6. Build a workflow using a Screen Scraper Wizard that scrapes text using the Tesseract OCR scraping method from an image and stores it in a Notepad.
- Search for "text images" in Google Images.
- Pick one image containing text from the search results.
- Scrape the text from the image using Tesseract OCR. - Store text in a Notepad file

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as "Screen Scraping".

Step 3: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 4: Name the Sequence activity as "Sequence – 'Demonstrating use of Screen Scraping Wizard'".

Step 5: Drag and drop an Open Browser activity, name it as "Open Browser - Google Images" and enter the URL – "www.google.com/images".

Step 6: In the Do container of the Open Browser activity, insert a Type Into activity and name it as "Type Into -Types 'Text Images' and hit 'Enter' key."

Step 7: Click on the "Indicate element on screen" link and select the search bar of the Google search engine.

Step 8: In the text area of the Type Into activity, enter the text as "text images[k(enter)]".

Step 9: Click the Screen Scraping button in the Design ribbon and select an image that contains some text from the search results.

Step 10 : From the Screen Scraper Wizard window, choose Tesseract OCR as the Screen Scraping method and change the Scale to 5.

Step 11: Drag and drop a Write Text File activity inside the Screen Scraping container (below Attach Browser), write the file name as "ScarpedText.txt" and pass the variable in which the extracted text is stored.

Step 12: Save and run the workflow.

7. Build a workflow using the Data Scraping wizard that scrapes blog post titles from the UiPath Blog from multiple pages.
- Open the UiPath Blog (https://www.uipath.com/blog).
- Extract all blog titles and URL by navigating through all pages.
- Store scraped data in an Excel file

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as "Data Scraping".

Step 3: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 4: Name the Sequence activity as "Sequence – 'Demonstrating the use of Data Scraping Wizard".

Step 5: Drag and drop an Open Browser activity and name it as "Open Browser - UiPath Blog website."

Step 6: Enter the URL – "www.uipath.com/blog".

Step 7: Click the Data Scraping button in the Design ribbon.

Step 8: Click Next in the Extract wizard window and select the title of the first blog post.

Step 9: Click Next again and select the title of the second blog post.

Step 10 : Click "Finish" in the preview window.

Step 11: Go to the blog page, and scroll down to the bottom to locate the Load More button.

Step 12: Go to the Indicate Next Link window, and click "Yes", indicate the Load More button.

Step 13: Insert a Write CSV activity in "Data Scraping" container, and in the "Write to what file:" text box, enter "ScrapedText.txt".

Step 14: In the Write From section pass the variable in which the extracted text is stored.

Step 15: Save and run the workflow.

8. Build a workflow using a Read PDF Text activity and extract only Email IDs and Phone Numbers from a PDF file and store it in an MS Word file.

- Read data from the PDF file using a Read PDF Text activity.

- Extract only Phone Numbers and email IDs from the PDF and store it in an MS Word file

Step 1: Open a new MS Word file.

Step 2: Open UiPath Studio.

Step 3: Create a new process and name it as "PDF Extraction".

Step 4: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 5: Name the Sequence activity as "Sequence – 'PDF Extraction"

Step 6: Drag and drop a Read PDF Text activity and name it as "Read PDF Text - Read 'challenge.pdf'".

Step 7: Click the folder button and select the pdf file located in the folder

Step 8: Create a variable through the Variables panel as shown below

Name	Variable type	Scope	Default
pdfOutput	String	Sequence – PDF Extraction	

Step 9: Enter the variable **pdfOutput** in the Output property of the Read PDF Text activity's Properties panel.

Step 10 : Insert a Matches activity and name it as "Matches – Email".

Step 11: Click the *Configure Regular Expressions* button.

Step 12: In the RegEx column, select **Email**.

Step 13: In the Quantifiers column, select Any (0 or more).

Step 14: Exit the RegEx Builder window after updating.

Step 15: In the Properties panel of the Matches activity, press **Ctrl + K** in the Result property, and enter a new variable called **emails**. Enter **pdfOutput** in the Input property.

Step 16: Insert another Matches activity and name it as “Matches – Phone Number”.

Step 17: Click the *Configure Regular Expressions* button

Step 18: In the RegEx column, select **Advanced**

Step 19: In Quantifiers, select Any (0 or more).

Step 20: In the Value column, enter the expression: **(407)([0-9])**

Step 21: Exit the RegEx Builder window

Step 22: In the Properties panel of the Matches activity, press **Ctrl + K** in the Result property, and enter a new variable called **phones**. Enter **pdfOutput** in the Input property.

Step 23: Drag and drop a For Each activity, name it as “For Each - Email”.

Step 24: In the first text box enter “**item”** and in the second box enter “**emails”**.

Step 25: Insert an Attach Window activity in the Body section of the For Each activity and name it as “Attach Window MS Word (Email)”.

Step 26: Click the hamburger button of the Attach Window and open Selector Editor. Replace the title with an asterisk (*). It will make the workflow run for MS Word files with varying names.

Step 27: Insert a Type Into activity in the Do section of the Attach Window activity and name it as “Type Into - MS Word File”.

Step 28: Click on the “Indicate element on screen” link and select the text editor area of MS Word.

Step 29: In the text area of the Type Into activity, enter **item.ToString + vbCr**.

Step 30: Insert another For Each activity, name it as “For Each – Phone Number”.

Step 31: In the first text box, enter **item** and in the second box enter **phones**.

Step 32: Insert an Attach Window activity in the Body section of the second For Each activity and name it as “Attach Window MS Word (Phone Number)”. Click the “Indicate window on screen” link and indicate the MS Word window.

Step 33: Click the hamburger button of the Attach Window activity and open Selector Editor. Replace the title with asterisk (*). It will make the workflow run for any MS Word files with varying names.

Step 34: Insert a Type Into activity in the Do section of the Attach Window activity and name it as “Type Into - MS Word File”.

Step 35: Click on the “Indicate element on screen” link and select text editor area of MS Word.

Step 36: In the text area of Type Into activity, enter **item.ToString + vbCr**

. Step 37: Save and run the workflow.

9. Build a workflow to display file names from a folder in the Output panel and also store names in an MS Word file.
- Locate and select a folder containing multiple files.
- List the directory path of all the files in the Output panel.
- Also, store the updated names in an MS Word file and save and close it.

Step 1: Open a new MS Word file.

Step 2: Open UiPath Studio.

Step 3: Create a new process and name it as "For Each Activity"

Step 4: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 5: Name the Sequence activity as "Sequence – "This code is to write all the file names in MS Word present in a particular folder.'"

Step 5: Insert a Select Folder activity in the Designer panel

Step 6 : In the Variables panel, define a new variable as shown below:

Name	Variable type	Scope	Default
FolderName	String	Sequence - 'This code is to write all the file names in MS Word present in a particular folder'	

Step 7: In the Properties panel of the Select Folder activity, enter **FolderName** in the Output property.

Step 8: Insert an Assign activity below the Select Folder activity and name it as "Assign - 'File List'".

Step 9: In the Variables panel, define a new variable as shown below:

Name	Variable type	Scope	Default
FileList	System.String[]	Sequence - 'This code is to write all the file names in MS Word present in a particular folder'	

Step 10 : In the Properties panel of the Assign activity, enter **FileList** in the Output property.

Step 11: In the Assign activity, enter the value as shown below:

To	Value
FileList	Directory.GetFiles.(FolderName)

Step 12: Click the hamburger button and select Edit Selector. In the bottom panel of the Selector Editor, rename the title of the MS Word to '*'. Click OK to save the changes.

Step 13: Drag and drop a For Each activity in the Do section of the Attach Window activity, insert **item** in the first text box, and the variable **FileList** in the second text box.

Step 14: Drag and Drop a Write Line activity in the Body section of the For Each activity. Enter the expression: **item.ToString.**

Step 15: Drag and Drop a Type Into activity in the Body section of the For Each activity and name it as "Type Into - 'File List into MS Word'".

Step 16: Click on the "Indicate element on screen" link and select the editor area of MS Word.

Step 17: In the text area of Type Into activity, enter the expression: item.ToString + "[k(enter)]"

Step 18: Drag a Send Hotkey activity and drop it below the For Each activity.

Step 19: Click on the "Indicate element on screen" link and select the editor area of MS Word.

Step 20: In the Send Hotkey activity, select *F12* from the dropdown for Key option.

Step 21: Insert a Click activity. Click "Indicate element inside window" link and select the Save button of the 'Save As' dialog box.

Step 22: Name the Click activity as "Click 'Save' button"**.**

Step 23: Insert second Send Hotkey activity and name it "Send Hotkey" activity as Send Hotkey - 'To close the application'"

Step 24: Click on the "Indicate element on screen" link and select the editor area of MS Word.

Step 25: In the Send Hotkey activity, check the box below Alt option, and choose *F4* from the dropdown of Key option.

Step 26: Save and run the workflow.

10.

Build a workflow using a Try Catch activity to do the following:

- Take Name, Gender, and Age as the user input.
- Subtract current year with Age value to get the Year of Birth.
- Handle an error that occurs due to a reckless user input of an incorrect age containing the 11-digit number.
- Continue the process to display the Name, Gender, and Year of Birth of the user in a message box.

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as "Try Catch activity"

Step 3: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 4: Name the Sequence activity as "Sequence – 'A workflow using Try Catch activity to catch certain errors'".

Step 5: Drag and drop a Try Catch activity and name it as "Try Catch - 'User Inputs'".

Step 6: In the Try Section, insert a Sequence activity and name it as "Sequence - 'User Inputs'".

Step 7: Insert an Input Dialog activity within the Sequence activity, name it as "Input Dialog - User Name" and enter the values as shown below:

Title	Label
"Name"	"Enter your Name"

Step 8: Insert a second Input Dialog activity below the previous "Input Dialog - User Name" activity, and name it as "Input Dialog - User Gender" and enter the values as shown below

Title	Label
"Gender"	"Enter your Gender"

.

Step 9: Insert a third Input Dialog activity below the previous “Input Dialog - User Gender” activity, and name it as “Input Dialog - User Age” and enter the values as shown below:

Title	Label
“Age”	“Enter your Age”

Step 10 : In the Variables panel, define five new variables as shown shown below:

Name	Variable type	Scope	Default
userName	String	Sequence - ‘A workflow using Try Catch activity to catch certain errors’	
userGender	String	Sequence - ‘A workflow using Try Catch activity to catch certain errors’	
intUserAge	Int32	Sequence - ‘A workflow using Try Catch activity to catch certain errors’	
intYearOfBirth	Int32	Sequence - ‘A workflow using Try Catch activity to catch certain errors’	
inputError	String	Sequence - ‘A workflow using Try Catch activity to catch certain errors’	

Step 11: In the Properties panel of “Input Dialog – User Name” activity, enter **userName** in the Output property

. Step 12: In the Properties panel of “Input Dialog – User Gender” activity, enter **userGender** in the Output property

.

Step 13: In the Properties panel of “Input Dialog – User Age” activity, enter **intUserAge** in the Output property.

Step 14: Insert an Assign activity below the “Input Dialog – User Age” activity and name it as “Assign - Year of Birth.” and enter values as shown below:

To	Value
intYearOfBirth	Date.Today.Year() - intUserAge

Step 15: In the Catches section of the Try Catch Activity, select “System.Exception” from the dropdown

Step 16: Insert an Assign activity in the Catches section and name it as “Assign – InputError” and enter values as shown below:

To	Value
inputError	exception.Message

Step 17: Insert a Message Box activity below the Try Catch activity and name it as “Message Box - User Details”.

Step 18: In the text area of the Message Box activity, enter the expression: **"Name: " + userName + vbCr + "Gender: "+ userGender + vbCr + "Year of Birth: " + intYearOfBirth.ToString + vbCr + "Error: " + inputError**

Step 19: Save and run the workflow