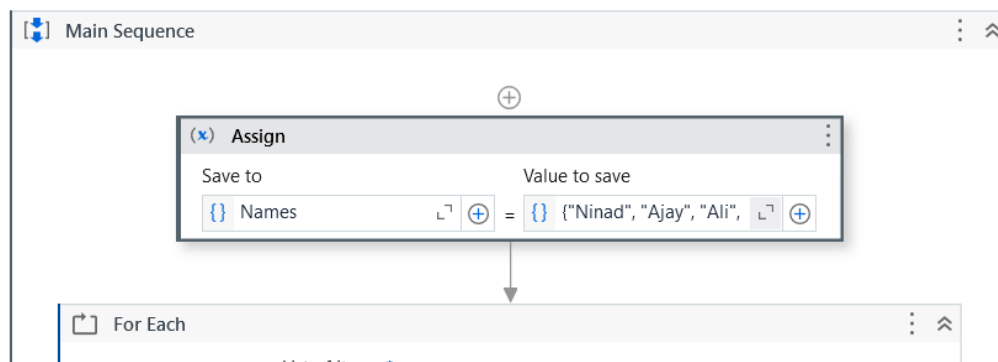
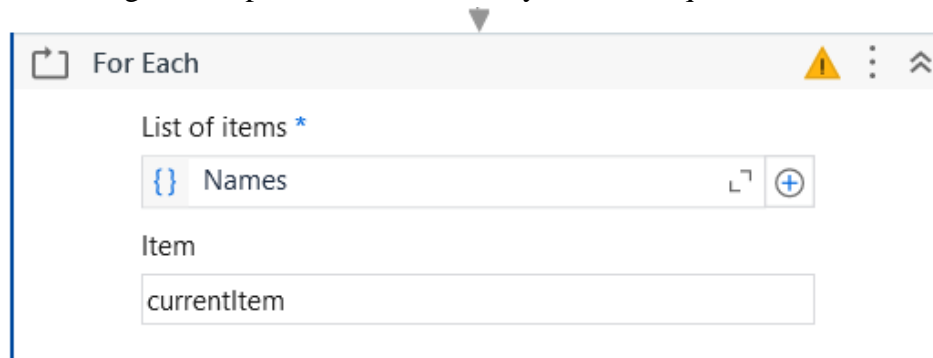
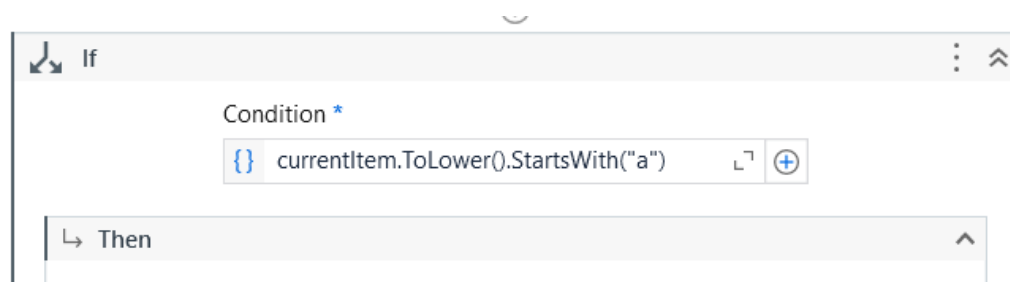
- Name the variable as CountA.
- Set the variable type to "Int32"

Name	Variable type	Scope	
Names	String[]	Main Sequence	E
CountA	Int32	Main Sequence	E
Create Variable			

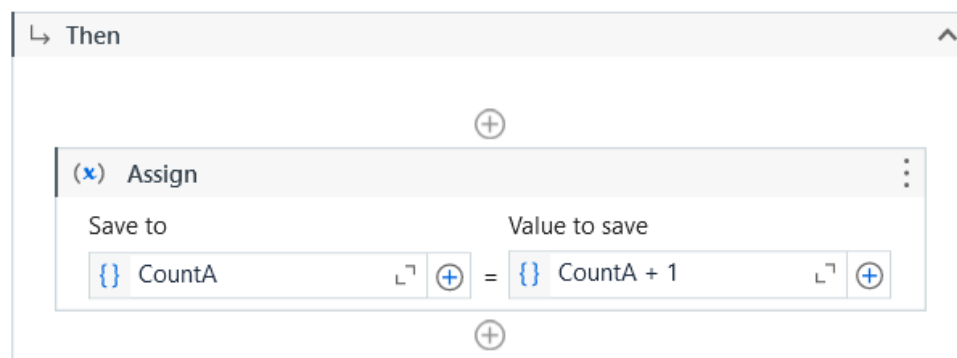
Step 2. Create a Name Variable:

- Name the variable (e.g., Names).
- Set the variable type to "Array of [your chosen data type]" -> "Array of String."

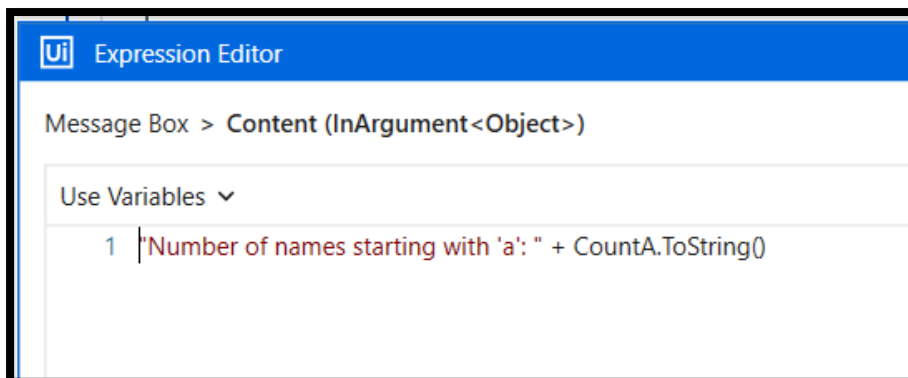
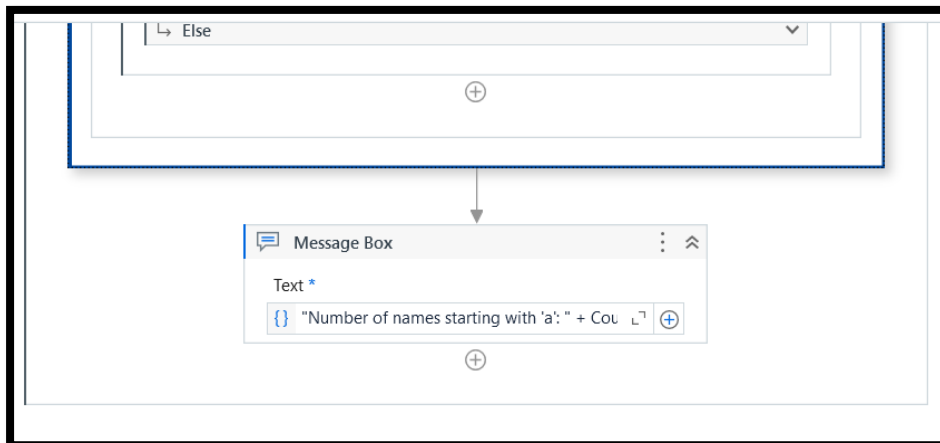


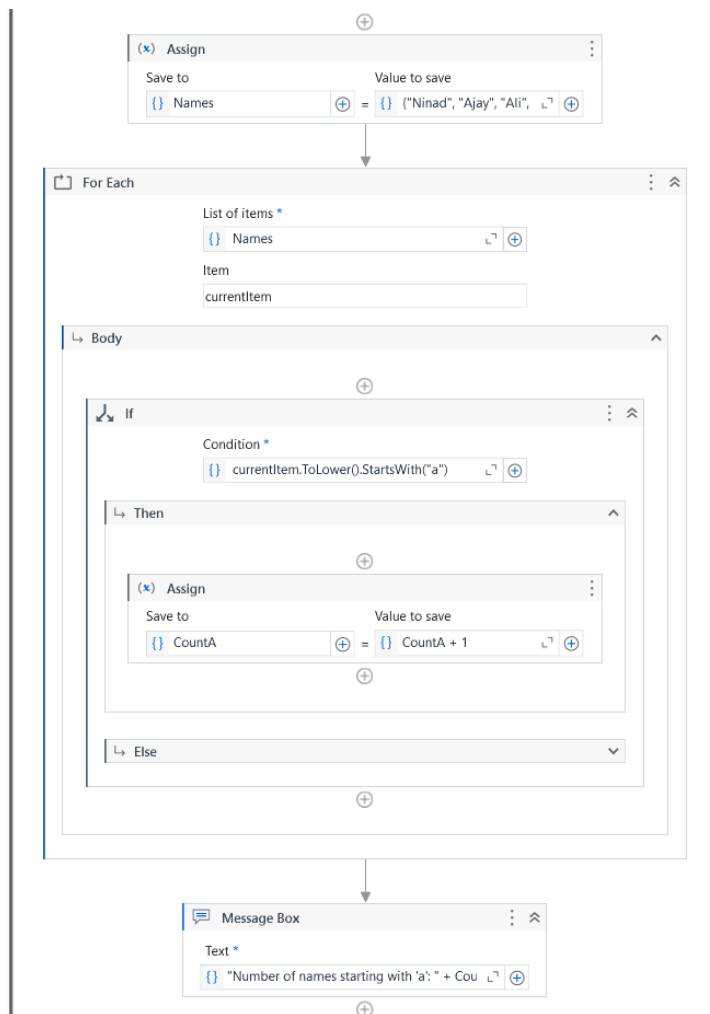
Step 3. Add an Assign Activity: (enter any names with { })**Step 4.** Drag and drop a "For Each" activity into the sequence.**Step 5.** Add an If Activity (Inside For Each):
currentItem.ToLower().StartsWith("a")

Step 6. Inside the "Then" section of the "If" activity, add an "Assign" activity.

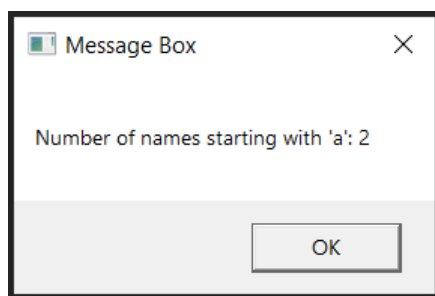


Step 7. Add a Message Box Activity (After For Each):





OUTPUT:



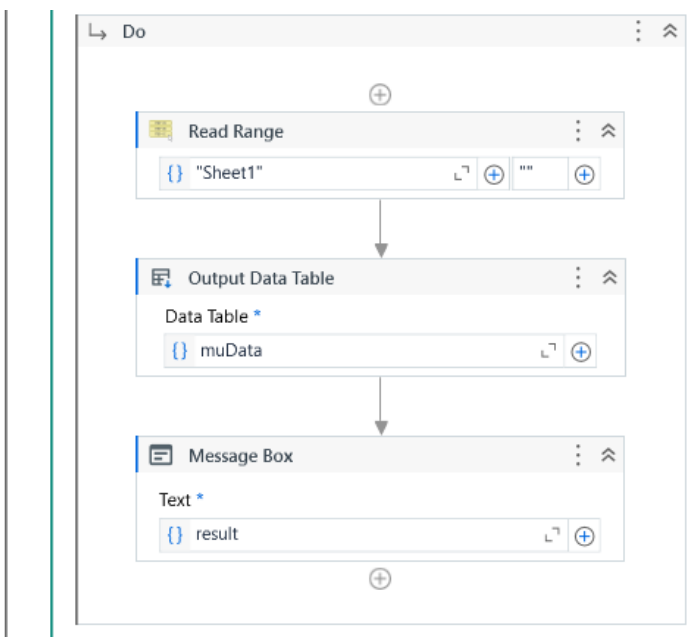
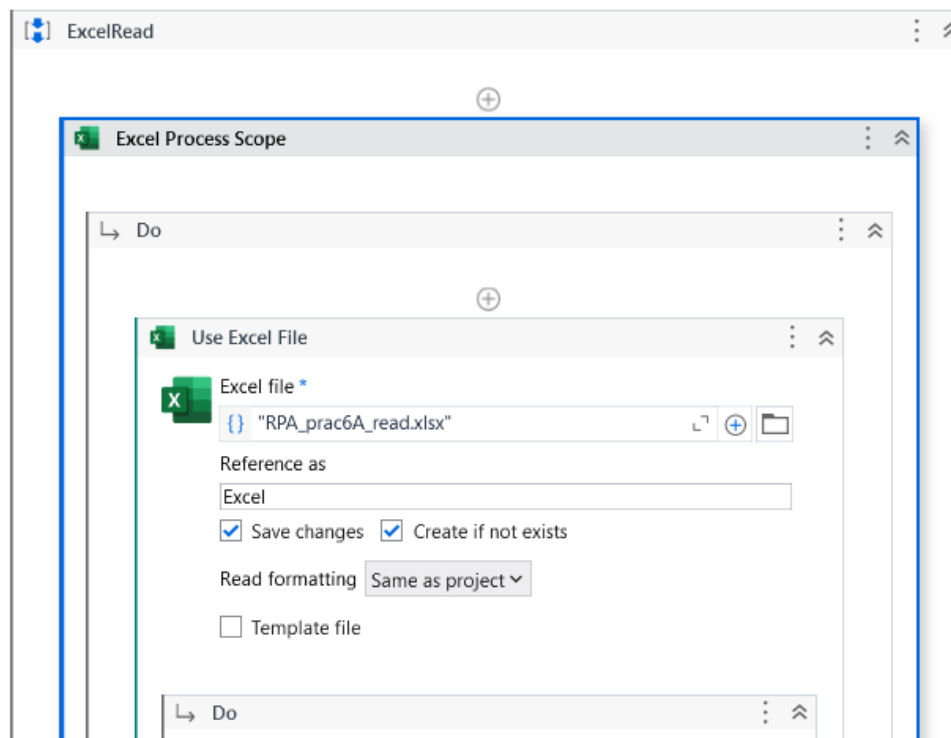
Learnings:

A. Count Names Starting with "a":

We learned to count the number of names starting with "a" in an array by creating and using a counter variable, employing a For Each loop, and implementing an If activity to check and increment the count accordingly. The result was then displayed in a Message Box.

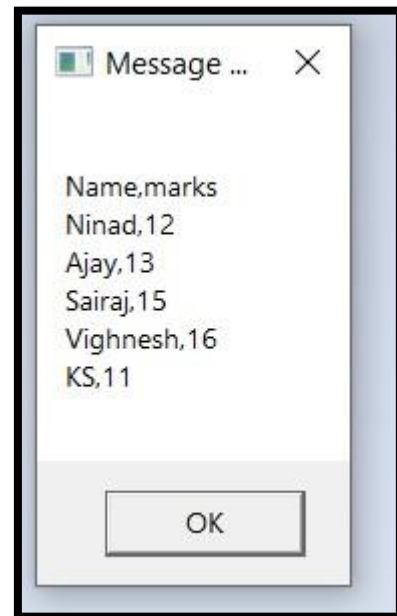
Practical No: 6**Excel Automation**

AIM: A) Create an application automating the read, write and append operation on excel file

ENABLE CLASSIC OPTION**READ**

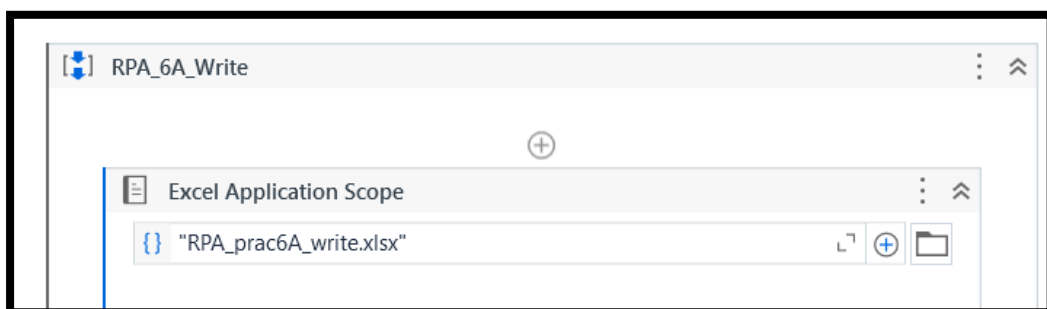
OUTPUT

	A	B	C
1	Name	marks	
2	Ninad	12	
3	Ajay	13	
4	Sairaj	15	
5	Vighnesh	16	
6	KS	11	
7			
8			
9			

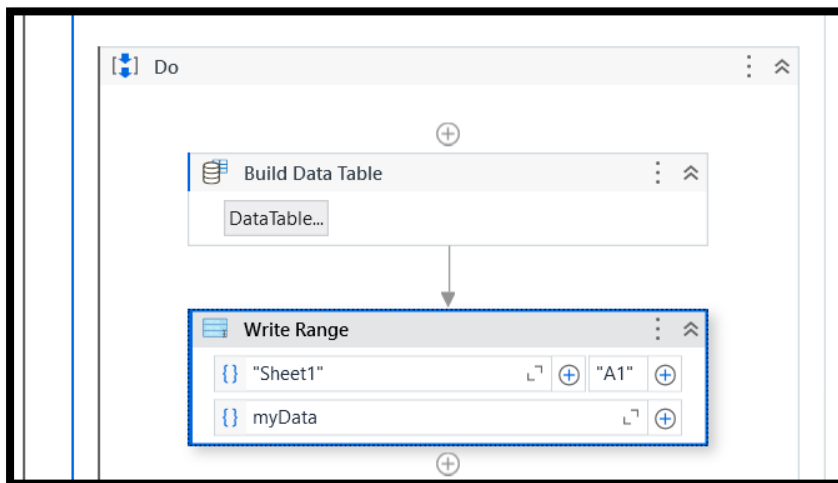


WRITE

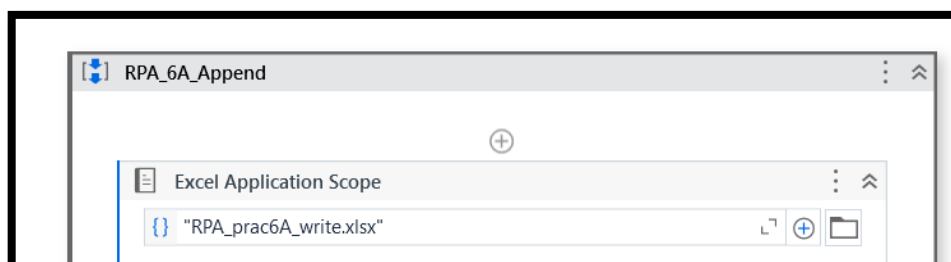
- Step 1.** Open main workflow
- Step 2.** Activities -> Excel application scope
- Step 3.** Add path of excel file

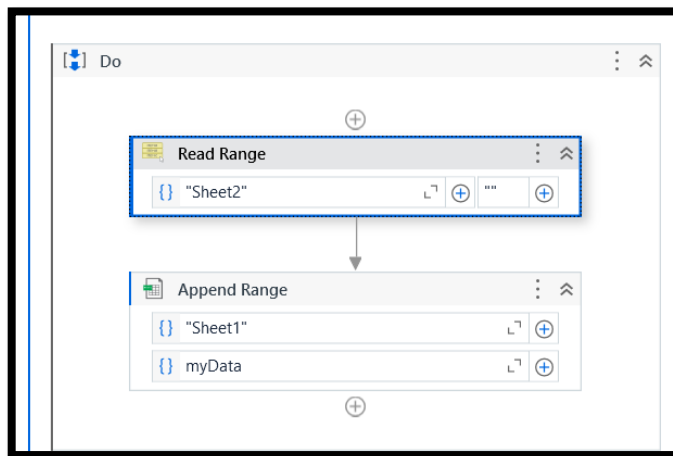


- Step 4.** In Do -> Add build data table -> add table data -> create variable(myData)

Step 5. Add write range**OUTPUT**

F19		
	A	B
1	Name	rollno
2	Ninad	46
3	KS	5
4	Vighnesh	50
5		
6		

Append**Step 1.** Open main workflow**Step 2.** Activities -> Excel application scope**Step 3.** Add path of excel file**Step 4.** In Do -> Add Read Range -> Sheet2 -> create variable(myData)

Step 5. Add Append range**OUTPUT**

	A	B	C
1	Name	rollno	
2	Ninad	46	
3	KS	5	
4	Vighnesh	50	
5	ABCD	65	
6			

AIM: B) Automate the process to extract data from an excel file into a data table and vice versa.

I. Excel to Datatable

Step 1. Prepare an Excel file with data.

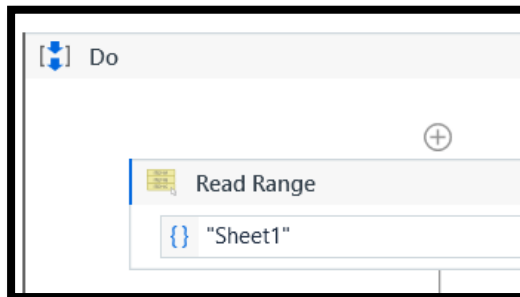
	A	B	C
1	Name	RollNO	
2	Ninad	12	
3	Sairaj	111	
4	KS	5	
5	VIGP	50	
6			

Step 2. Use the "Excel Application Scope" activity to specify the Excel file location.



Step 3. Read Range:

- a. Add the "Read Range" activity inside the Excel Application Scope.

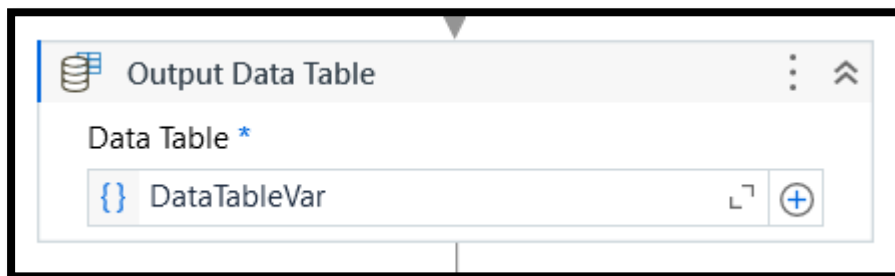


- b. Create a variable (e.g., DataTableVar) to store the output DataTable.



Step 4. Output DataTable:

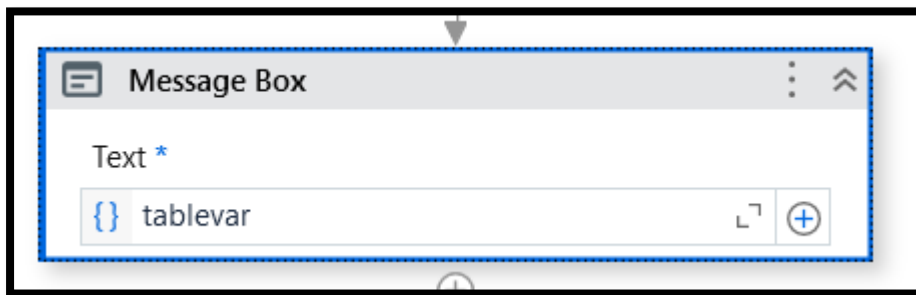
- a. Use the "Output Data Table" activity.
- b. Set the DataTableVar as the DataTable.



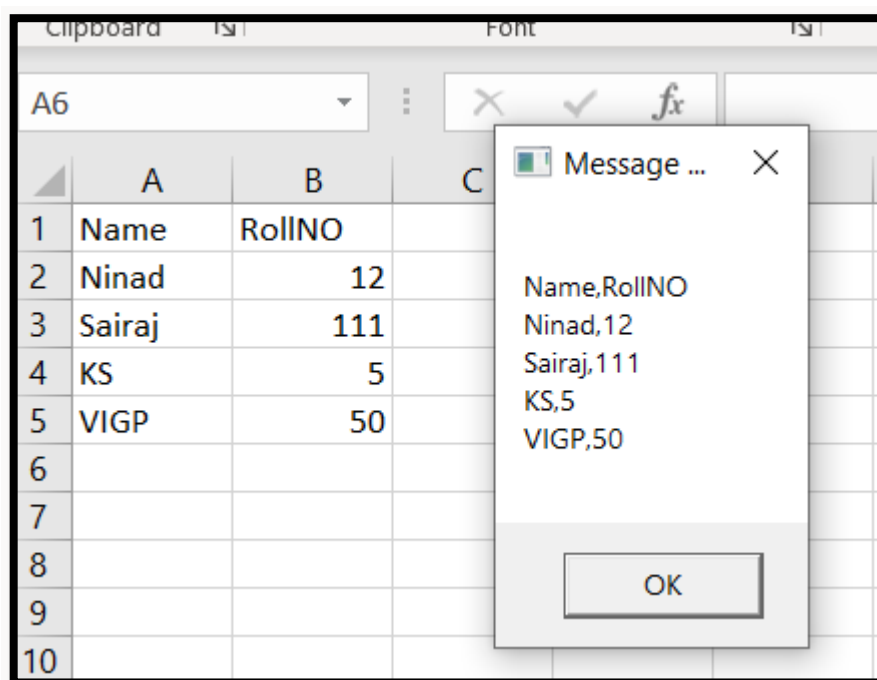
- c. Create a new variable (e.g., TableVar) for the output.



Step 5. Add a "Message Box" activity.

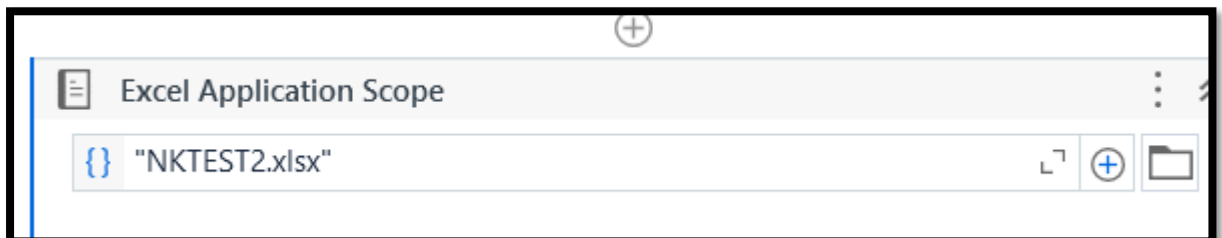


OUTPUT:

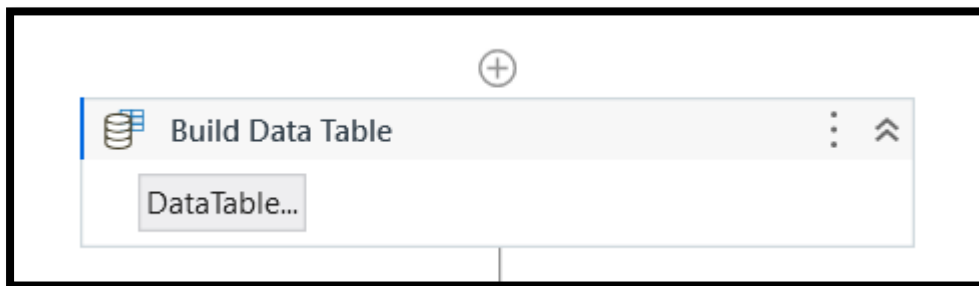


II. Datable to excel

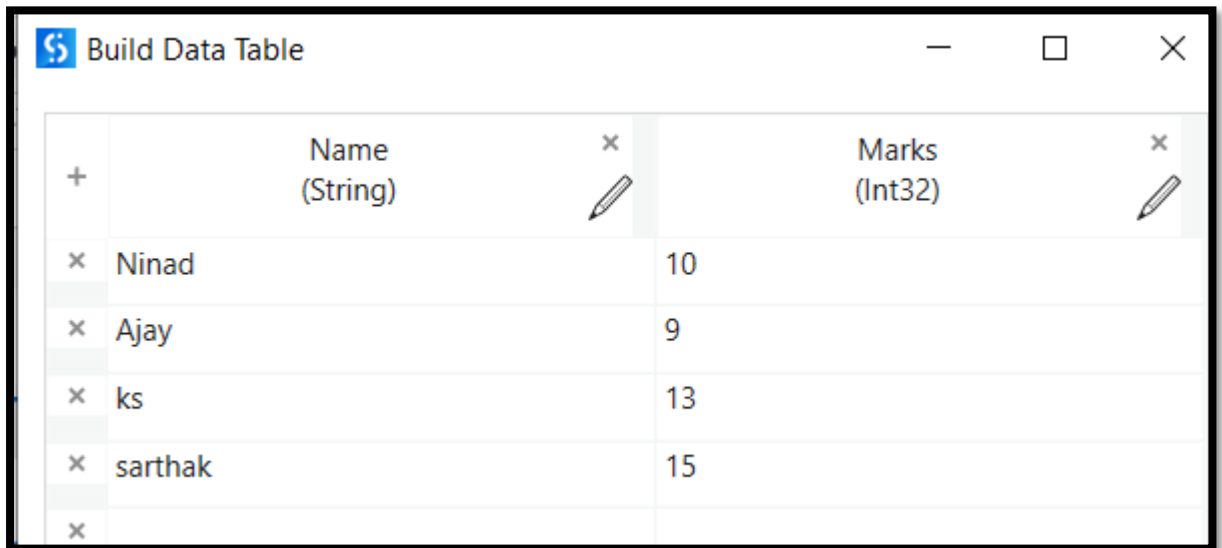
Step 1. Use the "Excel Application Scope" activity to specify the Excel file location.



Step 2. Add the "Build DataTable" activity.



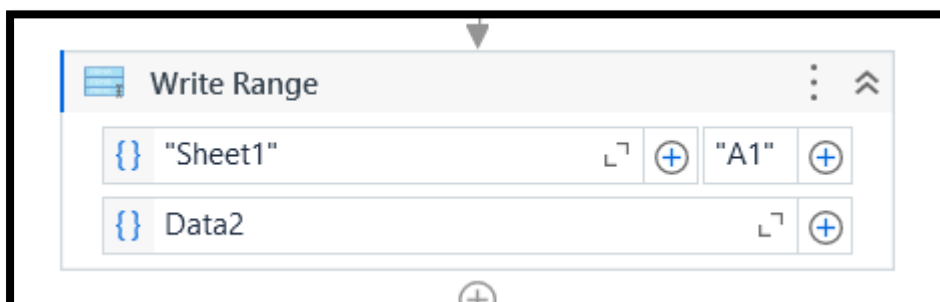
Step 3. Define the structure of the DataTable (columns).



Step 4. Create a variable (e.g., Data2) to store the DataTable.



Step 5. Add the "Write Range" activity inside the Excel Application Scope.



Step 6. Configure it to write the DataTable (Data2) to a specific sheet.

Destination	
SheetName	"Sheet1"
StartingCell	"A1"
Input	
DataTable	Data2
Misc	

Step 7. Save your workflow and run the sequence.

OUTPUT:

Ab			
	A	B	C
1	Name	RollNO	
2	Ninad	12	
3	Sairaj	111	
4	KS	5	
5	VIGP	50	
6			
7			

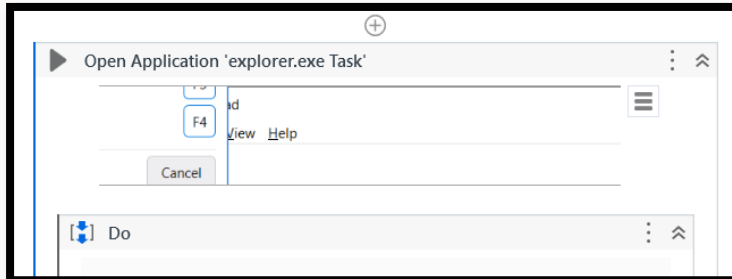
LEARNING:

Automating Excel data extraction involves specifying file locations, reading and outputting DataTables, and handling DataTable structures, enhancing efficiency in data manipulation.

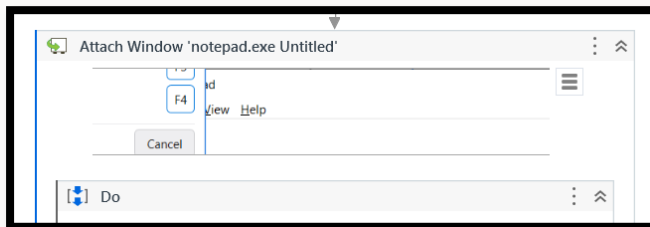
Practical No: 7

AIM: A) Implement the attach window activity.

Step 1. Use the "Open Application" activity to launch Notepad.

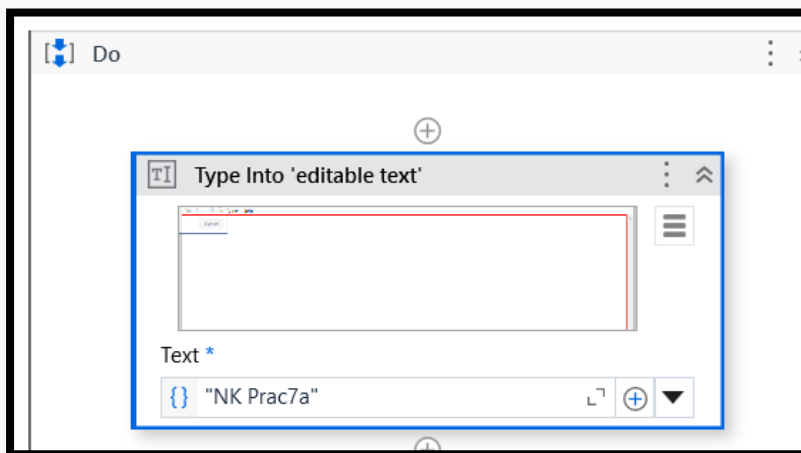


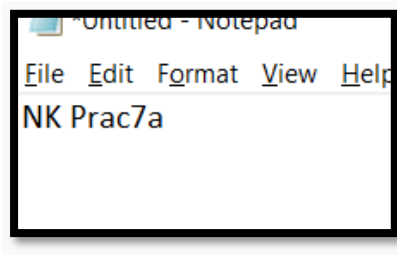
Step 2. Add the "Attach Window" activity to identify and attach to the Notepad window.



Step 3. Within the "Do" section of the "Attach Window" activity, include the "Type Into" activity.

Step 4. In the "Type Into" activity, input some text within quotation marks to be typed into the Notepad.

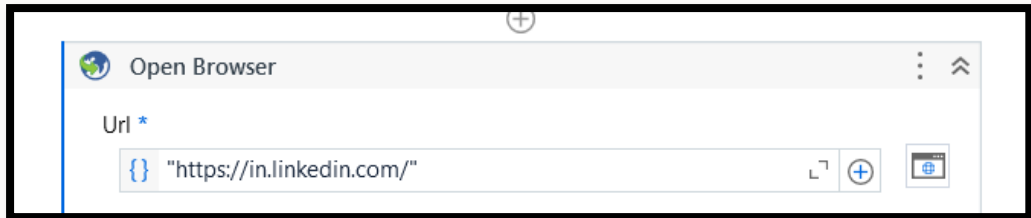


OUTPUT**Learnings:****A. Implement Attach Window Activity:**

By using the Attach Window activity, we learned to launch an application (Notepad), attach to its window, and perform actions within that window, such as typing text using the Type Into activity.

AIM: B) Find different controls using UiPath.**Steps with output**

Step 1. Utilize the "Open Browser" activity and input the LinkedIn URL.



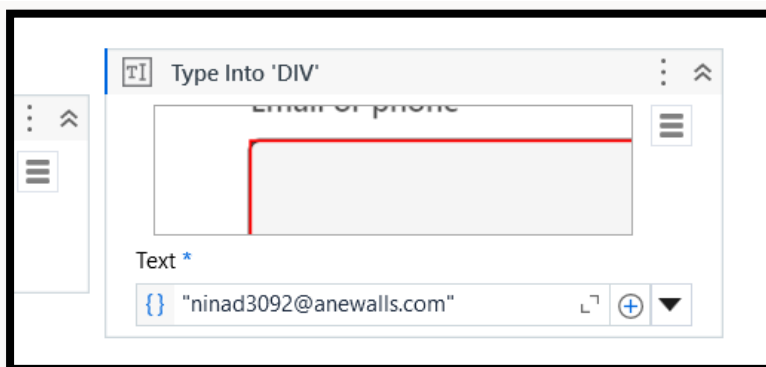
Step 2. Add an "Anchor Base" activity to locate elements relative to a reference anchor.

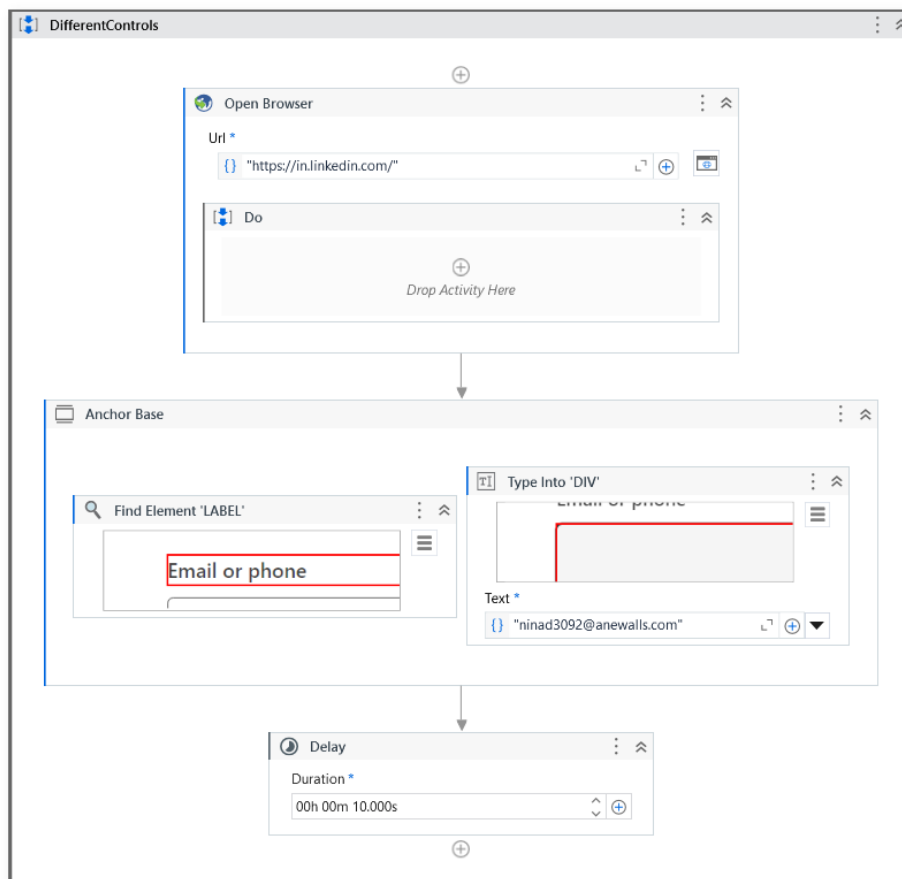


Step 3. Apply the "Find Element" activity as an anchor, indicating the "Email or phone" text on the login page.



Step 4. Integrate the "Type Into" activity within the "Anchor Base," targeting the email text box indicated by the anchor. Input the email within quotation marks.



**OUTPUT:**

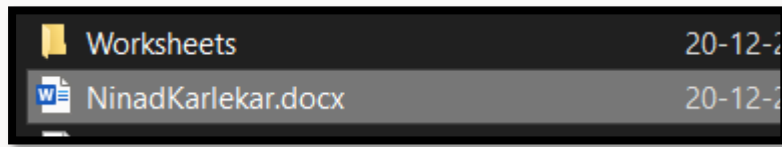
The screenshot shows a web form with a label 'Email or phone' and a text input field containing the email address `ninad3092@anewalls.com`. Below the input field, the label 'Password' is partially visible.

Learnings:**B. Finding Different Controls:**

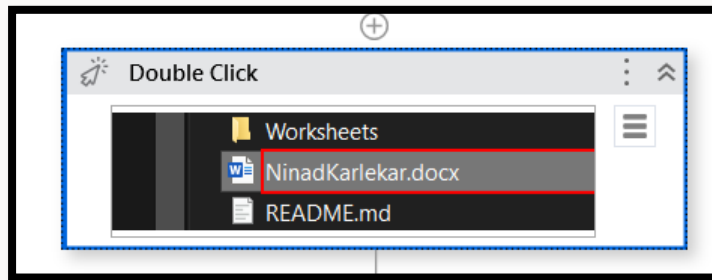
Using Open Browser, Anchor Base, Find Element, and Type Into activities, we learned how to locate and interact with specific elements on a web page, in this case, typing into the email text box on the LinkedIn login page.

AIM: C) Demonstrate the following activities in UiPath:**i. Mouse (click, double click and hover)**

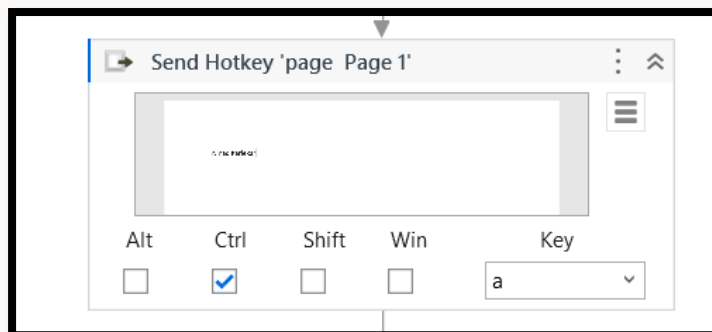
Step 1. Create an existing Word file with content.



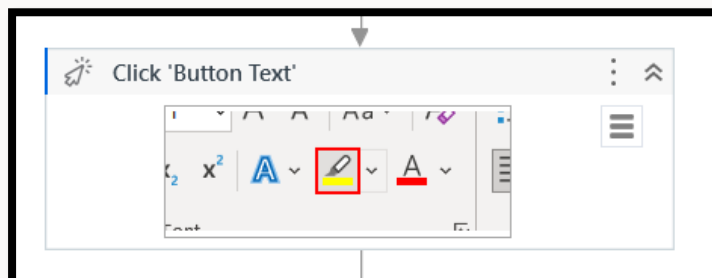
Step 2. Use the Double Click activity to open the Word file from File Explorer.



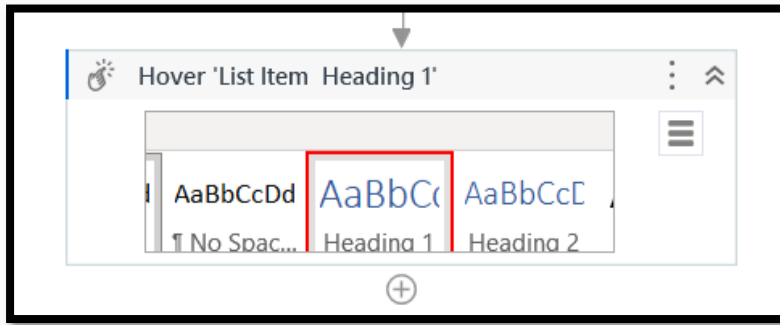
Step 3. Apply the Send Hotkey activity (Ctrl + A) to select all text in the Word document.



Step 4. Utilize the Click activity to click on the "Highlight" option in the Word application.



Step 5. Use the Hover activity to hover over the "Heading 1" option in the ribbon of the Word application.



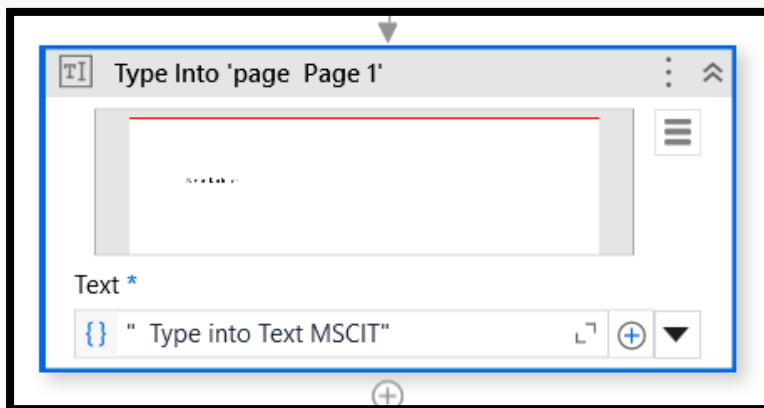
OUTPUT



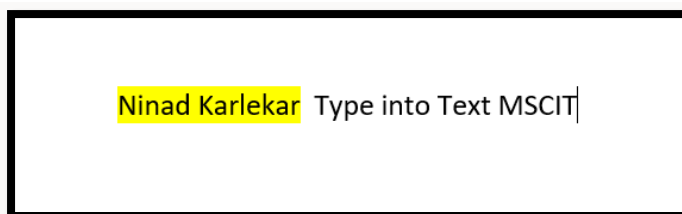
ii. Type Into

Step 1. Add type into activity after performing above steps

Step 2. Type some text into with quotation marks.



OUTPUT:

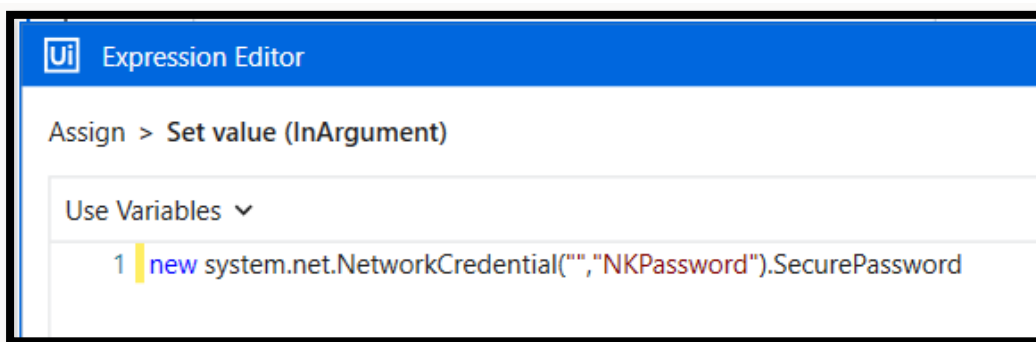


iii. Type Secure Text

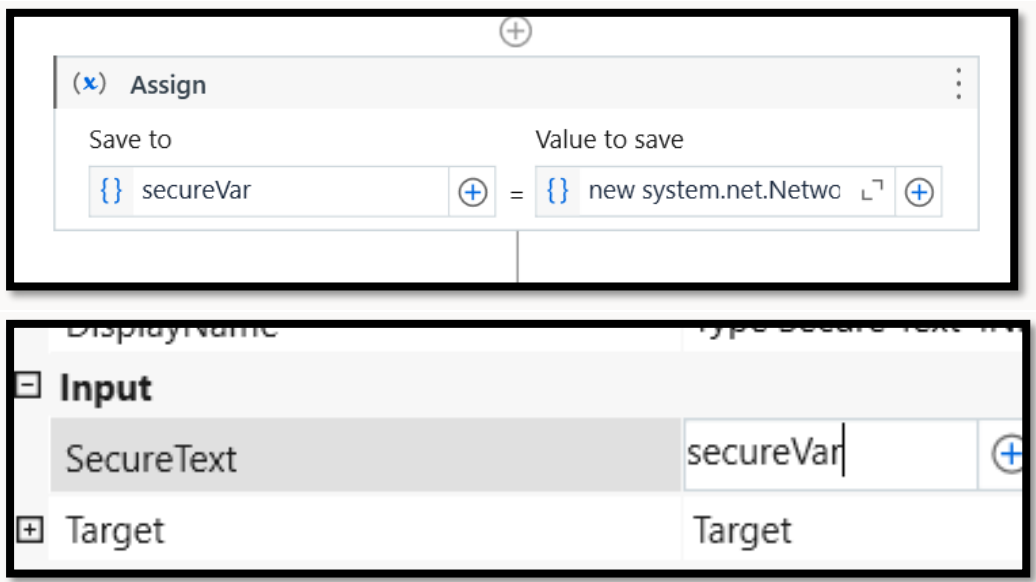
Step 1. Generate a fresh variable and modify its type to SecureString.

Name	Variable type	Scope
secureVar	SecureString	TypeSecureText
Create Variable		

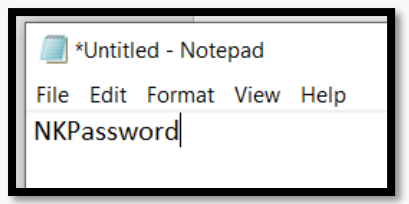
Step 2. Populate the variable with a Username/Password format using the VB Expression: new system.net.NetworkCredential("", "Password123").SecurePassword



Step 3. Integrate a Type Secure Text activity, targeting a password field within a web browser. Ensure to assign the previously created SecureString to the relevant property.



OUTPUT:

**Learnings:****C(i). Mouse Activities:**

We explored Mouse activities in UiPath, including Double Click to open a Word file, Send Hotkey to select all text, Click to highlight text, and Hover to interact with the ribbon.

C(ii). Type Into Activity:

We practiced using the Type Into activity to input text, enhancing our understanding of text entry automation.

C(iii). Type Secure Text:

We learned to secure sensitive information by creating a SecureString variable and using the Type Secure Text activity to input a password securely into a designated field in a web browser.

Practical No: 8**Triggering events, Screen Scraping, Plug-ins**

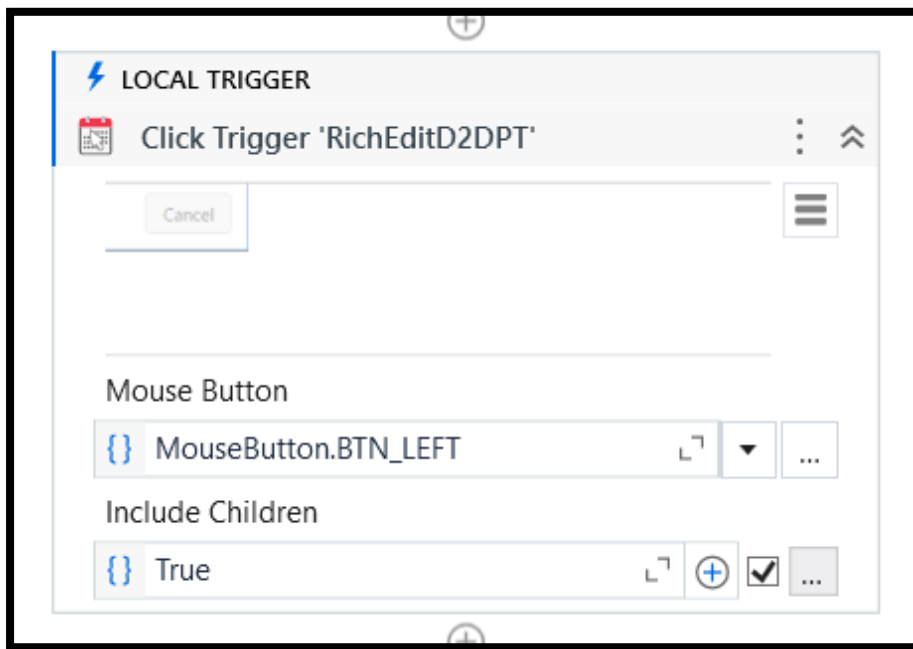
Aim: A) Demonstrate the following events in UiPath:

- i. Element triggering event**
- ii. Image triggering event**
- iii. System Triggering Event.**

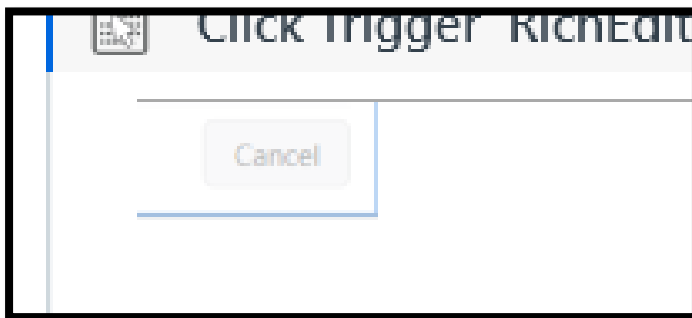
i. Element Triggering event

Step 1. Utilize the "Click Trigger" activity to set up a mouse click event.

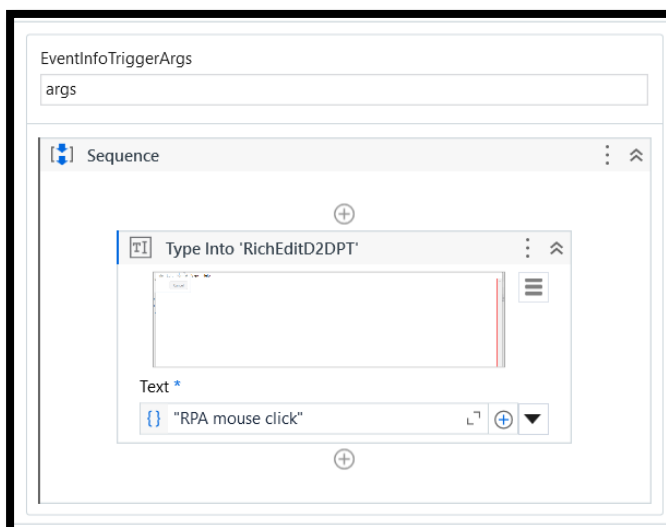
Step 2. Configure the mouse button to be the left mouse button.

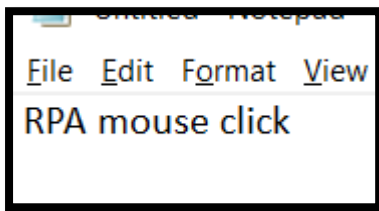


Step 3. Set the window scope for the click trigger by selecting the Notepad window.

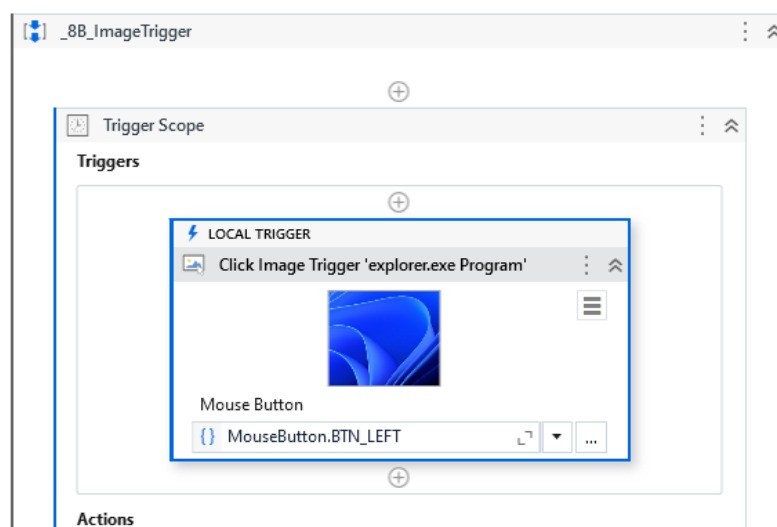


Step 4. Inside the event, include the "Type Into" activity to specify the text you want to be written upon the mouse click.



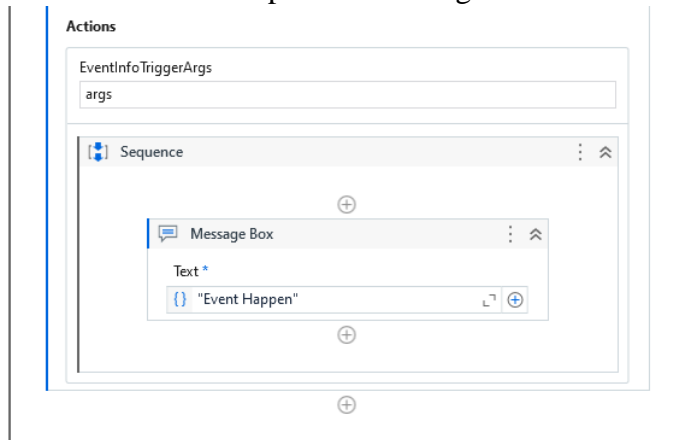
OUTPUT:**ii. Image Triggering event**

Step 1. Drag and drop the "Image Trigger" activity into your sequence.



Step 2. Click "Indicate Scope on screen" and choose a part of an image by dragging.

Step 3. Add a follow-up activity, like "Message Box," which will pop up when you click on the selected part of the image.

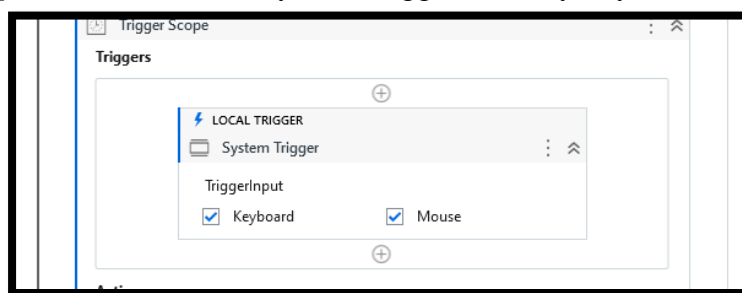


OUTPUT:

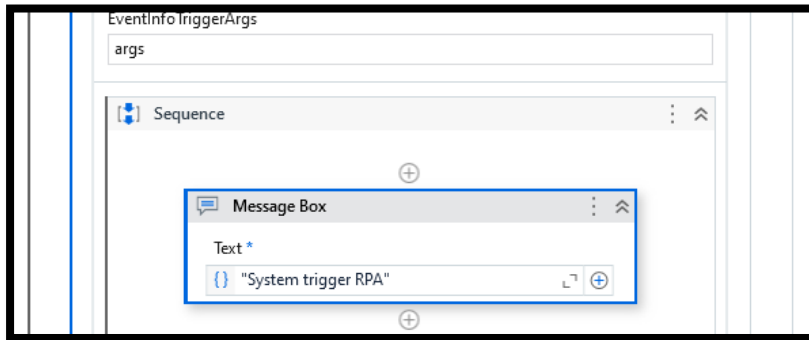


iii. System Triggering Events

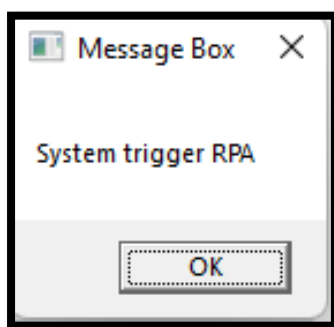
Step 1. Include the "System Trigger" activity in your workflow.



Step 2. Integrate a follow-up activity, such as "Message Box," to occur after the System Trigger. This activity will trigger when any key or mouse press occurs.



OUTPUT:



Learnings:

i. Element Triggering Event:

We learned to set up a mouse click event using the "Click Trigger" activity, configuring the mouse button, defining the window scope, and incorporating a follow-up "Type Into" activity to execute upon the mouse click.

ii. Image Triggering Event:

We explored the "Image Trigger" activity to create an event triggered by clicking on a selected part of an image, and added a follow-up "Message Box" activity to display a message when the image is clicked.

iii. System Triggering Events:

By using the "System Trigger" activity, we learned to set up an event triggered by any key or mouse press and added a follow-up "Message Box" activity to demonstrate the occurrence of the triggering event.

Aim: b) Automate the following screen scraping methods using UiPath

i. Full Text

ii. Native

iii. OCR

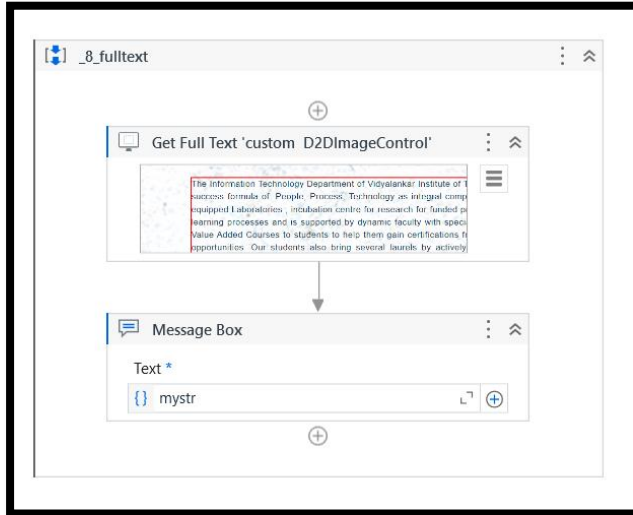
i. Full Text

Step 1. Drag and drop the "Get Full Text" activity into your sequence.

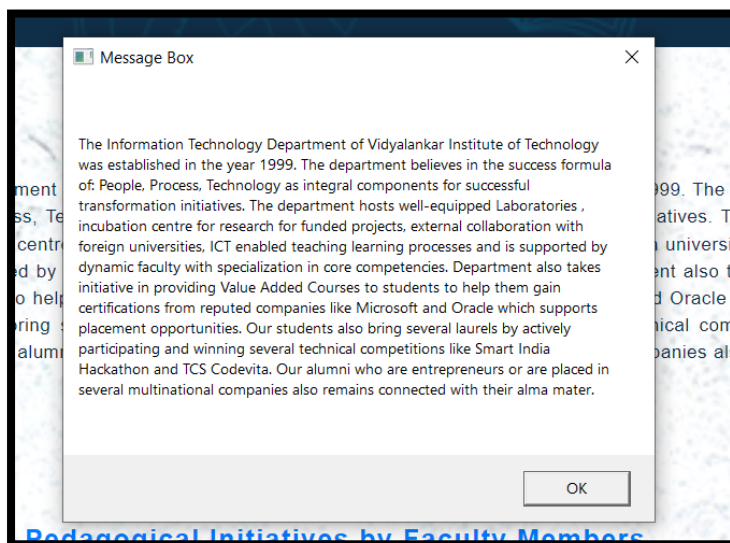
Step 2. Click "Indicate on Screen" within the activity.

Step 3. Select a text pane on the screen.

- Step 4.** In the properties of the "Get Full Text" activity, add a string variable to the output using [Ctrl + K].
- Step 5.** Add a "Message Box" activity to your sequence.
- Step 6.** Use the variable created in step 4 to print the extracted text in the message box.



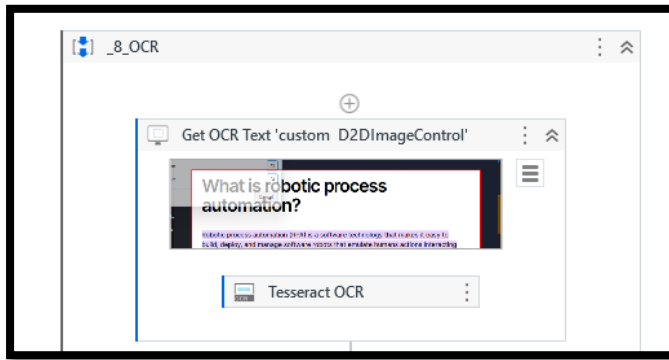
OUTPUT:



ii. OCR

- Step 1.** Drag and drop the "Get OCR Text" activity into your sequence.

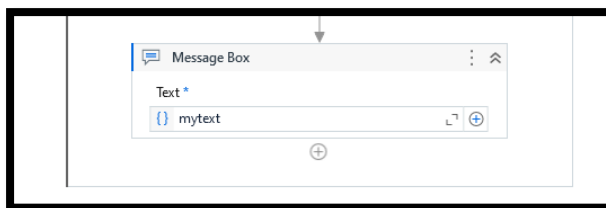
Step 2. Click "Indicate on Screen" within the activity.



Step 3. Select an image containing text that you want to extract.

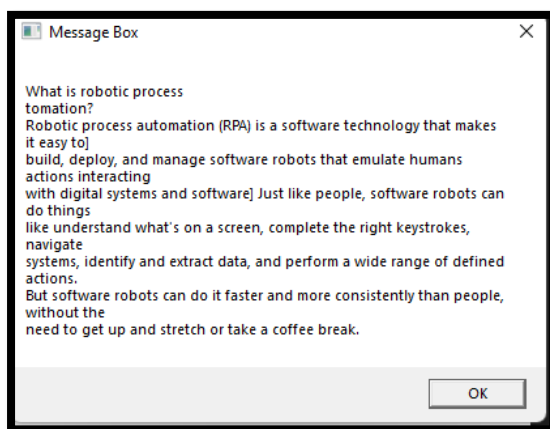
Step 4. Create a variable to store the OCR output.

Step 5. Add a "Message Box" activity to your sequence.



Step 6. Use the variable created in step 3 to display the OCR-extracted text.

OUTPUT:



Learnings:

i. Full Text:

We learned to use the "Get Full Text" activity to extract text from a specific pane on the screen, store it in a variable, and display it using a Message Box.

ii. OCR:

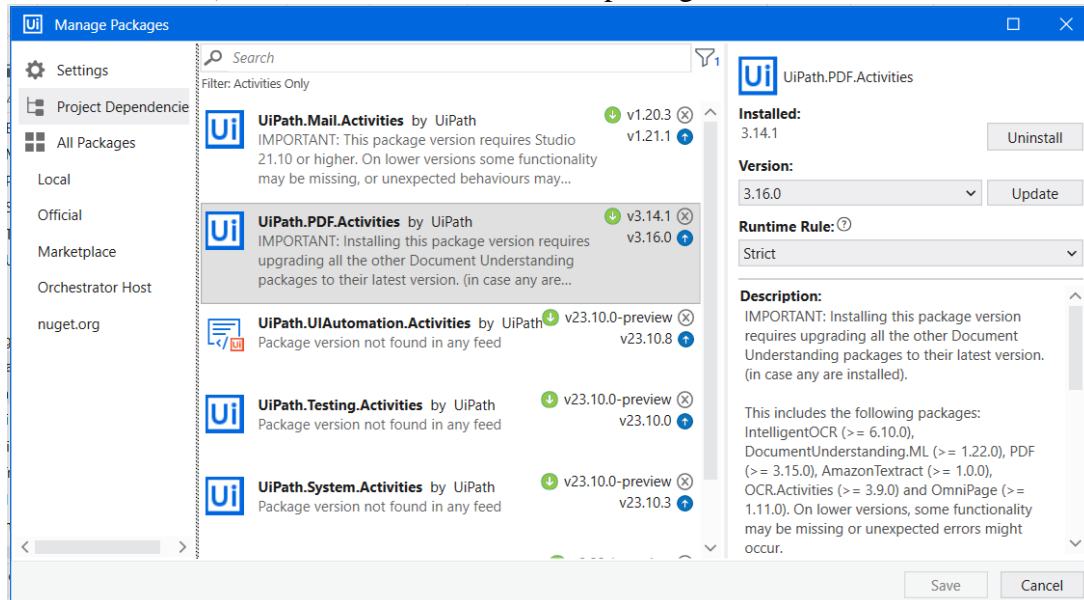
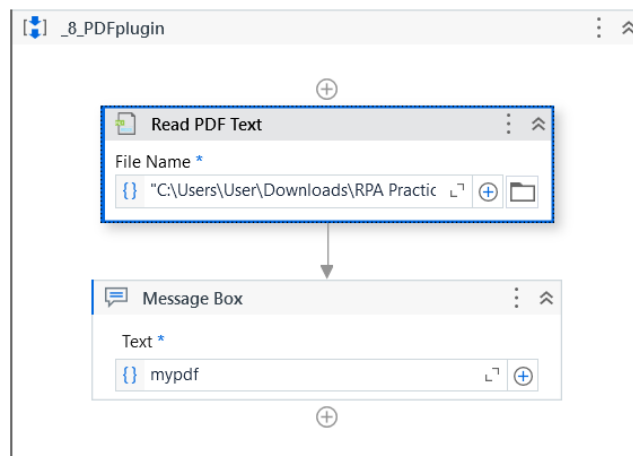
By employing the "Get OCR Text" activity, we learned to capture text from an image, save it in a variable, and showcase the extracted text in a Message Box, demonstrating the application of Optical Character Recognition (OCR) in UiPath.

Aim: 3. Install and automate any process using UiPath with the following plug-ins:

i. PDF Plugin

Step 1. Install UiPath.PDF Package:

- a) Open UiPath Studio.
- b) Navigate to "Manage Packages" > "All Packages."
- c) Search for "uipath.pdf" and select 'UiPath.PDF.Activities.'
- d) Click "Install" to install the package.

**Step 2.** Drag and drop the "Read PDF Text" activity into your sequence.**Step 3.** Within the activity, click on the field for selecting a PDF file and choose the desired PDF.**Step 4.** In the properties of the "Read PDF Text" activity, set the range of pages and select a string variable to store the extracted text.**Step 5.** Integrate a "Message Box" activity into your sequence.

Step 6. Use the string variable to display the extracted text in the message box.

Common	
DisplayName	Read PDF Text
File	
FileName	"C:\Users\User\C" + ...
Password	The password of + ...
Input	
PreserveFormatting	A flag used to sig + ...
Range	"1" + ...
Misc	
Private	<input type="checkbox"/>
Output	
Text	mypdf + ...

OUTPUT:

Message Box

INDEX
Roll No: _
PSIT3P4a Robotic Process Automation

No Title/Aim Date Faculty Grade Student
. Sign Sign

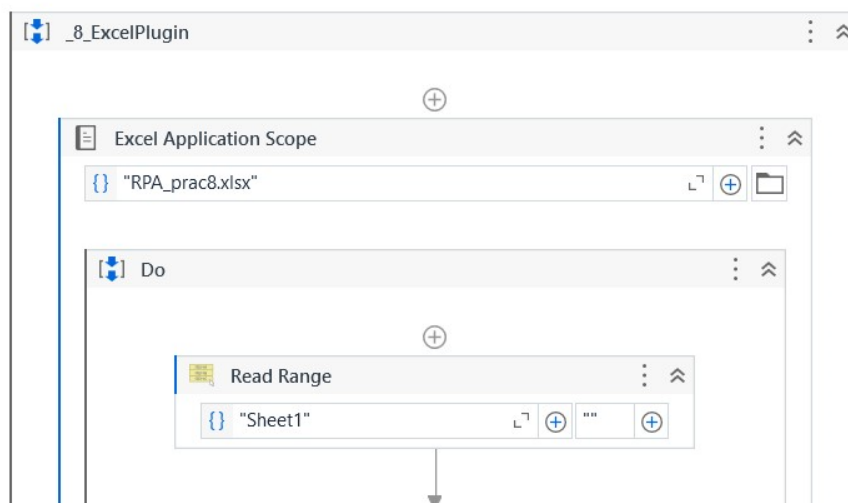
1 a) Create a simple sequence- based project.
b) Create a flowchart-based project.

2 a) Automate UiPath Number Calculation
(Subtraction, Multiplication, Division of
numbers).
b) Create an automation UiPath project using
different types of variables (number,
datetime, Boolean, generic, array, data
table)
3 a) Create an automation UiPath Project using
decision statements.
b) Create an automation UiPath Project using
looping statements.
4 a) Automate any process using desktop
recording.
b) Automate any process using web recording.
5 a) Consider an array of names. We have to find
out how many of them start with the letter
"a". Create an automation where the number
of names starting with "a" is counted and the
result is displayed
6 a) Create an application automating the read,
write and append operation on excel file.
b) Automate the process to extract data from
an excel file into a data table and vice versa
7 a) Implement the attach window activity.
b) Find different controls using UiPath.
c) Demonstrate the following activities in
UiPath:
i. Mouse (click, double click and hover)
ii. Type into

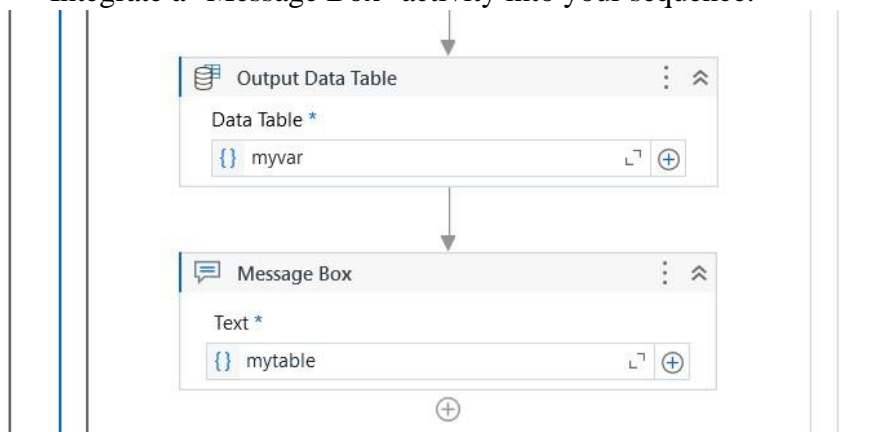
ii. EXCEL PLUGIN

Pre-requisite: Install UiPath.Excel.Activities

- Step 1.** Install Excel Plugin(Home > Tools > UiPath Extensions > Install 'Excel Add-in')
- Step 2.** Create an Excel file containing two columns: "Name" and "Marks."
- Step 3.** In UiPath Studio, add an "Excel Application Scope" activity to your sequence.
- Step 4.** Within the activity, select the Excel file you created in step 2.

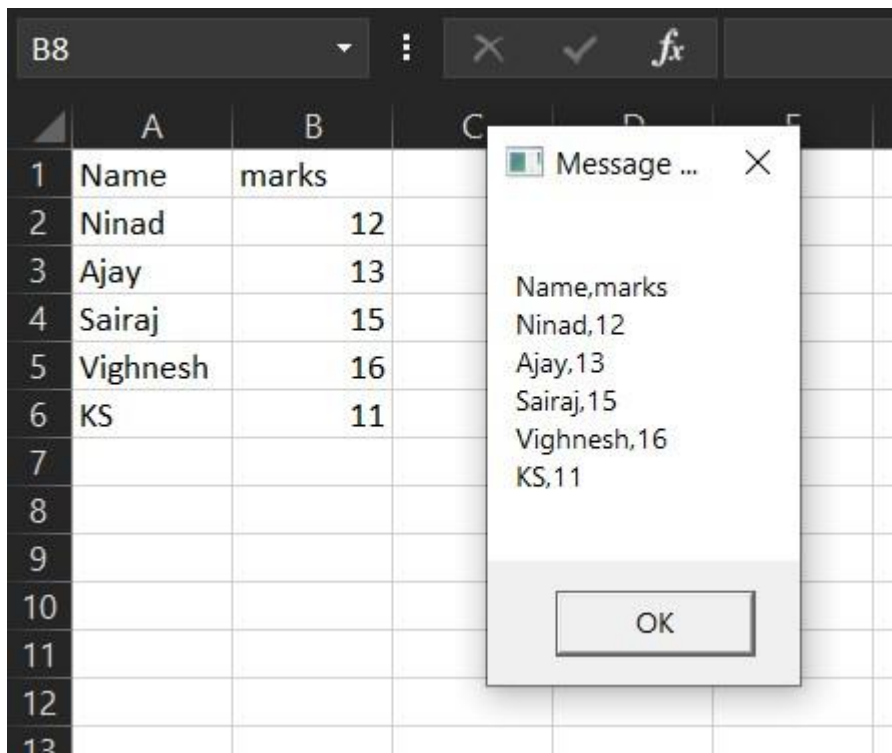


- Step 5.** Add a "Read Range" activity inside the Excel Application Scope.
- Step 6.** Integrate a "Message Box" activity into your sequence.



- Step 7.** Use the output of the "Read Range" activity to display the read data in the message box.

OUTPUT:



Learnings:

i. PDF Plugin:

We learned to install the UiPath.PDF package, use the "Read PDF Text" activity to extract text from a PDF file, and display the extracted text in a Message Box.

ii. Excel Plugin:

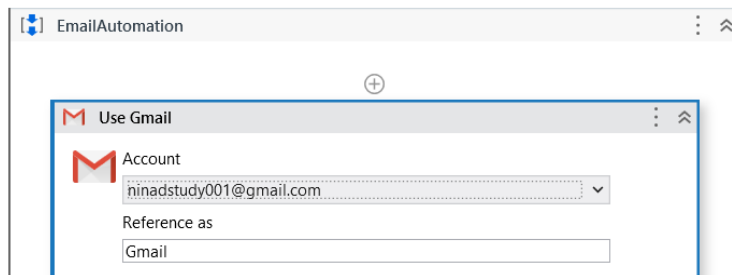
We grasped the process of installing the UiPath.Excel.Activities plugin, adding an Excel file containing data, utilizing "Excel Application Scope" and "Read Range" activities to read data, and displaying the read data in a Message Box using the output variable.

Practical No: 9

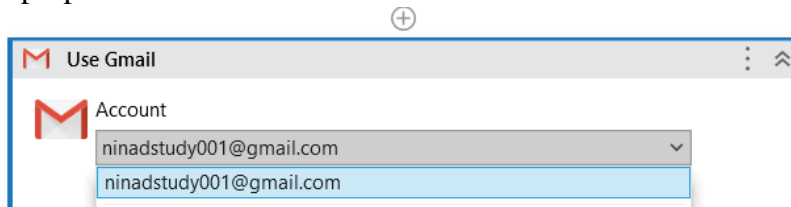
Email Automation

AIM: A) Automate the process of send mail event (on any email)

Step 1. Drag and drop the "Use Gmail" activity into your sequence.



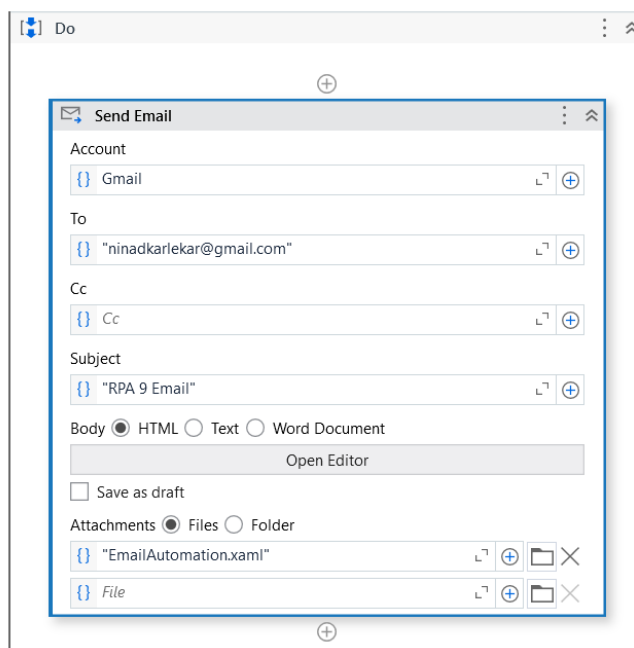
Step 2. Choose the default option and authenticate your Gmail account in the browser pop-up.

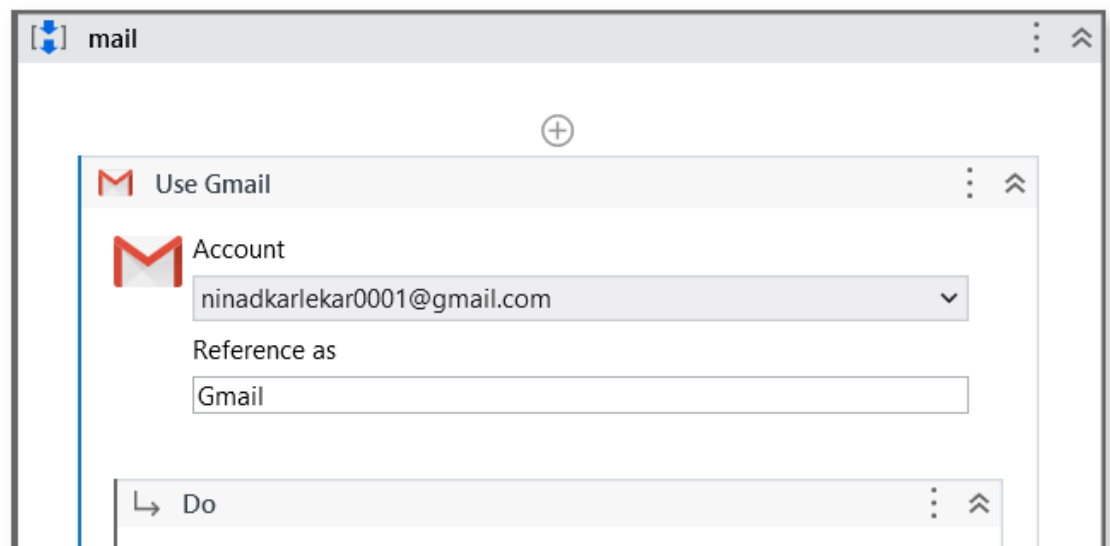


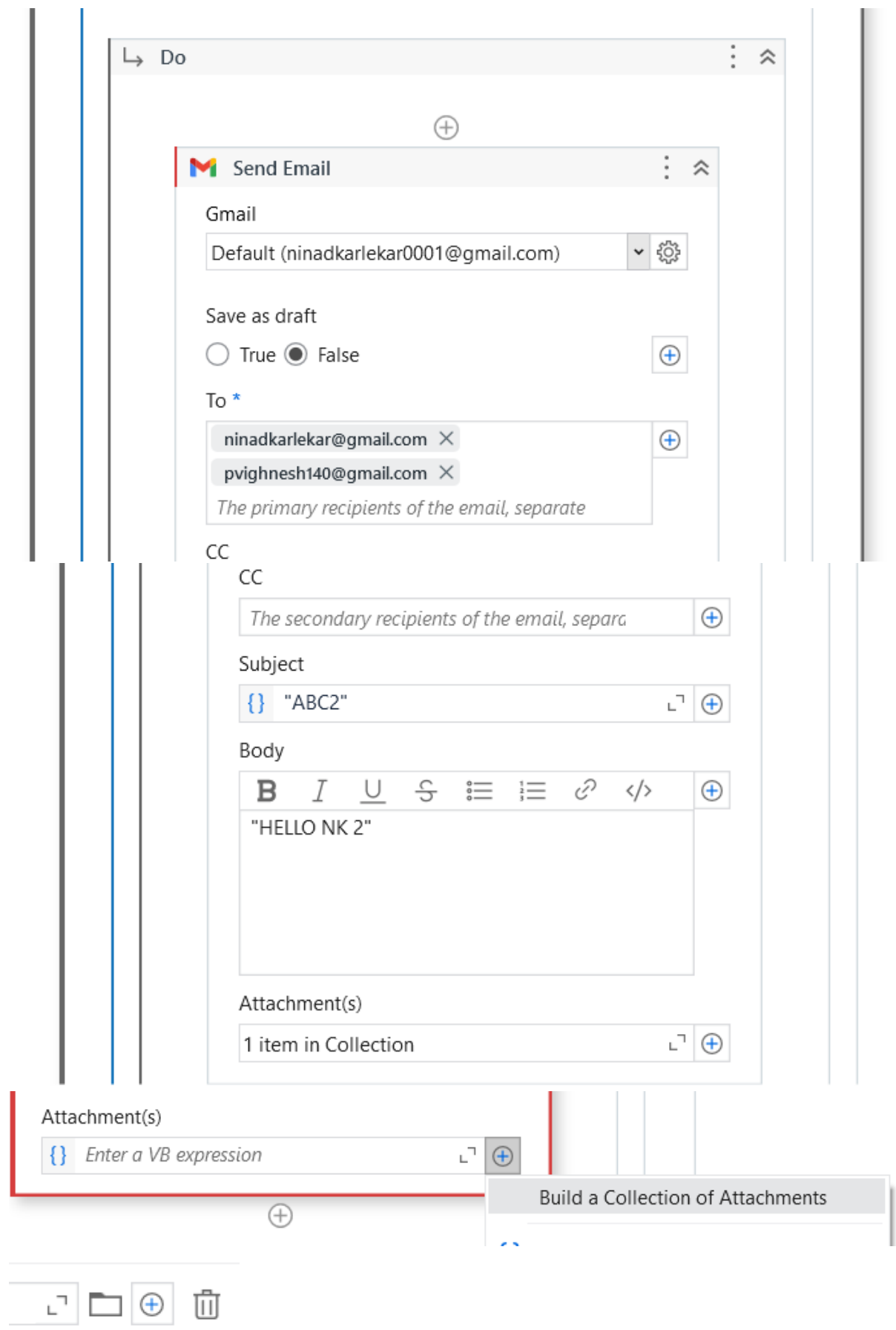
Step 3. Inside the "Do" section of the "Use Gmail" activity, add a "Send Email" activity.

Step 4. Select Gmail as the account.

Step 5. Enter the recipient's email address, subject, body, and attach a file if necessary.



OUTPUT:

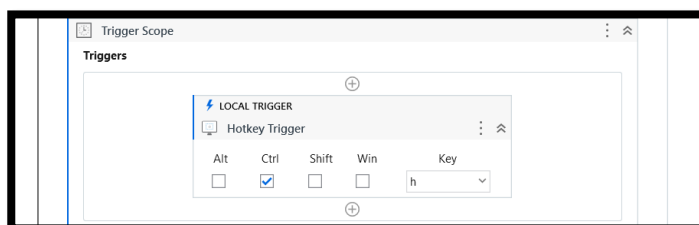


Learnings:**A. Automate Send Mail Event:**

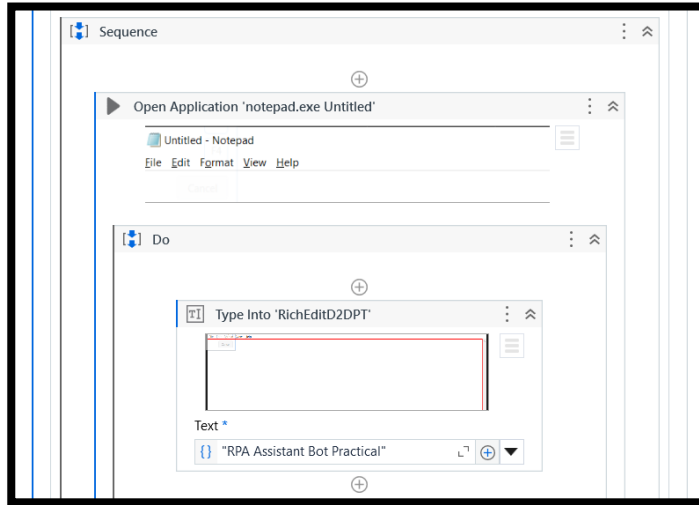
We learned to use the "Use Gmail" activity to send an email, authenticate a Gmail account, and configure email details, enabling automation of the email-sending process.

B: Automate the process of launching an assistant bot on a keyboard event

- Step 1.** Drag and drop the "Trigger Scope" activity into your sequence.
- Step 2.** Inside the Trigger Scope, add a "Hotkey Trigger" activity.
- Step 3.** Configure the Hotkey Trigger by selecting the desired hotkey combination.

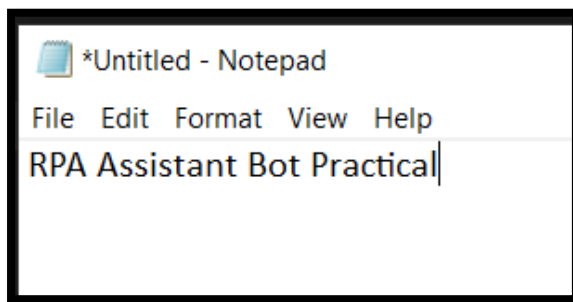


Step 4. In the Action section of the Trigger Scope, add the sequence of activities that you want to be executed when the specified hotkey combination is pressed.



Step 5. Save your workflow and run the sequence.

OUTPUT:



B. Automate Launching Assistant Bot on Keyboard Event:

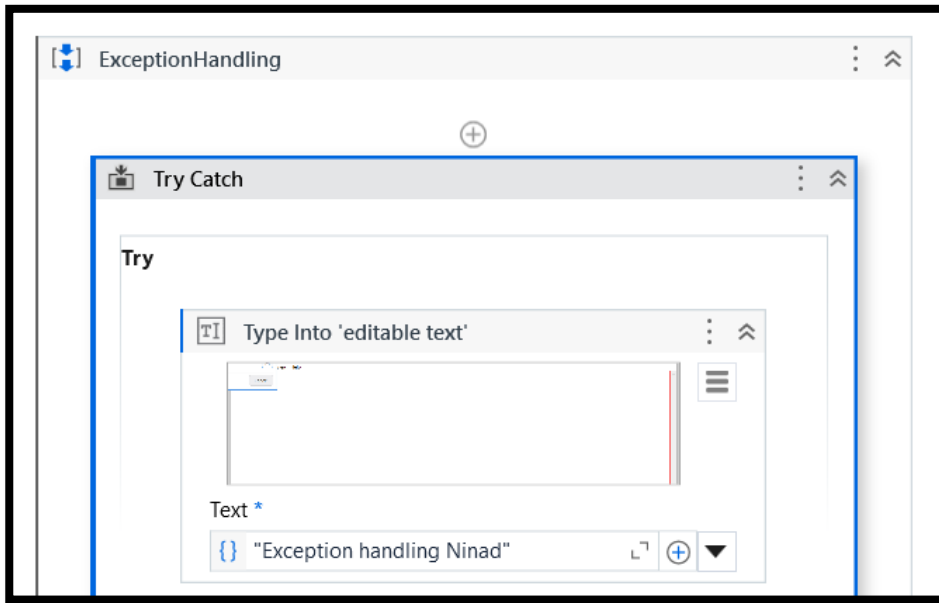
We grasped the concept of using the "Trigger Scope" and "Hotkey Trigger" activities to execute a sequence of actions when a specified hotkey combination is pressed, facilitating automation triggered by keyboard events.

Aim: c) Demonstrate the Exception handling in UiPath.

Step 1. Drag and drop the "Try Catch" activity into your sequence.

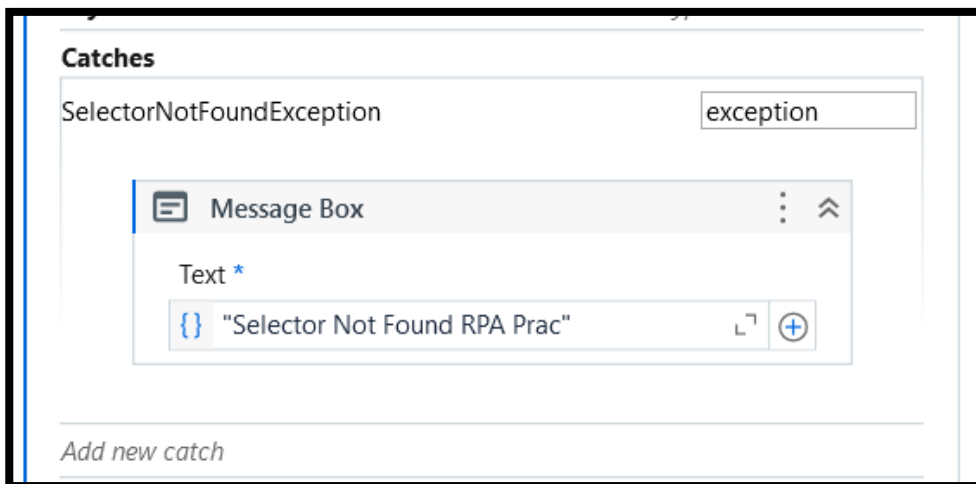
Step 2. Inside the Try block, add an activity that might throw an exception during execution. For example, use a "Type Into" activity to type text into an already running

Notepad window. If the Notepad window is not running, it will throw an exception.

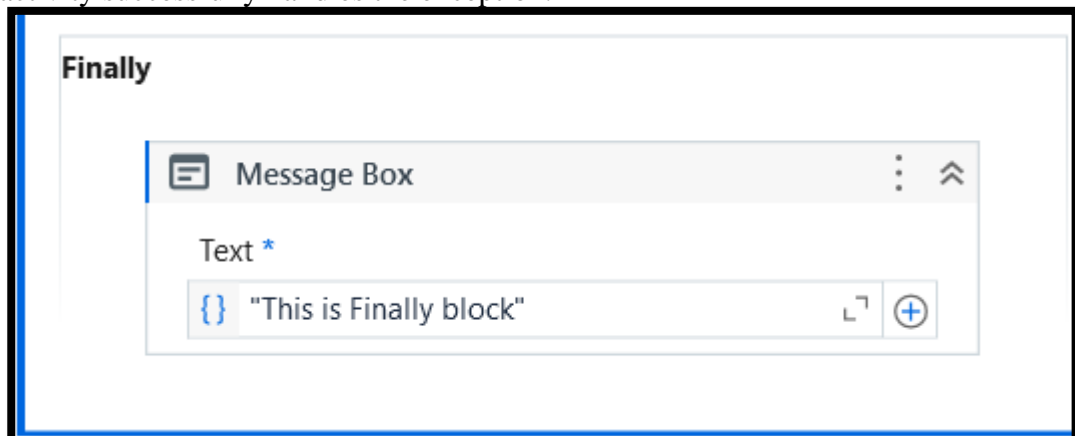


Step 3. In the Catch block, handle a specific exception. For example, take a "SelectorNotFoundException" exception.

Step 4. Add a "Message Box" activity inside the Catch block to notify you when an exception is thrown.



Step 5. In the Finally block, add a "Message Box" activity to test if the Try-Catch activity successfully handles the exception.



Step 6. Save your workflow and run the sequence.

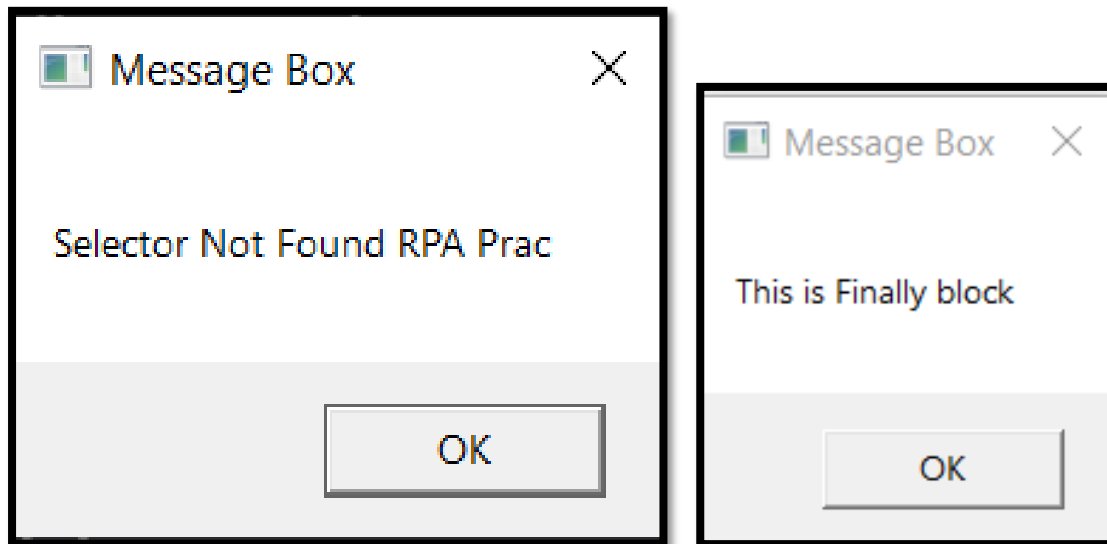
OUTPUT:

Case 1: Notepad is running



Case 2: Notepad is not in an active state

(In this situation, the UiPath application will pause for a duration of 30 seconds, attempting to find the Notepad window. If unsuccessful, it will then trigger an exception.)

**Learnings:****C. Demonstrate Exception Handling:**

We understood the implementation of the "Try Catch" activity for handling exceptions, demonstrated by attempting to type into a Notepad window and catching a specific exception (SelectorNotFoundException), with notification via a "Message Box" activity. The Finally block tested the overall success of handling exceptions.