***UI PATH* *Assignment Solutions***

**Primary Data types and Manipulations**

1. **Create a List and add 1-10 numbers in List using Loop and Print the List**

Open UiPath Studio and start a new Sequence.

Create a List Variable:

* Go to the "Variables" panel (at the bottom).
* Create a variable called MyList, set its type as List<Int32>.
* Initialize it by clicking Default and writing New List(Of Int32).

Add a Loop:

* Drag and drop a "For Each" activity.
* Set the range of the loop by writing Enumerable.Range(1, 10) in the loop's "Values".

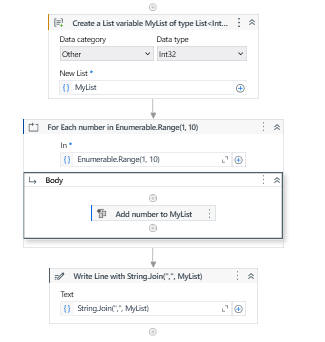
Add Numbers to the List:

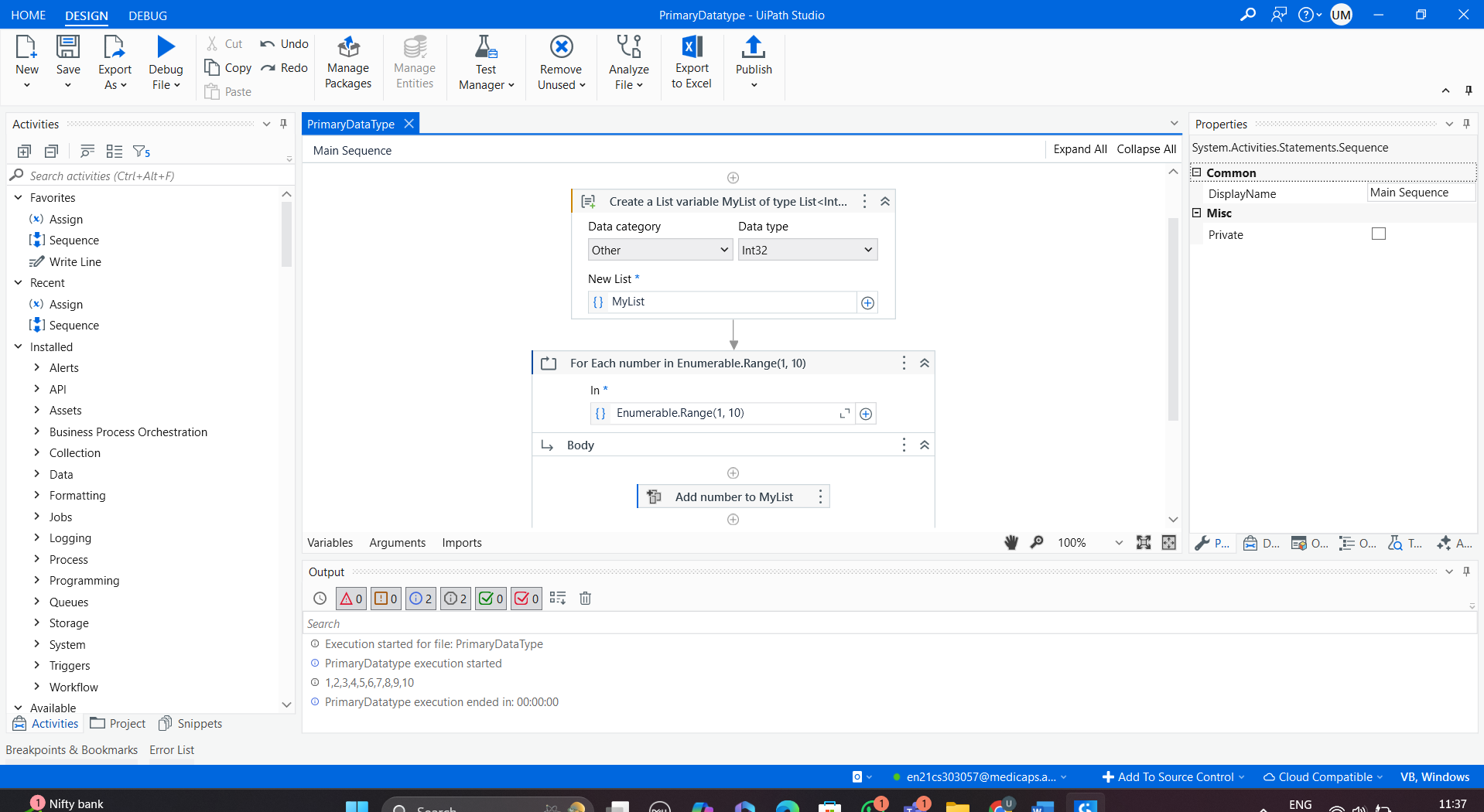
* Inside the loop, drag "Add to Collection" activity.
* Set the collection as MyList and the item as the current loop value (e.g., item).

Print the List:

* Drag "Write Line" activity.
* Use String.Join(",", MyList) to display all numbers in the list.

Run the program.

****

****

1. **Create a Datatable and Print the Row Count. Increment the Value till the Value is not equal to 10 using Do while loop.**

Create a Datatable:

* Drag the "Build Data Table" activity.
* Add a single column (e.g., "Column1").
* Add some rows with sample values (e.g., 1, 2, 3...).

Get Row Count:

* Use the "Assign" activity to store the row count in a variable:

RowCount = DataTable.Rows.Count

Create a Variable for Increment:

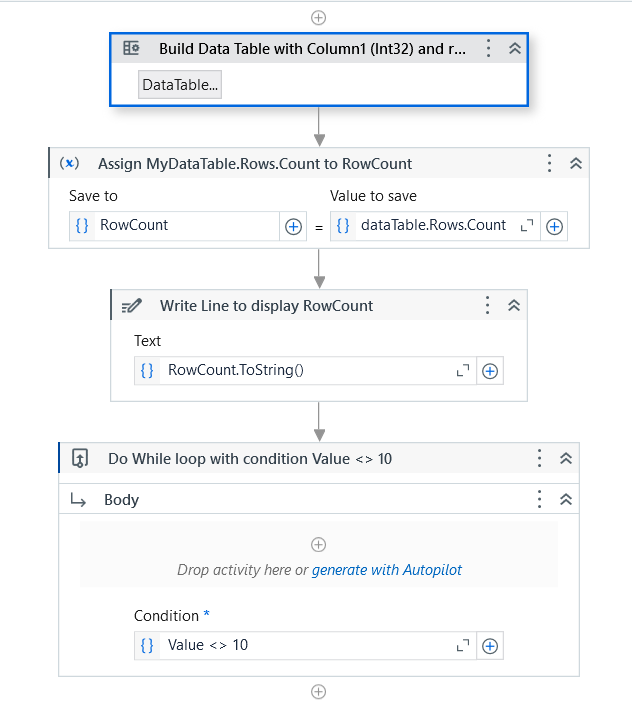
* Create a variable Value and set its default to 0.

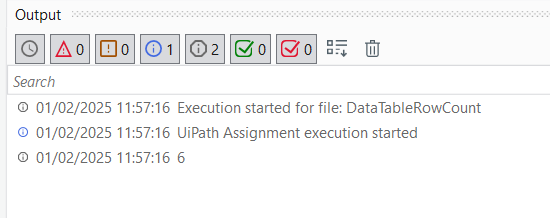
Use Do While Loop:

* Drag the "Do While" activity.
* Inside the loop, increment the value using Assign:
  + Value = Value + 1
* Add a condition for the loop to continue:
  + Value <> 10

Print Values:

* Use the "Write Line" activity to print Value in each iteration**.**

****

****

1. **Create a Dictionary for a class having 5 students and print the Marks obtained. for each student**

* Create a New Sequence: Open UiPath Studio and create a new sequence.
* Declare a Dictionary:
* Drag and drop an Assign activity.
* In the To field, write the name of the dictionary variable (e.g., studentMarks).
* In the Value field, use the following syntax to create and initialize the dictionary:

New Dictionary(Of String, Integer) From {

{"Alice", 85},

{"Bob", 92},

{"Cathy", 78},

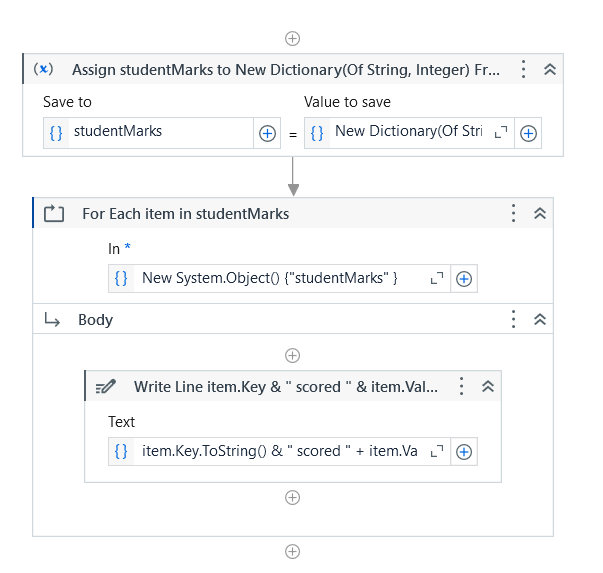
{"David", 89},

{"Ella", 95}

}

* Loop Through the Dictionary:
* Drag and drop a For Each activity.
* Set the TypeArgument of the For Each activity to KeyValuePair(Of String, Integer).
* Set the Values property of the For Each to studentMarks.
* Print Each Student's Marks:
* Inside the For Each loop, use a Write Line or Message Box activity to display the marks.
* Use the following syntax to display the marks:

item.Key & " scored " & item.Value.ToString()

****

1. **Consider a string H;e!llo;Eve;ry;one! and Print the output as - Hello Everyone.**

**Steps to Implement**

1. **Create a New Sequence**: Open UiPath Studio and create a new sequence.
2. **Define the Input String**:
   * Drag and drop an **Assign** activity.
   * In the **To** field, create a variable named inputString of type String.
   * In the **Value** field, set the value to:

"H;e!llo;Eve;ry;one!"

1. **Clean the String**:
   * Drag and drop another **Assign** activity.
   * Create a new variable cleanedString of type String.
   * In the **Value** field, write this expression to remove unwanted characters and format the string:

cleanedString = inputString.Replace(";", "").Replace("!", "").Trim()

1. Here's what happens:
   * **Replace(";", "")** removes all semicolons.
   * **Replace("!", "")** removes the exclamation marks.
   * **Trim()** ensures no leading or trailing spaces.
2. **Display the Cleaned String**:
   * Drag and drop a **Write Line** or **Message Box** activity.
   * Set the **Text** property to:

cleanedString

**Full Workflow Example**

1. **Assign Activity 1**:  
   To create the input string:

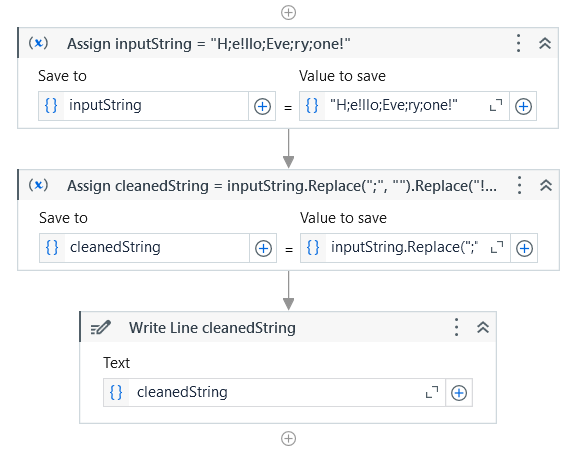
inputString = "H;e!llo;Eve;ry;one!"

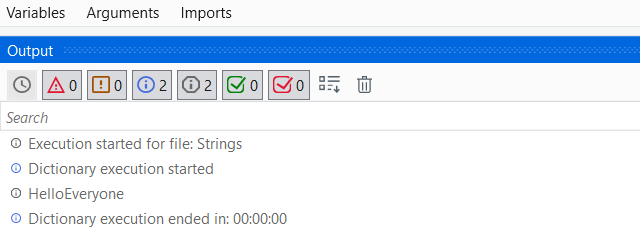
1. **Assign Activity 2**:  
   To clean the string and remove unwanted characters:

cleanedString = inputString.Replace(";", "").Replace("!", "").Trim()

1. **Write Line Activity**:  
   To display the cleaned string:

cleanedString

****

****

**Automation Activities**

**Activity 1**

1. **Open browser, search "Java" and print the text of the first URL. Return to Home page and now search "Selenium", click on the first URL.**
2. Open Browser:
   * Drag "Use Application/Browser" and indicate the browser (e.g., Chrome, Edge) window where you'll search.
   * Set the URL as "https://www.google.com".
3. Search for "Java":
   * Drag "Type Into" inside the browser activity. Indicate the Google search bar.
   * Enter the text "Java" + "[k(enter)]" (this types "Java" and hits Enter).
4. Get First URL:
   * Drag "Get Text" activity.
   * Indicate the first search result link. Use UiPath's selector tool to point to it.
   * Store the output in a variable (e.g., FirstJavaUrl).
5. Write Line (Print URL):
   * Drag "Write Line" or "Log Message" activity.
   * Enter the variable FirstJavaUrl to print it in the output pane.

Step 5: Return to the Home Page

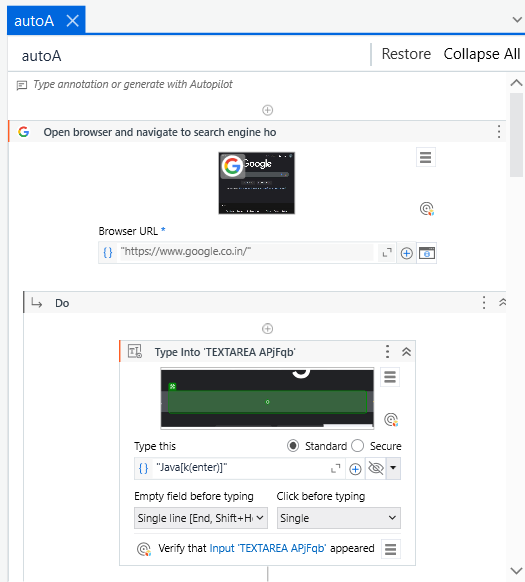
* Use another "Type Into" activity to navigate to Google’s home page by typing "https://www.google.com" in the browser and pressing Enter.

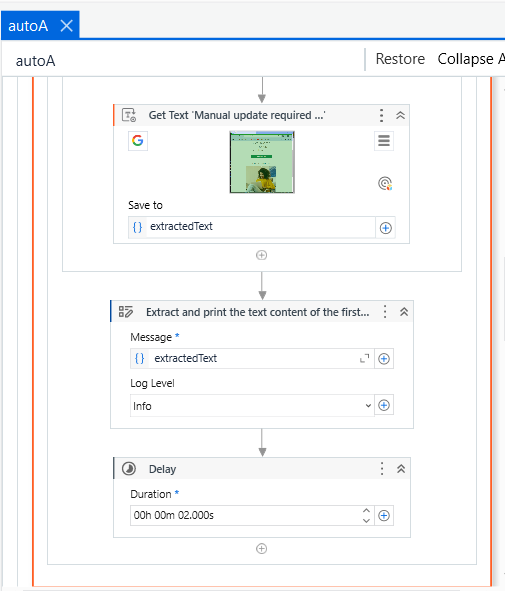
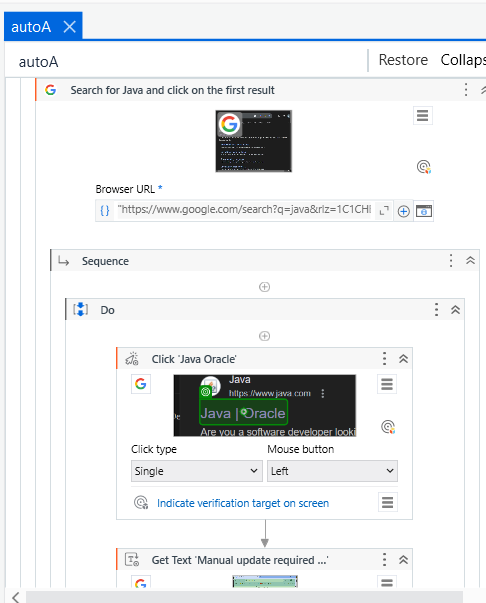
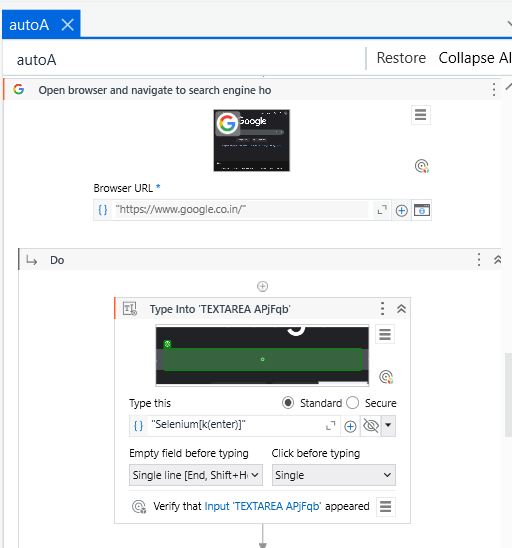
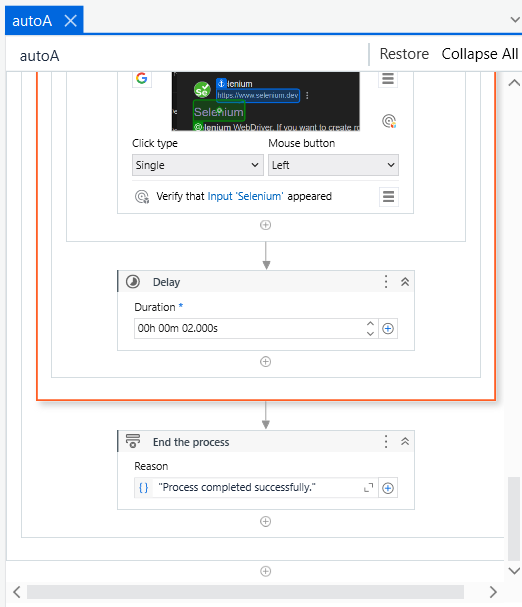
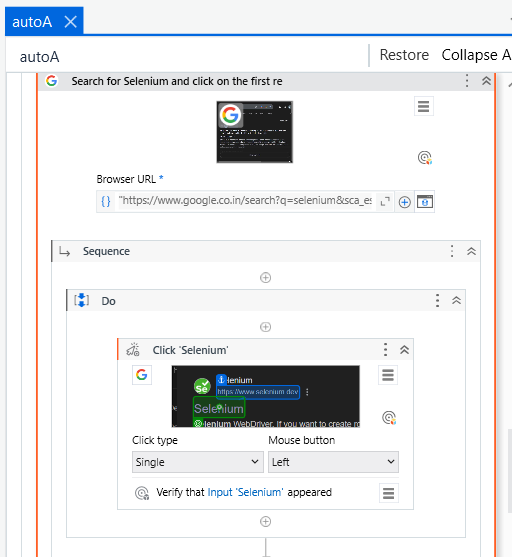
Step 6: Automate "Search Selenium and Click First URL"

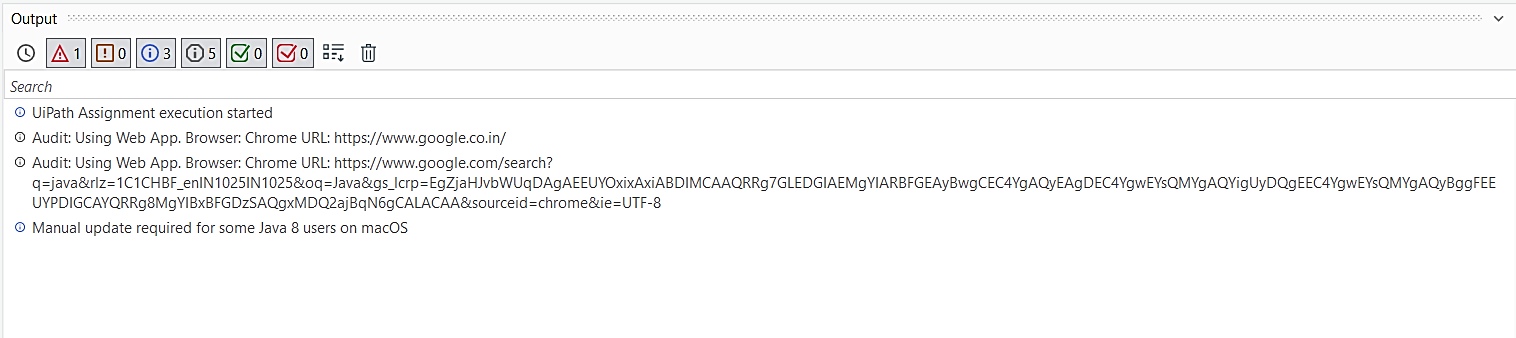
1. Search for "Selenium":
   * Drag another "Type Into" activity.
   * Indicate the Google search bar again.
   * Enter the text "Selenium" + "[k(enter)]".
2. Click on the First URL:
   * Drag "Click" activity.
   * Indicate the first search result link and ensure the selector is set to click the correct element.

Step 7: Save and Run the Automation

1. Save the workflow by clicking Save.
2. Click Run File in the top toolbar to test the automation. Ensure your browser (e.g., Chrome or Edge) is set as the default and the extension is installed and enabled.

****

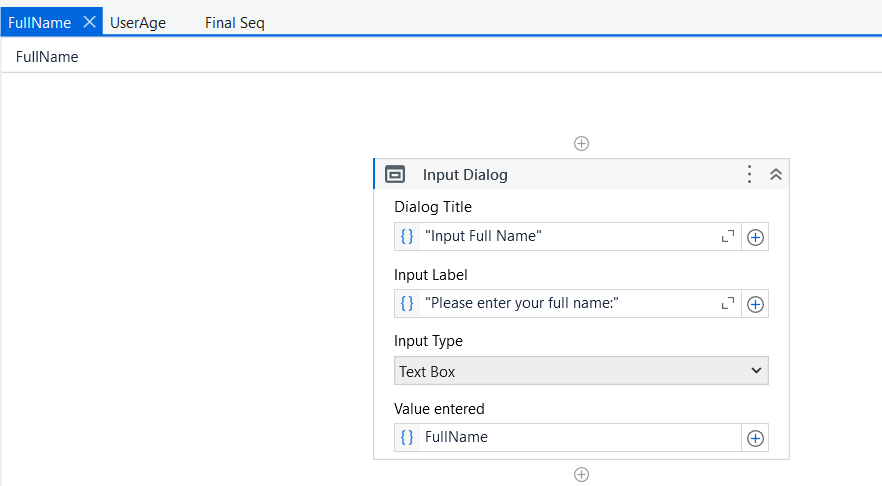
********

****

1. **Create three different sequences files:**

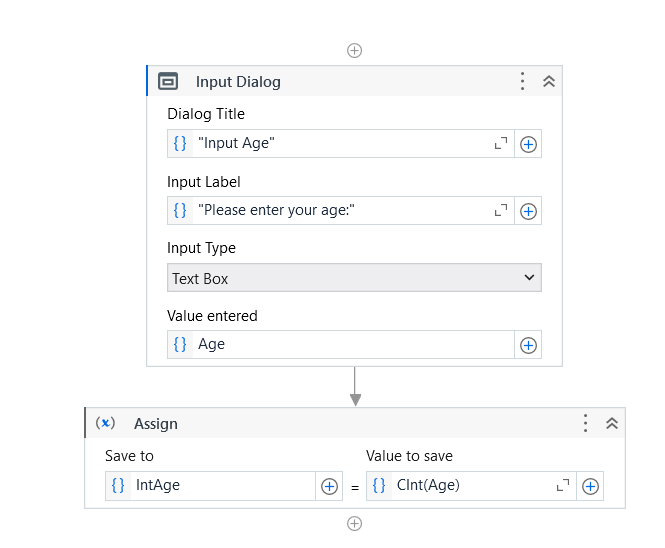
* **seq file 1 - Ask the user to input Full Name**
* **seq file 2 - Ask the user to input Age**
* **seq file 3 - Reuse the first two sequence files and print the first name of the user, also check if the user is an eligible voter or not.**

Sequence 1: Ask the User to Input Full Name (Seq 1)



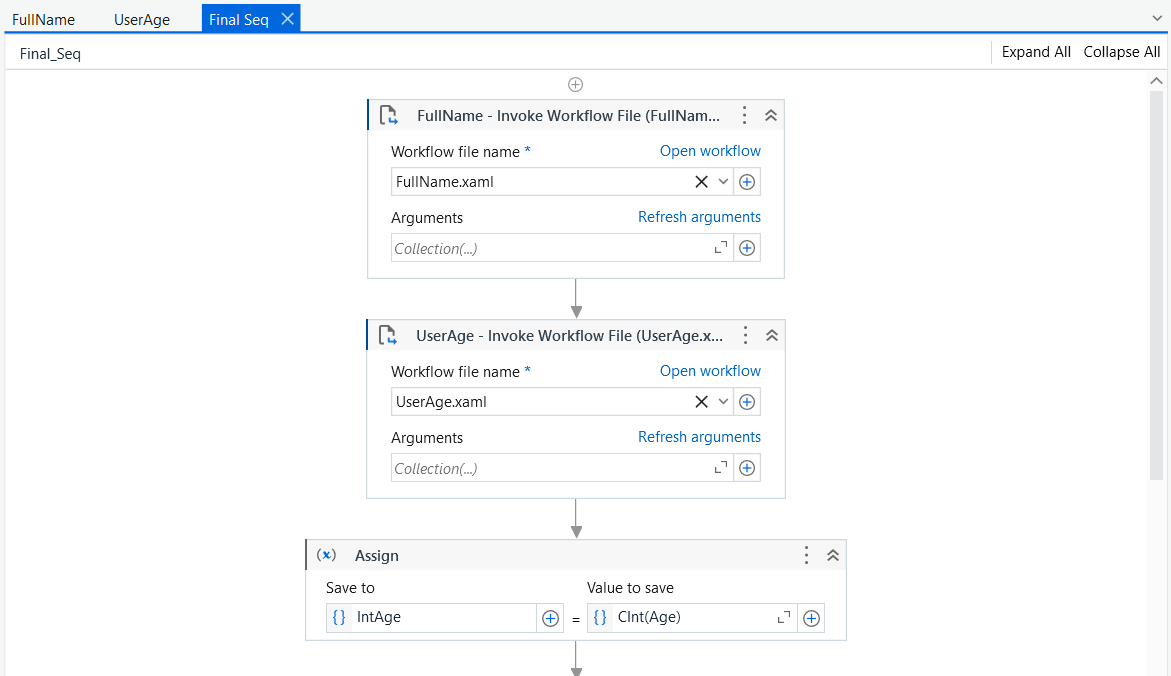
1. Open UiPath Studio and create a new process or project if you haven't already.
2. Create a New Sequence for asking the user’s full name.
3. Add an Input Dialog Activity:
   * In the Activities Panel, search for Input Dialog.
   * Drag Input Dialog into the sequence.
4. Configure the Input Dialog:
   * Title: Enter "Input Full Name".
   * Label: Enter "Please enter your full name:".
   * Result: Assign the output to a String variable (e.g., FullName).
5. Add a Message Box to display the output for testing.
   * Text: "Your Full Name is: " + FullName.
6. Publish or save this sequence.

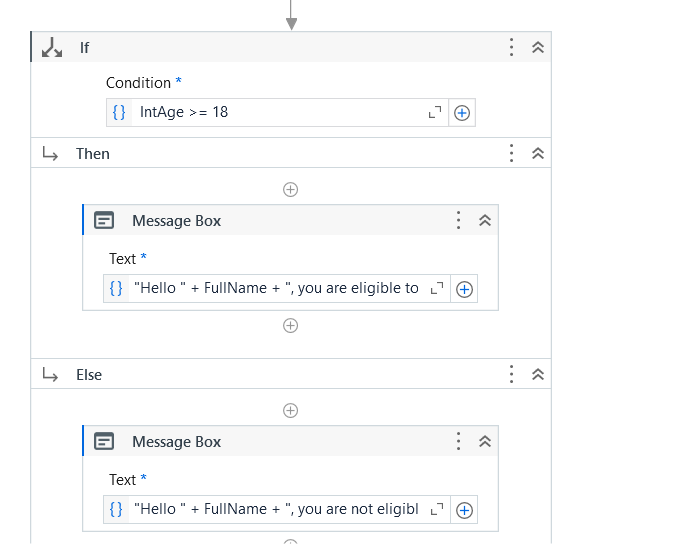
Sequence 2: Ask the User to Input Age (Seq 2)



1. Create a New Sequence for asking the user’s age.
2. Add an Input Dialog Activity:
   * Drag and configure it similarly to Seq 1.
3. Configure the Input Dialog:
   * Title: "Input Age".
   * Label: "Please enter your age:".
   * Result: Assign to a String variable (e.g., Age).
4. Convert the String Age to Integer:
   * After collecting the input, add an Assign activity below the Input Dialog:
   * To: IntAge (new integer variable)
   * Value: CInt(Age)
5. Add a Message Box to display the age for testing:
   * Text: "Your Age is: " + IntAge.ToString().
6. Publish or save this sequence.

Sequence 3: Reuse the First Two Sequences to Print Name and Check Voter Eligibility

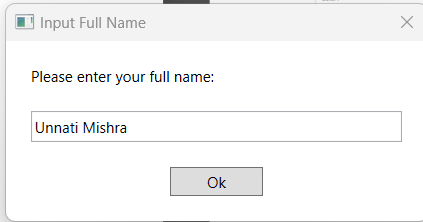


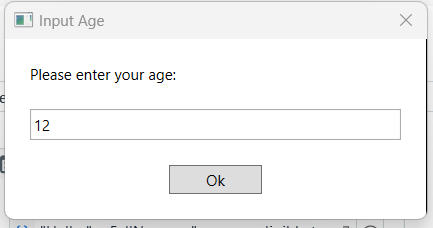


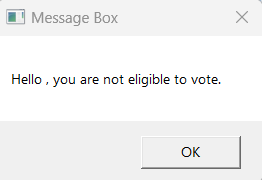
1. Create a New Sequence to combine both previous sequences.
2. Invoke the First Sequence (Seq 1 - Full Name):
   * Drag Invoke Workflow activity into this sequence.
   * Browse to select the previously created workflow for Seq 1.
   * Assign the result to the FullName variable.
3. Invoke the Second Sequence (Seq 2 - Age):
   * Drag another Invoke Workflow activity and select the Seq 2.
   * Assign the result to Age.
   * Convert Age to Integer:
     + Add an Assign activity to convert the Age string to an Integer:
       - To: IntAge
       - Value: CInt(Age).
4. Check Voter Eligibility:
   * Add an If Activity to check if the user is eligible to vote.
   * Condition: IntAge >= 18
5. Configure Message Box for Voting Eligibility:
   * In the Then section of the If activity:
     + Message: "Hello " + FullName + ", you are eligible to vote."
   * In the Else section:
     + Message: "Hello " + FullName + ", you are not eligible to vote."

Test the Process

1. Run the Sequences Individually:
   * Run Seq 1 and Seq 2 independently to make sure there are no errors in each.
2. Run the Combined Sequence (Seq 3):
   * Ensure the Full Name and Age are passed correctly between sequences.
   * Verify that the voter eligibility check works based on age.

****

****

****

**Activity 2**

1. **Download and save the attached excel at some location.**

[**Financial Sample**](https://geminisolutionsindpvtltd-my.sharepoint.com/:x:/g/personal/raja_arora_geminisolutions_com/ETg-NFj_gatKtWmwByyitOMB1qM_P7c--W6FPpDI8QIuqQ?e=dAIdPi)

1. **Open the excel and read the content.**
2. **Find the highest sale made by the country each year.**

**Example - In year 'YYYY' company 'xyz' made highest sale of $ 202.**

1. **Once you have identified the highest sale. Highlight the complete row in red color.**
2. **Find all the unique country name and Segment and store it in a separate excel Sheet.**

A. Download and Save the Excel

* Ensure the Excel file is located at the following path:

"C:\Users\unnat\Desktop\Re\_ Freshers -2025 Pre Training Material QA (Review)\UiPath Assignment\Financial Sample.xlsx"

(Since you’ve mentioned the file location, I assume it’s already in place.)

B. Open the Excel and Read the Content

1. Open UiPath Studio and create a new project.
2. Add Excel Application Scope:
   * Search for Excel Application Scope in the Activities Panel and drag it into your workflow.
3. Set the FilePath in Excel Application Scope:
   * In the Properties Panel, find the FilePath property.
   * Set the FilePath to your file location:

"C:\Users\unnat\Desktop\Re\_ Freshers -2025 Pre Training Material QA (Review)\UiPath Assignment\Financial Sample.xlsx"

1. Add Read Range Activity:
   * Inside the Excel Application Scope, drag the Read Range activity to read data from the Excel sheet.
   * In the Properties Panel of the Read Range:
     + SheetName: Specify the name of the sheet (or leave blank to default to the first sheet).
     + Range: Leave blank to read all data or specify a range (optional).
     + Output: Assign a DataTable variable (e.g., dtFinancial).

Example:

* + FilePath: "C:\Users\unnat\Desktop\Re\_ Freshers -2025 Pre Training Material QA (Review)\UiPath Assignment\Financial Sample.xlsx"
  + Output (DataTable): dtFinancial

C. Find the Highest Sale Made by Each Country Each Year

1. Create For Each Row Activity:
   * After the Read Range, drag the For Each Row activity to loop through each row in the DataTable (dtFinancial).
2. Initialize Variables:
   * Create these variables:
     + highestSale (Type: Decimal to store the maximum sale amount).
     + countryWithHighestSale (Type: String to store the country).
     + yearWithHighestSale (Type: String to store the year).
3. Process Each Row:
   * In the For Each Row activity, use the following logic to find the highest sale:

If Convert.ToDecimal(row("Sales")) > highestSale Then

highestSale = Convert.ToDecimal(row("Sales"))

countryWithHighestSale = row("Country").ToString()

yearWithHighestSale = row("Year").ToString()

End If

1. Display Results:
   * After the loop, use a Message Box or Log Message activity to display:

"In year " & yearWithHighestSale & " company " & countryWithHighestSale & " made highest sale of $ " & highestSale.ToString()

D. Highlight the Complete Row in Red Color Once You Have Identified the Highest Sale

1. Create Another Excel Application Scope to Edit the Excel File:
   * Drag another Excel Application Scope to modify the original file. Set the FilePath to the same file path used previously.
2. Highlight the Row:
   * Use For Each Row again, looping through dtFinancial.
   * When you find the row with the highest sale, use the Set Range Color activity to highlight the row in red. Here’s an example of the logic:

If row("Country").ToString() = countryWithHighestSale And row("Sales").ToString() = highestSale.ToString() Then

ExcelRange = "A" & rowIndex.ToString() & ":Z" & rowIndex.ToString()

SetRangeColor(ExcelRange, Color.Red)

End If

E. Find All Unique Country Names and Segments and Store Them in a Separate Excel Sheet

1. Extract Unique Values:
   * Use LINQ to get unique countries and segments:

Dim uniqueCountries = dtFinancial.AsEnumerable().Select(Function(row) row("Country").ToString()).Distinct().ToList()

Dim uniqueSegments = dtFinancial.AsEnumerable().Select(Function(row) row("Segment").ToString()).Distinct().ToList()

1. Create a New DataTable for Unique Values:
   * Create a new DataTable (dtUniqueCountriesSegments).
   * Add two columns for Country and Segment.
2. Populate the DataTable:
   * Use a For Each loop over uniqueCountries and uniqueSegments, adding the values to dtUniqueCountriesSegments.
3. Write Data to a New Excel Sheet:
   * Use Excel Application Scope to write dtUniqueCountriesSegments to a new sheet.
   * Inside Excel Application Scope, drag Write Range and set the Input as dtUniqueCountriesSegments.
   * You can specify the sheet name (e.g., "UniqueData").