
RPA - ROBOT PROCESS AUTOMATION (RPA) USING UIPATH - BEGINNER TO EXPERT

Section 1- Introduction to RPA and UIPATH

What is RPA? And its advantages in Software Industry

Robot Process Automation (RPA) using UIPATH

What is Robot Process Automation (RPA)?

- The process of automating day to day business operations with the help of robots to reduce human intervention is said to be Robot Process Automation (RPA).

What types of Work RPA can do?

- Extract Content from any documents like PDF's, word, Notepad, Excel or Email clients.
- Move files and folders in Operating System
- Can retrieve data from Web browsers with the Artificial Intelligence capabilities.
- Automate any desktop applications like SAP, Mainframes, Word etc.
- Automate any web applications
- Read and Write Databases

RPA robots can capture and manipulate any applications just like humans do.

RPA in Layman Terms

- Just like physical robots have been rapidly replacing industrial workers, software RPA robots will be replacing a high percentage of white-collar workers, sooner than you think.

How these Robots are created?

- There are RPA tools in the market which can design the virtual robots according to process requirements.

Where is RPA used in Business?

- Financial services
- Health Care
- Accounting
- Customer Service
- Human Resources
- Supply Chain Management

Popular RPA tools available in the market

- Uiopath
- Blue Prism
- Automation anywhere

Why UiPath is the best among all other RPA Tools?

<u>Features</u>	<u>UiPath</u>	<u>Blue Prism</u>	<u>Automation anywhere</u>
How will I practice?	Has a community Edition/Free Edition Available	No trial version available, have to purchase this tool to learn	Trial version available, but expires after 30 days
Learning Curve	Has a user friendly visual designer, hence easy to learn and implement	Has a visual designer, easier than Automation Anywhere	Developer Friendly – Requires Higher programming skills
Google Trends Popularity	Most popular tool	More Popular than Automation Anywhere	Least popular of the three

Link: <https://www.besanttechnologies.com/rpa-vs-selenium>

Instructions in Installing UIPATH Community Edition

- Download uipath and install the community edition
- Link: <https://www.uipath.com/>

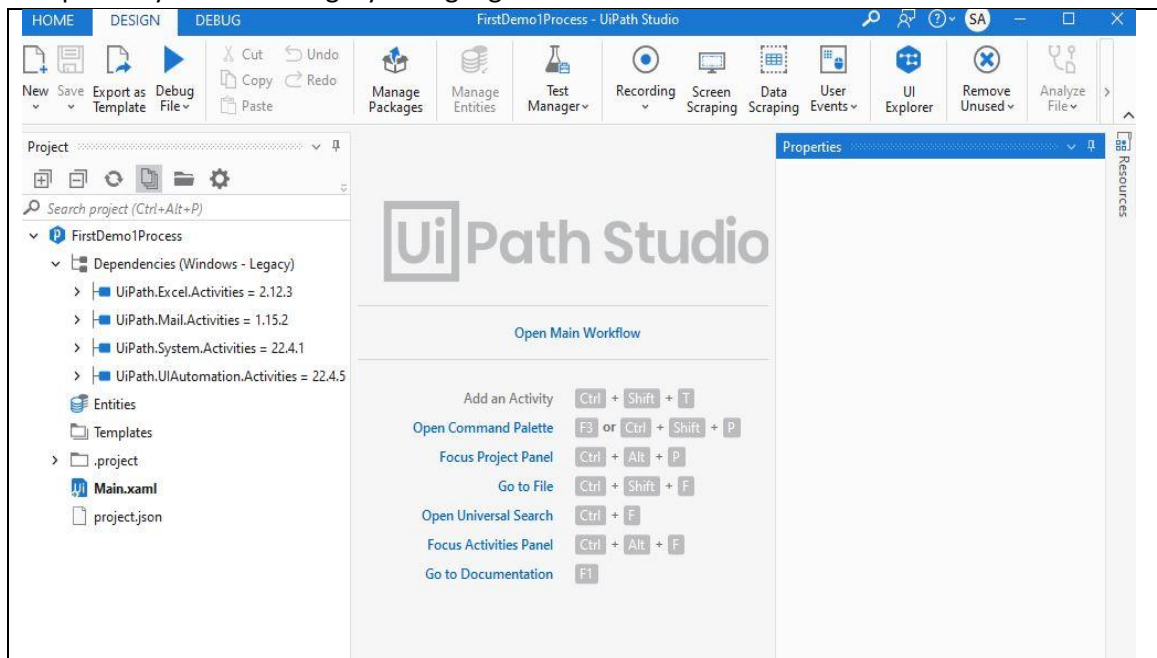
Section 2- Getting Started with UiPath features and Recording Modes

Creating blank Process in UiPath for automation

Studio Terminologies:

In the uipath go to settings->select Design->Use Modern for new projects->select No->select Start

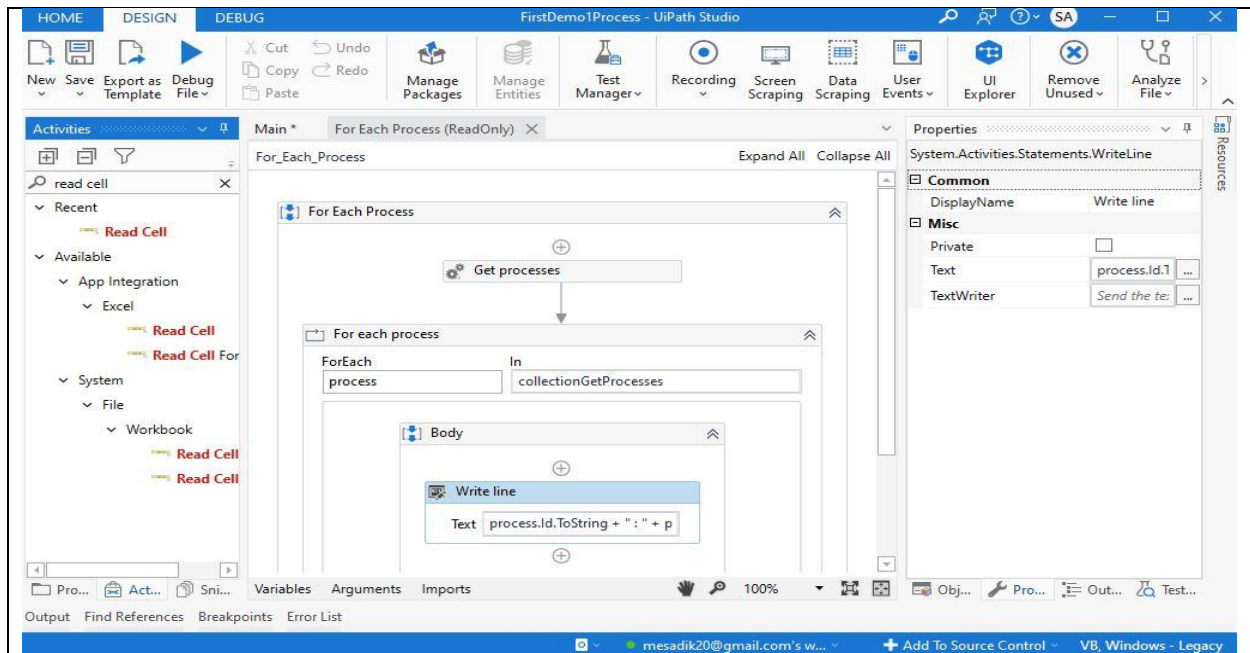
- **Process:** Project/workspace which stores all files.
Select process->Name: **FirstDemo1Process**->give some Description
Compatibility: Windows Legacy->Language: VB->select create



- **Dependencies:** Libraries to support Automation.
- **Xaml:** Design Automation Workflow
- **Project.json:** Project Information and metadata to publish
- **Activities:** uipath core packages for Designing workflow
- **Properties:** Properties list for selected Activity
- **Outline:** Output console to log Workflow results
- **Ribbon Menu:** uipath tool features

Understand UI features of UiPath and their significance

- **Activities:** uipath core packages for Designing workflow
To use a activities-> select activities ->search the activities and drag it and drop to the Workflow area



Recording Desktop application activities using UiPath

Different options for Recording:

- **Basic:** It will work on any app. It support all basic apps but not in depth, which is simple flow.

To create workflow of calculator

Select Basic->click on the desktop

Application Path: C:\WINDOWS\Explorer.EXE

Give Arguments: e.g. C:\Windows\System32\calc.exe

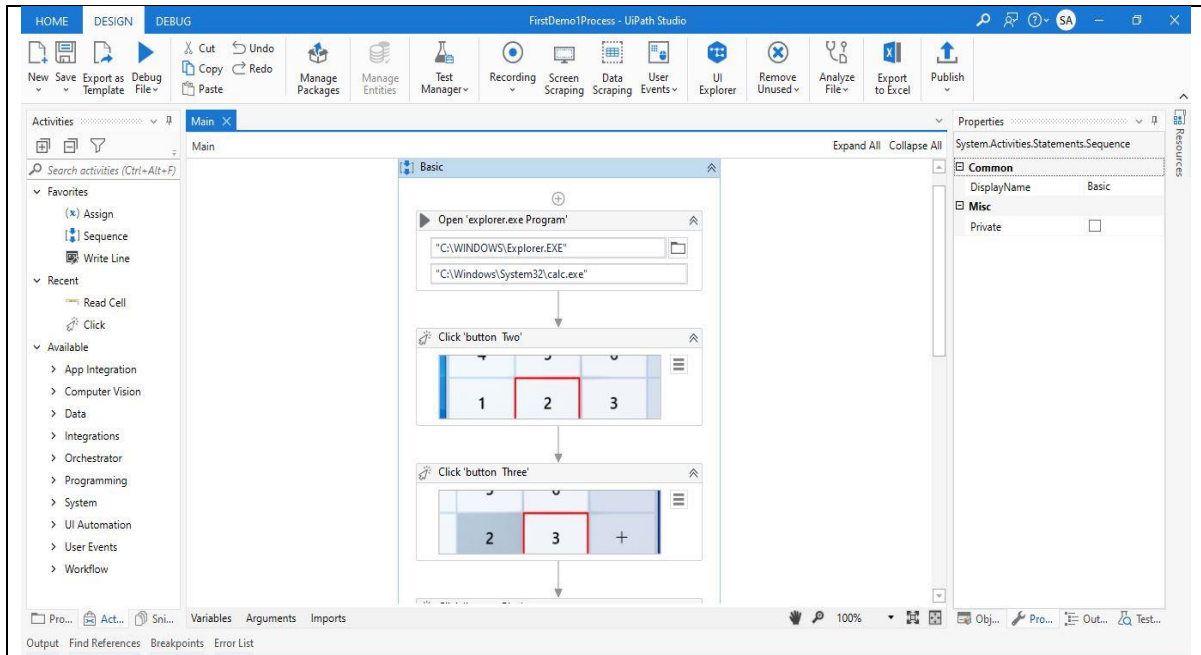
Select OK

Open the calculator->Select Record->do a calculation in the calculator->close the calculator

->give esc to come out of recording mode->select save and exit->it will save and create a workflow

To run the workflow->select dropdown Debug file->select Run File

- **Desktop:** It is for Desktop
- **Web:** It is for Web
- **Image:** It is for Image
- **Native Citrix:**
- **Computer Vision:** It will not have any properties. It will work on AI.



Understanding Sequence and flow chart workflow

- Select Main.xaml file->double click Open explorer.exe Program
->Here all are Activities like click button, click application. Which is used to build the calculator.
- To create another Activities for notepad application, creating another **.xaml** file.
Open notepad and type how are you? And then close it
- In the Uipath studio->select project->right click on the project name->select Add
->select Flowchart->select create
- There are two type of workflow: **flowchart** and **sequence**
- In flowchart workflow we can have both sequence and flow chart types.
But in sequence we can't have both so we can't have flow chart.
- Copy the sequence in **Main.xaml** file->and paste it in Flowchart->rename the sequence to Calculator->then connect the flowchart with the sequence
- In the flowchart workflow, multiple sequence can be run.

To create workflow of notepad

Select Basic->click on the desktop

Application Path: C:\WINDOWS\Explorer.EXE

Give Arguments: e.g. C:\Windows\System32\ notepad.exe

Select OK

Open the notepad->Select Record-> Open notepad, click on it and type: How are you? Then hit enter. Then in the notepad select format, select font, select 72 size and select Narrow Bold italic and click ok and close the notepad.

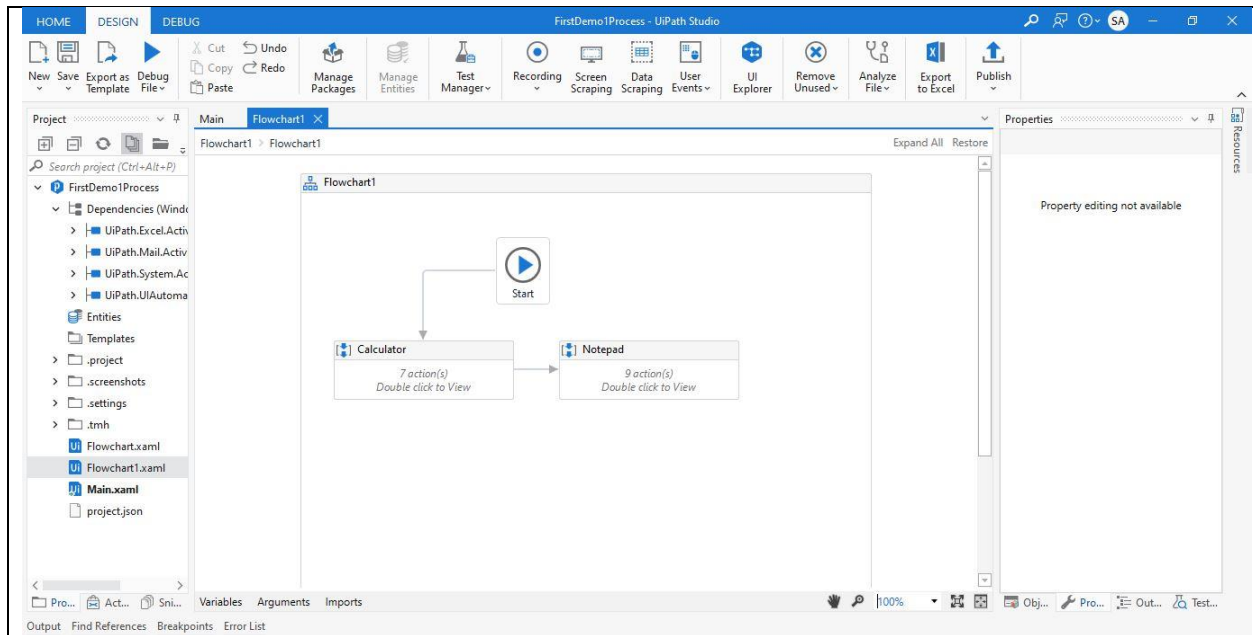
->give esc to come out of recording mode->select save and exit->it will save and create a workflow

Join the Start with Notepad to run the workflow of notepad

To run both Calculator and Notepad. First join Start with Calculator. Then join Calculator with Notepad.

To run the workflow->select dropdown Debug file->select Run File

First it will run workflow of calculator then will run workflow of notepad



Section 3- Code Download Instructions

Download Code used in the tutorial

How to import an automation workflow project into the UiPath project:

- Extract the zip project file which you want to import
- Open uipath->select **Open a Local Project**->choose the unzip project->inside the unzip project
->select **project.json**->import the **.json** file into the uipath->automatically all the project content
will be shown in the created workspace

Section 4- Understand UiPath Activities to build workflowsRobots

Automating Notepad and word Doc Activities using UiPath

Steps:

- Open notepad and type some text.
- Copy the text and save the file with any name then close the notepad
- Open word document.
- Paste the copied text from notepad to word document

Workflow:

To create workflow of Notepad to Word Copy

To get sequence->in the uipath created project->select Activities->search sequence ->select the sequence which is under Control->drag the sequence and place it on the flowchart

->rename Sequence name: Notepad to Word Copy

To focus only one on a sequence name: Notepad to Word Copy ->double click on it

->select Activities->search Open Application ->select the Open Application ->drag Open Application and place on the flowchart->place Open Application on sequence name: Notepad to Word Copy above the Drop Activity here->Double click Open Application ->make sure open the notepad on the background before clicking "indicate window on screen"-> after clicking "indicate window on screen" click on the notepad which was opened

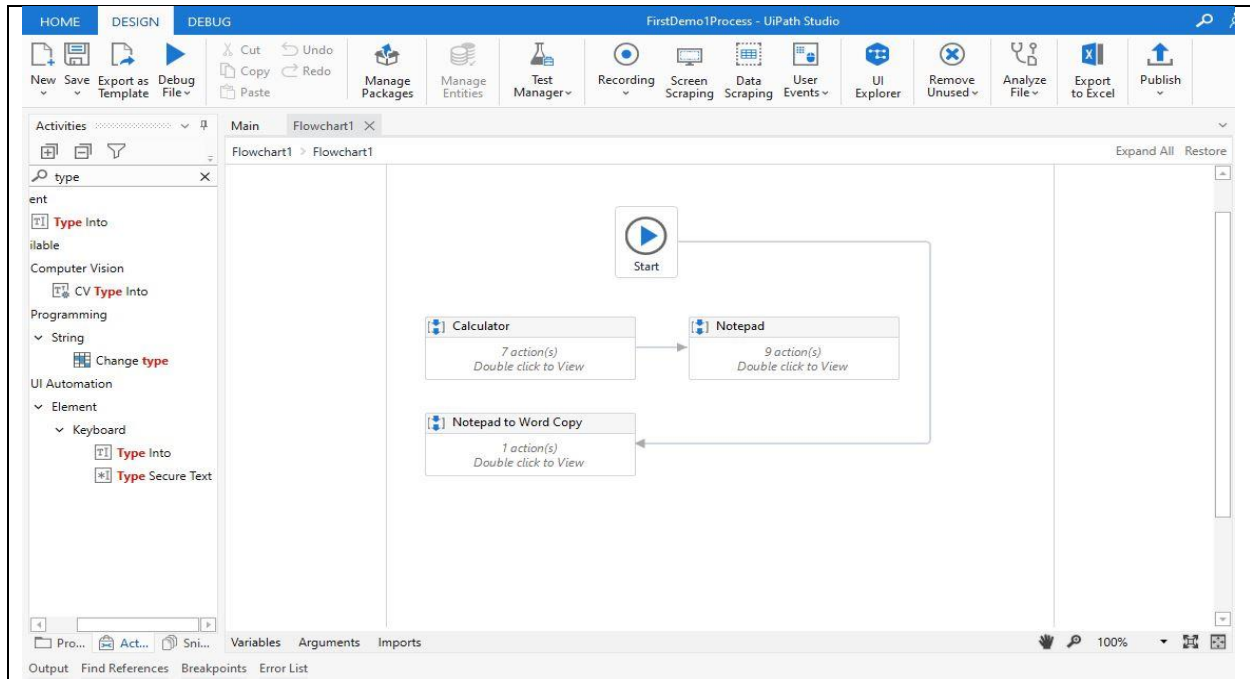
->select Activities->search "Type" ->select the Type Into ->drag Type Into and place on the flowchart Do "Drop Activity here"

->Double click Type into-> click "indicate window on screen"-> after clicking "indicate window on screen" click on the notepad which was opened without any typed text on the notepad

->in the uipath select text must be quoted-> type "UiPath learning"

->close the notepad which was open in the background

To run the workflow First join with Start then select dropdown Debug file->select Run File



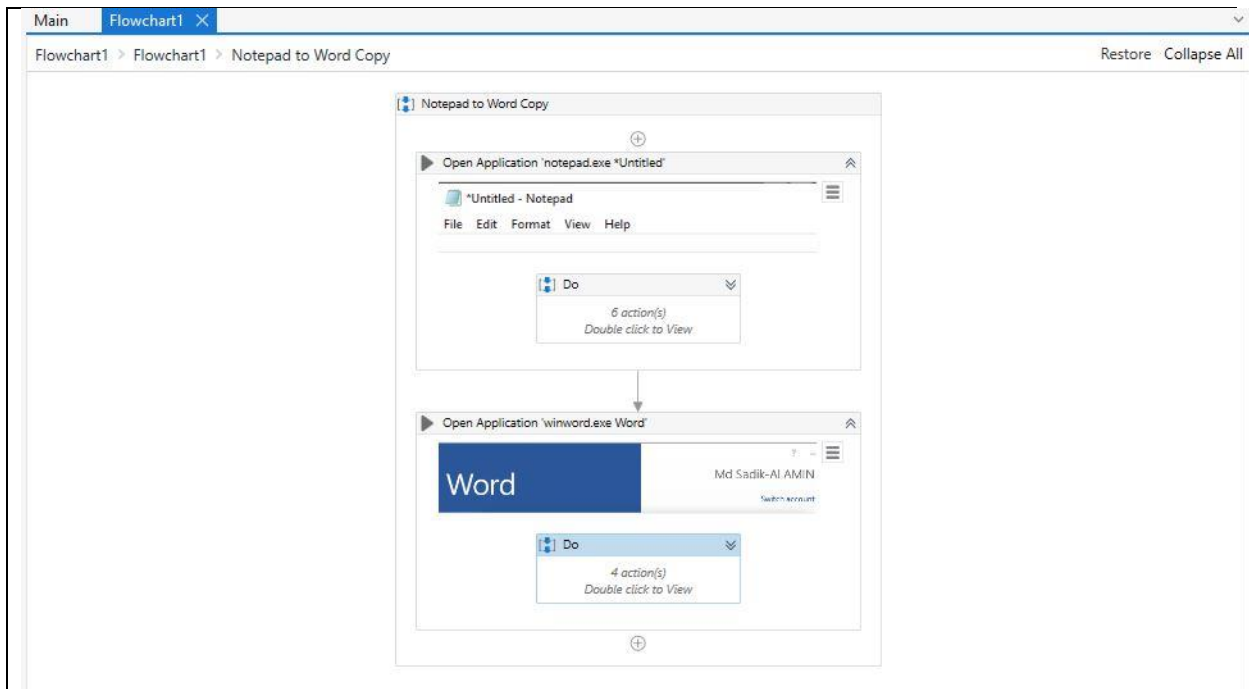
Finishing the Robot which can work between Notepad and Word Content

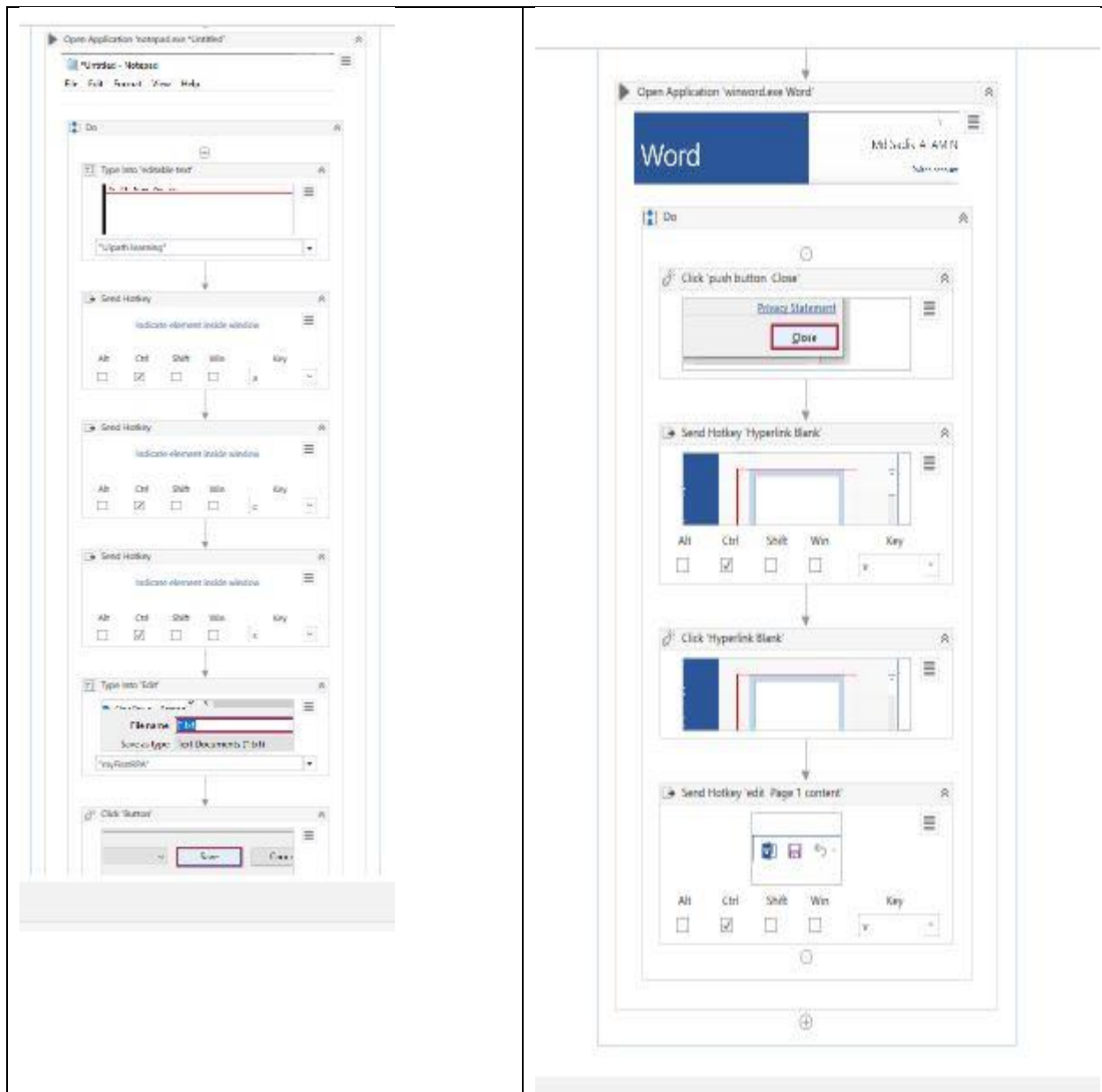
- Copy the text and save the file with any name then close the notepad
- Open the word document.
- Paste the copied text from notepad to word document

Copy the text, save the file then Paste the copied text to word document

- Normally to select all text : **ctrl + a**
->select + ->search "hotkey" ->select the Send Hotkey ->it will be automatically place on the flowchart Do "Drop Activity here"->click **ctrl** and key: **a**
-> for delay: shift + f2
- Normally to copy all text: **ctrl + c**
->select + ->search "hotkey" ->select the Send Hotkey ->it will be automatically place on the flowchart Do "Drop Activity here"->click **ctrl** and key: **c**
- Normally to save: **ctrl + s**
->select + ->search "hotkey" ->select the Send Hotkey ->it will be automatically place on the flowchart Do "Drop Activity here"->click **ctrl** and key: **c**
- ->select + ->search "**hotkey**" ->select the Send Hotkey ->it will be automatically place on the flowchart Do "Drop Activity here"->click **ctrl** and key: **a**
->Double click Type into-> click "indicate window on screen"-> after clicking "indicate window on screen" click on the location to type

- > type the file name in quotes: "myFirstRPA"
 - >to click on the save->select Activities->search "click" ->select the Click under mouse ->drag click and place on the flowchart Do "Drop Activity here"
 - > Double click click-> click "indicate window on screen"-> after clicking "indicate window on screen" click on the opened notepad save button
 - To close application
 - >select + ->search "**close application**" ->select the close application ->it will be automatically place on the flowchart Do "Drop Activity here"
 - >Double click close application -> click "indicate window on screen"-> after clicking "indicate window on screen" click on the location which to close
 - To open any application select Open Application activity.
 - >minimize the notepad open application
 - >select + outside the notepad open application->search "Open Application" ->select the Open Application ->it will be automatically place on the flowchart Do "Drop Activity here"
 - >rename: Open Word document
 - > To focus only one on a sequence name: Open Word document double click on it
 - >make sure to open the word document on the background before clicking "indicate window on screen"-> after clicking "indicate window on screen" click on the word document which was opened earlier
 - >select Activities->search "click" ->select the Click under mouse ->drag click and place on the flowchart Do "Drop Activity here"
 - > Double click click-> click "indicate window on screen"-> after clicking "indicate window on screen" click on the opened word document
 - Normally to paste: **ctrl + v**
 - >select + ->search "hotkey" ->select the Send Hotkey ->it will be automatically place on the flowchart Do "Drop Activity here"->click **ctrl** and key: **c**
 - > Double click click-> click "indicate window on screen"-> after clicking "indicate window on screen" click on the opened word document
 - >close the notepad and word document which was open in the background
- To run the workflow First join with Start then select dropdown Debug file->select Run File





Section 5- Screen Scrapping Techniques to extract text from ImagesWebDocuments

Example to Demonstrate Get Text Activity from the Browsers

Data Scrapping Techniques:

- Get text: Grab text from specific location
- Get OCR text: to extract or grab text from images like scanned copies
- Get Full text: Gets complete text from application
- Get visible text: activity is use for Native text. It will grab text which is only visible

Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**
->select add->select Flowchart->Name: **DataScrapping**->select create

Select Activity->search sequence->drag and drop sequence to the flowchart->double click sequence

->Rename: getText from element ->select + drop activity here

Select Activity->search **open browser**->drag and drop open browser to the + drop activity here

In the open browser properties->Input type->select BrowserType: **Chrome**

In the open browser->insert the url in quoted

Select Activity->search **get text**->drag and drop get text to the + drop activity here

Before selecting Indicate element inside browser ->install the uipath browsers plugin

To install uipath plugin->open uipath studio->select Home->select Tools->install the uipath browsers plugin extensions

To detect the uipath browsers ->in uipath studio->select **UI Explorer**->select options->select UI Frameworks->select **UI Automation**

->Select Indicate element inside browser->in the browser select the text which you want to grab

->Select Activity->search **message box**->drag and drop **message box** to + Add Activity

Select Get text container->the grab text is store in the Properties->**Output** value variable

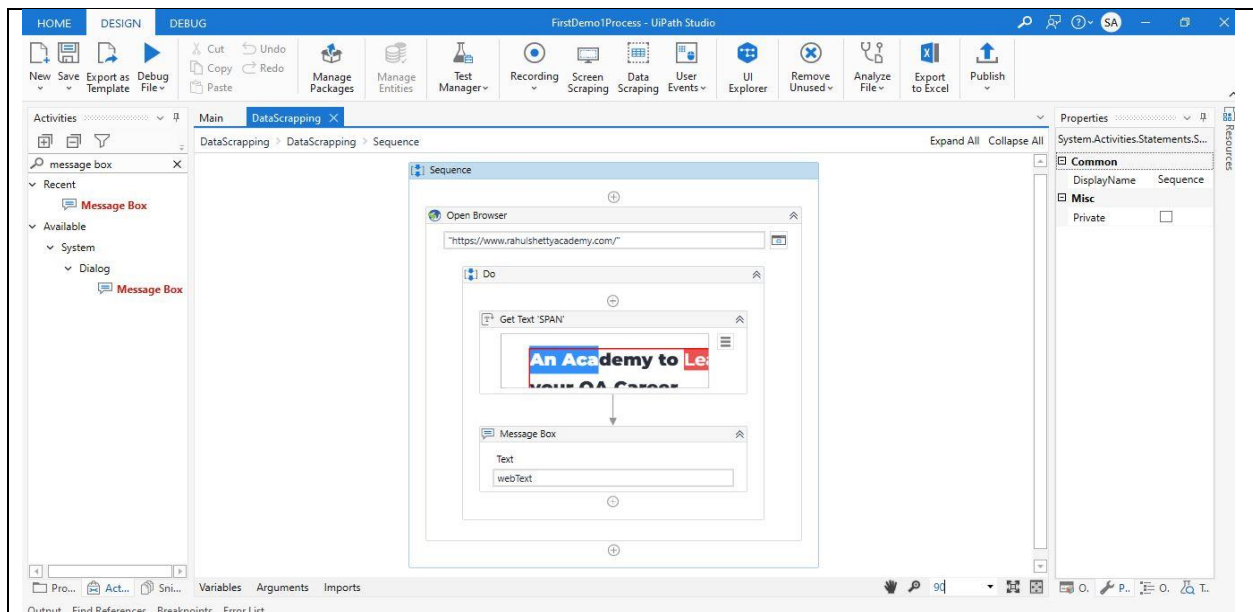
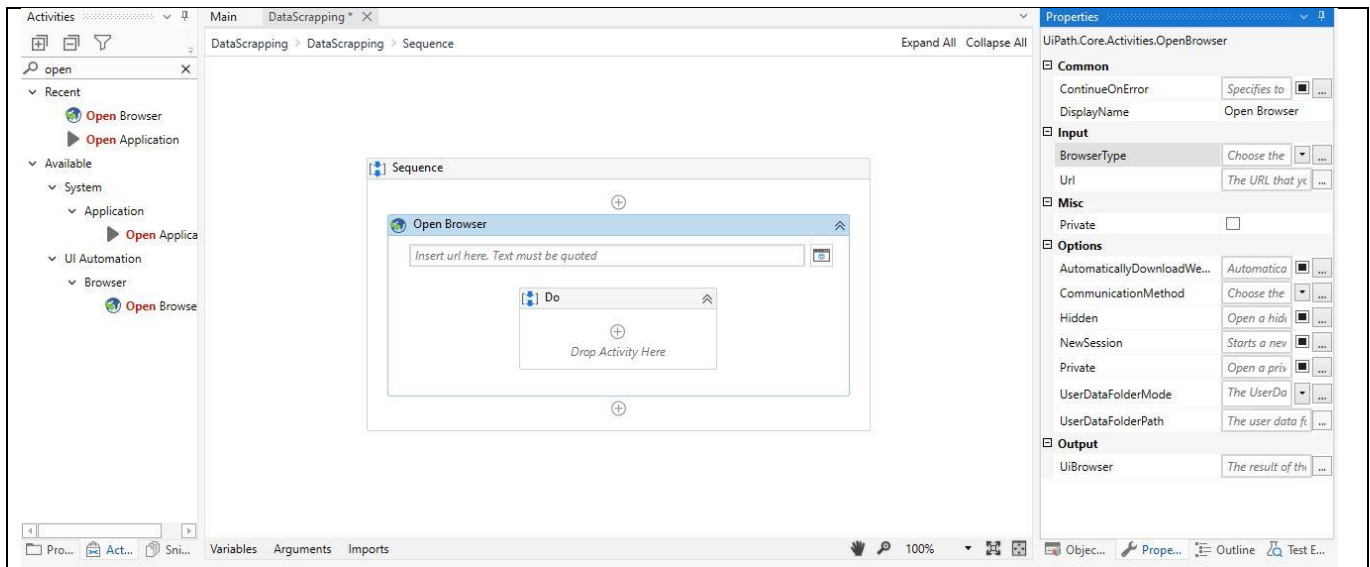
From the **Output** value variable->create a local variable to store the grab text

To create variable->select variable->select Create Variable->Name: **webText**, Variable type: **String**, Scope: **Do**

Again Select Get text container->the grab text is store in the Properties->give Output value: **webText**

In the flow chart->Select Message box->give the variable in Text: **webText**

To run the workflow First join with Start then select dropdown Debug file->select Run File



How to grab the text from Images using UiPath

Get OCR (Optical Character Recognition) text: to extract or grab text from images like scanned copies

Select Activity->search sequence->drag and drop sequence to the flowchart->double click sequence

->Rename: Get Text from Image ->select + drop activity here

Select Activity->search **OCR**->select Get OCR Text->drag and drop Get OCR Text to the + drop activity

here

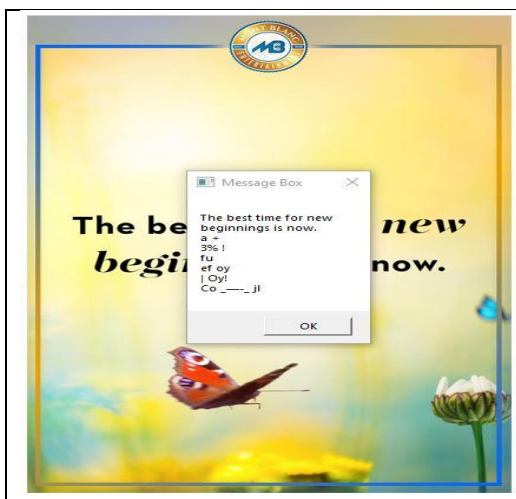
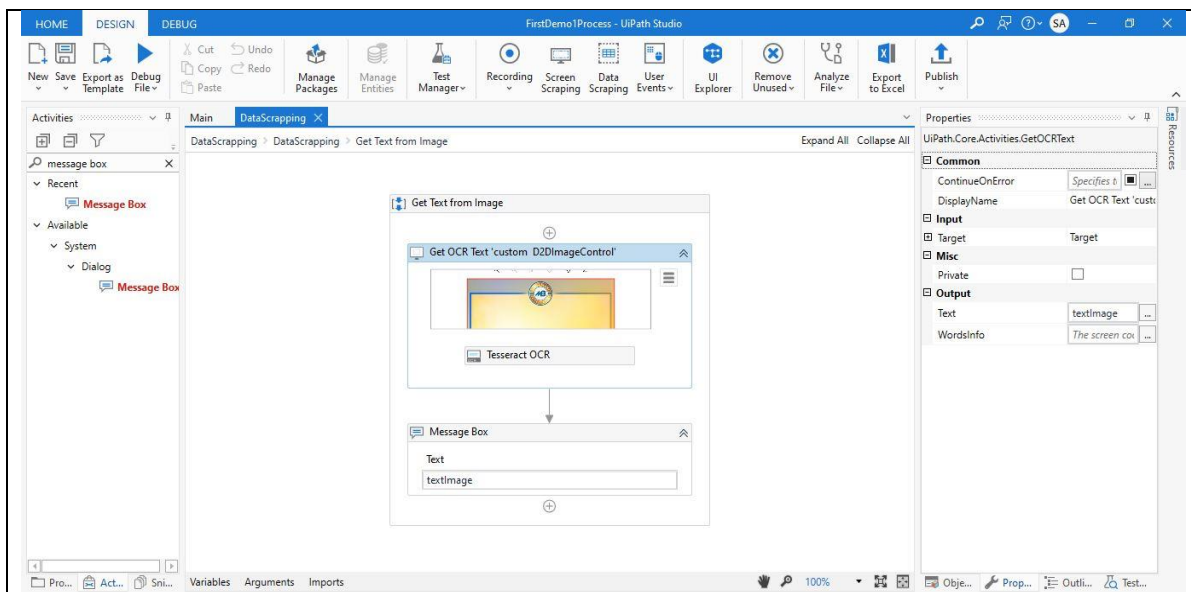
Open the image from where to grab text->in uipath flowchart->in Get OCR Text->select indicate on screen

Select Get OCR Text container->the grab text is store in the Properties->**Output** text variable

From the **Output** text variable->create a local variable to store the grab text
 To create variable->select variable->select Create Variable->Name: **textImage**, Variable type: **String**,
 Scope: **Get Text from Image**
 Again Select Get OCR Text container->the grab text is store in the Properties->give Output text:
textImage
 Select Activity->search **message box**->drag and drop **message box** to + Add Activity
 In the flow chart->Select Message box->give the variable in Text: **textImage**
 To run the workflow First join with Start->then Open the image from where to grab text
 ->select dropdown Debug file->select Run File

Note:

- Get OCR Text is GoogleOCR type use OCR engine to grab text from image
- To wait/pass for few seconds-> **shift + f2**



Example to grab the Text from Documents/Files

Screen Scraping:

Extract data from specific document using Fulltext methods

Open the notepad document from where to grab text->in the uipath->select **Screen Scraping**
->click on the notepad document from where to grab text->automatically a **Screen Scraper Wizard** will open->showing **Scrape Result Preview** of all the grab text

In the **Screen Scraper Wizard**->Scraping Method option have: **Native, FullText, OCR**

Native: Select Native-> select UI Element to scrape->select finish

It will not grab the entire text. It will grab the text which is only visible

FullText: Select FullText-> select UI Element to scrape->select finish

It will grab the entire text which is present in the document

OCR (Optical Character Recognition): Select OCR->select UI Element to scrape->select finish

It will extract or grab text from images like scanned copies

FullText: Select **FullText**-> select UI Element to scrape->select finish

->uipath will automatically create Screen Scraping sequence in the flowchart

->double click Screen Scraping sequence->Rename: FullText Screen Scraping

->The activity use for **FullText** scraping is-> Get Full Text

Whenever there is Attach Window->it will see already there is any opened application to scrape data.

Select Activity->search Fulltext->select Get full Text ->drag and drop to + Add Activity

Select Indicate element inside Window->select the already opened notepad

Select Activity->search write line->select **Write Line** ->drag and drop **Write Line** to + Add Activity

Select Get Full text container->the grab text is store in the Properties

->**Output** value variable: EditableText

In the flow chart->Select Write Line ->give the variable in Text: **EditableText**

To run the workflow First join with Start then select dropdown Debug file->select Run File

->The grab text is printed in the uipath output

Open the notepad document from where to grab text->in the uipath->select **Screen Scraping**

->click on the notepad document from where to grab text->automatically a **Screen Scraper Wizard** will open->showing **Scrape Result Preview** of all the grab text

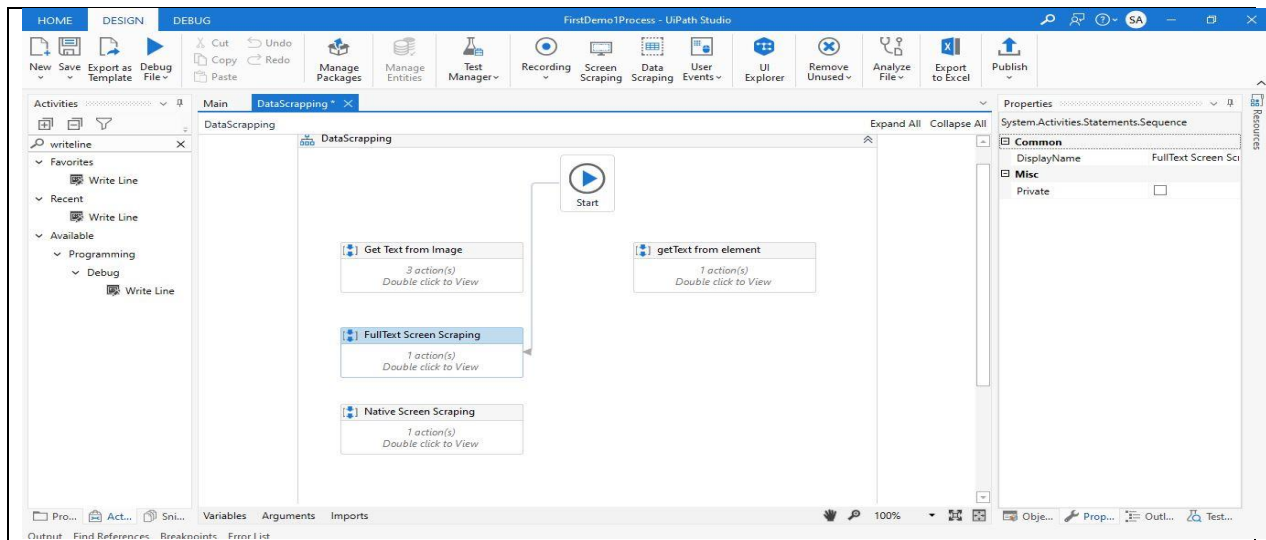
In the **Screen Scraper Wizard**-> Select **Native** -> select UI Element to scrape->select finish

->uipath will automatically create Screen Scraping sequence in the flowchart

->double click Screen Scraping sequence->Rename: Native Screen Scraping

->The activity use for **Native** scraping is-> Get Visible Text

Note: **Native** scraping method only works with apps that are built to render text with the Graphics Device Interface (GDI)



Section 6- Build a Robot which Checks Country GDP and suggest Business solutions

Demo - Understand the Business rules to Automate

Goal: I want to start Business in the Country If its Gross domestic product (GDP) is less than 10 Trillion

Business Solution: I will provide Country name and robot should tell me if I can invest there.

Business Rule:

Entry Country check GDP

If GDP>10 Trillion – Not Right place to start Business

If GDP<10 Trillion – Right place to start Business

Building the logic to get the GDP of country with User input

Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**

->select add->select Flowchart->Name: **GdpCheckRobot**->select create

Input Dialog activity: Display a dialog box that prompts the user with a label message and an input field

Select Activity->search input->select **Input Dialog**->drag and drop **Input Dialog** to the flowchart->double click on **Input Dialog** ->Give Dialog Title in double quote: "**GDP Check**"

->Give Input Label in double quote: "**Enter the Country Name**"

-> To run the workflow First join with Start then select dropdown Debug file->select Run File

->A dialog box will appear, where have to enter value->select OK



->Select Input Dialog container->the input value is store in the Properties->Output Result:

->to set the variable->ctrl + k -> Output Result: **country**

->select the **GdpCheckRobot** flowchart-> Select Activity->search write line->select **Write Line** ->drag and drop **Write Line** to the flowchart->join **Write Line** with **Input Dialog**

->select **Write line**->put the variable in Text: **country**

Select Activity->search **open browser**->drag and drop **open browser** and join with **Input Dialog**->double click **open browser**

In the **open browser** properties->Input type->select BrowserType: **Chrome**

In the open browser->insert the url in quoted

Select Activity->search **type into**->select **Type Into**->drag and drop **Type Into** to the + drop activity here

->Before selecting **Indicate element inside browser** ->install the uipath browsers plugin

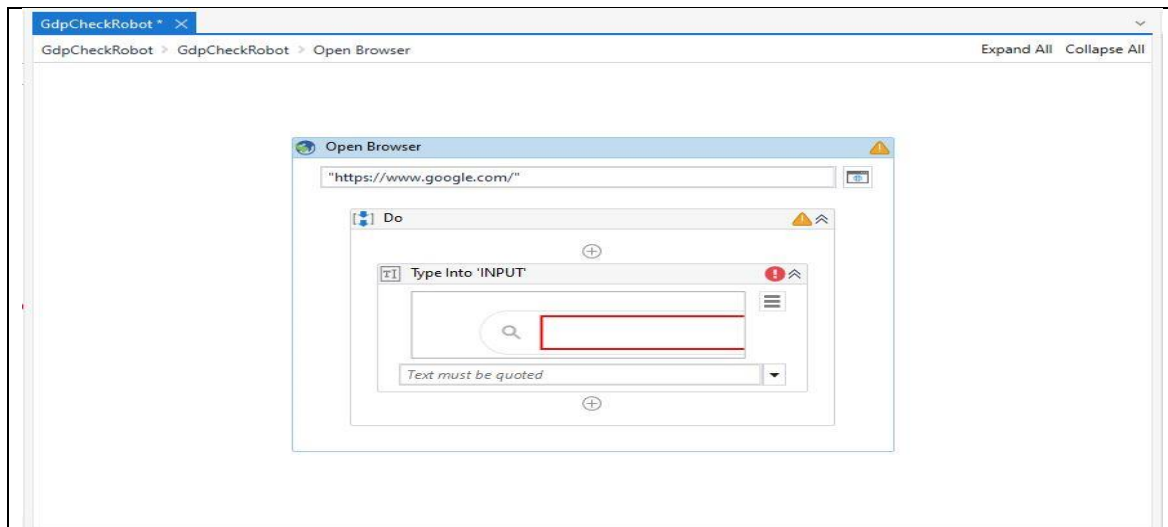
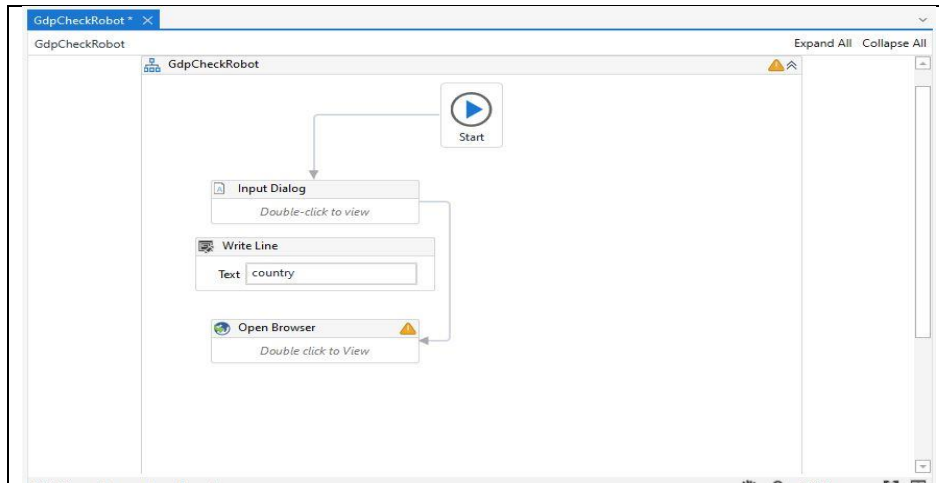
To install uipath plugin->open uipath studio->select Home->select Tools->install the uipath browsers plugin extensions

To detect the uipath browsers ->in uipath studio->select **UI Explorer**->select options->select UI Frameworks->select **UI Automation**

->Select Indicate element inside browser->in the browser select the location where to type

->in the Type Into container->select Click to open options menu->select Edit Selector

->select **Highlight**->to know the selector of that specific webpage



Perform Keyboard Actions and provide appropriate locators to identify the object

To replace the existing locator->in the **Type Into** container->select Click to open options menu->select **Indicate on screen**->then select the location to get the locator

->to check the locator element-> in the **Type Into** container->select Click to open options menu

->select Edit Selector->select **Highlight**

->select Type Into->in Text must be quoted: **country+" GDP"** ->country is input variable and GDP is a constant string. This will be typed in the edit box

Select Activity->search hot key->select **Send Hotkey**->drag and drop **Send Hotkey** to the + drop activity here->select Key: **enter**

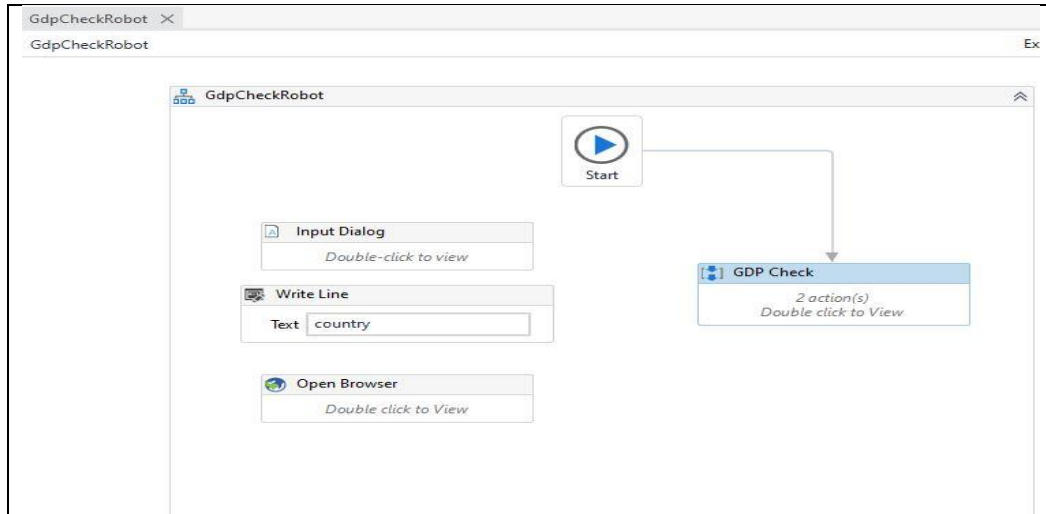
To key the Input Dialog and Open Browser in one sequence-> Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: **GDP Check**

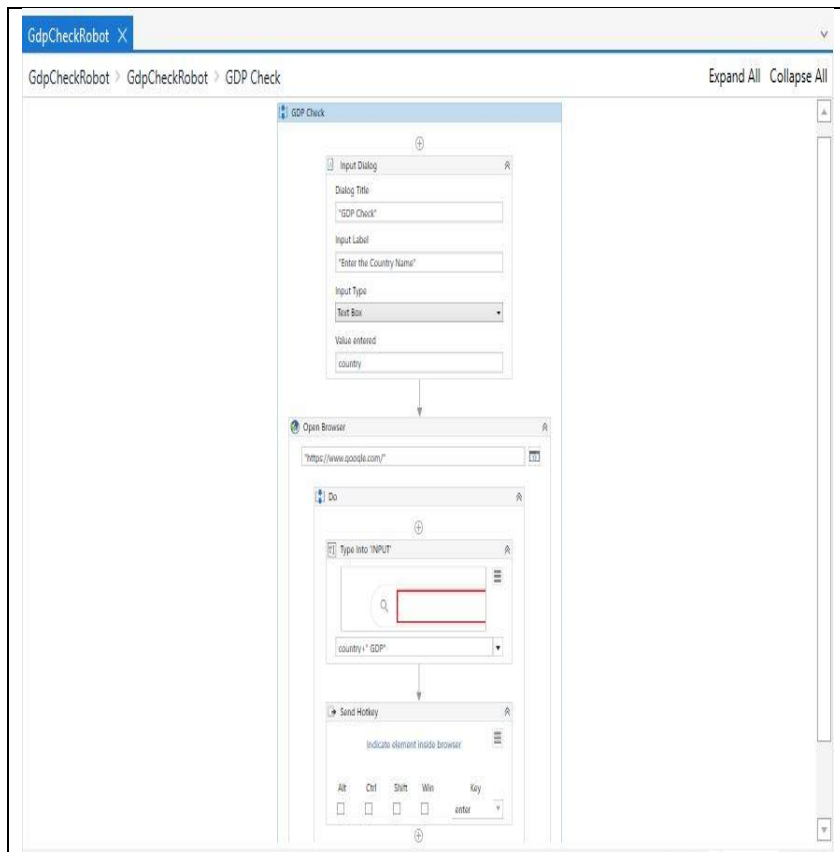
->In the sequence copy Input Dialog to the + drop activity here

->In the sequence copy Open Browser to the + drop activity here

->select variable->Scope: **GDP Check**

To run the workflow First join with Start then select dropdown Debug file->select Run File





Building Regular expressions from the Web properties to make them generic

Select Activity->search **get text** ->select **Get Text**->drag and drop **Get Text** to the + drop activity here
->select Indicate element inside browser->select the element location in the browser which was opened earlier

Select Get text container->the grab text is store in the Properties->**Output** value variable

From the **Output** value variable->create a local variable to store the grab text

To create variable->select variable->select Create Variable->Name: **gdpValue**, Variable type: **String**, Scope: **Do**

Again Select Get text container->the grab text is store in the Properties->give Output value: **gdpValue**

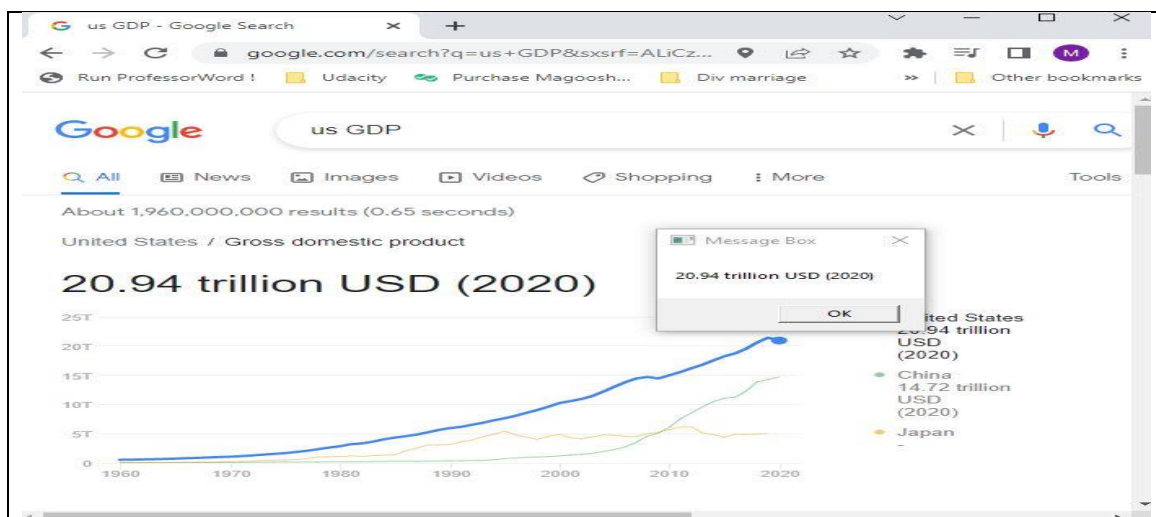
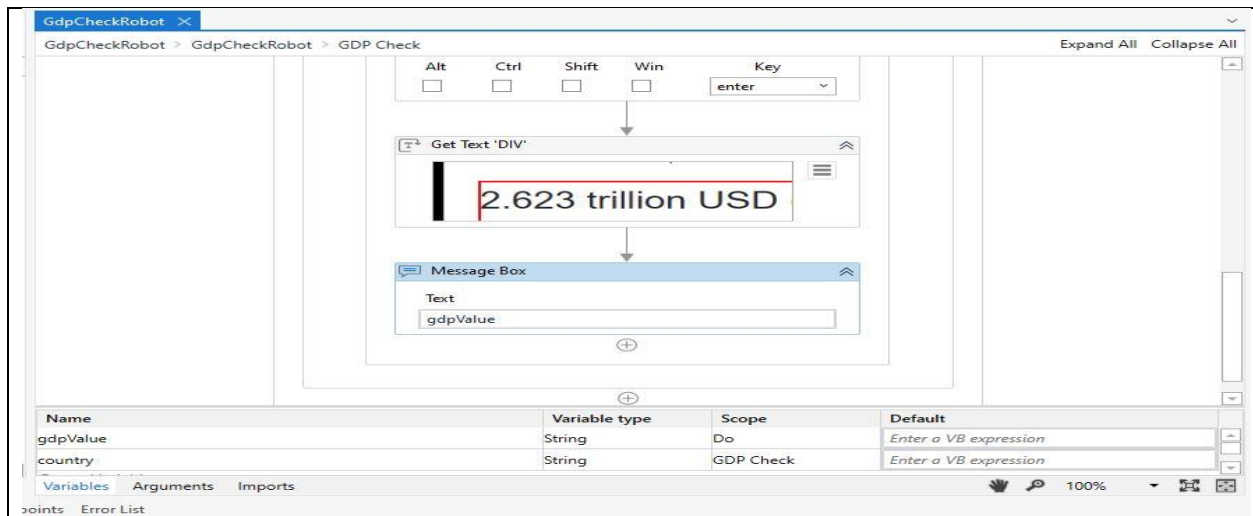
->in the Get text container->select Click to open options menu->select Edit Selector ->it shows **Highlight**

->search different country GDP in google-> in the Get text container->select Click to open options menu->select Edit Selector->select **Repair**->then select the new element location, which will update the new element location->select OK

->Select Activity->search **message box**->drag and drop **message box** to + Add Activity

In the flow chart->Select Message box->give the variable in Text: **gdpValue**

To run the workflow First join with Start then select dropdown Debug file->select Run File



Get The GDP Text from image using Optical Character Recognition?

- >Select Activity->search **OCR Text**->select **Get OCR Text**->drag and drop **Get OCR Text** between **Message Box** and **Get Text**
- >select **Get OCR Text**->select **Indicate element inside browser**->click and drag only portion of the text which is needed in the browser which was opened from earlier->it only extract the portion of text which was drag and selected
- >select **Get OCR Text**-> the grab text is store in the Properties->in the Output Text press **ctrl + k**
- > Output Text: **gdpActualNumber** ->then give enter
- >Select Activity->search **message box**->drag and drop **message box** to + Add Activity
- In the flow chart->Select Message box->give the variable in Text: **gdpActualNumber**

Business Rule:

Entry Country check GDP

If GDP>10 Trillion – Not Right place to start Business

If GDP<10 Trillion – Right place to start Business

Come out of GDP check by selecting GdpCheckRobot

-> Select Activity->search decision->select **Flow Decision**->drag and drop **Flow Decision** to the flowchart

->when select **Flow Decision** it shows two condition **True** and **False**

->For the variable **gdpActualNumber** in GDP Check->change the scope to: **GdpCheckRobot**

->select **Flow Decision** ->in the properties->give condition: **gdpActualNumber>10**

->select Expression Editor: **convert.ToDouble(gdpActualNumber)>10**->select OK

->join **GdpCheckRobot** sequence container with **Flow Decision**

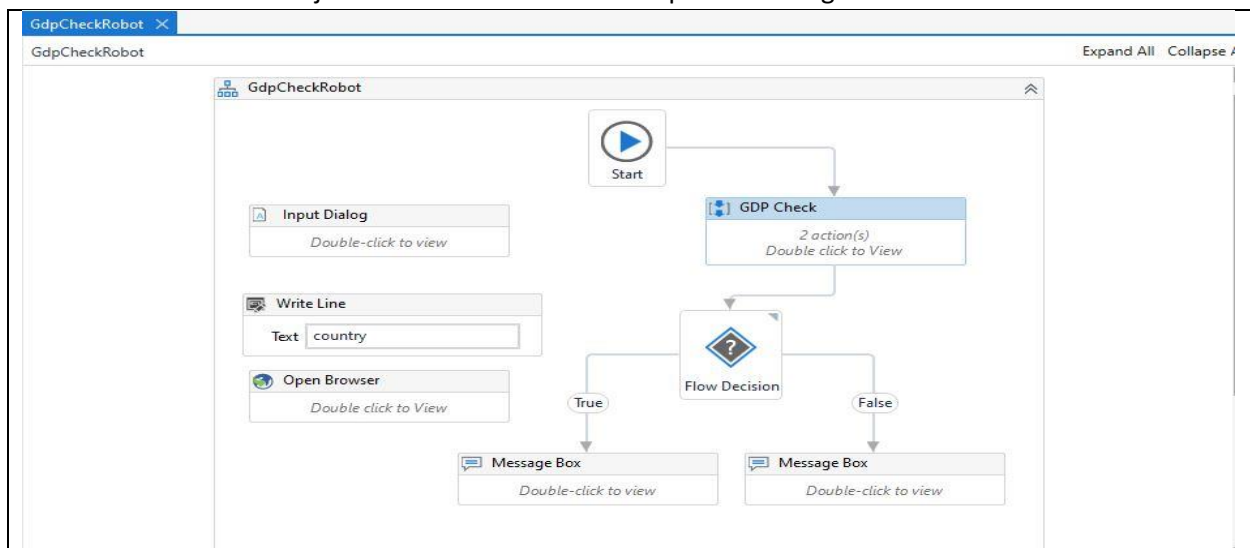
-> Select Activity->search **message box** ->select **message box** ->drag and drop **message box** to the flowchart->from the **Flow Decision** join the **True** with the **message box**->double click **message box**

->in the text: "Not Right place to start Business"

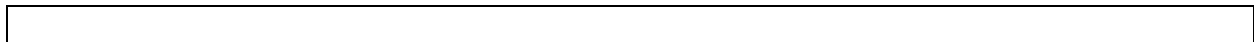
-> Select Activity->search **message box** ->select **message box** ->drag and drop **message box** to the flowchart->from the **Flow Decision** join the **False** with the **message box**->double click **message box**

->in the text: "Right place to start Business"

To run the workflow First join with Start then select dropdown Debug file->select Run File



Running the Robot with Successful Business logic Implementation



Section 7- UiPath Excel Activities in Depth to build Robots

How to read and write cell from excel using UiPath

UiPath Excel activities:

- Read cell
- Write cell
- Read range with Specific index
- Read range on full sheet
- Iterating over Data table using for each row

Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**
->select add->select Flowchart->Name: **Excelrobot**->select create

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: **Excel Basics**

Select Activity->search excel ->select **Excel Application Scope**->drag and drop **Excel Application Scope** to the + drop activity here->give Workbook path: "location of the excel file path"

Select Activity->search read cell ->select **Read Cell**->drag and drop **Read Cell** to the + drop activity here

->in the **Read Cell**->give the sheet number and cell number which to read

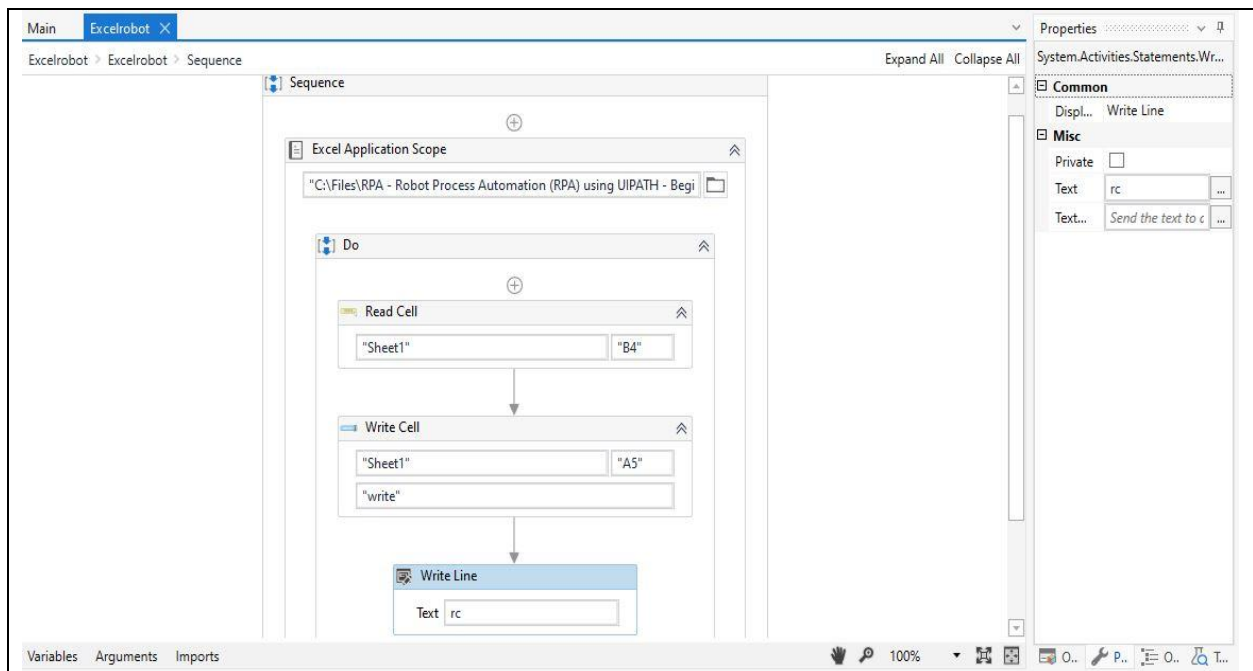
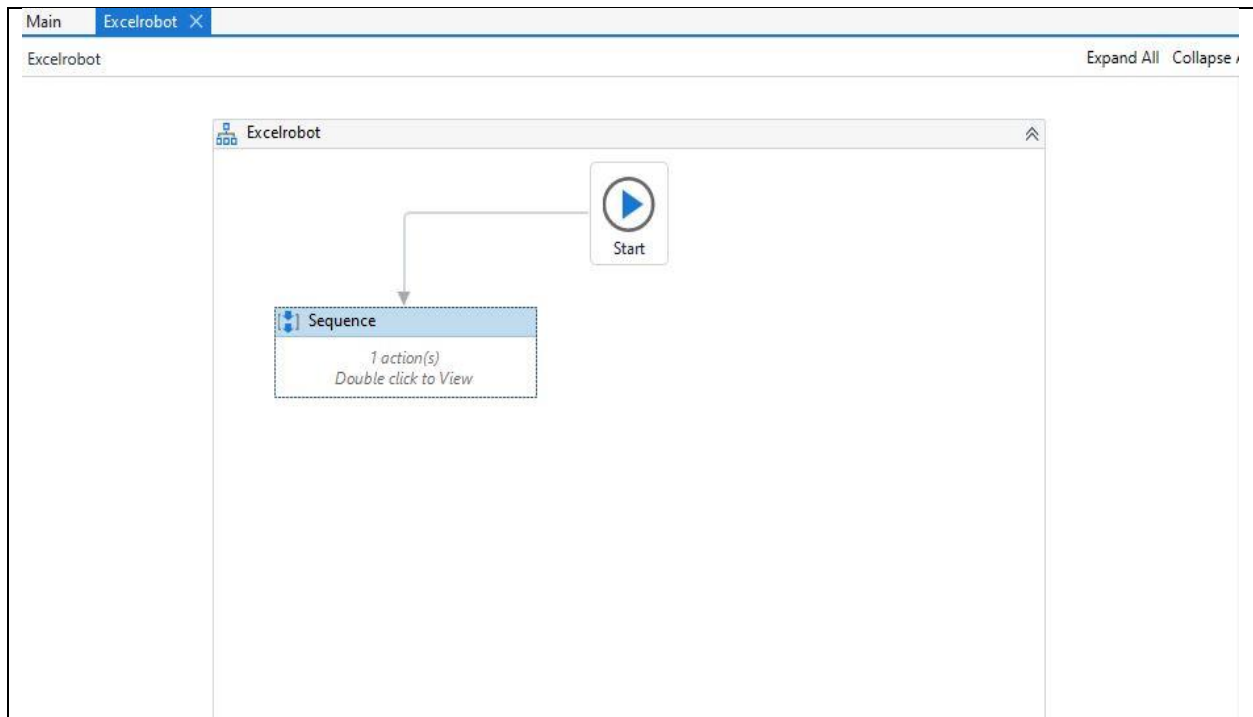
-> Select **Read Cell** container->the value in the cell number is store in the Properties->Output Result variable press **ctrl + k**-> Output Result: rc ->then give enter

Select Activity->search write line->select **Write Line**->drag and drop **Write Line** to the + drop activity here

->select **Write line**->put the variable in Text: rc

Select Activity->search write cell->select **Write Cell**->drag and drop **Write Cell** to the + drop activity here between **Read Cell** and **Write Line**->give the sheet number, cell number, text which to write

To run the workflow First join with Start then select dropdown Debug file->select Run File



Read range with given boundaries in excel using Output Data Table

Read range is how much range it have to read

Select Activity->search **read range**->select **Read Range**-> drag and drop **Read Range** to the + drop activity here between **Write Cell** and **Write Line**->give the sheet number and range of cell number which to read

Select **Read Range** container->the value in the cell number is store in the Properties->Output Data variable press **ctrl + k**-> Output Data: **rr** ->then give enter

Select Activity->search **write line**->select **Write Line**->drag and drop **Write Line** to the + drop activity here

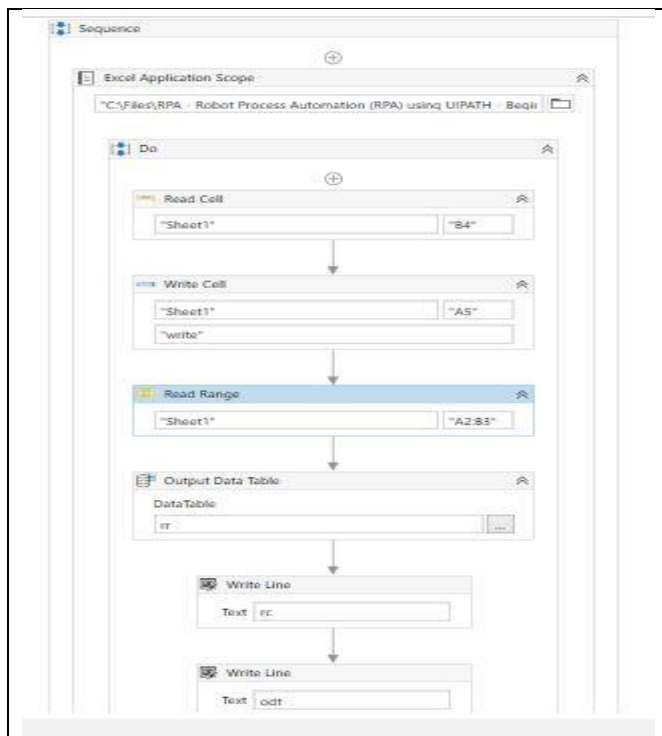
->select **Write line**->put the variable in Text: **odt**

Select Activity->search **output data**->select **Output Data Table**-> drag and drop **Output Data Table** to the + drop activity here between **Read Range** and **Write Line**->give the sheet number and range of cell number e.g. "A2:B3" which to read->if the range of cell number is kept blank e.g. "" then it will read complete excel data

->Select **Output Data Table** container->the value in the cell number is send as Input and store as Output in the Properties-> Input DataTable: **rr** ->then give enter

Select **Output Data Table** container->the value in the cell number is send as Input and store as Output in the Properties-> Output Text variable press **ctrl + k**-> Output Text: **odt** ->then give enter

To run the workflow First join with Start then select dropdown Debug file->select Run File



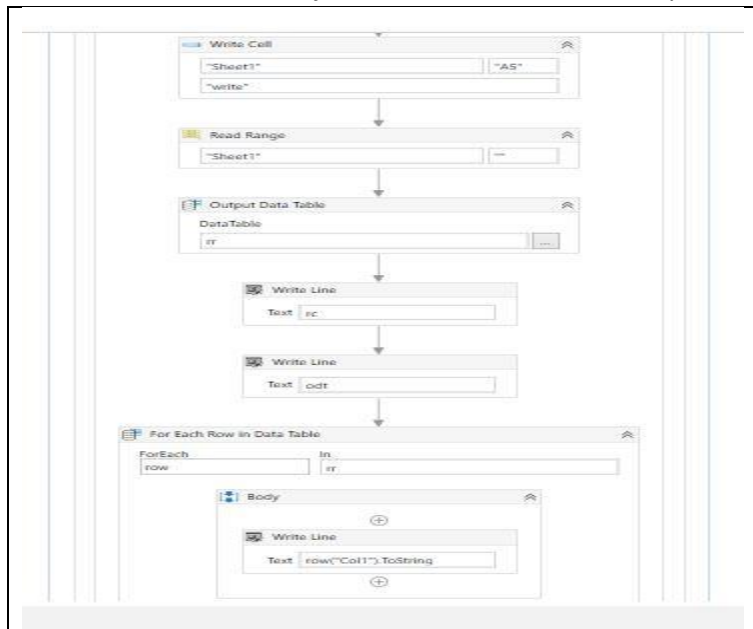
How to Iterate over Data Table and extract values

Select Activity->search **for each row**->select **For Each Row**-> drag and drop **For Each Row** to the + drop activity here ->give ForEach: **row** and In: **rr**

Select Activity->search **write line**->select **Write Line**->drag and drop **Write Line** to the + drop activity here in the Body section of the **For Each Row** container

->Select **Write Line** container->Text: row("of_column_number").ToString

To run the workflow First join with Start then select dropdown Debug file->select Run File



Section 8- Build a Robot to apply Concession for top merit students with given logic

Demo - Understand the Business rules to Automate

Business Scenario:

For students who ranked 1 in exam, school should provide discount of 300\$ in Fees and update the status to Concession Applied and let management know the name of the person who got concession.

Business Rules:

Input ID to the Robot so that robot does below job:

- Deduct 300\$ from fees
- Update the status to concession applied for the Id provides.
- And Print the message in below format
Discount is provided to "XXXX"(name) and update fees is "XXXX"

Implementing input Dialog and Row scanning Technique

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename:

Select Activity->search input ->select **Input Dialog**->drag and drop **Input Dialog** to the + drop activity here

Select **Input Dialog**->in **Title**: "Give a Title" ->in **Label**: "Give a label"

Select **Input Dialog**-> the value is store in the Properties->Output Result:

Select Variables->select Create Variable: **id**

Again Select **Input Dialog**-> the value is store in the Properties->Output Result: **id**

Select Activity->search excel ->select **Excel Application Scope**->drag and drop **Excel Application Scope** to the + drop activity here->give **Workbook path**: "location of the excel file path"

Select Activity->search read range->select **Read Range**-> drag and drop **Read Range** to the + drop activity here of the **Excel Application Scope** in the **Do** container

Select **Read Range** container->the value in the cell number is store in the Properties->Output DataTable variable press **ctrl + k**-> Output DataTable: **rrd** ->then give enter

Select Variables->in the Created Variable: **rrd**->change Scope: **Sequence**

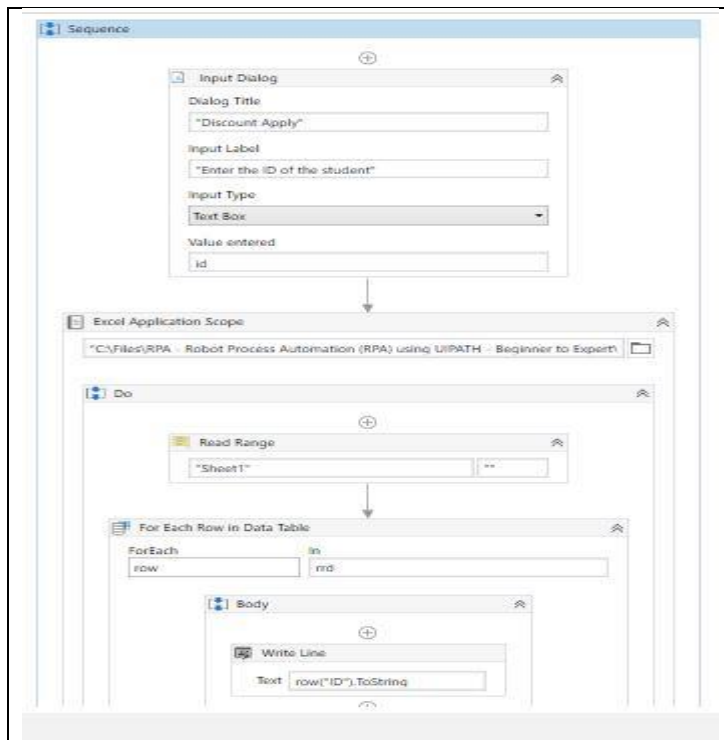
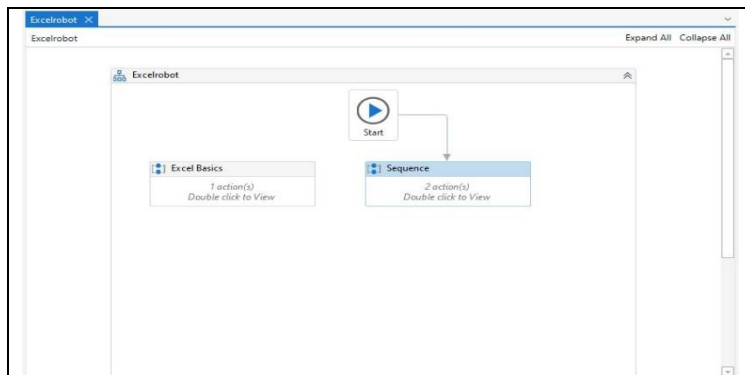
Select Activity->search for each row->select **For Each Row**-> drag and drop **For Each Row** to the + drop activity here ->give ForEach: **row** and In: **rrd**

Select Activity->search write line->select **Write Line**->drag and drop **Write Line** to the + drop activity here in the **Body** container of **For Each Row**

->Select **Write Line** container->Text: row("of_column_number").ToString

To run the workflow First join with Start then select dropdown Debug file->select Run File

	A	B	C	D
1	ID	Fee	Status	Name
2	123	500		Bob
3	456	500		Rob
4	324	500		Tom
5	780	500		Sam
6				
7				
8				
9				



Control flow Usage if else, For loop in UiPath to solve the problem 5

Select Activity->search **if**->select **if**->drag and drop **if** condition to the + drop activity here in the **Body** container of **For Each Row** container

Select **if**->in **Condition** : `row("ID").ToString.Equals(id)` ->if condition match it will go to **Then** and if the condition didn't match it will go to **Else**

Select Activity->search **assign** ->select **Assign**->drag and drop **Assign** to the + drop activity here in the **if** container of **Then**

Select **assign**-> the value is store in the Properties ->Value variable: `CInt(row("Fee"))-300` ->then give enter

Select Variables->select Create Variable: **updatedFee** ->variable Type: **Int** ->scope: **Sequence**

Select **assign**-> the value is store in the Properties ->To: **updatedFee** ->then give enter

Select Activity->search **Write Cell** ->select **Write Cell** ->drag and drop **Write Cell** to the + drop activity here in the **assign** container of **Then**

Select **Write Cell** ->in the Properties section->give the Destination **Range**: "B4"

Destination **SheetName**: "Sheet1", Input **Value**: "Concession Applied"

Build the Counter logic with smart debugging feature in UIPath

Select Variables->select Create Variable: **counter** ->variable Type: **Int** ->scope: **Do**

Select **For Each Row**->in the properties-> Output Index variable press **ctrl + k**-> Output Index: **counter**
->then give enter

Select Activity->search **message box** ->select **Message Box** ->drag and drop **Message Box** to the + drop activity here in the **Then** container between **Assign** and **Write Cell**

->select **Message Box** ->Text: **counter**

Select **Write Cell** ->in the Properties section->give the Destination **Range**: "B8"

Destination **SheetName**: "Sheet1", Input **Value**: updatedFee.ToString

Select Activity->search **assign** ->select **Assign** ->drag and drop **Assign** to the + drop activity here in the **Then** container between **message box** and **Write Cell**

Select **assign**-> the value is store in the Properties ->To: **counter** ->Value variable: **counter+2**->then give enter

Again Select **Write Cell** ->in the Properties section->give the Destination **Range**: "B"+counter.ToString

Destination **SheetName**: "Sheet1", Input **Value**: updatedFee.ToString

- To run the **.xaml** file, it will run Main.xaml file. To set a xaml file as main, right click on the .xaml file then select **Set as Main**.
- To Debug a .xaml file-> right click where to Debug and select **Toggle Breakpoint** ->select **Debug**

To run the workflow First join with Start then select dropdown Debug file->select Run File

Writing the data back to excel with the business logic result

Select Activity->search **Write Cell** ->select **Write Cell** ->drag and drop **Write Cell** to the + drop activity here in **Then container** after **Write Cell**

Select **Write Cell** ->in the Properties section->give the Destination **Range**: "C"+counter.ToString

Destination **SheetName**: "Sheet1", Input **Value**: "Concession Applied"

Select Activity->search **Write Cell** ->select **Write Cell** ->drag and drop **Write Cell** to the + drop activity here in **Else** container

In the **Else** container->Select **Write Cell** ->in the Properties section->give the Destination **Range**: "C"+counter.ToString,

Destination **SheetName**: "Sheet1", Input **Value**: "NA"

Remove the **Assign** activity which is in the **Then** container between **message box** and **Write Cell**
 Remove the **message** activity which is in the **Then** container between **Assign** and **Write Cell**

Select Activity->search assign ->select **Assign** ->drag and drop **Assign** to the + drop activity here in the **For Each Row** container before the **If** condition

In the **For Each Row** container before the **If** condition -> Select assign-> the value is store in the Properties ->To: **counter** ->Value variable: **counter+2**->then give enter

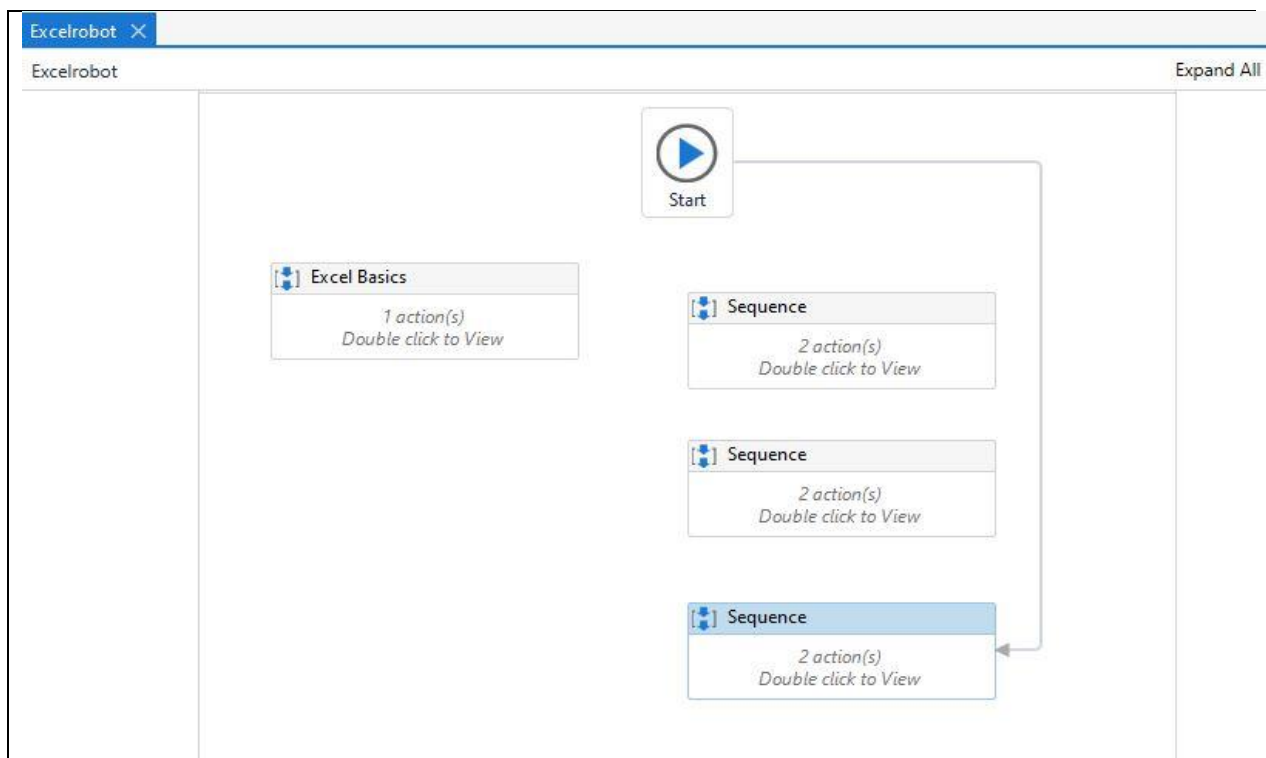
Select Activity->search message box ->select **Message Box** ->drag and drop **Message Box** to the + drop activity here in the **Then** container after **Write Cell**

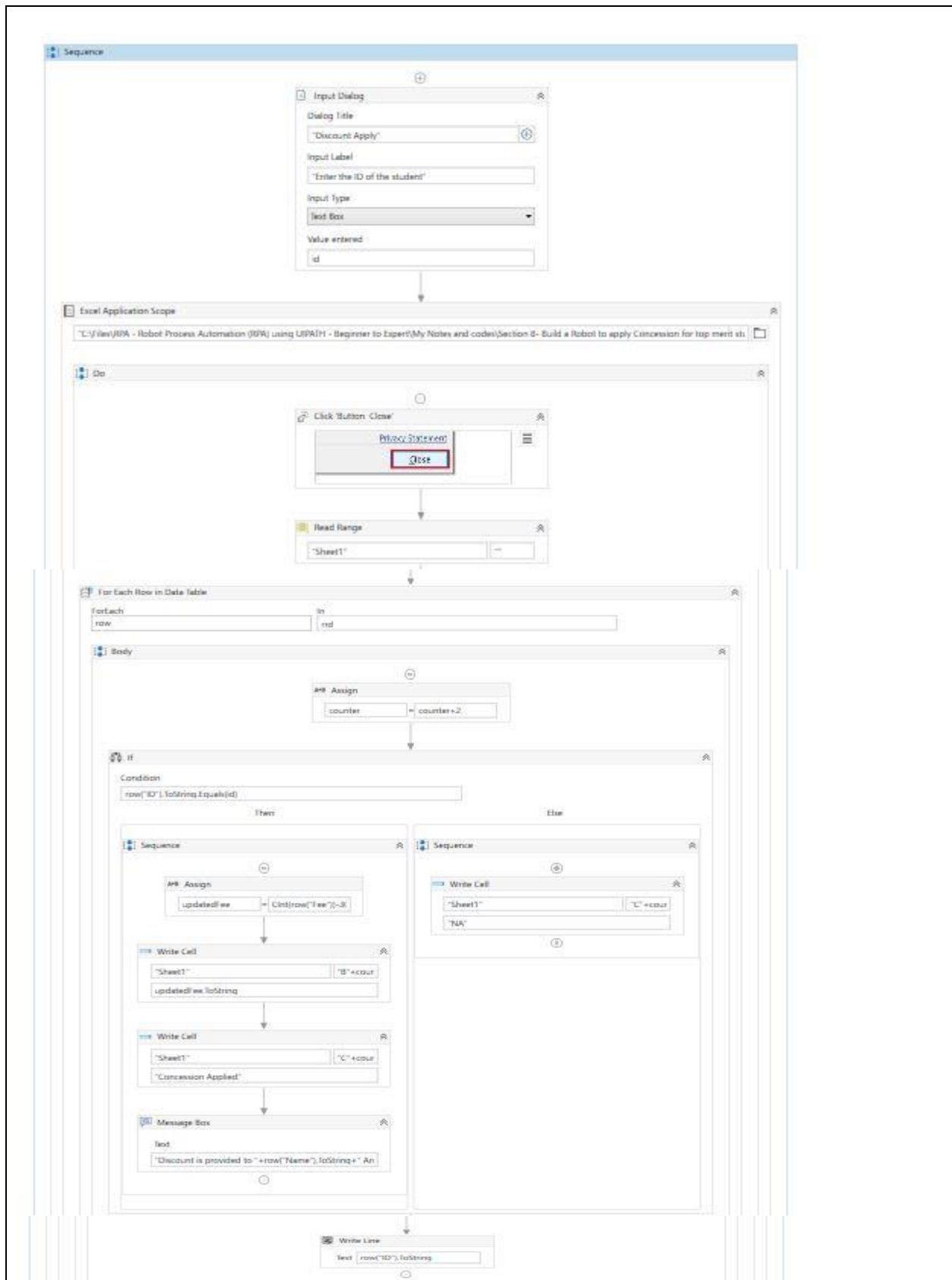
Select message box

->Text: "Discount is provided to "+row("Name").ToString+" And update fees is "+updatedFee.ToString

To run the workflow First join with Start then select dropdown Debug file->select Run File

	A	B	C	D
1	ID	Fee	Status	Name
2	123	500		Bob
3	456	500		Rob
4	324	500		Tom
5	780	500		Sam
6				
7				
8				
9				





Misconceptions in DataTable and Excel updates

Section 9- Techniques to Read PDFScanned Images using UIPath

Read PDF Text activity to read content in PDF

PDF Activities:

- **Read PDF Text**- Read all Characters from Specific PDF file and Stores in a string
- **Read PDF with OCR**- Read all Characters from Specific PDF file using OCR Technologies
- **Get Text**- Extract text value from specified UI Element

Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**
->select add->select Flowchart->Name: **InvoiceExtractPDF**->select create

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: Read PDF Text Activity Demo

Select Activity->search pdf->if the activity isn't there then have to install the package

To install a package->select Manage Packages->select All Packages->search pdf

->select **UiPath.PDF.Activities**->select install->select save

Select Activity->search pdf->select **Read PDF Text**->drag and drop **Read PDF Text** to the + drop activity

Select **Read PDF Text**->in the Properties->File, **FileName**: "give location of PDF file"

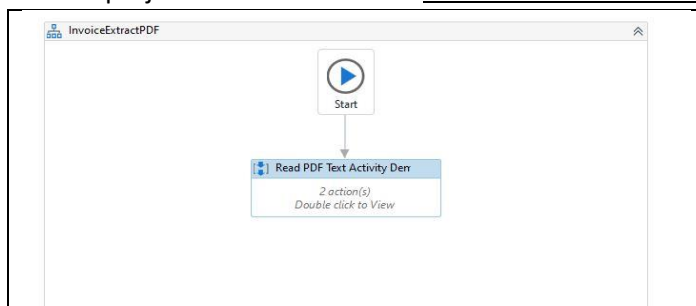
->Output, **Text**: press ctrl + k-> Output **Text**: pdfTextExtract->then give enter

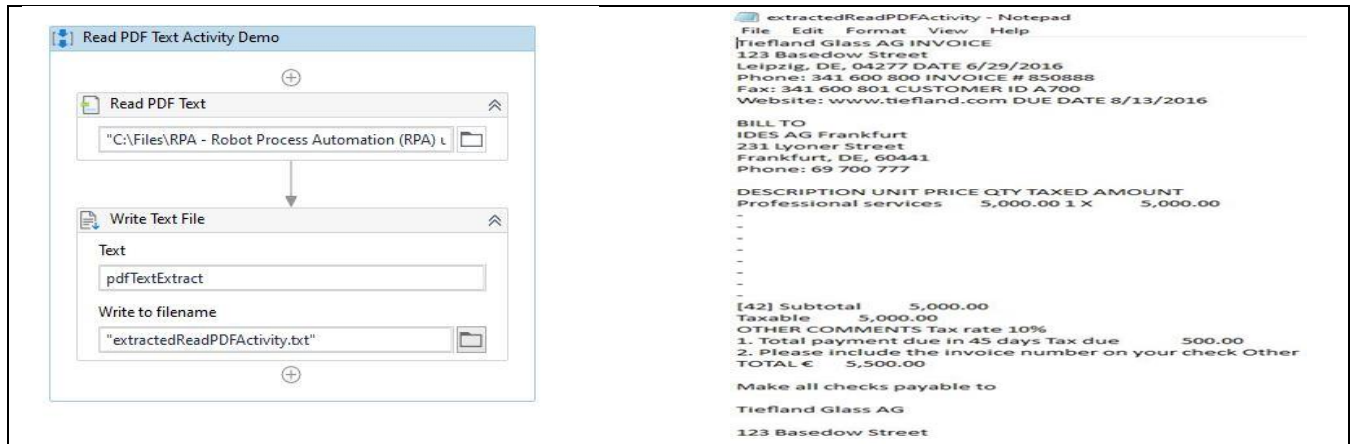
Select Activity->search write text file->select **Write Text File**->drag and drop **Write Text File** to the + drop activity

Select **Write Text File**->Text: pdfTextExtract ->Write to filename: "extractedReadPDFActivity.txt"

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File

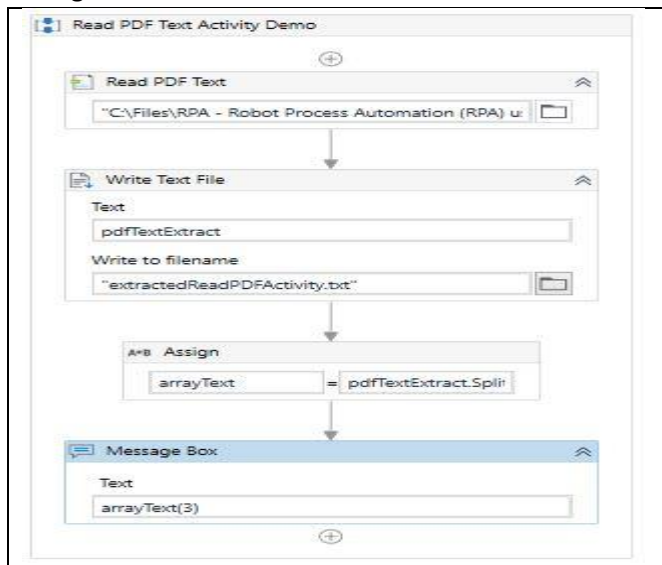
Refresh project->a file name with "extractedReadPDFActivity.txt" is created

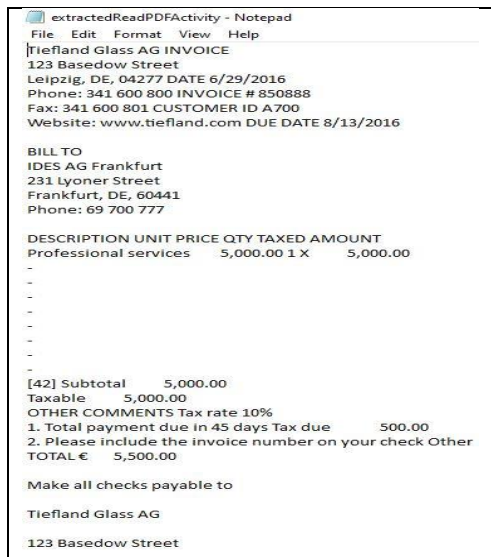




String functions to extract content from pdf based on the logic applied

Select Activity->search **assign**->select **Assign**->drag and drop **Assign** to the + drop activity
 Select **Assign**->in the Properties->Value:pdfTextExtract.Split(Environment.NewLine.ToArray,StringSplitOptions.RemoveEmptyEntries)
 Select Variables->in the Created Variable: **arrayText**->Variable Type: select Array of [T]->select String->select ok
 Select Assign->in the Properties->To: arrayText
 Select Activity->search **message box**->select **Message Box**->drag and drop **Message Box** to the + drop activity
 Select **Message Box**->Text:arrayText(3) ->->to extract the value from index
 To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File





Read PDF with OCR Technology to extract item with image

Using **Read PDF Text** activity can extract any value from PDF

To get the invoice no from pdf->Select **Message Box**->Text: `arrayText(3).Split("#").ToCharArray()(1)`

To get the phone no from pdf->Select **Message Box**->Text: `arrayText(3).Split("#").ToCharArray()(0)`

To extract data from image copy which is converted to PDF use: **Read PDF with OCR**

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: Read PDF Demo 2

Select Activity->search read pdf with ocr->select **Read PDF With OCR**

->drag and drop **Read PDF With OCR** to the + drop activity

Select Activity->search read pdf with ocr->select **Read PDF With OCR**

->drag and drop **Read PDF With OCR** to the + drop activity

Select activity->search **Microsoft OCR**->select **Microsoft OCR**-> drag and drop **Microsoft OCR** to the Drop OCR Engine

Select **Read PDF With OCR** ->in the Properties->File, **FileName**: "give location of PDF file"

->Output, **Text**: press ctrl + k-> Output **Text**: pdfTextImage->then give enter

Select Activity->search write text file->select **Write Text File**->drag and drop **Write Text File** to the + drop activity

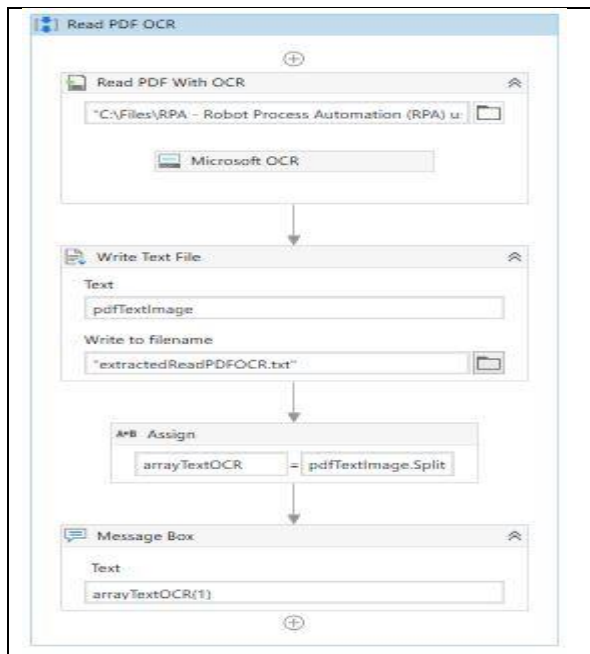
Select **Write Text File**->Text: pdfTextExtract ->Write to filename: "extractedReadPDFOCR.txt"

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File

Refresh project->a file name with "extractedReadPDFOCR.txt" is created

Select Activity->search assign->select **Assign**->drag and drop **Assign** to the + drop activity

Select **Assign**->in the Properties->Value: pdfTextImage.Split(Environment.NewLine.ToArray,StringSplitOptions.RemoveEmptyEntries)
 Select Variables->in the Created Variable: **arrayTextOCR**->Variable Type: select Array of [T]->select String->select ok
 Select Assign->in the Properties->To: arrayTextOCR
 Select Activity->search **message box**->select **Message Box**->drag and drop **Message Box** to the + drop activity
 Select **Message Box**->Text:arrayTextOCR(1) ->to extract the value from index
 To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File



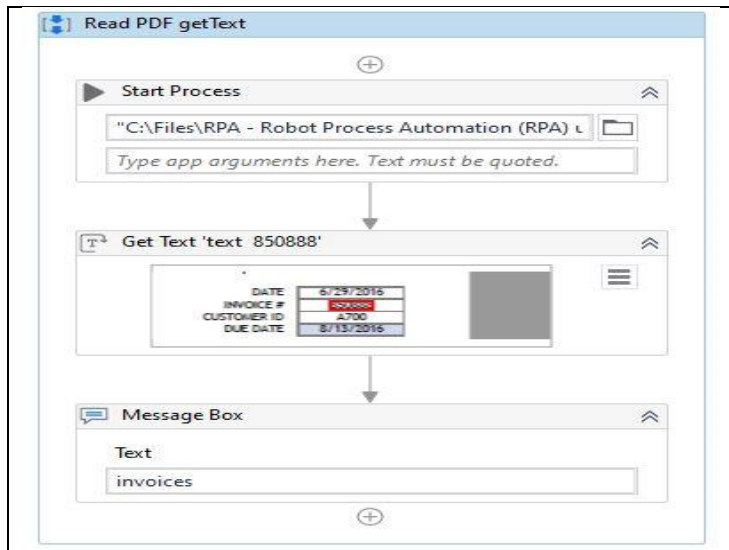
Get Text Activity to extract content from PDF

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: Read PDF getText
 Select Activity->search **Start Process**->select **Start Process**->drag and drop **Start Process** to the + drop activity
 Select **Start Process**->in the properties->Input, **FileName**: "give location of PDF file"
 Select Activity->search **get text**->select **Get Text**->drag and drop **Get Text** to the + drop activity
 At first manually open the pdf file then Select **Get Text** ->select **indicate on screen**->select the element on the pdf which will extract
 To confirm the unique locator which will be extracted from pdf->select **Get Text**->select **click to open options menu**->select **edit selector**->select **Highlight**
 Select **Get Text**->in the Properties ->Output, Value: press ctrl + k-> Value: **invoices** ->then give enter
 Select Activity->search **message box**->select **Message Box** ->drag and drop **Message Box** to the + drop activity
 Select **Message Box**->Text: **invoices**

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File

Note:

- **Read PDF Text** and **Read PDF with OCR** activity : use this when there are multiple validation
- **Get Text** activity: use this when processing hundreds of pdf or when is prone to failure for invalid format



Section 10- Build a robot to Scan PDF documents and Automate Desktop Apps

Demo - Understand the Business rules to Automate

Business Scenario:

Extract Invoices and Amount details from hundreds of PDF's generated every day and place them into Company internal tools.

Maintain a copy of extracted Details in Excel for further Data Analytics and Research.

For each loop to go through every file present in Folder

Step1:

- Invoices and Amount
- Open file
- Read invoice
- Read amount
- Close file
- Print extracted details

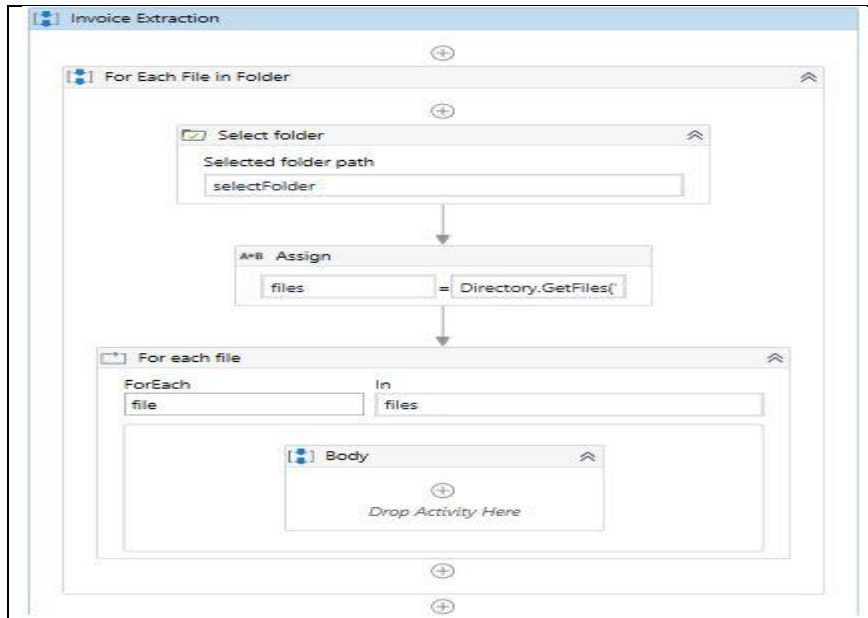
Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**
->select add->select Flowchart->Name: **InvoiceExtractPDF**->select create

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: **Invoice Extraction**

To get the readymade Files in Uipath->Select **Snippets**->double click Loops->select **For Each Files in Folder**-> drag and drop **For Each Files in Folder** to the + drop activity

Select **For Each Files in Folder**->select **Assign**->in the Properties

->Value, Directory.GetFiles("give location of PDF Folder")->all the pdf files will be stored in the file variable



Identify properties in PDF using Google OCR Technology

Select **For Each Files in Folder**->select **Body**->Delete the **Write Line** activity

Select Activity->search **Start Process**->select **Start Process**->drag and drop **Start Process** to the + drop activity of the **Body** in the **For Each Files in Folder**

Select **Start Process** -> in the Properties->**Input**, FileName: **file**

Select Activity->search **Close Application**->select **Close Application** ->drag and drop **Close Application** to the + drop activity

Open the **NPO Invoice1.pdf**->select **Close Application**->select **Indicate on screen**->then select the pdf which is opened

select **Close Application**->select **click to open options menu**->select **Edit Selector**->in the Edit Selector -> **title='NPO Invoice*.pdf'** ->select ok

Select **For Each Files in Folder** ->Delete the **Select Folder**

Select Snippets->search **Delay**->select **Delay 3 Seconds** ->drag and drop **Delay 3 Seconds** to the + drop activity between **Start Process** and **Close Application**

Select Activity->search **Get Text**->select **Get Text** ->drag and drop **Get Text** to the + drop activity between **Sequence Delay** and **Close Application**

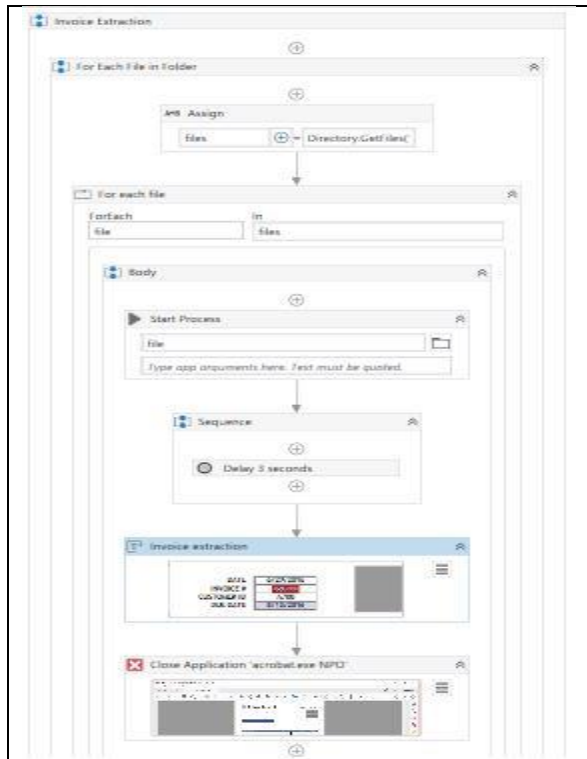
Open the **NPO Invoice1.pdf** ->Select **Get Text**->select **Indicate on screen**->>then select the invoice number from the pdf which is opened

Double click **Get text**->Rename: **Invoice extraction**

Select **Invoice extraction** ->select **click to open options menu**->select **Edit Selector**->in the Edit Selector -> **title='NPO Invoice*.pdf'** -> **name='8*'** ->then select ok

Select **Invoice extraction** ->in the properties-> Output, Value: press ctrl + k-> Value: **invoice** ->then give enter

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select **Debug file**->select **Run File**



How to handle Dynamic Elements in PDF with regular expressions

Select Activity->search **Get Text**->select **Get Text** ->drag and drop **Get Text** to the + drop activity between **Invoice extraction** and **Close Application**

Select **Get Text** ->in the properties->**Common**, DisplayName: **Amount Extract**

Open the **NPO Invoice1.pdf** ->Select **Amount Extract** ->select **Indicate on screen**->>then select the invoice number from the pdf which is opened

Select **Amount Extract** ->select **click to open options menu**->select **Edit Selector**->in the Edit Selector -> **title**='NPO Invoice*.pdf'->then select ok

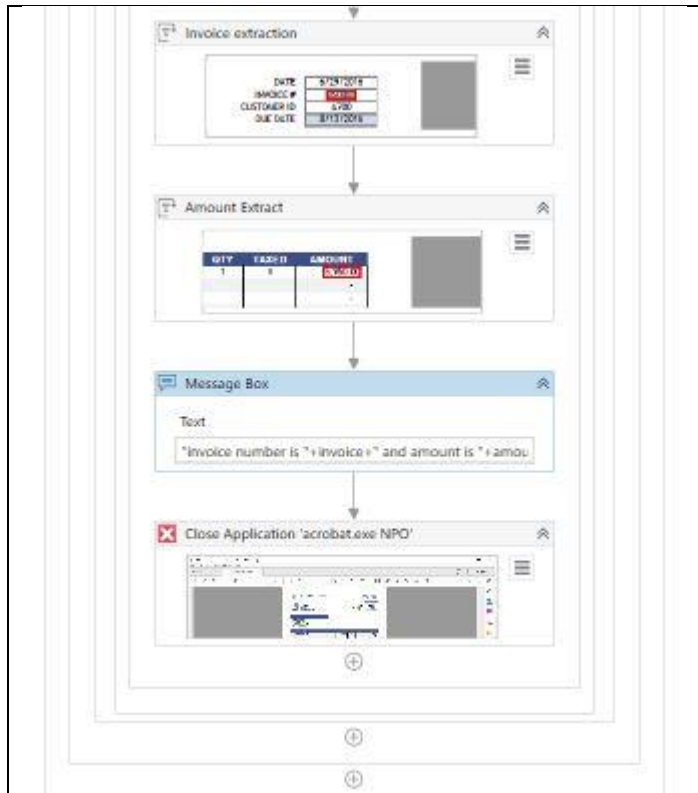
To get the amount from all the pdf, open each pdf and click the repair->Select **Amount Extract** ->select **click to open options menu**->if it is not validate->then select **Repair**->then select ok->Do this for all the pdf

Select **Amount Extract** ->in the properties-> Output, Value: press ctrl + k-> Value: **amount** ->then give enter

Select Activity->search **Message Box**->select **Message Box** ->drag and drop **Message Box** to the + drop activity

Select **Message Box**->Text: "invoice number is "+**invoice**+" and amount is "+**amount**

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select **Debug file**->select **Run File**



Extracting the values from PDF to Excel using UiPath

- Create an excel file. Name: **Book3**

Select Activity->search **Excel Application Scope**->select **Excel Application Scope** ->drag and drop **Excel Application Scope** to the + drop activity between **Amount Extract** and **Message Box**

Select **Excel Application Scope**->in the properties, **File**->Workbook path: give the location of the excel file

Select Activity->search **Write Cell**->select **Write Cell** ->drag and drop **Write Cell** to the + drop activity of **Do** sequence of **Excel Application Scope**

Select **Write Cell** ->in the properties->Destination->Range: **"Sheet1"**

->SheetName: **"A"+counter.ToString**->Input, Value: **invoice**

Select Variables->in the Created Variable: **counter**->Variable Type: select **int32**

->scope: **InvoiceExtractPDF**->Default: **2**

Select Activity->search **Assign**->select **Assign**->drag and drop **Assign** to the + drop activity after **Write Cell**

Select **Assign**->in the properties->Misc, To: **counter** ->Value: **counter+1**

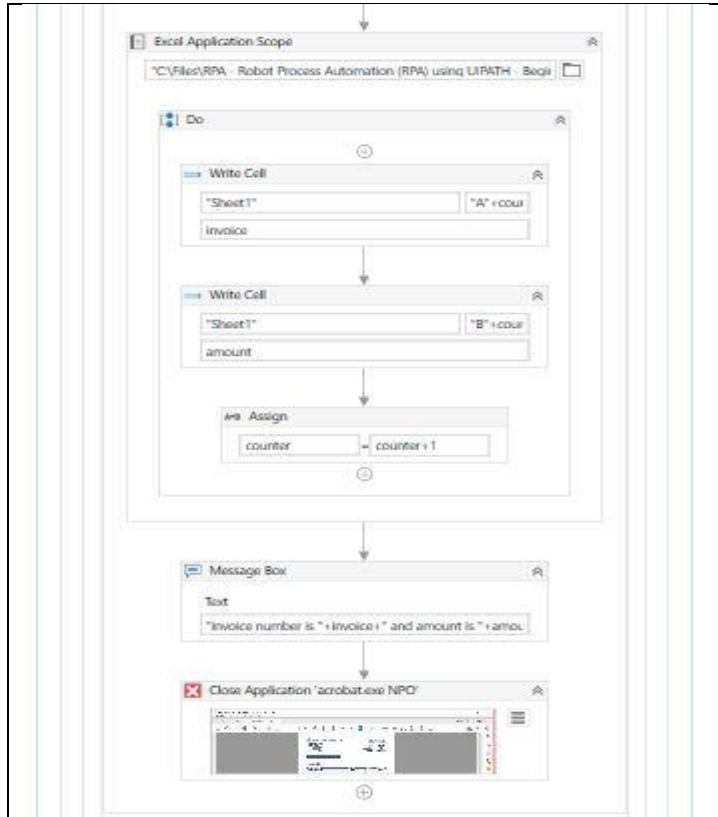
Select Activity->search **Write Cell**->select **Write Cell** ->drag and drop **Write Cell** to the + drop activity of **Excel Application Scope** between **Write Cell** and **Assign**

Select **Write Cell** ->in the properties->Destination->Range: "**Sheet1**"

->SheetName: "**B**" + **counter.ToString** -> Input, Value: **amount**

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File

It will extract the invoice and amount from the pdf files and place it in the excel file Name: **Book3**



Pushing the values from pdf/excel to Desktop Internal tools

Download the internal tool: **AnyTeller** application

Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**

->select add->select Flowchart->Name: **InvoiceExtractPDF**->select create

Select Activity->search **sequence**->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: **Invoice Extraction 2**

To get the readymade Files in Uipath->Select **Snippets**->double click Loops->select **For Each Files in Folder**-> drag and drop **For Each Files in Folder** to the + drop activity

Select **For Each Files in Folder**->select **Assign**->in the Properties

->Value, Directory.GetFiles("give location of PDF Folder")->all the pdf files will be stored in the **file** variable

Select **For Each Files in Folder** ->Delete the **Select Folder**

Select Activity->search **Sequence**->select **Sequence**->drag and drop **Sequence** to the + drop activity after **For each file** activity

Select Activity->search **Excel Application Scope**->select **Excel Application Scope** ->drag and drop **Excel Application Scope** to the + drop activity of **Sequence** activity

Select **Excel Application Scope**->in the properties, **File**->Workbook path: give the location of the excel file

Select Activity->search **Read Range**->select **Read Range**->drag and drop **Read Range** to the + drop activity of **Excel Application Scope** activity

Select **Read Range**->in the properties-> Output, DataTable: press ctrl + k-> DataTable: **rr** ->then give enter

To open the internal tool: **AnyTeller** application -> Select Activity->search **open Application**->select **Open Application**->drag and drop **Open Application** to the + drop activity after **Read Range** activity

First open the **AnyTeller** application then->select **Open Application**->select **Indicate window on screen** ->select the **AnyTeller** application which was opened

Select Activity->search **For Each Row**->select **For Each Row** ->drag and drop **For Each Row** to the + drop activity of **Do** of **Open Application** activity

Select **For Each Row** ->ForEach: **row**->in: **rr**

Select Activity->search **Type Into**->select **Type Into** ->drag and drop **Type Into** to the + drop activity of **Body** of the **For Each Row** activity

First open the **AnyTeller** application then ->Select **Type Into**->select **Indicate element inside window** ->select **Cash In** for input in the **AnyTeller** application

Select **Type Into**->Text: **row("Amount").ToString**

Select Activity->search another **Type Into**->select **Type Into** ->drag and drop **Type Into** to the + drop activity of after **Type Into** activity

First open the **AnyTeller** application then ->Select **Type Into**->select **Indicate element inside window** ->select **On Us Check** for input in the **AnyTeller** application

Select **Type Into**->Text: **row("Invoice").ToString**

Select Activity->search another **Click**->select **Click** ->drag and drop **Click** to the + drop activity of after the second **Type Into** activity

First open the **AnyTeller** application then ->Select **Click** ->select **Indicate element inside window**->select **Accept** in the **AnyTeller** application

--

Section 11- Best Practices for Maintaining UiPath Coding standards

Optimization of workflow with proper conventions

Best Practices for Maintaining the Code:

- Making your workflow readable
- Build Reusable workflows
- Commenting techniques
- Exception Handling

Bring all the files to Project level so that directly give the name of that folder instead of giving the complete path.

Select FirstDemo1Process->right click->select **open Project Folder**->copy the **invoices** folder paste in the Project Folder

Similarly copy the **excel file** and paste in the **Project Folder**

In the Uipath->select **Refresh**->then all the files are present

--

Exception handling mechanism in UiPath

When one of the file is corrupted and data is not displaced properly then Automation workflow will break. So, Exception handling mechanism is used, with **Try and Catch block.**

Put a corrupted pdf invoice name: NPO Invoice7 in the invoices folder

Select Activity->search **Try Catch**->select **Try Catch** ->drag and drop **Try Catch** to the + drop activity between **Invoice extraction** and **Delay**

If something fails in **Try block** then the script will not stop and it will go to the **Catch block**

Move the **Invoice extraction** and **Amount extraction** in the **Try Catch**

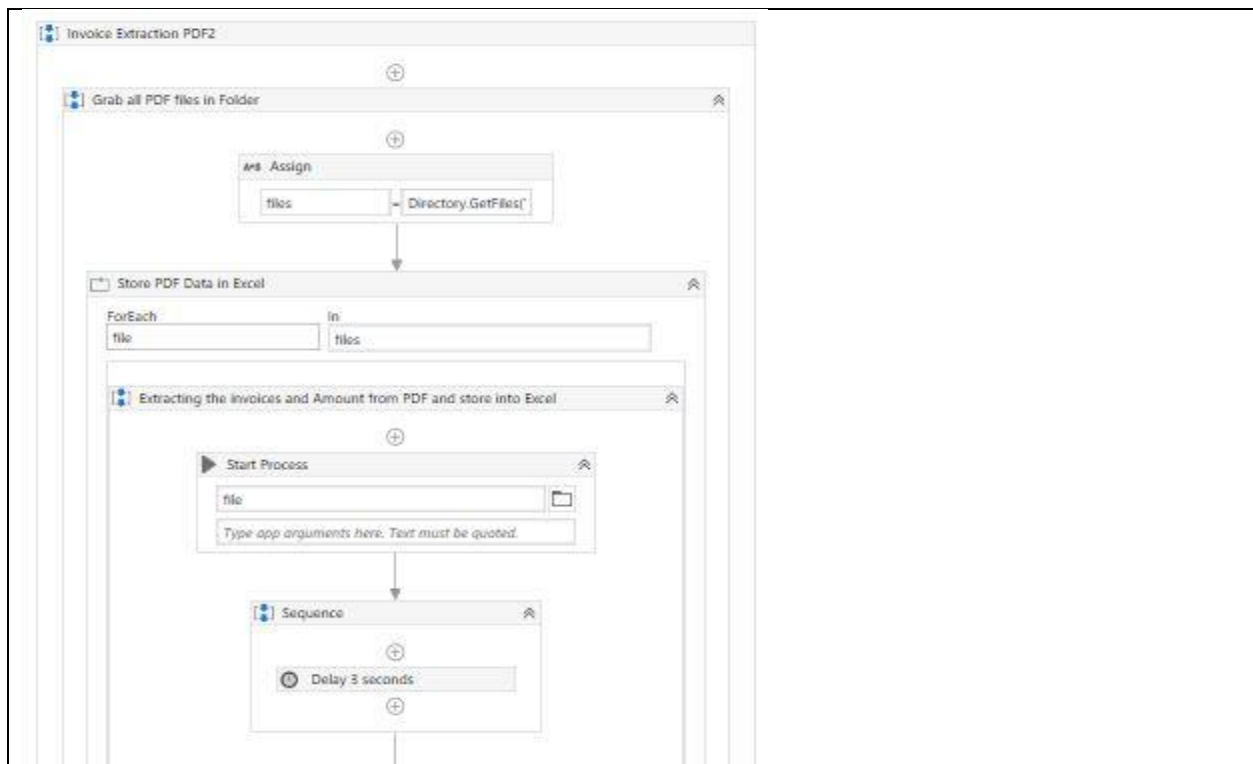
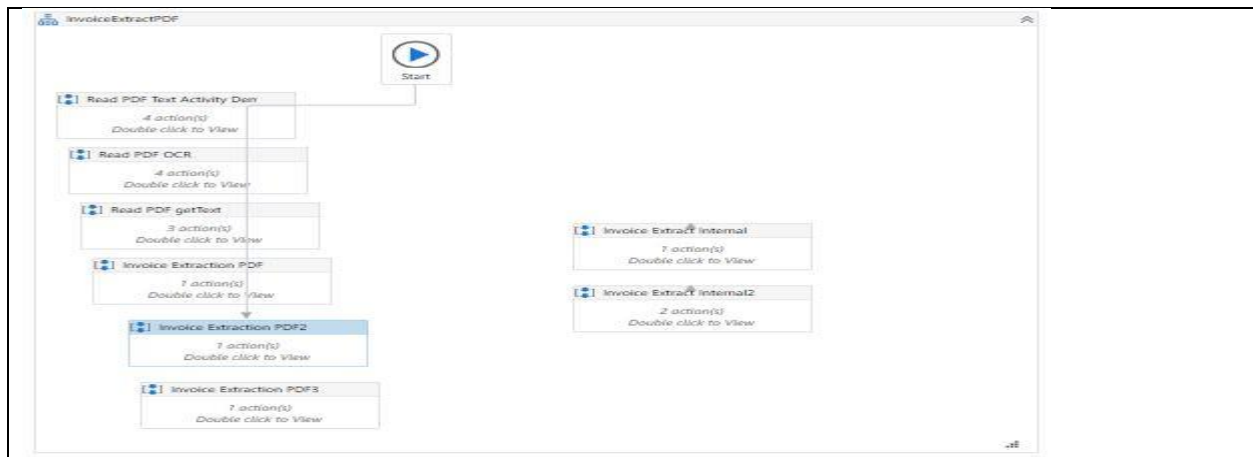
In the **Catch block** ->select the Type of exception will be receiving

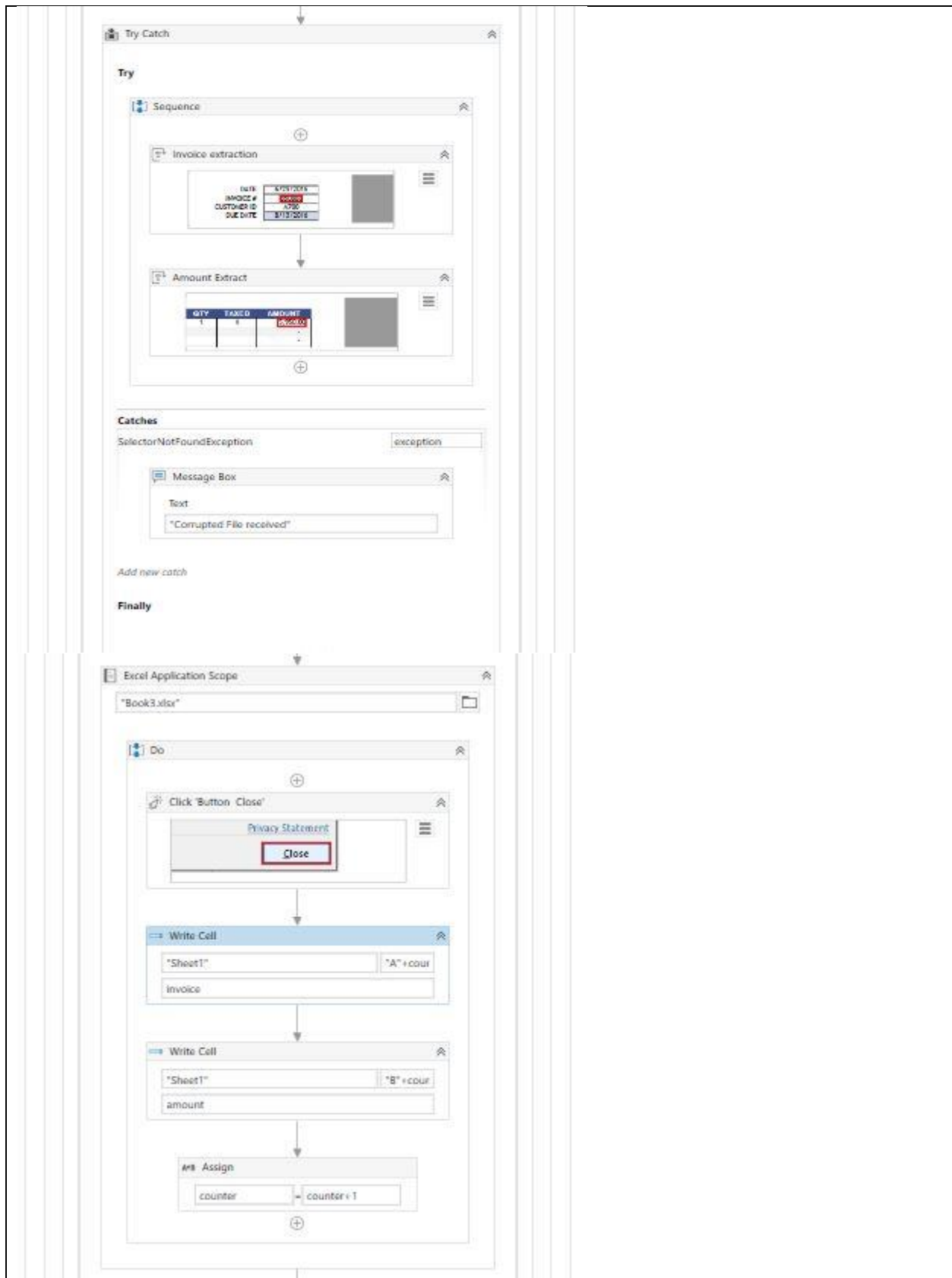
->Exception: **UiPath.Core.SelectorNotFoundException**

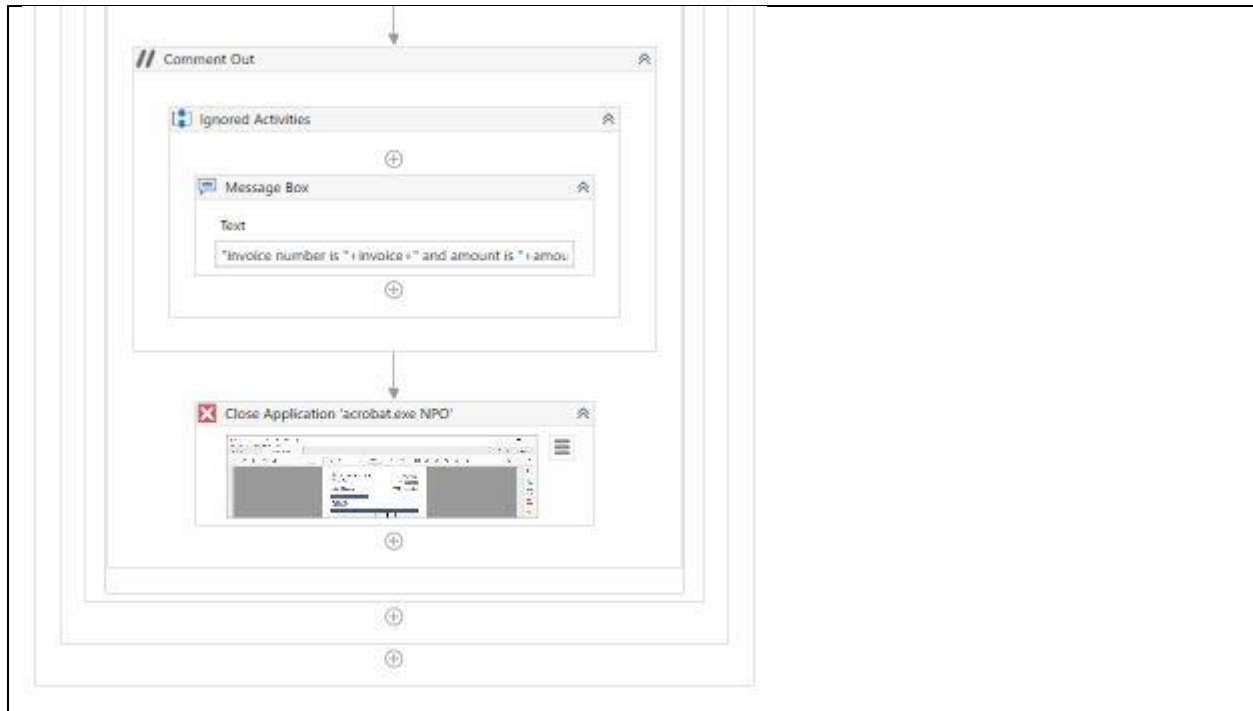
Select Activity->search **message box**->select **Message Box** ->drag and drop **Message Box** to the + drop activity of **Catch block**

Select **Message Box** ->Text: "Corrupted File received"

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File







Running Multiple xaml files from Main xaml file

- Running Multiple Workflows (xaml) files from Main.xaml file
- Running Workflows in Parallel

To run two or three (xaml) files together->to call two or three (xaml) files in a new workflow and then execute everything in a sequence.

Right click on the **Project**->select **Add**->select **Sequence**->Name: **Automation Runner**->select create

Place all the (xaml) files which want to run into the **Automation Runner** sequence

Select Activity->search **Invoke Workflow File**->select **Invoke Workflow File** ->drag and drop **Invoke Workflow File** to the + drop activity of **Automation Runner**

Select **Invoke Workflow File** ->select **Browser for file**->select the (xaml) files which want to execute

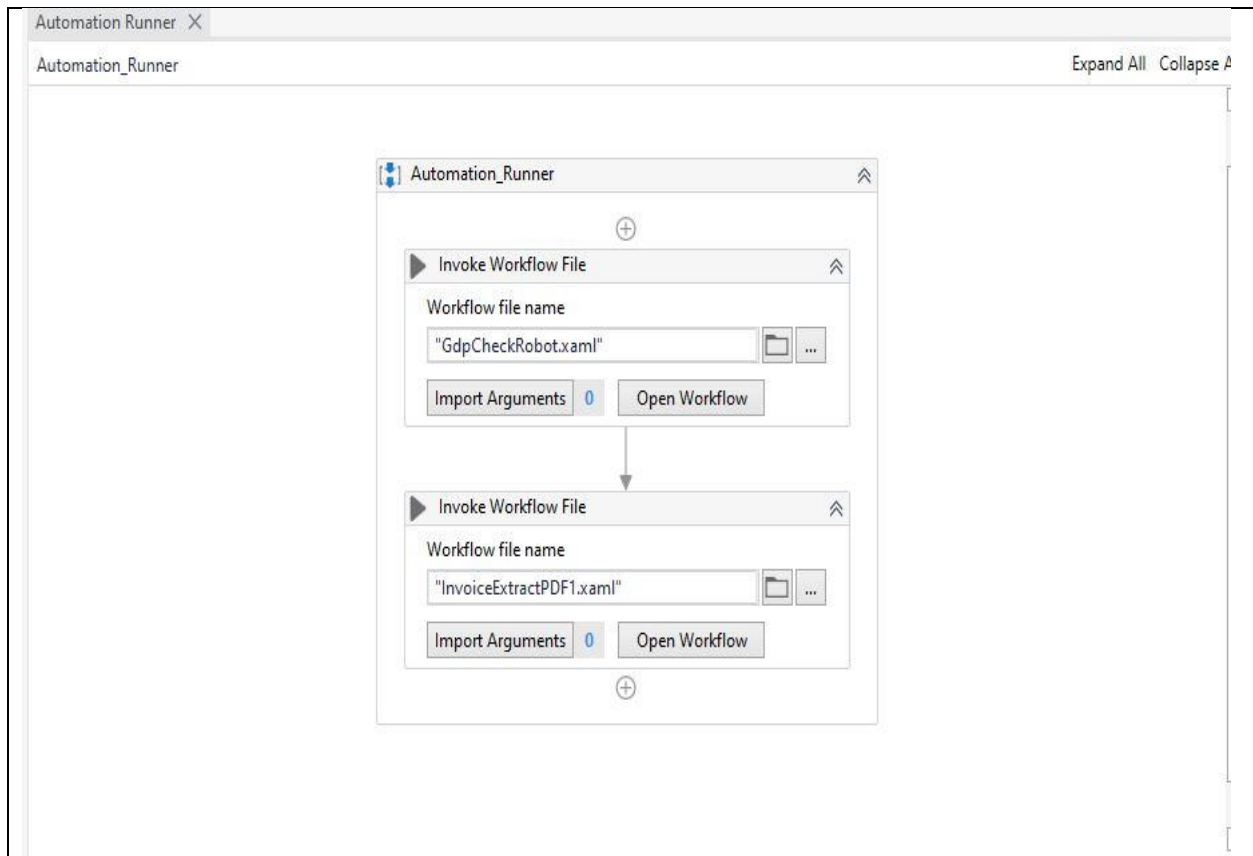
Select Activity->search another **Invoke Workflow File**->select **Invoke Workflow File** ->drag and drop

Invoke Workflow File to the + drop activity of **Automation Runner** after the first **Invoke Workflow File**

Then Select the second **Invoke Workflow File** ->select **Browser for file**->select the (xaml) files which want to execute

To run the workflow->right click on the particular (xaml) file which will run->select **Set as Main**

->select Debug file->select Run File



Running Robots (Workflows) in Parallel mode from UIPath

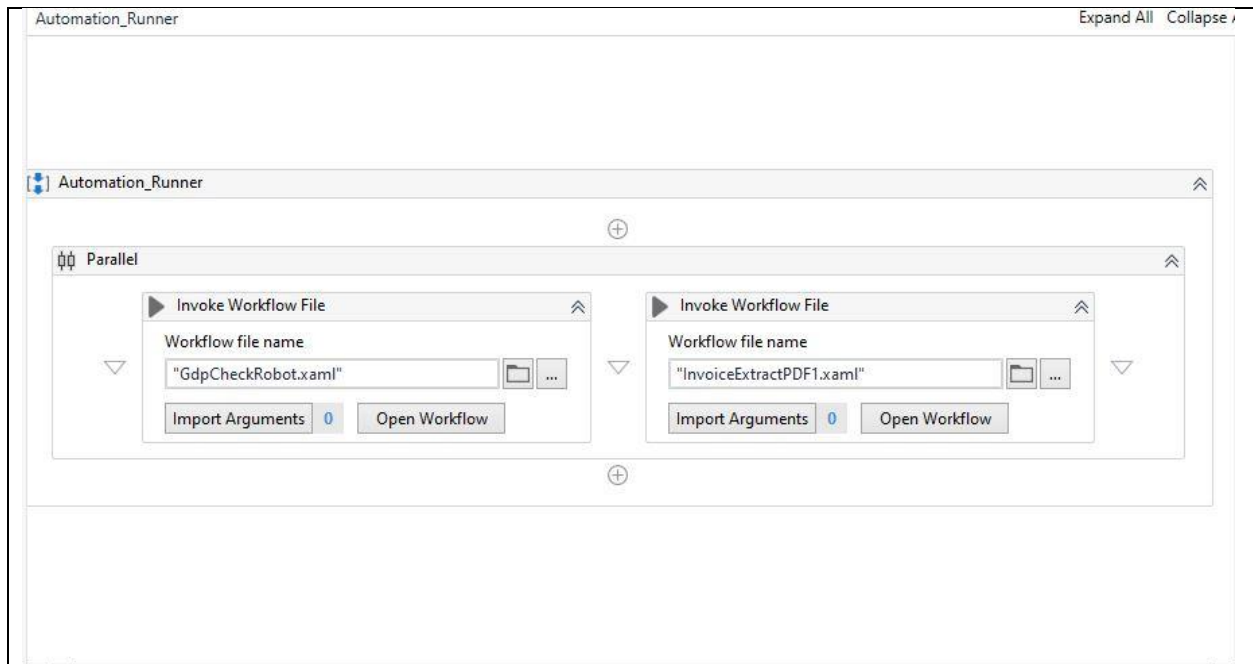
- Running Workflows in Parallel

Select Activity->search **Parallel**->select **Parallel** ->drag and drop **Parallel** to the + drop activity of **Automation Runner** before **Invoke Workflow File**

Then drag and drop the workflow which want to run in parallel in the **Parallel** activity

To run the workflow->right click on the particular (xaml) file which will run->select **Set as Main**

->select Debug file->select Run File



Comment Activity Techniques to UiPath workflows

- Commenting techniques

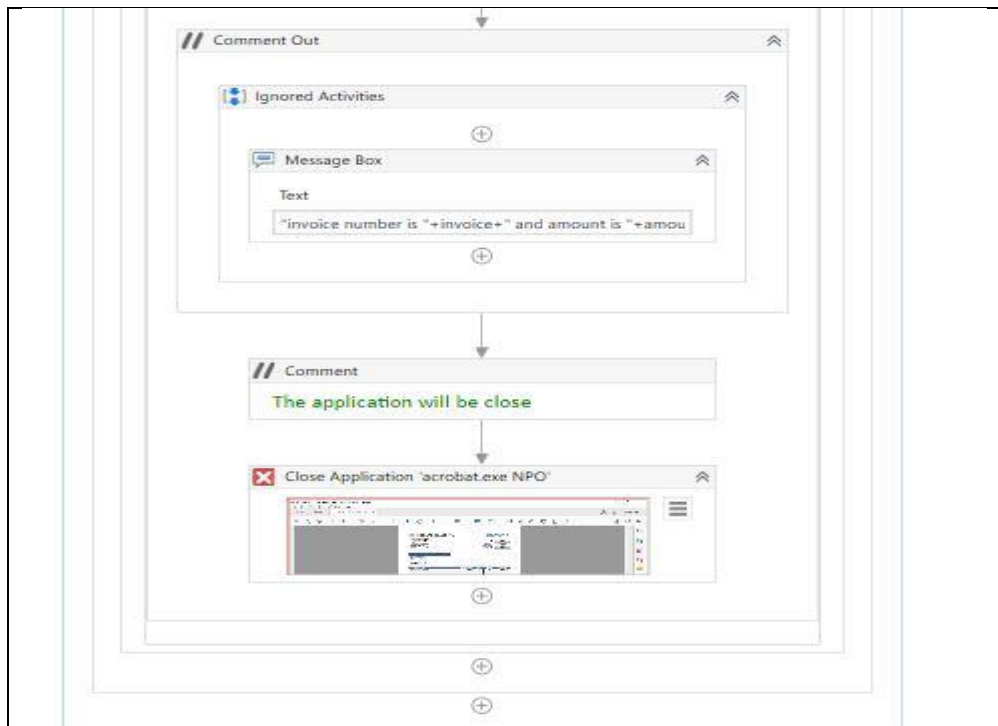
Select Activity->search **Comment**->select **Comment** ->drag and drop to anywhere in the container for comment

Select **Comment** ->write the comments in the text property

Select Activity->search **Comment Out**->select **Comment Out** ->drag and drop to anywhere in the container for comment

Select **Comment Out** ->drag and drop the activities which will be ignored in the **Comment Out**. So that particular activity will be disable

To disable an activity->right click on an activity->select **Disable Activity**->automatically that activity will be placed inside comment out



Section 12- Build Robots for Gmail Automation

Demo - Understand the Business rules to Automate

- Create a Robot which Monitors Incoming Emails and save attachments and recipient details if Subject name of mail contains "Selenium"
- Consolidate all the recipient emails with subject Selenium and email to Recruiter

Important settings for Gmail Account before proceeding

- Link: <https://docs.uipath.com/studio/docs/enabling-gmail-for-email-activities>

Enabling POP3/IMAP from Gmail

- Go to <https://mail.google.com> and log in with the email you want to use in your automation.
- Click Settings > Settings. The Settings page is opened.
- In the Forwarding and POP/IMAP tab, select the Enable POP and Enable IMAP check boxes.

Generate and Use an App Password from Google

- Go to <https://security.google.com/settings/security/apppasswords>
- From the Select app drop-down select Other (Custom Name). A text box (Custom Name) is displayed.
- Fill in the text box with a custom app name, such as UiPath, and click Generate.
A 16-character password is generated that can be used in UiPath Studio instead of the original password.

How to read Emails in Gmail Server using UiPath

Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**
->select add->select Flowchart->Name: **MailRobot**->select create

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: Retrieve Emails and Attachments

Select Activity->search Get IMAP->select **Get IMAP Mail Messages**->drag and drop **Get IMAP Mail Messages** to the flowchart

In the Gmail account->create a new label->name: **jobs**

Select **Get IMAP Mail Messages**->in the properties

->select **Logon**->Email: give_email

->select Host->MailFolder: "**jobs**", Port: **993**, Server: "**imap.gmail.com**"

Select Activity->search Get Password->select **Get Password**->drag and drop **Get Password** to the flowchart before **Get IMAP Mail Messages** activity

Select **Get Password**->in the properties->select **Misc**->Password: give 16-character generated password

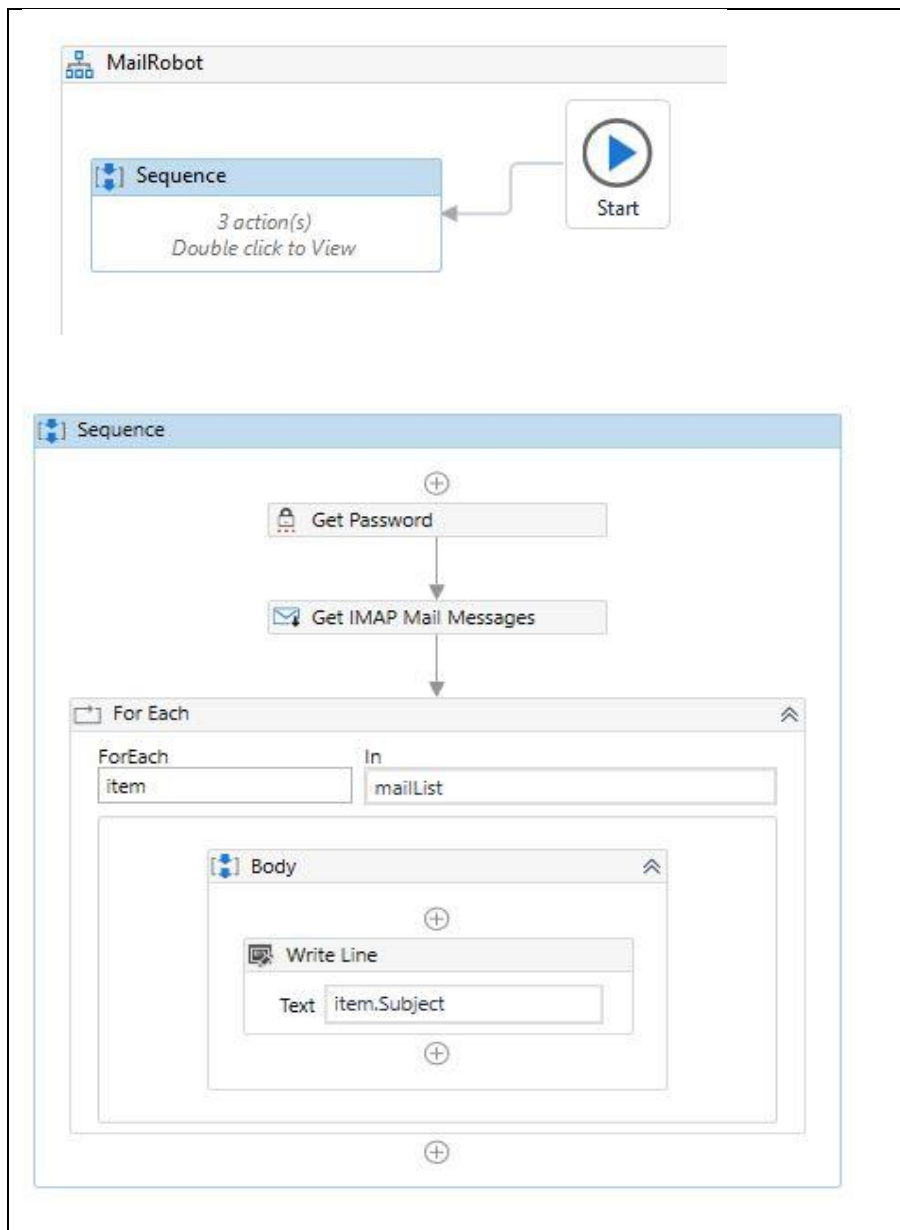
-> Result: press ctrl + k-> Result: **pwd** ->then give enter

Select Variables->select **pwd**->scope: **MailRobot**

Select **Get IMAP Mail Messages**->in the properties->select **Logon**->Password: **pwd**

->select **Options**->MarkAsRead: click, Top: 5,

->select **Output**->Messages: press ctrl + k-> Messages: **mailList** ->then give enter
Select Activity->search For Each->select **For Each**->drag and drop **For Each** to the flowchart after **Get IMAP Mail Messages** activity
Select **For Each**->in the properties->select **Misc**->List of items: mailList
->TypeArgument: System.Net.MailMessage
Select Activity->search Write Line->select **Write Line** ->drag and drop **Write Line** to the flowchart in Body of **For Each** activity
Select **Write Line**->Text: item.Subject
To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File
In the output, it will give the subject of the email





Save Attachments and Receipt Email address with logical condition

Disable the **Write Line** activity which is in the body of the **For Each** activity

Select Activity->search **If**->select **If** ->drag and drop **If** to the flowchart in Body of **For Each** activity

Select **If** ->condition: **item.Subject.Contains("Selenium")**

Select Activity->search **Append Line**->select **Append Line** ->drag and drop **Append Line**

->to the **If** activity in the **Then** section

Select **Append Line** ->Text: **item.From.ToString** ->Write to filename: "**FromDetails.txt**"

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select

Debug file->select Run File->a file name: **FromDetails** will be generated in the Project which contains the email which are sent

Select Activity->search **Sequence**->select **Sequence** ->drag and drop **Sequence**

->to the **If** activity in the **Then** section

Create a folder name: **attachments**->a place the folder in the

Select Activity->search **Save Attachments**->select **Save Attachments** ->drag and drop **Save Attachments**

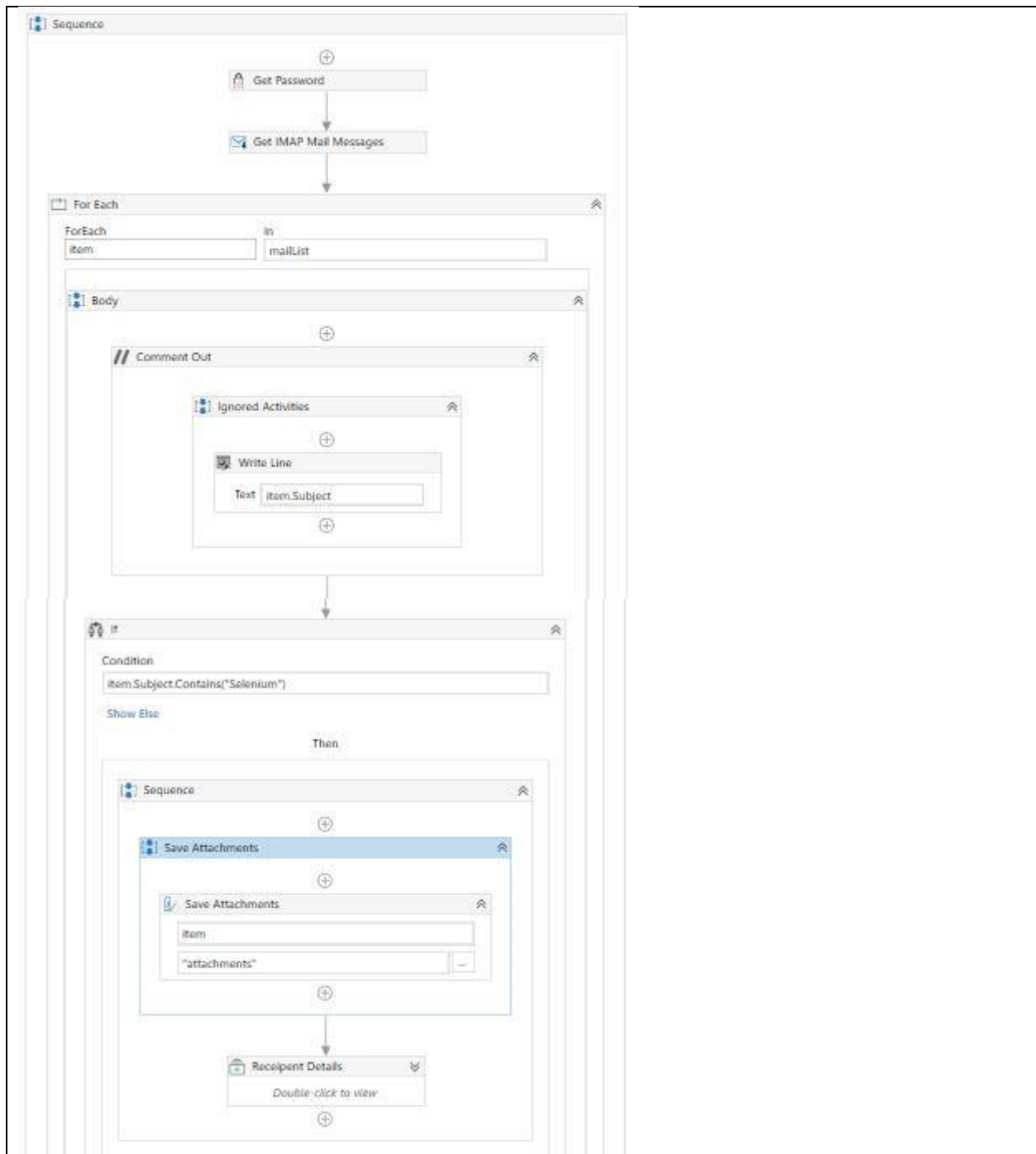
->to the **Sequence** activity of the **Then** section before **Append Line** activity

Select **Save Attachments** ->in the properties->select **Input**->Message: **item** , FolderPath: "**attachments**"

Rename the **Sequence** activity with their task

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select

Debug file->select Run File->email which contain the attachments are placed in the attachments folder



Mini Robot to Send Emails with attachments to any User

Select Activity->search Delete->select **Delete** ->drag and drop **Delete** between the **Get Password** activity and **Get IMAP Mail Messages** activity

Select **Delete** ->in the properties->select **File**->Path: "**FromDetails.txt**"

Select **Delete** ->right click->select **Surround with Try catch**

Select **Catches**->select Exception: System.IO.FileNotFoundException

Select Activity->search Write Line->select **Write Line** ->drag and drop **Write Line** in the **Catches**
Select **Write Line** ->Text: "File not found"

Open uipath->select Flowchart->Name: **MailRobot**

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename: **Send Email of Recipients to Recruiter**

Select Activity->search Send SMTP Mail Message->select **Send SMTP Mail Message** ->drag and drop **Send SMTP Mail Message** to the flowchart

Select **Send SMTP Mail Message** ->in the properties

->select **Host**->Port: **465**, Server: "**smtp.gmail.com**"

->select **Logon**->Email: give_email, Password: **pwd**

->select **Options**->IsBodyHtml: click

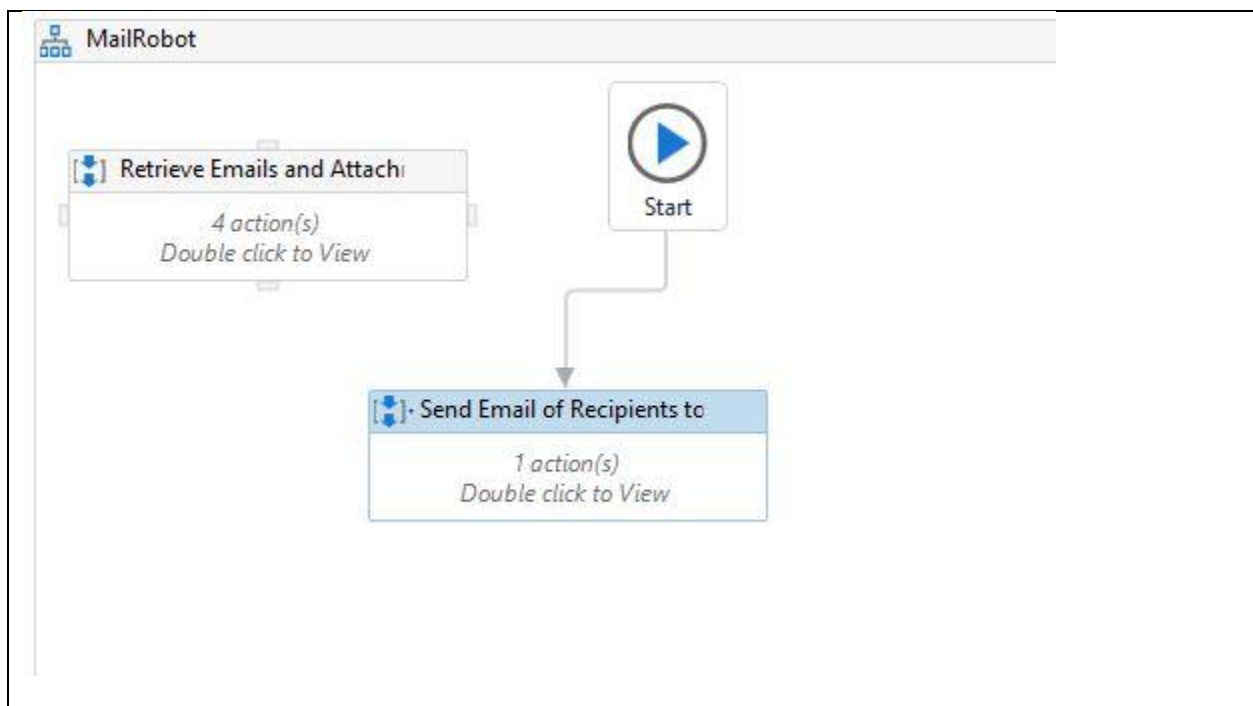
->select **Receiver**->To: receiver_email

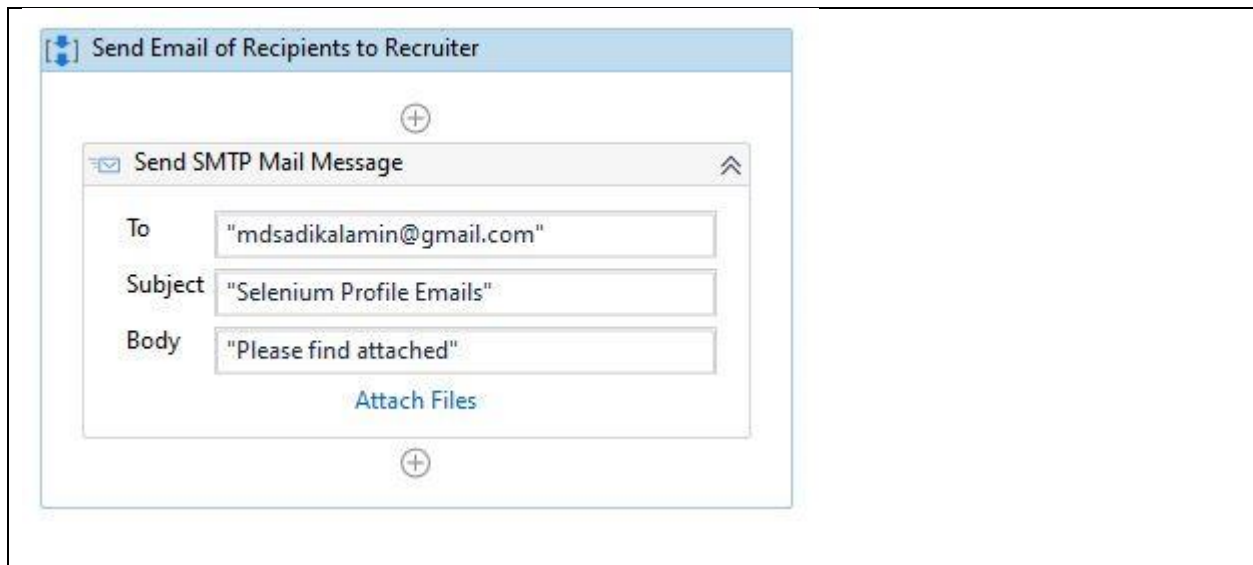
->select **Sender**->From: sender_email, Name: sender_name

->select **Email**->Subject: Subject_of_email, Body: body_of_email

Select **Send SMTP Mail Message**->select **Attach Files**->select create Argument->Value:"**FromDetails.txt**"

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File





Section 13- Data Scrapping with AI Techniques in UiPath

What is Data Scrapping and its Intelligence in extracting Data

Data Scrapping: Extract structured data from a browser, application or document to a datatable

Open uipath->select **FirstDemo1Process** project->select **Data Scrapping**-> **Extract Wizard** will open with instruction to follow->Follow the instruction from the wizard

->open browser, application or document and navigate to where to extract data from->Press Next in this dialog and hover the mouse cursor over a data source field->Click that field->Again **Extract Wizard** will open with instruction to follow->Follow the instruction from the wizard

->Select Second Element preferable the last in collection-> Again **Extract Wizard** will open

->Select **Extract Text**->select **Extract URL**->Text Column Name: **Title**->URL Column Name: **url** ->select Next-> Again **Extract Wizard** will open with Preview Data->select Finish->Another pop up will open asking for next pages->select No

->Automatically new Sequence will be created with a xaml file->Name: **DataScrappingDemo**->select create

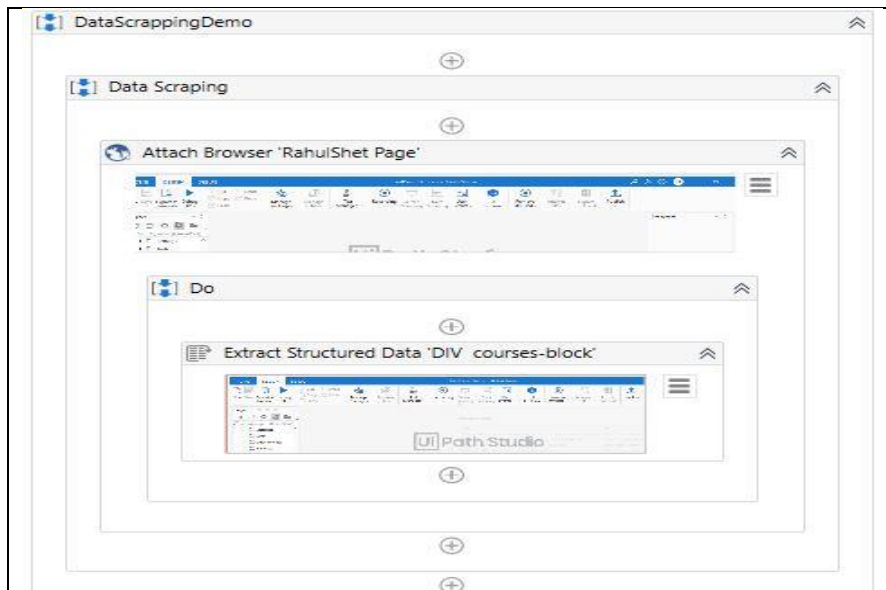
Preview Data

Title	url
All-Access Membership-Complete Access To 25+ Courses (And Counting!)	https://courses.rahulshettyacademy.c
Cypress-Modern Automation Testing From Scratch + Framework	https://courses.rahulshettyacademy.c
Core Java For Automation Testers + Interview Programs	https://courses.rahulshettyacademy.c
JavaScript SDET- Automation Testing Package	https://courses.rahulshettyacademy.c
Learn Postman For API Automation Testing With Javascript	https://courses.rahulshettyacademy.c
Software Quality Assurance Engineer - 4 Courses Bundle	https://courses.rahulshettyacademy.c
Test Architect /Devops For QA - Package	https://courses.rahulshettyacademy.c
Selenium Design Patterns/Best Practices For Framework Design	https://courses.rahulshettyacademy.c
Spring Boot Fundamentals With Unit Testing (MockMVC/Mockito)	https://courses.rahulshettyacademy.c
REST API Testing(Automation) From Scratch-Rest Assured Java	https://courses.rahulshettyacademy.c
Python SDET- Automation Testing Package	https://courses.rahulshettyacademy.c
Azure Devops Fundamentals For Testers -CI/CD+Project Boards	https://courses.rahulshettyacademy.c

Maximum number of results (0 for all) 100

Extract Correlated Data

Help < Back Finish Cancel



End to End Data Scrapping Techniques with Live example

To extract additional fields.

Open uipath->select **FirstDemo1Process** project->select **Data Scrapping**-> **Extract Wizard** will open with instruction to follow->Follow the instruction from the wizard

->open browser, application or document and navigate to where to extract data from->Press Next in this dialog and hover the mouse cursor over a data source field->Click that field->Again **Extract Wizard** will open with instruction to follow->Follow the instruction from the wizard

->Select Second Element preferable the last in collection-> Again **Extract Wizard** will open

->Select **Extract Text**->select **Extract URL**->Text Column Name: **Title**->URL Column Name: **url** ->select Next-> Again **Extract Wizard** will open with Preview Data

->To extract more fields->select **Extract Correlated Data**->Click that field->Again **Extract Wizard** will open with instruction to follow->Follow the instruction from the wizard

->Select Second Element preferable the last in collection-> Again **Extract Wizard** will open

->Select **Extract Text**->Text Column Name: **Price**>select Next-> Again **Extract Wizard** will open with Preview Data which have Title, url, Price

Select Activity->search **Excel Application Scope**->select **Excel Application Scope** ->drag and drop **Excel Application Scope** to the + drop activity after **Attach Browser**

Select Activity->search **Write Range** of Excel->select **Write Range** ->drag and drop **Write Range** to the + drop activity of **Do** of the **Excel Application Scope**

Select **Excel Application Scope** ->in the properties->Workbook path: "DataScrappingMethod.xlsx"

Select **Write Range** -> in the properties

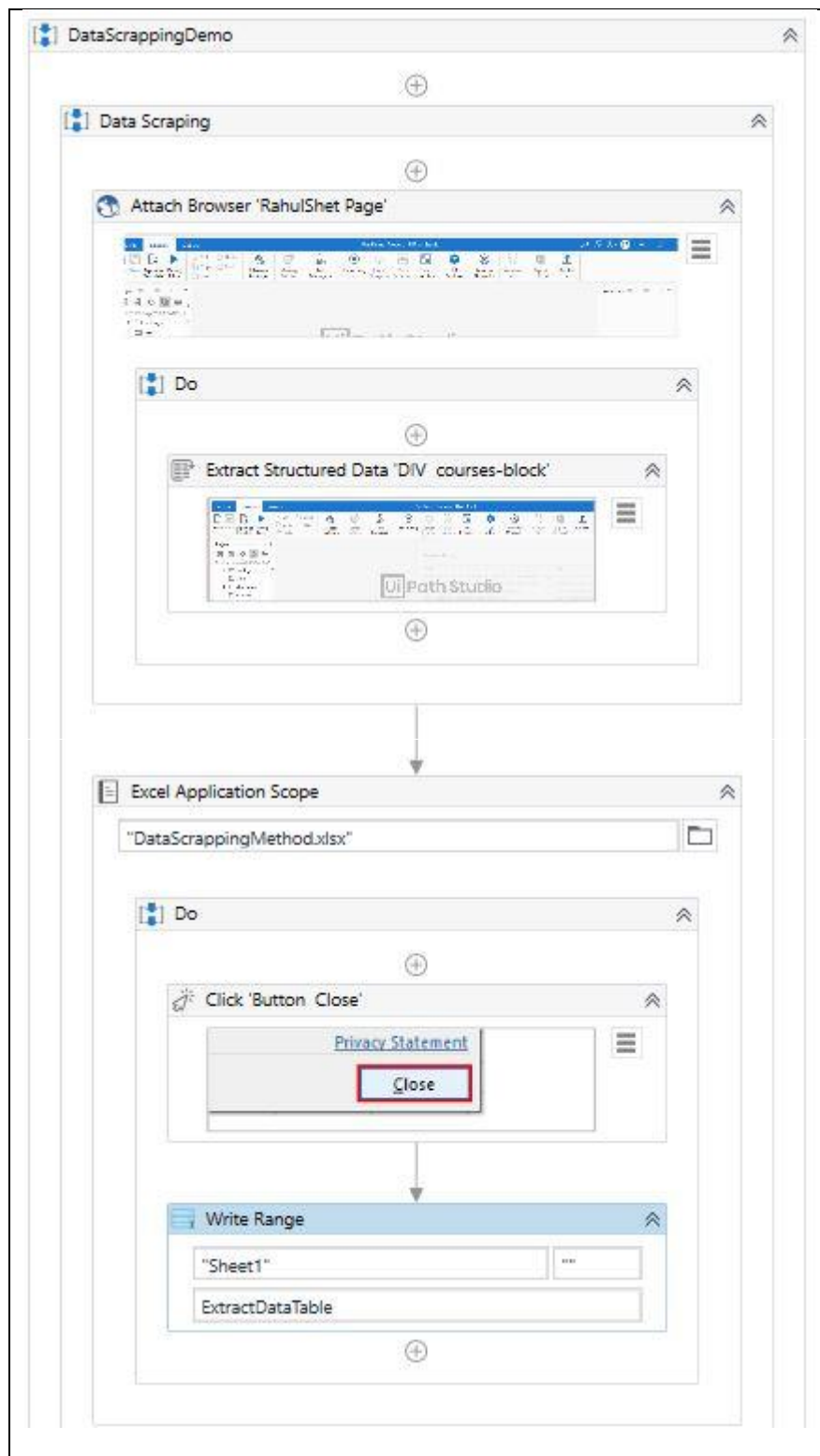
->select **Destination**->StartingCell: ""

->select **Input**->DataTable: ExtractDataTable

->select **Options**->AddHeader: selected

To run the workflow ->select Debug file->select Run File

->Automatically the data are extracted and place in the DataScrappingMethod.xlsx excel file



Section 14- Build a Robot for Data Analytics (Filter, Sort, Limit) using UiPath

Demo - Understand the Business rules to Automate

Business Scenario:

Extract the Top 3 lowest income holders with Designation of Employee and Student and Submit into Web Portal

Applying Filter to Excel Table Data with UiPath Activity

Open uipath->select **FirstDemo1Process** project->select project->right click on **FirstDemo1Process**
->select add->select Flowchart->Name: **SortFilter**->select create

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the flowchart->double click on **Sequence**->Rename:

Copy the excel file with data name: **Income.xlsx** and place in the project

Select Activity->search Excel Application Scope->select **Excel Application Scope** ->drag and drop **Excel Application Scope** to the flowchart

Select **Excel Application Scope**->in the properties->Workbook Path: "Income.xlsx"

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the **Do** of the **Excel Application Scope**->double click on **Sequence**->Rename: **Filter the Table with Student and Employee values**

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the **Do** of the **Excel Application Scope** after the **Filter the Table with Student and Employee values**->double click on **Sequence**

->Rename: **Sort on Income Column**

Select Activity->search Filter Table->select **Filter Table** ->drag and drop **Filter Table** to the **Filter the Table with Student and Employee values**

->Select **Filter Table** ->in the properties

->select **Input**->TableName: "**Table1**", ColumnName: "**Occupation**",
FilterOptions: {"Student", "Employee"}

A	B	C	D
Name	Gender	Occupation	Income
Sam	Female	Student	920
Josh	Male	Employee	2000
Sri	Female	Student	2100
Lam	Female	Entrepreneur	4000
Tam	Female	Employee	3000
Mike	Male	Entrepreneur	2200
Debang	Female	Student	3200
chris	Male	Entrepreneur	12000
Jenni	Female	Student	5500
Ram	Male	Employee	6000
Gam	Male	Employee	9000
Sai	Male	Entrepreneur	20000
Rahul	Male	Employee	9500

Applying Sort to Excel Table Data with UiPath Activity

Select Activity->search Sort Table->select **Sort Table**->drag and drop **Sort Table** to the sequence, **Sort on Income Column**

Select **Sort Table**->in the properties->select **Input**->TableName: "**Table1**"->CoumnName: "**Income**"

Select Activity->search sequence->select **Sequence**->drag and drop **Sequence** to the **Sort on Income Column**>double click on **Sequence**->Rename: **Read Content in Table**

Select Activity->search Read Range->select **Read Range** ->drag and drop **Read Range** to the **Read Content in Table**

Select **Read Range** -> in the properties

->select **Input**->Range: "**A2:D14**"

->select **Options**->AddHeaders: **selected**, UseFilter: **selected**

->select **Output**-> Output DataTable variable press **ctrl + k**-> Output DataTable: **extractedData** ->then give enter

->Select Variables->select **extractedData**->change Scope to: Sequence

Select Activity->search Output Data Table->select **Output Data Table** ->drag and drop **Output Data Table** in the sequence **Read Content in Table** after **Read Range**

Select **Output Data Table** -> in the properties->

select **Input**->DataTable: **extractedData**

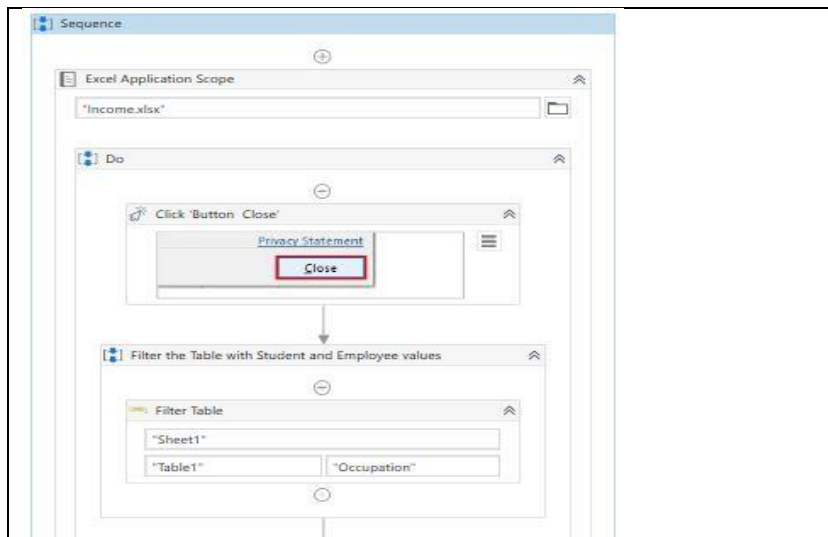
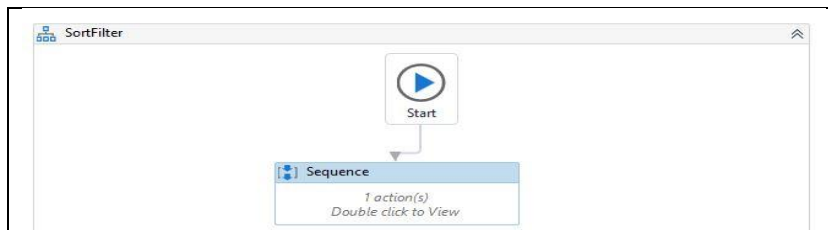
select **Output**->Text: **data**

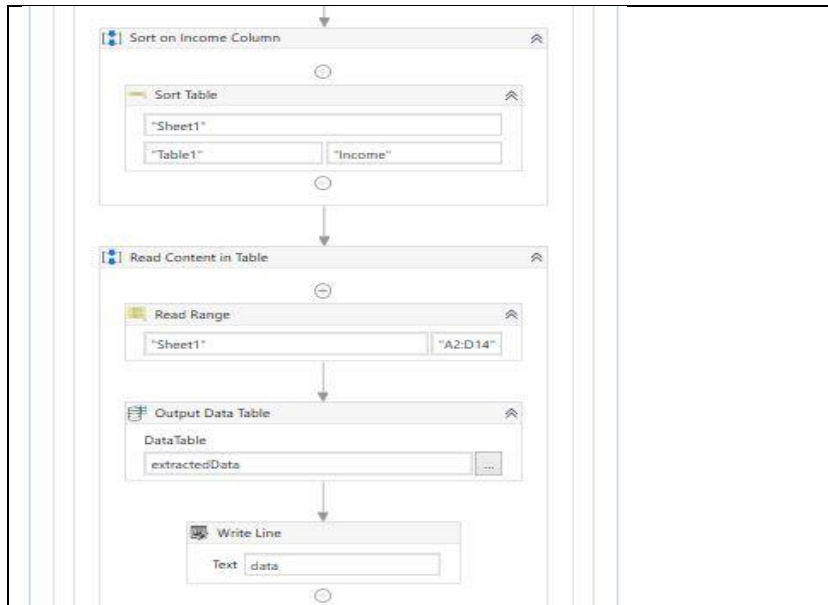
Select Activity->search Write Line->select **Write Line** ->drag and drop **Write Line** in the sequence **Read Content in Table** after **Output Data Table**

Select **Write Line** -> in the properties->select **Output**->Text: **data**

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select Debug file->select Run File

A	B	C	D
Name	Gender	Occupation	Income
Sam	Female	Student	920
Josh	Male	Employee	2000
Sri	Female	Student	2100
Tam	Female	Employee	3000
Debang	Female	Student	3200
Jenni	Female	Student	5500
Ram	Male	Employee	6000
Gam	Male	Employee	9000
Rahul	Male	Employee	9500





Get Table Range in Excel Dynamically using UiPath Activity

Select Activity->search Get Table Range->select **Get Table Range** ->drag and drop **Get Table Range** in the sequence **Read Content in Table** before **Read Range**

Select **Get Table Range** ->in the properties

->select **Input**->TableName: "**Table1**"

->select **Output**-> Output variable press **ctrl + k**-> Range: "**EmployeesRange**"

Select **Read Range** -> in the properties->select **Input**->replace Range-> Range: "**EmployeesRange**"

Business Scenario:

Extract the Top 3 lowest income holders with Designation of Employee and Student and submit into Web Portal

Select Activity->search Sequence->select **Sequence** ->drag and drop **Sequence** after **Read Content in Table**->double click on **Sequence**->Rename: **Submit values into Portal**

Select Activity->search Open Browser->select **Open Browser** ->drag and drop **Open Browser** in **Submit values into Portal**

Select **Open Browser** ->in the properties

->select **Input**->Url: "<https://rahulshettyacademy.com/angularpractice/>"

->select **Input**->BrowserType: **Chrome**

Select Activity->search For Each Row->select **For Each Row in Data Table** ->drag and drop **For Each Row in Data Table** in **Do** of **Open Browser** activity

Select **For Each Row in Data Table**-> in the properties

->Select **Input**->DataTable: **extractedData**

Select Activity->search Type Into->select **Type Into** ->drag and drop **Type Into** in to the **Body of For Each Row in Data Table**

Open the Url link from before->Select **Type Into** ->click **Indicate element inside browser**->click on the input of Name

Select **Type Into** ->in the properties

->select **Input**->Text: "row("Name").ToString"

->select **Options**->EmptyField: True

--

Build Front End Application Logic and send data from Excel

Select Activity->search Assign->select **Assign** ->drag and drop **Assign** in to the **Body of For Each Row in Data Table** after **Type Into**

Select **Assign** -> in the properties

->Select **Misc**->To variable press **ctrl + k**-> To: **designation**

->Select **Misc**->Value: **row("Occupation").ToString**

Select Activity->search If->select **If** ->drag and drop **If** in to the **Body of For Each Row in Data Table** after **Assign** ->double click on **If**->Rename: **Select Radio Button based on Designation**

Select **Select Radio Button based on Designation** ->Condition: **designation.Equals("Employee")**

Select Activity->search Click->select **Click** ->drag and drop **Click** in to the **Then of Select Radio Button based on Designation**

Open the Url link from before ->Select **Click** ->click **Indicate element inside browser**-> click on the radio button of Employed

Select Activity->search Click->select **Click** ->drag and drop **Click** in to the **Else of Select Radio Button based on Designation**

Open the Url link from before ->Select **Click** ->click **Indicate element inside browser**-> click on the radio button of Student

Select Activity->search Click->select **Click** ->drag and drop **Click** after **Select Radio Button based on Designation** ->double click on **Click**->Rename: **Submit the Form**

Open the Url link from before ->Select **Click** ->click **Indicate element inside browser**-> click on the submit

Limit the number of Rows returned to Web Portal with control flow logic

->Select Variables->Name: **counter**, Variable type: **Int32**, Scope: **Submit values into Portal**, Default: **1**

Select Activity->search Sequence->select **Sequence** ->drag and drop **Sequence** in to the **Body of Submit values into Portal** after **click**

Select Activity->search Assign->select **Assign** ->drag and drop **Assign** in **Sequence** activity

->Select **Assign** ->in the properties

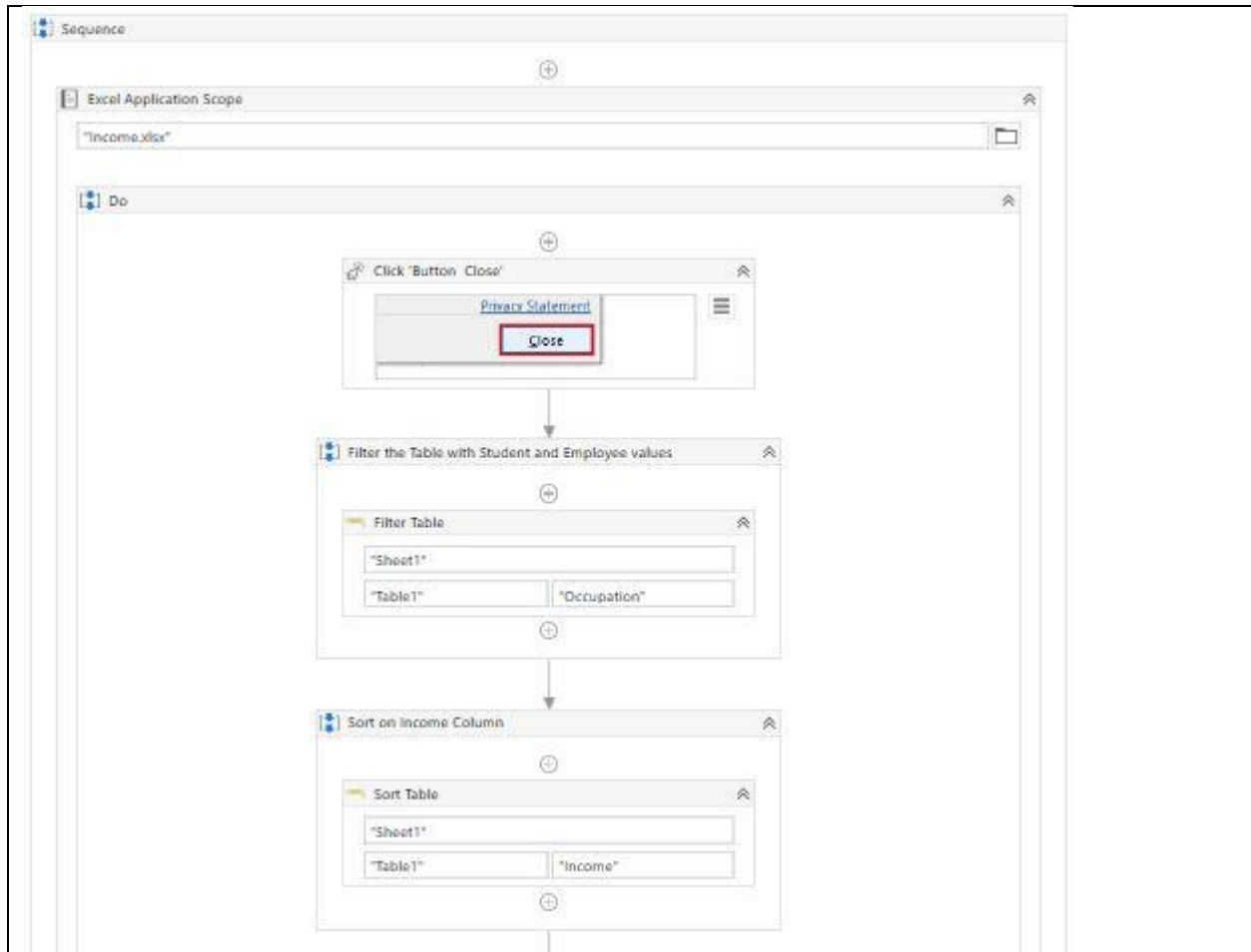
->select **Misc**-> To: **counter**, Value: **counter+1**

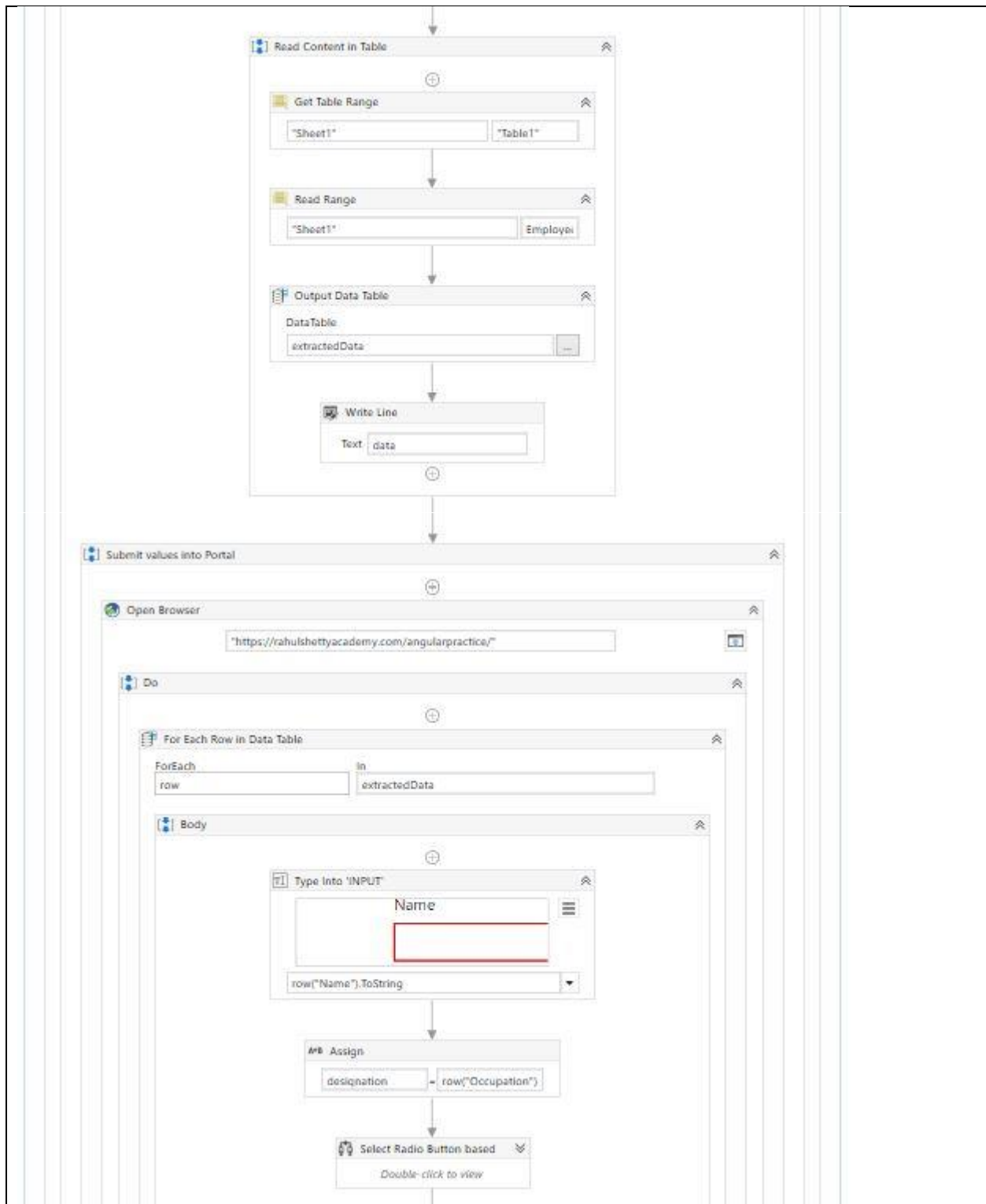
Select Activity->search If->select **If** ->drag and drop **If** in to the **Body** of **Submit values into Portal** after **Assign**

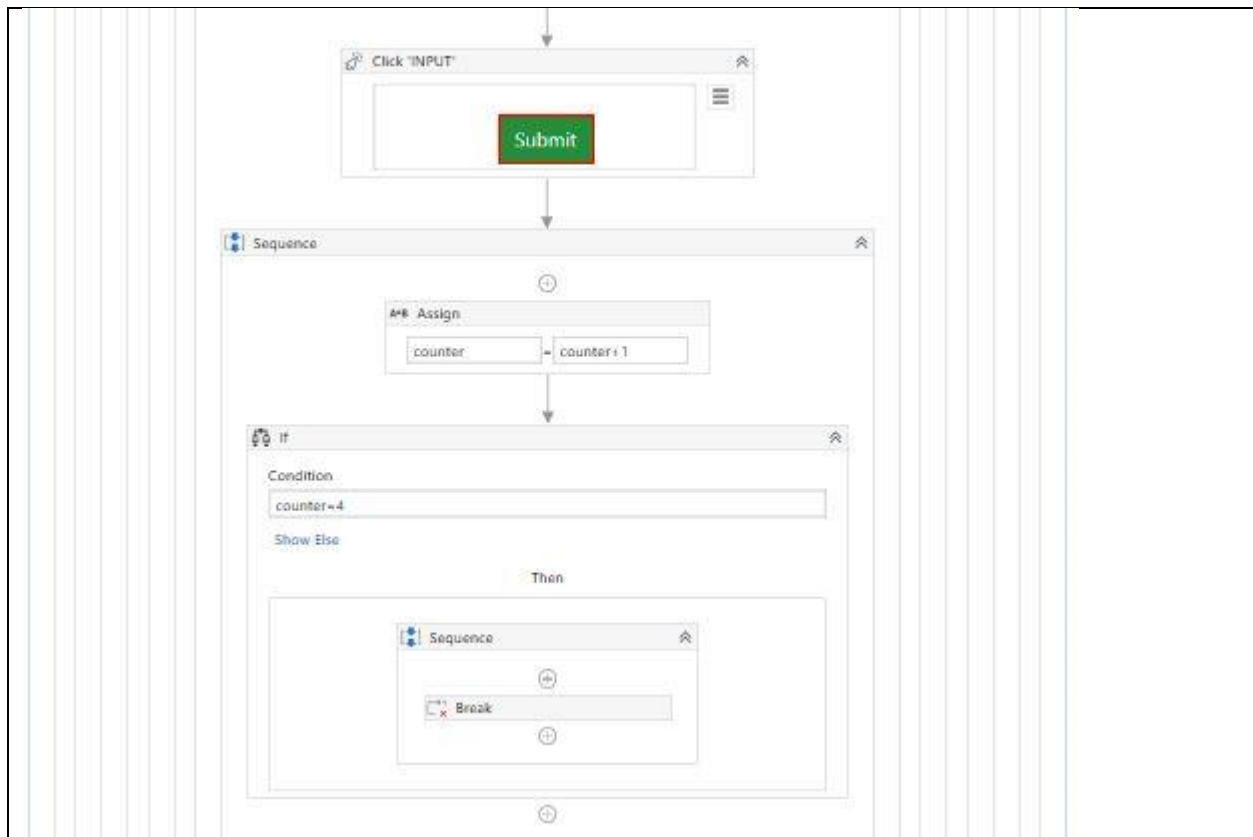
Select **If** -> in the properties->Condition: **counter=4**

Select Activity->search Break->select **Break** ->drag and drop **Break** in to the **Then** of **If** condition

To run the workflow->right click on the particular file which will run->select **Set as Start Node**->select **Debug file**->select **Run File**







Section 15- Orchestrator- Robot Management DashBoard

What is Orchestrator and its advantages with UiPath

UiPath Orchestrator

It is centralized robot management dashboard where you can easily deploy, secure and manage your **UiPath Robots** at scale.

Setup Instructions:

Log into UiPath Portal and select Service

Create the Robot in Orchestra

Add Robot to Environment

Copy the Machine key from Machines Page

Open Robot Tray (C:\Users\Owner\AppData\Local\UiPath)

Add Machine Key and OrchestraUrl in Robot tray to connect

Publish Package/Deploy process

Robots in Orchestrator will run the Process

Run the Robot Job

- Go to : <https://platform.uipath.com/>
- Login to the account
- Select Orchestrator-> Select Tenant->select Machines

Running Robots from Orchestrator Dashboard on Demand

- Go to : <https://platform.uipath.com/>
- Login to the account
- Select Orchestrator-> Select Tenant->select Automations->select Process
- Select Add Process->select Package Source Name->select Entry Point

Notes from others

- First close the UiPath studio and sign out UiPath Assistance
- Open the UiPath Assistance->select Preference->select Orchestrator Settings->select Connection type drop down and select Machine Key->Copy the Machine name
- Go to : <https://platform.uipath.com/>
- Login to the account

Create a new Tenant:

- Select Create new Orchestrator services
- Select Admin->Select Add Tenant-> Tenant Name: **TenantSadik** ->Select all Provision Services
->select save->A Tenant is created

- Select Orchestrator->select Tenant->select Machines->select Add Machine
->Select **Machine Template**->in the Template name: paste the copied Machine name
e.g: LAPTOP-VJJTB407
->In Production (Unattended): 1
->In Testing: 1
->Process Type: All
->Process Compatibility: All
->select Provision
->**Client ID/Machine Key** is generated->copy the Machine Key->paste the Machine Key in the Uipath Assistance Preference Machine Key
- In the Uipath Assistance give the Orchestrator URL which is until the url link orchestrator_/
e.g. <https://cloud.uipath.com/naaicurl/TenantSadik/orchestrator/>
- Select Orchestrator->select Tenant->select Folders->select New Folder->give Name: **IT**
->in the Process packages source->select **Tenant package feed**->select Create
- To find Domain user name->open Command prompt->type: **whoami**->give enter
->it shows the domain user name, eg: laptop-vjjtb407\mesad->copy the domain user name
- Select the Folder **IT**->in the Users->select edit->Select all the Roles->select next
->Select Enable Attended robot
-> Select Enable Unattended robot->Select Machine Login credentials->paste the copied domain user name->Password: give the window login password->select next->select update
->then robot setup is done for the user
- Select Tenant->select Folders->select IT->select Machines->select **Manage Machines in Folder**
->select the added machine->select update
- In the Uipath Assistance->select Connect and Sign in
->Status should show, Status: Connected, Licensed
- Open uipath studio->select **FirstDemo1Process** project->select project->select **IT**->select Publish->select next->select Publish to: **Orchestrator Tenant Process feed**->select next
->select publish
- Select Orchestrator->select Tenant->select Packages->in the package->select view versions-> to check the packages
- Select Orchestrator->select Tenant ->select Folders->select IT->select Process
->select **Add Process**->select Package Source Name->select Entry point->select next->select next->select create->then the process is added to the folder->select Start Now
->select the particular **Account**->select the particular **Machine** ->select the particular **Hostname**
->select Start->the job will be running->It can be run from Uipath studio, Uipath Assistance and Orchestrator

